



Universitat d'Alacant
Universidad de Alicante

SAFE HUMAN-ROBOT INTERACTION
BASED ON MULTI-SENSOR FUSION
AND DEXTEROUS MANIPULATION
PLANNING

Juan Antonio Corrales Ramón

Tesis **Doctorales**

www.eltallerdigital.com

UNIVERSIDAD de ALICANTE



Universitat d'Alacant
Universidad de Alicante
Escuela Politécnica Superior
Departamento de Física, Ingeniería de Sistemas y Teoría de la Señal

SAFE HUMAN-ROBOT INTERACTION BASED ON MULTI-SENSOR FUSION AND DEXTEROUS MANIPULATION PLANNING

Ph.D. Thesis

Doctorado Interuniversitario en Automática y Robótica

Author:

Juan Antonio Corrales Ramón

Supervisors:

Fernando Torres Medina
Francisco Andrés Candelas Herías

Alicante, June 2011

Abstract

The main goal of this thesis is to develop human-robot interaction tasks where human operators and robotic manipulators can cooperate in the same workspace. In order to do so, this thesis proposes several techniques to fulfill two requirements which are essential for this human-robot interaction: the guarantee of the human safety during human-robot interaction and the development of dexterous manipulation which improves the skills of robotic manipulators with robotic hands.

For the first requirement, this thesis develops new safety strategies which stop the normal behavior of the robot and activate a special safety behavior when the human-robot distance is below a safety threshold. A precise full-body human tracking system is required in order to compute on real-time this human-robot distance. This thesis makes a comparison of the current technologies of human motion capture systems and proposes to use an inertial motion capture system to track the human operator who collaborates with the robot. Although the relative rotational measurements of this system are very accurate and are applied over a skeleton; the global position of this skeleton accumulates errors. Thus, this thesis uses an additional localization system based on UWB (Ultra-wideband) pulses in order to correct this error. This thesis presents three novel fusion algorithms based on Bayesian filtering in order to combine the global position measurements of both systems. The first algorithm is based on a Kalman filter, the second algorithm is based on a particle filter and the third algorithm is based on a combination of a Kalman filter and a particle filter. All these algorithms have a new structure based on recalculating the transformation matrix between the coordinate frames of both systems each time a new UWB measurement is received. This structure improves the computational cost of the proposed algorithms in comparison with previous similar approaches because it considers the complementariness between the features of both systems: the rotational accuracy and the high sampling rate of the inertial system with the positional accuracy and the low sampling rate of the UWB system.

This thesis also proposes to cover the bones which compose the skeleton registered by the previous human tracking system by a hierarchy of bounding volumes in order to obtain a precise and efficient approximation of the human-robot distance. This hierarchy is composed by three levels which contain different bounding volumes depending on the required level of detail. The first level is composed by general AABBs (Axis-Aligned Bounding Boxes) which cover with one bounding volume all the body of the human and all the body of the robot. The second level is composed by local AABBs which cover the main limbs/links of the human and the robot. The third level is composed by SSLs (Swept-Sphere Lines) which cover each bone of their skeletons. Thereby, when the human and the robot are far away from each other, only the first level is needed to compute a good approximation of the human-robot distance. Nevertheless, when they are close to each other, the second or even the third levels are required. This hierarchical representation not only improves the computational cost of the human-robot distance calculation by reducing the number of pairwise distance tests between bounding volumes, but it also improves the approximation of the human and robot

surfaces by using the SSLs of the third level of the hierarchy in comparison with previous human-robot interaction systems based on spherical bounding volumes. In addition, the radii of these SSLs are modified dynamically according to the linear velocity of the bone which is covered by each SSL. This variation supposes an increase of the human safety since the bounding volumes of the third level of the hierarchy not only represent an approximation of the surface of the human and the robot bodies but also an estimation of their displacement between two execution steps of the safety strategy. This hierarchy of dynamic bounding volumes is used in safety strategies which have been applied successfully in several real human-robot interaction tasks.

For the second requirement of flexible human-robot interaction tasks, this thesis proposes a novel dexterous manipulation planner based on in-hand movements generated by multi-fingered robotic hands installed at the end-effector of robotic manipulators which cooperate with human operators. This planner receives as input an initial grasp of an object and a desired final configuration (i.e. position and orientation) of the object. This planner computes the movements of the fingers which are required to drive the object to the final configuration. It is built with a modular structure based on two levels: global and local planner. The global planner generates a trajectory of the object composed by intermediate configurations between the initial and final configurations. The local planner applies a novel contact evolution model based on a triangle mesh representation of the object and fingers surfaces in order to determine the possible evolution of the contacting primitives (i.e. vertices, edges and faces of the triangle meshes). This model provides a more general representation of the object and fingers surfaces than previous manipulation planners based on planar or parametric models. Afterwards, the local planner generates movements of the fingers which cause this contact evolution by applying the pseudo-inverse of the Jacobian matrix of each finger. Furthermore, this thesis adds an additional finger readjustment algorithm after each execution of the local planner in order to guarantee that the contacting fingers apply enough pressure to the object so that unstable grasps are avoided. This planner has been successfully applied in several real experiments of manipulation tasks with a three-fingered robotic hand equipped with tactile sensors.

Resumen

El principal objetivo de esta tesis es desarrollar tareas de interacción humano-robot en las que operadores humanos y robots manipuladores puedan cooperar en el mismo espacio de trabajo. Para ello, esta tesis propone varias técnicas para cumplir dos requisitos que son esenciales para esta interacción humano-robot: la garantía de la seguridad humana durante la interacción humano-robot y el desarrollo de técnicas de manipulación diestra que mejoren las habilidades de los robots manipuladores con manos robóticas.

Para el primer requisito, esta tesis desarrolla nuevas estrategias de seguridad que paran el comportamiento normal del robot y activan un comportamiento de seguridad especial cuando la distancia humano-robot está por debajo de un umbral de seguridad. Es necesario un sistema de seguimiento del humano preciso y para todo el cuerpo para calcular la distancia humano-robot en tiempo real. Esta tesis hace una comparativa de las tecnologías actuales de los sistemas de captura de movimiento y propone usar un sistema inercial de captura de movimiento para seguir al operador humano que colabora con el robot. Aunque las medidas rotacionales de este sistema son muy precisas y se aplican sobre un esqueleto; la posición global de este esqueleto acumula errores. Por ello, esta tesis usa un sistema de localización adicional basado en pulsos UWB (Ultra-wideband) para corregir este error. Esta tesis presenta tres nuevos algoritmos de fusión basados en el filtro de Bayes para combinar las medidas de posición global de ambos sistemas. El primer algoritmo se basa en un filtro de Kalman, el segundo algoritmo se basa en un filtro de partículas y el tercer algoritmo se basa en una combinación de un filtro de Kalman y un filtro de partículas. Todos estos algoritmos tienen una nueva estructura basada en recalculer la matriz de transformación entre los sistemas de coordenadas de ambos sistemas cada vez que se recibe una nueva medida del sistema UWB. Esta estructura mejora el coste computacional de los algoritmos propuestos en comparación con propuestas similares anteriores porque considera la complementariedad entre las características de ambos sistemas: la precisión rotacional y la alta frecuencia de muestreo del sistema inercial con la precisión posicional y la baja frecuencia de muestreo del sistema UWB.

Esta tesis también propone cubrir los huesos que componen el esqueleto registrado por el sistema de seguimiento humano anterior con una jerarquía de volúmenes envolventes para obtener una aproximación precisa y eficiente de la distancia humano-robot. Esta jerarquía está compuesta por tres niveles que contienen diferentes volúmenes envolventes dependiendo del nivel de detalle necesario. El primer nivel está compuesto por AABBs (Axis-Aligned Bounding Boxes) generales que cubren con un volumen envolvente todo el cuerpo del humano y todo el cuerpo del robot. El segundo nivel se compone de AABBs locales que cubren los principales miembros/eslabones del humano y del robot. El tercer nivel se compone de SSLs (Swept-Sphere Lines) que cubren cada hueso de sus esqueletos. De este modo, cuando el humano y el robot están alejados, sólo se necesita el primer nivel para calcular una buena aproximación de la distancia humano-robot. Sin embargo, cuando están cerca, se necesita el segundo o incluso el

tercer nivel. Esta representación jerárquica no sólo mejora el coste computacional del cálculo de la distancia humano-robot al reducir el número de tests de distancia entre pares de volúmenes envolventes, sino que también mejora la aproximación de las superficies del humano y del robot al utilizar los SSLs del tercer nivel de la jerarquía en comparación con los sistemas anteriores de interacción humano-robot basados en volúmenes envolventes esféricos. Además, los radios de estos SSLs se modifican dinámicamente según la velocidad lineal del hueso que es cubierto por cada SSL. Esta variación supone un incremento en la seguridad del humano ya que los volúmenes envolventes del tercer nivel de la jerarquía no sólo representan una aproximación de la superficie de los cuerpos del humano y del robot sino que también es una estimación de su desplazamiento entre dos ejecuciones de la estrategia de seguridad. Esta jerarquía de volúmenes dinámicos se usa en estrategias de seguridad que se han aplicado satisfactoriamente en varias tareas reales de interacción humano-robot.

Para el segundo requisito de tareas de interacción humano-robot flexibles, esta tesis propone un nuevo planificador de manipulación diestra basado en movimientos internos generados por una mano con múltiples dedos instalada en el efector final de los robots manipuladores que cooperan con operadores humanos. Este planificador recibe como entrada un agarre inicial de un objeto y una configuración final deseada (posición y orientación) del objeto. Este planificador calcula los movimientos de los dedos que son necesarios para llevar el objeto a la configuración final. Está construido con una estructura modular basada en dos niveles: un planificador global y uno local. El planificador global genera una trayectoria del objeto compuesta por configuraciones intermedias entre las configuraciones inicial y final. El planificador local aplica un nuevo modelo de evolución del contacto basado en una representación de las superficies del objeto y de los dedos como mallas de triángulos que permiten determinar la posible evolución de las primitivas en contacto (vértices, aristas y caras de las mallas de triángulos). Este modelo proporciona una representación más general de las superficies del objeto y de los dedos que planificadores de manipulación anteriores basados en modelos planos o paramétricos. Después, el planificador local genera movimientos de los dedos que causan dicha evolución del contacto al aplicar la pseudo-inversa de la matriz Jacobiana de cada dedo. Además, esta tesis añade un algoritmo adicional de reajuste de los dedos después de cada ejecución del planificador local para garantizar que los dedos en contacto aplican suficiente presión al objeto de tal modo que se evitan agarres inestables. Este planificador se ha aplicado satisfactoriamente en varios experimentos reales de tareas de manipulación con una mano robótica de tres dedos equipada con sensores táctiles.

Resum

El principal objectiu d'aquesta tesi és desenvolupar tasques d'interacció humà-robot en les quals operadors humans i robots manipuladors puguin cooperar en el mateix espai de treball. Per a això, aquesta tesi proposa diverses tècniques per complir dos requisits que són essencials per a aquesta interacció humà-robot: la garantia de la seguretat humana durant la interacció humà-robot i el desenvolupament de tècniques de manipulació de braços que millorin les habilitats dels robots manipuladors amb mans robòtiques.

Per al primer requisit, aquesta tesi desenvolupa noves estratègies de seguretat que paren el comportament normal del robot i activen un comportament de seguretat especial quan la distància humà-robot està per sota d'un llindar de seguretat. És necessari un sistema de seguiment de l'humà precís i per a tot el cos per calcular la distància humà-robot en temps real. Aquesta tesi fa una comparativa de les tecnologies actuals dels sistemes de captura de moviment i proposa usar un sistema inercial de captura de moviment per seguir a l'operador humà que col·labora amb el robot. Encara que les mesures rotacionals d'aquest sistema són molt precises i s'apliquen sobre un esquelet; la posició global d'aquest esquelet acumula errors. Per això, aquesta tesi usa un sistema de localització addicional basat en polsos UWB (Ultra-wideband) per corregir aquest error. Aquesta tesi presenta tres nous algorismes de fusió basats en el filtre de Bayes per combinar les mesures de posició global d'ambdós sistemes. El primer algorisme es basa en un filtre de Kalman, el segon algorisme es basa en un filtre de partícules i el tercer algorisme es basa en una combinació de un filtre de Kalman i un filtre de partícules. Tots aquests algorismes tenen una nova estructura basada en recalculer la matriu de transformació entre els sistemes de coordenades d'ambdós sistemes cada vegada que es rep una nova mesura del sistema UWB. Aquesta estructura millora el cost computacional dels algorismes proposats en comparació de les propostes similars anteriors perquè considera la complementaritat entre les característiques d'ambdós sistemes: la precisió rotacional i l'alta freqüència de mostreig del sistema inercial amb la precisió posicional i la baixa freqüència de mostreig del sistema UWB.

Aquesta tesi també proposa cobrir els ossos que componen l'esquelet registrat pel sistema de seguiment humà anterior amb una jerarquia de volums envolupants per obtenir una aproximació precisa i eficient de la distància humà-robot. Aquesta jerarquia està composta per tres nivells que contenen diferents volums envolupants dependent del nivell de detall necessari. El primer nivell està compost per AABBs (Axis-Aligned Bounding Boxes) generals que cobreixen amb un volum envolupant tot el cos de l'humà i tot el cos del robot. El segon nivell es compon de AABBs locals que cobreixen els principals membres/enllaços de l'humà i del robot. El tercer nivell es compon de SSLs (Swept-Sphere Lines) que cobreixen cada os dels seus esquelets. D'aquesta manera, quan l'humà i el robot estan allunyats, només es necessita el primer nivell per calcular una bona aproximació de la distància humà-robot. No obstant això, quan estan a prop, es necessita el segon o fins i tot el tercer nivell. Aquesta representació jeràrquica no només millora el cost computacional del càlcul de la distància humà-robot en reduir

el nombre de tests de distància entre parells de volums envolupants, sinó que també millora l'aproximació de les superfícies de l'humà i del robot en utilitzar els SSLs del tercer nivell de la jerarquia en comparació dels sistemes anteriors d'interacció humà-robot basats en volums envolupants esfèrics. A més, els radis d'aquests SSLs es modifiquen dinàmicament segons la velocitat lineal de l'os que és cobert per cada SSL. Aquesta variació suposa un increment en la seguretat de l'humà ja que els volums envolupants del tercer nivell de la jerarquia no només representen una aproximació de la superfície dels cossos de l'humà i del robot sinó que també és una estimació del seu desplaçament entre dues execucions de l'estratègia de seguretat. Aquesta jerarquia de volums dinàmics s'usa en estratègies de seguretat que s'han aplicat satisfactòriament en diverses tasques reals d'interacció humà-robot.

Per al segon requisit de tasques d'interacció humà-robot flexibles, aquesta tesi proposa un nou planificador de manipulació destra basat en moviments interns generats per una mà amb múltiples dits instal·lada en l'efector final dels robots manipuladors que cooperen amb operadors humans. Aquest planificador rep com a entrada un agarri inicial d'un objecte i una configuració final desitjada (posició i orientació) de l'objecte. Aquest planificador calcula els moviments dels dits que són necessaris per portar l'objecte a la configuració final. Està construït amb una estructura modular basada en dos nivells: un planificador global i un local. El planificador global genera una trajectòria de l'objecte composta per configuracions intermèdies entre les configuracions inicial i final. El planificador local aplica un nou model d'evolució del contacte basat en una representació de les superfícies de l'objecte i dels dits com a malles de triangles que permeten determinar la possible evolució de les primitives en contacte (vèrtexs, arestes i cares de les malles de triangles). Aquest model proporciona una representació més general de les superfícies de l'objecte i dels dits que planificadors de manipulació anteriors basats en models plans o paramètrics. Després, el planificador local genera moviments dels dits que causen aquesta evolució del contacte en aplicar la pseudo-inversa de la matriu Jacobiana de cada dit. A més, aquesta tesi afegeix un algorisme addicional de reajustament dels dits després de cada execució del planificador local per garantir que els dits en contacte apliquen suficient pressió de tal manera que s'eviten agarris inestables. Aquest planificador s'ha aplicat satisfactòriament en diversos experiments reals de tasques de manipulació amb una mà robòtica de tres dits equipada amb sensors tàctils.

Index of Contents

1 Introduction	1
1.1 Motivation	1
1.1.1 Safe Human-Robot Interaction.....	3
1.1.2 Dexterous Manipulation	4
1.2 Framework	5
1.2.1 Research Project DPI2005-06222	5
1.2.2 Research Project DPI2008-02647	6
1.3 Contributions.....	7
1.4 Structure.....	11
 2 Human Tracking for Human-Robot Interaction.....	 13
2.1 Introduction	13
2.2 Human Tracking Technologies	15
2.2.1 Mechanical Motion Capture Systems	16
2.2.2 Magnetic Motion Capture Systems	17
2.2.3 Optical Motion Capture Systems.....	18
2.2.4 Inertial Motion Capture Systems.....	21
2.2.5 Comparison of Human Tracking Systems.....	23
2.3 Architecture of the Human Tracking System	24
2.3.1 Inertial Tracking System	24
2.3.1.1 Rotational Accuracy of the Inertial Tracking System	26
2.3.1.2 Global Translational Accuracy of the Inertial Tracking System	29
2.3.2 Ultra-wideband Localization System.....	30
2.4 Conclusions	33
 3 Sensor Fusion for Human Tracking.....	 35
3.1 Introduction	35
3.2 Review of Sensor Fusion Techniques	37
3.2.1 Sensor Fusion Algorithms based on Kalman Filters	39
3.2.2 Sensor Fusion Algorithms based on Particle Filters	44
3.3 First Approach for Sensor Fusion in Human Tracking	49
3.3.1 Coordinate Transformation.....	49
3.3.2 Transformation Recalculation Algorithm	50
3.4 Bayesian Filtering for Human Tracking	52
3.4.1 Kalman Filter Fusion Approach.....	52
3.4.2 Particle Filter Fusion Approach.....	54
3.4.3 Kalman-particle Combination Approach	57
3.5 Experimental Results	59
3.5.1 Comparison between Transformation Recalculation Algorithm and Sensor Fusion based on Kalman Filter	59
3.5.2 Comparison between Sensor Fusion Algorithms based on Bayesian Filtering	61
3.5.3 Application of Sensor Fusion in Human-Robot Interaction Tasks.....	66
3.6 Conclusions	70

4 Safety Strategies for Human-Robot Interaction	73
4.1 Introduction	73
4.2 Review of Human and Robot Modeling in Cooperative Tasks	75
4.3 Hierarchy of Bounding Volumes	77
4.4 Dynamic Bounding Volume Algorithm.....	81
4.4.1 General Description of the Algorithm	81
4.4.2 Joint Angular Rate Estimation	83
4.4.3 Link Linear Velocity Computation.....	84
4.5 Safety Strategies.....	86
4.5.1 Human-Robot Distance Computation.....	86
4.5.2 Software Architecture.....	88
4.6 Experimental Results	90
4.6.1 Human-Robot Interaction Task 1: Changing a light bulb of a streetlamp	90
4.6.2 Human-Robot Interaction Task 2: Disassembly of a home appliance.....	94
4.6.3 Human-Robot Interaction Task 3: Assembly of a metallic structure	98
4.7 Conclusions	104
 5 Dexterous Manipulation for In-Hand Object Reconfiguration	107
5.1 Introduction	107
5.2 Review of Planning Techniques for Dexterous Manipulation	109
5.2.1 Manipulation Planning Strategies based on Graph Search Techniques.....	109
5.2.2 Manipulation Planning Strategies based on Hybrid Control Systems.....	112
5.2.3 Manipulation Planning Strategies based on Probabilistic Techniques.....	114
5.3 Representation of the Surfaces of the Hand and the Object	116
5.4 Geometric Manipulation Planner for In-Hand Manipulation.....	117
5.4.1 General Structure of the Geometric Manipulation Planner.....	117
5.4.2 Global Planner	119
5.4.3 Local Planner	120
5.5 Contact Evolution between Surface Primitives	122
5.5.1 Contact Evolution Graph.....	122
5.5.2 Contact Transitions Generated by Rolling.....	123
5.6 Finger Readjustment based on Tactile Information.....	125
5.7 Experimental Results	127
5.7.1 In-Hand Manipulation with Shadow Hand	127
5.7.2 In-Hand Manipulation with Barrett Hand	130
5.8 Conclusions	135
 6 Conclusions.....	137
6.1 Conclusions	137
6.2 Publications	141
6.2.1 JCR Journals.....	142
6.2.2 Book Chapters and non-JCR Journals.....	144
6.2.3 International Conferences.....	145
6.2.4 National Conferences.....	148
6.3 Future Work	149

APPENDICES

A. Modeling of the Barrett Hand	153
A.1 Introduction	153
A.2 Kinematic Model	154
A.2.1 Direct Kinematic Model	154
A.2.2 Inverse Kinematic Model	154
A.3 Dynamic Model	155
A.3.1 Inertia Matrix	156
A.3.2 Coriolis-Centrifugal Vector.....	157
A.3.3 Gravitational Vector	158
A.4 Breakaway Model	158
B. Resumen	161
B.1 Introducción	161
B.2 Resultados Obtenidos	164
B.2.1 Resultados en Estrategias de Seguridad para Interacción Humano-Robot.....	164
B.2.2 Resultados en Manipulación Robótica.....	167
B.3 Discusión de los Resultados.....	169
B.3.1 Discusión de los Resultados en Seguridad para Interacción Humano-Robot.....	169
B.3.2 Discusión de los Resultados en Manipulación Robótica	171
B.4 Conclusiones	172
References	175

Index of Figures

Figure 2.1. Mechanical motion capture systems: (a) *Animazoo Gypsy-7* mechanical system based on exoskeleton, Source: [Animazoo-2011a]; (b) action parsing and goal inference system based on the *Gypsy* system for collaborative human-robot tasks, Source: [Gray-2005]; (c) movement imitation system based on *Sarcos Sensuit*, Source: [Ijspeert-2002].16

Figure 2.2. Magnetic motion capture systems: (a) *Ascension MotionStar* system, Source: [Ascension-2011]; (b) *Polhemus Liberty* system, Source: [Polhemus-2011]; (c) hand motion capture system, Source: [Mitobe-2006].18

Figure 2.3. Optical motion capture systems based on passive markers: (a) passive markers and (b) camera of the *Vicon MX* system, Source: [Vicon-2011]; (c) online imitation of human motion by a humanoid robot, Source: [Montecillo-Puente-2010]...19

Figure 2.4. Optical motion capture systems based on active markers: (a) active marker and (b) camera of the *PhaseSpace Impulse* system, Source: [PhaseSpace-2011]; (c) physical motion analysis system in driving based on high intensity LEDs, Source: [Hanai-2009].19

Figure 2.5. Markerless optical motion capture systems: (a) *iPi Desktop* software, Source: [iPiSoft-2011]; (b) camera of the *Xbox Kinect* system, Source: [Microsoft-2011]; (c) action recognition and imitation of human motion, Source: [Azad-2007].20

Figure 2.6. Inertial motion capture systems: (a) *Xsens MTx* IMU, Source: [Xsens-2011a], (b) *Xsens MVN motion capture* suit and (c) Teleoperation system for the *Robonaut* robot, Source: [Miller-2004]; (d) real-time teaching system for novice violinists based on the *Animazoo IGS-190-M* inertial suit, Source: [Linden-2009].22

Figure 2.7. Components of the inertial motion capture system: (a) hardware architecture; (b) skeleton.25

Figure 2.8. Experimental setup for rotational accuracy analysis of the inertial motion capture system: (a) Mitsubishi PA-10 robotic manipulator; (b) InertiaCube3 IMU over a graduated rotating disk.....27

Figure 2.9. Areas of magnetic influence of the PA-10 over the inertial motion capture system: (a) spheres defined around the joint motors; (b) 3D surface around the manipulator.29

Figure 2.10. Global translational error of the motion capture system in three different paths.29

Figure 2.11. Advantages of the UWB technology: (a) immunity to multipath fading; (b) immunity to interferences.....31

Figure 2.12. Hardware components of the *Ubisense* system: (a) sensor (front and side views); (b) tag.32

Figure 2.13. <i>Ubisense</i> hardware architecture: (a) interconnection of sensors; (b) installation in workplace.	32
Figure 3.1. Diagram of the operation of the Kalman filter based on two steps (prediction and correction).	42
Figure 3.2. Multi-sensor adaptations of the Kalman filter: (a) complementary Kalman filter; (b) Kalman filter with multiple correction channels.	43
Figure 3.3. Sensor fusion systems for human tracking based on Kalman filters: (a) navigation system <i>NavShoe</i> , Source: [Wan-2010]; (b) wearable Augmented Reality kit, Source: [Ribo-2002].	43
Figure 3.4. Evolution of the different particle sets generated during the execution of the SIR particle filter.	46
Figure 3.5. Sensor fusion systems for human tracking based on particle filters: (a) arm tracking system for rehabilitation, Source: [Tao-2008]; (b) full body tracking with inertial sensors; Source: [Zhang-2009].	48
Figure 3.6. Coordinate systems of the tracking systems (UWB system and inertial motion capture system).	49
Figure 3.7. Diagram of the fusion algorithm based on transformation recalculation.	51
Figure 3.8. Linear trajectory experiments: (a) original position measurements in experiment 14; (b) position measurements obtained from the transformation recalculation algorithm in experiment no. 14; (c) position estimates obtained from the prediction and correction steps of the Kalman filter algorithm in experiment no. 14; (d) mean error and standard deviation of the position estimates from the two fusion algorithms in the 15 linear path experiments.	60
Figure 3.9. Rectangular trajectory experiments: (a) original position measurements from both tracking systems in experiment no. 3; (b) position measurements obtained from the transformation recalculation algorithm in experiment no. 3; (c) position estimates obtained from the prediction and correction steps of the Kalman filter algorithm in experiment no. 3; (d) mean error and standard deviation of the position estimates from the two fusion algorithms in the 15 rectangular path experiments.	61
Figure 3.10. Linear trajectory experiments: (a) original position measurements in experiment 10; (b) position estimates from the Kalman filter algorithm in the prediction and correction steps; (c) position estimates from the particle filter algorithm in the prediction and resampling steps; (d) position estimates from the Kalman-particle filter algorithm in the Kalman filter (KF) and the particle filter (PF) steps.	62
Figure 3.11. Mean error and standard deviation of the position estimates from the three fusion algorithms (KF: Kalman filter, PF: Particle filter, KF-PF: Kalman-particle filter) in the linear path experiments.	63
Figure 3.12. Computational cost of the linear path experiments: (a) number of measurements registered by the two tracking systems (inertial and UWB); (b) total	

execution time of the three fusion algorithms (KF: Kalman filter, PF: Particle filter, KF-PF: Kalman-particle filter).63

Figure 3.13. Rectangular trajectory experiments: (a) original position measurements in experiment 7; (b) position estimates from the Kalman filter algorithm in the prediction and correction steps; (c) position estimates from the particle filter algorithm in the prediction and resampling steps; (d) position estimates from the Kalman-particle filter algorithm in the Kalman filter (KF) and the particle filter (PF) steps.64

Figure 3.14. Mean error and standard deviation of the position estimates from the three fusion algorithms (KF: Kalman filter, PF: Particle filter, KF-PF: Kalman-particle filter) in the rectangular path experiments.65

Figure 3.15. Computational cost of the rectangular path experiments: (a) number of measurements registered by the two tracking systems (inertial and UWB); (b) total execution time of the three fusion algorithms (KF: Kalman filter, PF: Particle filter, KF-PF: Kalman-particle filter).65

Figure 3.16. Experimental setup for the development of the proposed human-robot interaction tasks.66

Figure 3.17. Frame sequence of a human-robot interaction task where a human operator gives an object to the robot. Each frame is composed by a photograph of the real task on the left and a 3D representation of the human operator's skeleton on the right.68

Figure 3.18. Frame sequence of a human-robot interaction task where a human operator collaborates with a robot in the disassembly of a metallic structure. Each frame is composed by a photograph of the real task on the left and a 3D representation of the human operator's skeleton on the right.69

Figure 4.1. Most commonly used bounding volumes.75

Figure 4.2. Bounding volumes in robotic applications: (a) safety strip of spheres that cover a robotic manipulator during a cleaning task of an I-beam of a bridge, Source: [Xu-2007b]; (b) spherical approximation of a persona and two manipulators that collaborate in a task, Source: [Martinez-Salvador-2003].....76

Figure 4.3. Model of the human operator: (a) Human operator with the tracking systems; (b) links/skeleton of the human; (c) SSL bounding volumes which cover the links of the human.78

Figure 4.4. Model of the robotic manipulator: (a) PA-10 manipulator; (b) links/skeleton of the robot; (c) SSL bounding volumes which cover the links of the robot.79

Figure 4.5. Hierarchy of bounding volumes which cover the body of the human operator.....80

Figure 4.6. 3D representation of the hierarchy of bounding volumes for the human operator: (a) Global AABB (level 1); (b) local AABBs (level 2); (c) SSLs (level 3).....80

Figure 4.7. Hierarchy of bounding volumes which cover the structure of the robotic manipulator.81

Figure 4.8. 3D representation of the hierarchy of bounding volumes for the robotic manipulator: (a) Global AABB (level 1); (b) local AABBs (level 2); (c) SSLs (level 3).....	81
Figure 4.9. Diagram of the dynamic bounding volume algorithm based on the linear velocities of the links.....	83
Figure 4.10. Evolution of the angular rate of the first joint of the robotic manipulator in a human-robot interaction task: (a) finite difference method; (b) adaptive windowing method (with $\varepsilon = 0.3$ rad).	84
Figure 4.11. Components of the software architecture which implements the human-robot interaction behavior.....	89
Figure 4.12. Experimental setup for the light bulb replacement task.....	91
Figure 4.13. Phases of the disassembly task of a streetlamp for bulb replacement: (a) Disassembly of the streetlamp; (b) transportation of the blown light bulb; (c) collaboration of the human operator; (d) installation of the new light bulb.	92
Figure 4.14. Evolution of the trajectories during the task: (a) Position of the robot's end-effector during the original trajectory; (b) position of the end-effector when the safety behavior is activated; (c) Minimum human-robot distance; (d) 3D error between the real position and the desired position of the end-effector.....	93
Figure 4.15. Sequence of frames of phases 2 and 3 of the task when the human enters the workplace and leaves the new light bulb over the turning table.....	94
Figure 4.16. Experimental setup for the fridge disassembly task.....	95
Figure 4.17. Sequences of frames of the fridge disassembly task. Each frame is composed by a photograph of the real experiment and a 3D representation of the bounding volumes of the third level of the hierarchy.....	96
Figure 4.18. (a) Evolution of the human-robot distance; (b) Evolution of the distance error from the hierarchy of bounding volumes.....	97
Figure 4.19. Histogram of the number of distance tests required for the minimum human-robot distance computation.	98
Figure 4.20. (a) components of the metallic structure which are used during the assembly task; (b) experimental setup of the metallic structure assembly task.....	99
Figure 4.21. Components which are assembled to the metallic structure during the three phases of the assembly task.....	99
Figure 4.22. Sequence of frames of the metallic structure assembly task. Each frame is composed by a photograph of the real experiment and a 3D representation of the dynamic bounding volumes of the third level of the hierarchy.....	101
Figure 4.23. Evolution of the radii of the bounding volumes which cover the last three links (E2, W1 and W2) of the first robotic manipulator.....	102
Figure 4.24. Evolution of the radii of the bounding volumes which cover (a) the left foot and (b) the right hand of the human operator.	102

Figure 4.25. Evolution of the minimum human-robot distance during the metallic structure assembly task.	103
Figure 4.26. (a) Evolution of the distance error from the hierarchy of bounding volumes; (b) evolution of the difference between computing the distance with and without bounding volumes.....	103
Figure 4.27. Histogram of the number of distance tests required for the minimum human-robot distance computation.	104
Figure 5.1. (a) sequence of CGs for manipulating a rectangular object; (b) the same CGs with a real hand, Source: [Zhang-1996].....	110
Figure 5.2. Manipulation of a sphere by three spherical fingers with finger relocation, Source: [Vass-2005].....	111
Figure 5.3. (a) Manipulation of an egg-shaped object with four fingers, Source: [Goodwine-1999]; (b) reorientation of a circle by three linear fingers, Source: [Xu-2008].	113
Figure 5.4. (a) Reconfiguration of an ellipsoidal object with three planar fingertips, Source: [Yashima-2003]; (b) reorientation of a sphere with a four-fingered hand including regrasping, Source: [Saut-2006].	115
Figure 5.5. Triangle mesh representation of the used multi-fingered robotic hands: (a) <i>Shadow</i> hand; (b) <i>Barrett</i> hand.....	117
Figure 5.6. Diagram of the general structure of the developed geometric manipulation planner.	118
Figure 5.7. Control law of each fingertip to steer it towards the desired position.	122
Figure 5.8. Contact evolution graph for new contact states due to rolling.....	122
Figure 5.9. Contact transition from contract state e_o / f_{f_i} to contact state f_o / f_{f_i}	123
Figure 5.10. Contact transition from contract state f_o / v_{f_i} to contact state f_o / f_{f_i}	124
Figure 5.11. Diagram of the general structure of the developed dexterous manipulation planner after including the finger readjustment algorithm between the local and global levels of the geometric planner.	126
Figure 5.12. Simulator software of the <i>Shadow</i> hand with a window for updating joint values (left) and a panel with a 3D visualization of the hand and the object loaded in the environment (right).....	127
Figure 5.13. Simulation of translation manipulation tasks of a cube: (a) linear trajectory in the X axis; (b) linear trajectory in the Y axis; (c) linear trajectory in the Z axis. First frame is above and last frame is below.	128
Figure 5.14. Simulation of rotation manipulation tasks: (a) rotation of a hexagonal prim around the Y axis; (b) rotation of a icosphere round the Z axis. First frame is above and last frame is below.....	129

Figure 5.15. Experimental setup for dexterous manipulation planning: (a) <i>Barrett</i> hand with 7 tactile sensors; (b) 3D representation of the pressure measurements registered by the arrays of tactile sensors.	130
Figure 5.16. Software simulator of the Barret hand where the geometric planner has been implemented.....	131
Figure 5.17. Sequence of frames of a manipulation task of a cylinder by applying the geometric planner with finger readjustements.....	132
Figure 5.18. Evolution of the manipulation task of a cylinder: (a) Evolution of the motor counts of each finger; (b) evolution of the maximum contact pressure registered by the tactile sensors for each finger.....	132
Figure 5.19. Sequence of frames of a manipulation task 1 of a sphere by applying the geometric planner with finger readjustements.....	133
Figure 5.20. Evolution of the manipulation task 1 of a sphere: (a) Evolution of the motor counts of each finger; (b) evolution of the maximum contact pressure registered by the tactile sensors for each finger.....	133
Figure 5.21. Sequence of frames of the manipulation task 2 of a sphere by applying the geometric planner with finger readjustements.....	134
Figure 5.22. Evolution of the manipulation task 2 of a sphere: (a) Evolution of the motor counts of each finger; (b) evolution of the maximum contact pressure registered by the tactile sensors for each finger.....	134
Figure A.1. Coordinate frames assigned to each finger of the <i>Barrett</i> hand.	154
Figure A.2. Geometric relations of the links of a finger.	155

Index of Tables

Table 2.1. Comparison between the motion capture technologies.....	23
Table 2.2. Placement of IMUs in the <i>GypsyGyro-18</i> system.....	25
Table 2.3. Statistical evolution of error in yaw angle with regard to distance with motor S1.....	28
Table 2.4. Statistical evolution of error in yaw angle with regard to distance with motor E1.....	28
Table 2.5. Statistical evolution of error in yaw angle with regard to distance with motor E2.....	28
Table 2.6. Statistical evolution of the global translational error of the motion capture system.....	30
Table 3.1. Comparison of the two tracking systems: inertial system and UWB system.	36
Table 3.2. Algorithm of the SIR particle filter.	47
Table 3.3. Fusion algorithm based on Kalman filter.....	54
Table 3.4. Systematic Resampling Algorithm.	56
Table 3.5. Fusion algorithm based on SIR particle filter.....	56
Table 3.6. Fusion algorithm based on the combination of a Kalman filter and a SIR particle filter.	58
Table 4.1. Comparison of the space usage (volume) of the bounding volumes based on SSLs and spheres with regard to the real volumes of the links of a <i>PA-10</i> robot.	78
Table 4.2. Minimum distance algorithm based on the three-level hierarchy of bounding volumes.....	87
Table 5.1. Evolution of execution time of the local planner depending on the complexity of the mesh.....	129
Table A.1. Denavit-Hartenberg parameters for each finger k where $r = [1, 1, 0]$ for $k = [1, 2, 3]$	154
Table A.2. Dimension parameters of the links of the <i>Barrett</i> hand.....	154
Table A.3. Mass and inertia parameters of the links of the <i>Barrett</i> Hand.	156
Table A.4. Algorithm for the detection of the activation of the breakaway mechanism.....	159

Index of DVD

This thesis includes a DVD with the videos of the experiments and the publications developed in this thesis. This DVD contains the following videos:

- *Video_3.1.avi*: This video shows a human-robot interaction task where a human operator takes an object from a storage box which is outside the robot's workspace and gives this object to a robotic manipulator. This video not only contains the real sequence of the experiment but also a 3D representation of the skeleton of the human in order to verify that the fusion algorithm developed in [Chapter 3](#) is able to track on real-time the movements of the human. More information can be read at [Section 3.5.3.1](#).
- *Video_3.2.mpg*: This video shows a human-robot interaction task where a human operator cooperates with a robotic manipulator in the disassembly of a connector from a metallic structure. This video shows the real sequence of the experiment. More information about this experiment and the data obtained from the human tracking system can be read at [Section 3.5.3.2](#).
- *Video_4.1.mpg*: This video shows an execution of a task which involves changing a blown light bulb of a streetlamp with a robotic manipulator. This video shows the original trajectory which is executed by the robot in order to leave the blown light bulb inside a storage box when there is no human intervention. More information can be read at [Section 4.6.1](#).
- *Video_4.2.mpg*: This video shows an execution of a task which involves changing a blown light bulb of a streetlamp with a robotic manipulator. This video shows how the original trajectory of the robot is changed because of the safety behavior activation when the human is near the robot. It also shows a 3D representation of the skeletons of the human operator and the robot and a plot with the evolution of the error in the robot's trajectory when the safety behavior is activated. More information can be read at [Section 4.6.1](#).
- *Video_4.3.avi*: This video shows an execution of an assembly task of a metallic structure where a human operator collaborates with two robotic manipulators. This video not only shows the sequence of the real experiment but also a 3D representation of the dynamic bounding volumes which are generated in the third level of the hierarchy of bounding volumes. More information can be read at [Section 4.6.3](#).
- *Video_5.1.avi*: This video shows a simulation of a dexterous manipulation task of a cube by the *Shadow* hand. This task involves a displacement of the cube by 20mm in the X axis. More information can be read at [Section 5.7.1](#).

- *Video_5.2.avi*: This video shows a simulation of a dexterous manipulation task of a hexagonal prism by the *Shadow* hand. This task involves a rotation of the prism by 5° around the Y axis. More information can be read at [Section 5.7.1](#).
- *Video_5.3.avi*: This video shows a simulation of a dexterous manipulation task of an icosphere by the *Shadow* hand. This task involves a rotation of the sphere by 5° around the Z axis. More information can be read at [Section 5.7.1](#).
- *Video_5.4.mpeg*: This video shows a dexterous manipulation task of a cylinder by a *Barrett* hand. This task ends when the maximum pressure of finger 3 is below the safety threshold. This video shows two different views of the real task and three plots: the evolution of the finger joints, the 3D representation of the pressure values registered by the tactile array and the evolution of the maximum pressure registered by the tactile sensors of each finger. More information can be read at [Section 5.7.2](#).
- *Video_5.5.mpeg*: This video shows a dexterous manipulation task of a sphere by a *Barrett* hand. This task ends when the maximum pressure of finger 2 is below the safety threshold. This video shows two different views of the real task and three plots: the evolution of the finger joints, the 3D representation of the pressure values registered by the tactile array and the evolution of the maximum pressure registered by the tactile sensors of each finger. More information can be read at [Section 5.7.2](#).
- *Video_5.6.mpeg*: This video shows a dexterous manipulation task of a sphere by a *Barrett* hand. This task ends when the object reaches the desired final configuration. This video shows two different views of the real task and three plots: the evolution of the finger joints, the 3D representation of the pressure values registered by the tactile array and the evolution of the maximum pressure registered by the tactile sensors of each finger. More information can be read at [Section 5.7.2](#).

This DVD also contains a folder, called *Publications*, which contains all the pdf files of the accepted publications developed in this thesis (see [Section 6.2](#) for more details). The pdf file of this thesis can be found in the main folder of the DVD.

Acknowledgements

First of all, I would like to thank my supervisors, Prof. Fernando Torres Medina and Dr. Francisco Andrés Candelas Herías, for their continuous guidance and revision of my research work along the development of this thesis. I am very grateful for their advice and their suggestions for developing new research lines in this thesis.

Secondly, I would like to thank all the members of the *Automatics, Robotics and Computer Vision* (AUROVA) research group of the *University of Alicante*: Dr. Jorge Pomares Baeza, Dr. Pablo Gil Vázquez, Dr. Santiago Timoteo Puente Méndez, Dr. Gabriel Jesús García Gómez, Dr. Carlos Alberto Jara Bravo and Iván Perea Fuentes. All of them have helped me very much with their advice, their support and their comments about my research in order to find new paths for solving the problems arisen during the development of this thesis. I would like to thank Dr. Gabriel Jesús García Gómez for his invaluable help not only with many of the experiments developed in this thesis but also with its formatting. I would also like to thank other colleagues who have worked at the AUROVA laboratory and at the *University of Alicante* during the development of this thesis: María José Blanes Payá, Juan Carlos Molina Giménez, Dr. Carolina Soledad Díaz, Manuel Domenech Beltrán and Gonzalo Lorenzo Lledó. It has been a pleasure working with you and sharing so many hours. I would also like to thank the members of the *Physics, Systems Engineering and Signal Theory Department* of the *University of Alicante* for their academic and administrative support along this thesis.

Thirdly, I would like to express my gratitude to Prof. Véronique Perdereau for making possible my research stay at the *University Pierre et Marie Curie* (UPMC) and helping me with the research on dexterous manipulation. I would like to thank some people who have helped me very much during my research stay at the UPMC: Prof. Giovanni Legnani, Dr. Guillaume Walck, Kien Cuong Nguyen, Hoang Lan Pham, Dr. David Flavigné, Urbain Prieur, Ella Bouquet and Brigitte Cord.

I would also like to thank my family (Silvia Ramón Castelló, Antonio Corrales Requena and Encarnación Castelló Caparrós) for their support. They have always encouraged me to overcome all the problems and difficulties which have appeared during the development of this thesis.

Finally, I would like to thank some institutions whose financial support has made this thesis possible. I would like to thank the Valencian Government for its support by the research project “Infraestructura 05/053”. I would like to thank the Spanish Ministry of Science and Innovation for its support by the pre-doctoral grant AP2005-1458 and the research projects DPI2005-06222 and DPI2008-02647, which constitute the research framework of this thesis.

THANKS TO EVERYONE.

Alicante, June 2011,

Juan Antonio Corrales Ramón

Introduction

This first chapter constitutes a general introduction to this thesis. First to all, it presents the main motivations which have favored the conception and the development of this thesis. In particular, the main goal of this thesis is the development of human-robot interaction tasks where human operators can collaborate with robotic manipulators. In order to make this vision real, two requirements are needed: safety strategies which avoid collisions between human operators and robots and dexterous manipulation techniques which enable robots to manipulate correctly the objects that are shared with humans. These two requirements are the two main lines of research of this thesis. This chapter also explains the research framework where the research of this

1.1 Motivation	1
1.1.1 Safe Human-Robot Interaction	3
1.1.2 Dexterous Manipulation	4
1.2 Framework	5
1.2.1 Research Project DPI2005-06222	5
1.2.2 Research Project DPI2008-02647	6
1.3 Contributions.....	7
1.4 Structure.....	11

thesis has been developed. It explains the goals of the two research projects where the results of this thesis have been implemented. Next, this chapter enumerates the main contributions of this thesis, which will be described and justified in detail in the subsequent chapters. Finally, a summary of the chapter organization of this thesis is developed and the main points of each chapter are explained briefly.

1.1 Motivation

Robotic manipulators are more and more widespread in current industries [IFR-2008] due to their positional precision, repeatability and durability. They are usually applied in repetitive assembly tasks where all the components are always in the same position and only a position control of the robot is needed. Nevertheless, the application of robotic manipulators which can react to changes in their environment may improve the productivity of the current industrial processes. In particular, context-aware robotic manipulators do not need to be completely isolated from the rest of the production line [Kulic-2004] and thus the cost of the manufacturing process can be reduced not only in terms of space (by removing the current fenced robotic cells) but also in terms of time (by redefining the production processes so that much more subtasks can be performed simultaneously). In order to develop this context-awareness, robotic manipulators are gradually incorporating additional sensors which register important information from the environment in order to adapt their behavior according to the changes of the environment. For instance, cameras and force sensors are usually installed at the end-effector of the robot [García-2009b] in order to determine the trajectory which has to be tracked by the robot depending on the evolution of the visual features registered by the camera (i.e. visual servoing techniques) and the contact forces/torques detected by the force sensor (i.e. force control).

This context-awareness of robotic manipulators enables the participation of new agents during the production such as human operators who can develop those subtasks which cannot be completed by robots. In fact, robots and humans present complementary features for the development of common tasks. On one hand, the robot completes those subtasks which are exhausting or dangerous for the human. On the other hand, the human performs the subtasks which cannot be executed by the robot because of their complexity. For instance, humans are able to perform specialized tasks which require intelligence and dexterity. This synergy between humans and robots enables the development of more flexible and complex tasks which cannot be performed individually by a human or a robot [Green-2008]. Furthermore, the application of human-robot interaction in industrial environments can also improve the productivity of the system because the robot can continue its normal work while human operators are inside the workcell [Ebert-2005]. Thereby, industrial tasks can be improved substantially by making humans and robots collaborate in the same workspace. The main goal of this thesis is fulfilling several requirements which are necessary to transform this vision of human and robotic manipulators working together into reality.

The first requirement which should be fulfilled is to guarantee the safety of human operators who collaborate with robotic manipulators. This safety requirement is particularly imperative in human-robot interaction systems where humans cooperate with industrial robots because these robots usually have large dimensions, are heavy and operate at high velocities. Therefore, collisions between humans and industrial robotic manipulators may be very dangerous for humans and they must be completely avoided.

The second requirement which should be fulfilled is to enable the handling of objects between operators and robotic manipulators [Edsinger-2007]. One of the most common applications of industrial robots is the assembly or disassembly of products. This assembly/disassembly process usually entails the manipulation of parts of the product in order to insert or remove them from a specific place of the product. In automatic assembly tasks without human-robot interaction, these parts are always placed at the same position and orientation so that a simple position control of the end-effector of the robot is sufficient to perform the task. When human-robot interaction takes place, the participation of the human introduces more uncertainties about the manipulated object which require increasing the flexibility of the robot's manipulation skills. For instance, when the human gives a part to the robot, this part could be orientated differently from its final configuration in the product. This situation makes the robot change the part's orientation from the initial value to the desired one. Another typical situation which requires an adjustment of the manipulation of an object is when the robot has to give several objects to a human but the human is situated in different positions each time. In this case, the robot has also to orientate the object in the best configuration to be grasped by the human.

Both requirements are necessary in order to develop human-robot interaction tasks which increase the flexibility of industrial systems composed by robotic manipulators. In the following sections, these two requirements are analyzed in detail in order to establish the different techniques which should be designed and implemented in this thesis in order to enable the development of real human-robot interaction tasks.

1.1.1 Safe Human-Robot Interaction

As stated in the previous section, human safety is an essential requirement for any human-robot interaction system [Bicchi-2008]. The traditional solution to the safety problem is based on isolating the robot with physical barriers such as fences or light curtains (i.e. opto-electronic sensors) which stop the robot immediately when crossed by a human. The first robot safety standards, such as ISO-10218:1992 [ISO-1992], were based on this paradigm. The segregation between robots and humans avoids any risk of collision but limits extremely the flexibility of the task. In fact, this safety solution cannot be applied in tasks where the human and the robot must share the same physical workspace or in tasks where the task completion requires collaboration.

These drawbacks have triggered the development of new safety standards which permit the coexistence of human and robots in the same workspace under certain circumstances [Lafvas-2007]. For instance, the standard ISO-10218-1:2006 [ISO-2006], which constitutes the first part of the revision of the previous standard ISO-10218:1992, defines a group of new collaborative operation modes where purposely designed robots work in direct cooperation with humans but always respecting hard limits of maximum velocity (250 mm/s) and static force (150 N). While the standard ISO-10218-1:2006 is based on the design and construction of the robot itself and is useful for robot manufacturers; a new standard ISO-10218-2:2011 [ISO-2011], which is now under development and constitutes the second part of the revision of ISO-10218:1992, will determine the safety requirements for the installation of robotic cells. These standardization efforts show the increasing interest of industries in human-robot collaboration.

Several approaches have been developed in previous research in order to implement safety in human-robot interaction tasks. The first approach is based on redesigning mechanically the robot so that it does not cause any injury to humans in case there is a collision. Generally, this safe robot designs include using a whole-body visco-elastic covering for the robot [Lim-1999] [Yamada-1997] and/or mechanical compliant joints based on cable transmissions, springs or even with flexible links [Luca-2000]. Nevertheless, these compliant transmissions are rarely applied in industrial manipulators because they cause slow responses and increased oscillations that reduce the position accuracy and the velocities of the robot. In addition, visco-elastic covers are useless for heavy industrial robots whose inertias are too high because collisions with humans continue to be too risky. Other solution for human-robot collisions is motion controllers that decrease the contact forces when a collision with the human is detected by force/torque sensors installed at the end-effector and/or at the joints of the robot [Lu-2005] [Heinzmann-2003]. All these techniques are post-collision strategies that reduce the effects of collision but they do not avoid them. Nevertheless, in industrial environments where human collaborate with robots, the collisions should always be avoided because they can disturb the human operator while he/she is performing the task.

Therefore, the implementation of human-robot interaction tasks requires pre-collision planners which modify the trajectories of robotic manipulators depending on the distance to human operators. When this distance is too small, the planner should change the trajectory of the robot in order to avoid any risk of collision. In addition, since these systems need to track the movements of these human operators in order to compute the human-robot distance, more flexible interaction tasks can be performed

by employing this context information. Thus, human-robot interaction tasks become more natural and unobtrusive because trajectories of the robots are dynamically changed depending on the human operators' movements.

1.1.2 Dexterous Manipulation

As described in the previous sections, the manipulation of objects by robotic manipulators is another important requirement to be fulfilled for the development of flexible human-robot interaction tasks. Most human-robot interaction systems are based on simple grasping strategies where robotic manipulators perform pick-and-place operations by restraining completely the object with simple grippers based on parallel jaws. Nevertheless, in the last years, multi-fingered robotic hands have been more and more used as end-effectors of robotic manipulators [Yoshikawa-2010]. Multi-fingered robotic hands have two important features in order to replicate the manipulating skills of human beings: dexterity and anthropomorphism [Melchiorri-2008]. Dexterity denotes the capability of the end-effector to autonomously perform tasks with a certain level of complexity while anthropomorphism denotes the capability of a robotic end-effector to mimic the human hand in terms of shape, size, consistency and general aspect. Both features increase the flexibility of multi-fingered hands in comparison with traditional parallel grippers. In fact, multi-fingered robotic hands are able not only to perform grasping tasks of objects with complex shapes but also to complete manipulation tasks which apply controlled movements to the manipulated objects.

These manipulation tasks that can be performed by robotic multi-fingered hands are divided into three different types depending on their main goal [Han-1998]:

- *Object manipulation:* This type of manipulation task involves achieving a desired object configuration (i.e. position and orientation of the object) without considering contact configurations (i.e. relative positions of the contact points over the surfaces of fingers and the object). Thereby, this manipulation task is usually based on making the object evolve towards the desired configuration by movements of the fingers but with fixed contact points.
- *Grasp adjustment:* This type of manipulation task involves obtaining a desired contact configuration without considering the object configuration. This manipulation task is based on changing the contact points with the aim of improving the grasp quality or generating new grasps when the fingers reach their kinematic limits (i.e. finger gaiting). In this manipulation task, the configuration of the object is unimportant because its main goal is assuring a stable grasp.
- *Dexterous manipulation:* This type of manipulation task involves achieving a goal configuration for the object and the contacts. This is the most general manipulation task since it is based on moving the object towards a desired configuration but also including changes over the contact points (such as rolling or sliding).

In the case of human-robot interaction tasks, the reconfiguration of the manipulated object is usually required for the development of assembly tasks where the object

has to be placed in the assembly in a specific configuration or for the development of handling tasks where the object has to be given to the human in a configuration easy to be grasped. Therefore, these manipulation tasks for human-robot interaction are not based on grasp adjustment but on object manipulation or dexterous manipulation. However, the kinematic limitations of multi-fingered hands make difficult the application of object manipulation tasks since they usually require moving the fixed contact points to unreachable positions for the fingers [Han-1998]. To sum up, dexterous manipulation [Okamura-2000] becomes the most general and flexible solution for implementing manipulation tasks for human-robot interaction because it not only permits to modify the configuration of the object to a desired value but it also permits to modify the contact points (i.e. rolling and/or sliding) so that the kinematic limitations of multi-fingered hands are reduced.

1.2 Framework

The research described in this thesis has been developed in the framework of two different research projects. Firstly, the author of this thesis have collaborated with the research project DPI2005-06222 as a research technician from 22-03-2006 to 30-04-2006 and as a Ph.D. student funded by the Spanish Ministry of Education and Science through the pre-doctoral grant AP2005-1458 from 01-05-2006 to 30-12-2008. Afterwards, the author has collaborated with the research project DPI2008-02647 as a Ph.D. student funded by the grant AP2005-1458 from 01-01-2009 to 31-04-2010 and as a research assistant from 13-05-2010 till now. The author has also performed a research stay at the Institut des Systèmes Intelligents et de Robotique (ISIR) of the University Pierre et Marie Curie (Paris, France) from 05-10-2010 to 08-01-2011. All of these collaborations constitute the experimental framework where the research techniques developed in this thesis are applied. The following sections explain the main objectives of each of these projects and their relation with this thesis.

1.2.1 Research Project DPI2005-06222

The project DPI2005-06222, entitled “Design, Implementation and Experimentation of Intelligent Manipulation Scenarios for Automatic Assembly and Disassembly Applications”, is a Spanish research project funded by the Spanish Ministry of Education and Science which has been developed at the University of Alicante (San Vicente del Raspeig, Spain) from 31-12-2005 to 30-12-2008. This project is formed by a group of 9 researchers and its principal researcher is Prof. Fernando Torres Medina.

The main goal of this project is the design, implementation and experimentation of a new intelligent scenario with specialized sensors where complex manipulation tasks (such as assembly and disassembly) can be performed autonomously by a multi-robot system and with collaboration of human operators. The scientific and technical objectives of this project are the following ones:

- *Design and implementation of an intelligent scenario:* This scenario integrates several sensors not only to control the movements of two robots (such as cameras and force sensors), but also to track the movements of human operators who

collaborate in the developed tasks (such an inertial motion capture system and a UWB localization system).

- *Multi-sensor robotic techniques:* New techniques of visual servoing and force control are developed in order to control the interaction of robotic manipulators with unknown objects in unstructured environments.
- *Computer vision techniques for object processing:* New algorithms of computer vision have been developed in order to remove shines and shadows in images of objects and to detect occlusions. Thereby, the objects to be manipulated by the robots can be correctly segmented.
- *Complex manipulation algorithms and human-robot collaboration:* New algorithms for grasping complex objects with occluded parts are developed. In addition, manipulation tasks are improved by including the collaboration of human operators.
- *Experimentation of the previous results:* All the techniques described in the previous paragraphs have been applied in real applications of automatic assembly and disassembly.

The research about human-robot interaction developed in this thesis is directly related with the fourth objective of this research project. In fact, as described above, the guarantee of the human safety is an important requirement for the development of the human-robot collaboration tasks which are implemented in this project. The safety strategies proposed in this thesis use the measurements from the sensors of the intelligent scenario implemented in this project in order to track the movements of the human operator and thus, stop the normal behavior of the robot if it is too close to the human. In addition, the experiments which have been performed to verify the performance of the safety strategies developed in this thesis are part of the assembly and disassembly tasks of this project.

1.2.2 Research Project DPI2008-02647

The project DPI2008-02647 entitled “Dexterous Manipulation through Haptic Perception and visual servoing by using an articulated structure over a robotic manipulator”, is a Spanish research project funded by the Spanish Ministry of Education and Science which is being developed at the University of Alicante (San Vicente del Raspeig, Spain) from 01-01-2009 to 31-12-2011. This project is formed by a group of 7 researchers and its principal researcher is Prof. Fernando Torres Medina.

The main goal of this project is the implementation of advanced manipulation techniques where a mini-robot with a camera is installed at the end-effector of the robotic manipulator in order to guarantee that the manipulated object is always in the field of view of the camera. Therefore, the movements of the robotic manipulator which has to execute the manipulation task can be adapted to the configuration of the object to be manipulated. The scientific and technical objectives of this project are the following ones:

- *Design and construction of a mini-robot with a camera:* This mini-robot with 3 rotational joints is installed at the end-effector of the robot so that it is able to focus its camera towards the object to be manipulated. This configuration avoids occlusions which may happen during the development of a manipulation task.
- *Multi-sensor robotic techniques for manipulation:* New techniques of visual servoing, force control and tactile control are developed in order to control the movements of the robotic manipulator during the execution of manipulation tasks.
- *Algorithms for dexterous manipulation planning:* The robotic manipulator is equipped with a three-fingered robotic hand at its end-effector in order to execute manipulation tasks. New techniques for performing in-hand movements of the grasped object are developed. These techniques will be extended in future to deformable objects and objects with variable center of mass (such as containers with liquid).
- *Object Recognition:* A RFID (Radio-Frequency Identification) system is applied in order to recognize the objects that are present in the environment. This recognition is used to inform the manipulation system about the objects that can be manipulated.

The research about dexterous manipulation developed in this thesis is directly related with the third objective of this research project. As described above, dexterous manipulation is an important requirement which has to be fulfilled in order to develop flexible human-robot interaction where human operators and robots share objects. This thesis is focused on the development of a dexterous manipulation planner which permits to modify the configuration of the grasped object by controlled in-hand movements. This in-hand manipulation is very useful for reorienting objects in order to verify the restrictions of the developed task and is based on the representation of the fingers' and object's surfaces as triangle meshes. The author of this thesis has developed this geometric dexterous manipulation planner during his research stay at the University Pierre et Marie Curie. This planner has been applied on a software simulator of a multi-fingered robotic hand (*Shadow* hand). Afterwards, a finger readjustment technique based on tactile information has been added to this geometric planner in order to guarantee that the stability of the grasp is maintained during the execution of in-hand manipulation tasks. This finger readjustment technique has been tested on a *Barrett* robotic hand [Barrett-2011] with tactile sensors, which constitutes one of the main components of the experimental setup of the project DPI2008-02647.

1.3 Contributions

The contributions of this thesis can be organized in two different groups: contributions on safety for human-robot interaction and contributions on dexterous manipulation.

In the field of human safety for human-robot interaction the following contributions can be emphasized:

- *Analysis of human tracking systems for human-robot interaction in industrial applications:* As described in [Section 1.1.1](#), human safety is the first requirement to be fulfilled in human-robot interaction systems. Since pre-collision strategies are required in order to adapt the behavior of the robot to the localization of the human, a human tracking system is necessary. Nevertheless, in human-robot interaction tasks where humans and robots collaborate directly over the same objects, global localization systems which give a global position of the human in the environment are not sufficient. In these cases, it is necessary to track precisely all the limbs of the human body by using motion capture systems. This thesis compares the existing motion capture technologies and analyzes their advantages and disadvantages for their application on human-robot interaction tasks with industrial robotic manipulators.
- *Development of a precise full-body human tracking system:* The previous analysis determines that inertial motion capture systems are the most suitable tracking systems for human-robot interaction in industrial applications. As a result, this thesis proposes the use of an inertial motion capture suit composed by 18 sensors. This system obtains precise relative rotational measurements for each joint of the human body and they are applied over a skeleton which represents the full-body of the human operator. Nevertheless, an analysis of the accuracy of the global position of this skeleton shows that it accumulates important errors for direct human-robot interaction. Due to this drawback, this thesis proposes including an additional localization system based on Ultra-wideband (UWB) pulses with a higher global translational accuracy. The combination of both systems generates a new full-body human tracking system which collects the advantages of both tracking systems: the relative rotational accuracy and the high sampling rate of the inertial motion capture system and the global position accuracy of the UWB system.
- *Three novel sensor fusion algorithms based on Bayesian filtering:* The global position measurements of the two tracking systems described in the previous paragraph have to be combined in order to obtain a precise estimate of the global position of the human operator with a high sampling rate. This thesis proposes three novel algorithms based on Bayesian filtering in order to implement this sensor fusion. In particular, the first algorithm is based on a Kalman filter, the second algorithm is based on a particle filter and the third algorithm is based on a combination of a Kalman filter and a particle filter. These three algorithms have a similar general structure which is based on the recalculation of the transformation matrix between both tracking systems each time a new measurement of the UWB system is registered. This structure not only reduces the computational cost of the fusion algorithms in comparison with previous similar approaches, but it also permits to harmonize the different sampling rates of both tracking systems. In addition, these three algorithms can be applied in

different human-robot interaction tasks depending on the requirements of accuracy and time execution.

- *Design and implementation of a hierarchy of dynamic bounding volumes for computing human-robot distance:* The skeleton of the human operator registered by the human tracking system is covered by a group of bounding volumes in order to calculate efficiently the distance between the human operator and the robot. The structure of the robotic manipulator is also covered by bounding volumes. In order to reduce the number of distance pair-wise tests between all these bounding volumes which are required to calculate the minimum human-robot distance, this thesis proposes a novel hierarchy of bounding volumes composed by three levels. Depending on the human-robot distance, a different level of this hierarchy is used. Each level is composed by a different group of bounding volumes, which covers the human and the robot bodies more precisely than the previous level but increases the required number of pairwise distance tests. Therefore, the implemented human-robot distance algorithm determines the level which combines an approximation of the human-robot distance with a sufficient degree of accuracy and an optimal computational cost based on a reduced number of pair-wise distance tests. Finally, the dimensions of the bounding volumes of the third level (i.e. the more detailed bounding volumes) are increased proportionally to the linear velocity of the link/limb which is covered by them. Thereby, each bounding volume not only represents the dimensions of the covered link but also constitutes an estimate of the possible displacements of the corresponding link. This displacement estimation entails an improvement of the safety in human-robot interaction tasks where quick movements of the robot or the human are performed when they are close to each other.
- *Design and implementation of safety strategies for human-robot interaction tasks:* The human-robot distance computation algorithm based on the hierarchy of bounding volumes described in the previous paragraph is applied in safety strategies. These strategies calculate the human-robot distance at each execution step of the robot controller and if this value is below a pre-established distance threshold, the normal behavior of the robot is stopped and a special safety behavior is executed. This safety behavior may involve stopping the robot's motion or moving the robot away from the human. This safety behavior will guarantee that no collision takes place. Several real human-robot interaction tasks have been performed in order to verify the correct behavior of these safety strategies.

In the field of dexterous manipulation, the following contributions can be emphasized:

- *Modeling of the surfaces of the fingers and the objects to be manipulated as triangle meshes:* Previous dexterous manipulation planners are usually applied on simple objects because they are based on a specific representation of their surfaces. Some of them can only be applied on planar objects. Others need a parametric

model of the surfaces (such as spheres or ellipsoids) because they need the equations of this model in order to compute the evolution of the contact points. In addition, these planners also establish restrictions over the shapes of the fingers. For instance, some of them suppose that the fingers are linear so that only one contact point is possible. Others require spherical fingertips. This thesis proposes a more flexible model where the surfaces of the fingers and the objects to be manipulated are standard triangle meshes. This model not only is able to represent more complex and varied shapes but it is also able to adjust the precision of the surfaces by increasing or reducing the number of triangles. Thereby, the computational cost for analyzing these surfaces can be adapted to the necessities of each manipulation task (i.e. the level of detail of the contact evolution between the surfaces and/or real-time computation restrictions).

- *Design of a contact evolution model for rolling between triangle-mesh surfaces:* This thesis proposes a new contact evolution model between the surfaces of the fingers and the objects represented as triangle meshes. First of all, a contact evolution graph is proposed in order to represent the transition between pairs of adjacent contacting primitives (i.e. vertices, edges and faces of the triangle meshes). Each node of the graph is associated with a specific contact configuration identified by two contacting primitives (one over the finger's surface and the other over the object's surface). Each edge between two nodes of the graph is associated with a transition from one contact configuration towards another contact configuration caused by the rolling around one of the initial contacting primitives until an adjacent new primitive comes into contact.
- *Design and implementation of a geometric dexterous manipulation planner for in-hand object reconfiguration:* This thesis develops a new dexterous manipulation planner based on the contact evolution model described in the previous paragraph. This planner receives as input an initial grasp of the object and a desired final configuration of the object and computes the fingers movements which are necessary to drive the object to the desired configuration. It is implemented in two different steps: global planner and local planner. The global planner generates intermediate configurations of the object. The local planner implements the previous contact evolution model in order to determine the evolution of contacting primitives between each pair of intermediate configurations from the global planner. Finally, the local planner applies the pseudo-inverse of the Jacobian matrix of each finger in order to obtain the movements of the fingers which are required to generate the desired evolution of the contacting primitives. This planner has been verified in several manipulation tasks developed in a software simulator of a five-fingered robotic hand (*Shadow hand*).
- *Readjustment technique of fingers movements for grasp stability during dexterous manipulation:* The main limitation of the dexterous manipulation planner summarized in the previous paragraph is that it is only based on the geometric information of the fingers and object surfaces but it does not take into account contact forces. This limitation makes impractical the use of the planner for

long manipulation tasks because the subsequent manipulation steps reduce progressively the applied contact force until the grasp becomes unstable. This thesis proposes to use pressure information from tactile sensors installed over the phalanxes of the fingers in order to overcome this drawback. In particular, the developed technique performs small readjustments after each iteration of the local planner in order to assure that each contacting finger applies a minimum contact pressure. When the real pressure is below the minimum threshold, the manipulation task finishes because a risk of grasp instability is present. This technique has been verified on a *Barrett* robotic hand.

1.4 Structure

This thesis has been organized in the following chapters in order to explain and justify the contributions enumerated in [Section 1.3](#):

- *Chapter 2- "Human Tracking for Human-Robot Interaction"*: This chapter compares the different motion capture systems depending on their sensor technologies (mechanical, magnetic, optical and inertial) and determines their advantages and disadvantages for full-body tracking of human operators who collaborate with robotic manipulators. From the result of this comparison, this chapter proposes the use of an inertial motion capture system but the analysis of its accuracy reveals the accumulation of drift in the global position of the human. Therefore, this chapter presents an additional localization system based on UWB pulses whose main goal is to correct this inertial drift.
- *Chapter 3- "Sensor Fusion for Human Tracking"*: This chapter presents the fusion algorithms proposed in this thesis in order to combine the global position measurements from the previous tracking systems. First of all, it enumerates previous similar fusion systems and explains in detail the foundations of the most common Bayesian filtering techniques. Afterwards, it describes in detail the principles which are used as common basis for the three novel fusion algorithms and enumerates their main steps. Finally, an assessment of the accuracy and computational efficiency of these algorithms is performed. They are also applied in two real human-robot interaction tasks.
- *Chapter 4- "Safety Strategies for Human-Robot Interaction"*: This chapter first makes a review of previous techniques for modeling the surfaces of humans and robots that work cooperatively. Next, it proposes a new hierarchy of bounding volumes and enumerates the main components of each of its levels. This hierarchy is extended with dynamic bounding volumes and the process of the calculation of their dimensions is detailed. Finally, this chapter explains how several safety strategies have been implemented by specifying the human-robot distance computation algorithm based on the hierarchy of bounding volumes and the software architecture which controls the robot. These safety strategies are applied in three real human-robot interaction systems and their results in terms of distance accuracy and computational efficiency are analyzed.

- *Chapter 5- “Dexterous Manipulation for In-Hand Object Reconfiguration”*: This chapter analyzes deeply the different planning techniques which have been implemented in previous dexterous manipulation systems. From this analysis, it determines the restrictive assumptions which are established in previous research about the surfaces of the fingers and the object to be manipulated. Next, it presents a new surface model that is based on a triangle mesh representation. A new dexterous manipulation planner based on this triangle mesh representation is presented and its main steps (global and local planners) are enumerated. The contact model which is used by the local planner is presented afterwards. Finally, this chapter describes the finger readjustment technique which is added to the local planner in order to guarantee that the stability of the grasp is kept during the manipulation task. The geometric dexterous planner is verified on a simulator of a five-fingered robotic hand (*Shadow* hand) while the readjustment technique is verified on real in-hand manipulation experiments with a three-fingered robotic hand (*Barrett* hand).
- *Chapter 6- “Conclusions”*: This last chapter summarizes and analyzes the different contributions of the previous four chapters and draws some final conclusions from them. In addition, it enumerates the different publications which have been written from the research of this thesis. Finally, it presents new research that will be done in the future in order to extend the research topics of this thesis. After this chapter, two appendices are added: one about the modeling of the *Barrett* hand and the other one is a summary in Spanish of this thesis. Finally, the bibliography of this thesis is enumerated so that the origin of all the citations referenced in the text of the thesis is clarified.

Human Tracking

2 Human Tracking for Human-Robot Interaction

One of the main requirements for the development of human-robot interaction tasks is the precise localization of the human operator who is collaborating with the robot. Motion capture systems are the most suitable solution for human tracking in this case since these systems are able to register the position of all the limbs of the human body. Nevertheless, there are different technologies (mechanical, magnetic, optical and inertial) for implementing motion capture systems for human tracking. In this chapter, not only their fundamentals are presented but their advantages and disadvantages are also analyzed for their application in human-robot interaction tasks. By taking into consideration this comparison between technologies, an architecture of a human tracking system based on an inertial motion capture system and an Ultra-WideBand (UWB) localization system is proposed.

2.1 Introduction	13
2.2 Human Tracking Technologies	15
2.2.1 Mechanical Motion Capture Systems	16
2.2.2 Magnetic Motion Capture Systems	17
2.2.3 Optical Motion Capture Systems	18
2.2.4 Inertial Motion Capture Systems	21
2.2.5 Comparison of Human Tracking Systems	23
2.3 Architecture of the Human Tracking System	24
2.3.1 Inertial Tracking System	24
2.3.1.1 Rotational Accuracy of the Inertial Tracking System	26
2.3.1.2 Global Translational Accuracy of the Inertial Tracking System	29
2.3.2 Ultra-wideband Localization System	30
2.4 Conclusions	33

This chapter describes the main features of the proposed human tracking system and finally it shows the necessity of combining their measurements. The next chapter will describe in detail the fusion algorithm which has been developed to do this.

2.1 Introduction

As described in the previous chapter, one of the main goals of this thesis is the development of human-robot interaction tasks. In order to do this, robotic systems need to be context-aware in order to adapt their tasks to the different states of their environment. This context-awareness does not only imply the detection of the objects which are near the robots but it also includes the tracking of people who collaborate with them. Thus, human-robot interaction tasks become more natural and unobtrusive because robots are able to change their behavior depending on this context information.

This human tracking process is very important in industrial environments because the collaboration between humans and industrial robots may be dangerous for the humans. In fact, industrial robots generally work at high speeds, they usually manipulate heavy objects and thus, a collision with a human could have fatal consequences. Therefore, a precise localization of human operators is needed. Nevertheless, most indoor localization systems only register the global position and orientation of a person but they do not consider the positions of the person's limbs. Motion capture systems

do not have this drawback because they are able to measure full-body movements and because of this feature, they are suitable for implementing tracking systems for human-robot interaction in industrial environments.

Motion capture is a technique for digitally registering the movements of an actor (generally, a human being). Motion capture systems consider the human body as a collection of rigid segments (bones) which are connected through joints which enable relative movements between the bones. The actor whose movements want to be registered usually wears some markers (or sensors) near her/his joints. The motion capture system registers the translations and rotations between these markers and computes the location (position and orientation) of all the bones of the body by applying a kinematic model of the human body.

Motion capture systems are currently used in a wide variety of applications where a precise analysis of the human motion is necessary. Some examples of these applications are:

- *Biomechanics*: Biomechanics aims to study the mechanical structure of living beings (especially, that of human beings). This study is usually applied in three different fields: medicine, sports and ergonomics. Medical biomechanics studies the human motion in order to detect and solve pathologies. It is one of the first fields where motion capture systems were applied [Andriacchi-2000]. Sports biomechanics analyses the kinematic features of sports activities with the purpose of improving the performance of sportswomen and sportsmen by developing optimized training techniques. Ergonomics biomechanics analyses how humans use products in order to adapt their design to the necessities and capacities of the human body.
- *Computer animation and graphics*: This is the most widespread application of motion capture systems nowadays. These systems register the movements of a human actor and afterwards, computer animators apply them to a digital character of a videogame or a film. In traditional computer animation (*keyframing*), the computer animator has to create manually some key frames of the animation and then the animation software interpolates the intermediate frames and generates a final animation. However, in many cases, the movements obtained by this technique are not very natural. This problem is solved with motion capture systems since the movements of the character are directly obtained from a real actor and the computer animator has only to clean and refine the registered data [Izani-2004].
- *Teleoperation of robots*: This is a field of application which is still under development and the current systems have mainly been used in research settings. These systems aim to remotely control a robot by the reproduction of the movements performed by a human operator and registered by a motion capture system. For instance, [Miller-2004] presents a teleoperated system which controls a humanoid robot through a motion capture system based on inertial sensors.

- *Human-robot interaction:* Similarly to the previous application, motion capture systems for human-robot interaction are still under development. The motion capture systems are used to register the movements of human operators who collaborate with robots in order to adapt the behavior of these robots to them. This type of systems have been developed for industrial [Esteves-2005] and service [Onishi-2006] applications.

In this thesis, motion capture systems are used for this last application: human-robot interaction. Nevertheless, these systems are implemented by applying different sensor technologies which have different advantages and limitations depending on the physical principles on which they rely. The ideal sensor technology should permit the development of motion capture systems which have the following features:

- *Small-sized devices:* As markers are usually attached to the human's body, they should be as small as possible. The human operator should feel comfortable and markers must not constrict her/his movements anywhere.
- *Reduced infrastructure:* The quantity of equipment to be mounted in the environment should be minimal, so that installation would be easy and effortless. In many cases, cluttered environments cannot be modified in order to install a complex sensor infrastructure. Finally, the installed infrastructure should not hinder the normal operation of the rest of components of the workplace.
- *Complete and accurate measures:* The measurements obtained by the motion capture system should include position and orientation information (6 DOF) to completely identify the location of every human's joint. Furthermore, the obtained measurements should be accurate in order to guarantee a correct localization of the human operator.
- *Fast response:* The latency, which is the delay between the human's movements and their recording, should be less than 100ms in order to guarantee a real-time response of the system. An increase in the latency of the system may involve dangerous situations where the system is not able to react in time to unexpected events.
- *Robustness:* The system should be able to operate properly even with the presence of disturbances, erroneous measures and occlusions.

Considering these desired characteristics, the main motion capture technologies are analyzed in the following section in order to select the most suitable one for developing human-robot interaction tasks in industrial environments.

2.2 Human Tracking Technologies

Current motion capture systems are based on different sensor technologies [Welch-2002] [Field-2011]: mechanical, magnetic, optical and inertial. They have different advantages and disadvantages depending on the physical properties of each type of sen-

sor. In the following subsections, all these motion capture technologies are described and analyzed.

2.2.1 Mechanical Motion Capture Systems

Mechanical motion capture systems are composed of a set of articulated rigid segments (i.e. an exoskeleton) attached to the human operator's limbs and interconnected between them through electromechanical transducers (such as potentiometers or rotary encoders). When the human operator moves, this exoskeleton is forced to move and the output electrical signal of each transducer identifies the relative configuration between the segments which are linked by the joint where this transducer is installed. In the case of transducers based on potentiometers, the output voltage of the potentiometer is proportional to the corresponding joint angle. In the case of transducers based on rotary encoders, the encoder provides an analog or digital code which represents the angular position of the joint. In both cases, the mechanical system obtains a precise measurement of the relative rotation of each joint of the human operator's body. These joint angles are used in conjunction with the length of the exoskeleton segments in order to compute the position of each limb of the human operator.

An example of a mechanical motion capture system is the *Gypsy-7* [Animazoo-2011a] from Animazoo. This system contains 15 joint sensors based on potentiometers with a resolution of 0.36° and an update rate up to 120 Hz. This system also includes two gyroscopic sensors for determining the orientation of the chest and coccyx. Figure 2.1.a depicts the exoskeleton which composes the Animazoo *Gypsy-7* system. Figure 2.1.b shows the application of this mechanical system in an action parsing system [Gray-2005] which interprets the movements of the human in order to infer the common goals the human wants to achieve in a collaborative task with the *Leonardo* robot. Figure 2.1.c shows a movement imitation system for teaching tennis swings to a humanoid robot [Ijspeert-2002]. The human teacher wears the mechanical system *Sarcos Sensuit* [Sarcos-2011], which is able to record the joint angles of 35 DOFs of the human body at 100Hz using Hall Effect sensors. These joint values are used to encode the desired trajectories as a relation between the angles of shoulder adduction-abduction, shoulder flexion-extension and elbow flexion-extension.

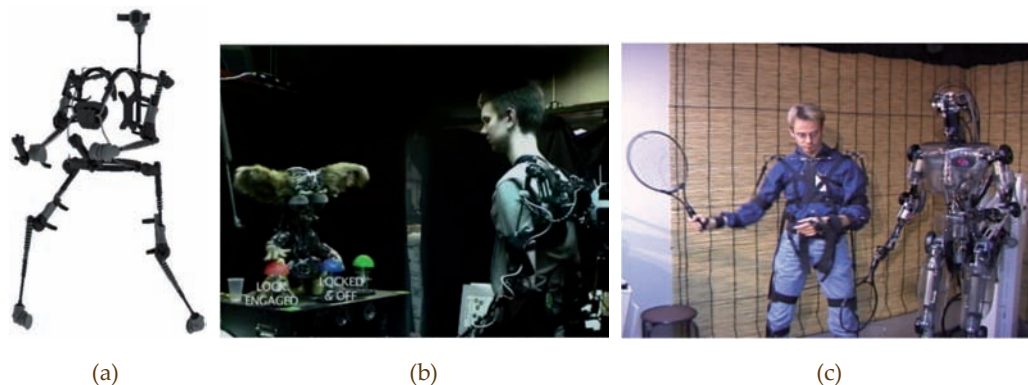


Figure 2.1. Mechanical motion capture systems: (a) Animazoo *Gypsy-7* mechanical system based on exoskeleton, Source: [Animazoo-2011a]; (b) action parsing and goal inference system based on the *Gypsy* system for collaborative human-robot tasks, Source: [Gray-2005]; (c) movement imitation system based on *Sarcos Sensuit*, Source: [Ijspeert-2002].

Mechanical motion capture systems have accurate joint angle measurements, low latencies and no interference problems from light or magnetic fields. However, the use of an exoskeleton can be uncomfortable for the user since the rigid segments must be tightened to the user's limbs and this may limit the user's movements. In addition, these systems do not register the absolute global position and orientation of the human operator in the environment because all the measurements from the exoskeleton are relative rotational joint values.

2.2.2 Magnetic Motion Capture Systems

Magnetic motion capture systems are based on the use of sensors which measure the strength and/or direction of the local magnetic field vector which is around the sensor. Two different magnetic sensors are used in this type of motion capture systems: magnetometers for quasi-static magnetic fields and electromagnetic coils for variable magnetic fields generated by active sources. Each group of three magnetic sensors are usually packaged in a single sensor unit and oriented orthogonally so that each component of the magnetic field vector can be registered. The magnetic field which energizes the sensors may come from the earth magnetic field or from a source unit which is composed of several coils. Most magnetic motion capture systems are based on this second approach and thus they use alternating (AC) or direct (DC) current signals to excite sequentially each coil of the source unit and thus generate different magnetic field vectors, which are measured by the sensor units. Thereby, the magnetic motion capture system is able to estimate the position and orientation (i.e. 6 DOF) of the sensor unit with respect to the source unit.

Two well-known magnetic motion capture systems are: *MotionStar* [Ascension-2011] from Ascension (see Figure 2.2.a) and *Liberty* [Polhemus-2011] from Polhemus (see Figure 2.2.b). The *MotionStar* system can handle a maximum of 108 magnetic sensors (18 sensors per human operator) with an accuracy of 0.762 cm in position and 0.5° in orientation, a resolution of 0.0762 cm in position and 0.1° in orientation and an update rate of 120 Hz. The *Liberty* system can handle up to 16 magnetic sensors, with an accuracy of 0.071 cm in position and 0.15° in orientation, a resolution of 0.046 cm in position and 0.035° in orientation and an update rate of 240 Hz. Both systems have a specific source unit for generating the magnetic fields but the *MotionStar* system applies DC current signals while the *Liberty* system applies AC current signals. One example of an application of the *Liberty* system is the hand motion capture system depicted in Figure 2.2.c which uses 16 magnetic sensors in order to learn the piano playing skills of an expert pianist [Mitobe-2006] [Rahman-2011].

The main advantage of magnetic motion capture systems is the real-time calculation of the absolute positions and orientations of the sensors since the magnetic source unit is always installed at a known fixed position of the environment from which all the sensor measurements are represented. In addition, magnetic systems do not have line of sight restrictions like optical systems and they are usually more affordable. Nevertheless, the accuracy of these systems is negatively influenced by ferromagnetic and conductive materials which can distort the magnetic fields generated by the source unit. For instance, girders and reinforcing metallic bars of the floor may affect the magnetic field of the source unit as well as electrical devices (such as motors, monitors, transformers, etc.) whose induced magnetic fields may cause interferences. Further-

more, the range of operation of magnetic systems is small because the strength of the magnetic field decreases proportionally to the squared distance from the magnetic source. In fact, the *Ascension MotionStar* system has coverage of 3 m for one source unit and an operating area of 3 m x 4.5 m for an extended configuration with two source units. The *Polhemus Liberty* system has a normal range of operation of 1.52 m although it can be increased until 4.6 m with a significant reduction in the accuracy and resolution of the registered measurements.

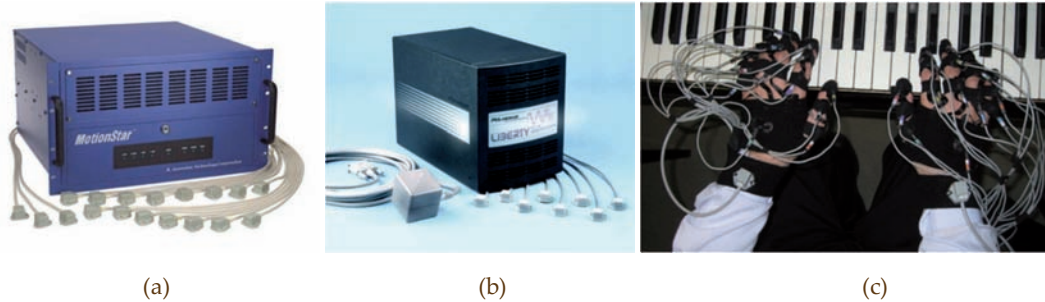


Figure 2.2. Magnetic motion capture systems: (a) *Ascension MotionStar* system, Source: [Ascension-2011]; (b) *Polhemus Liberty* system, Source: [Polhemus-2011]; (c) hand motion capture system, Source: [Mitobe-2006].

2.2.3 Optical Motion Capture Systems

Optical motion capture systems are based on the installation of several calibrated cameras which record images from some markers attached to the human operator's body. The position of each marker is triangulated by using three or more images that contain the marker. The orientation is deduced from the relative orientation between three or more markers. There are two types of markers which determine the two main types of optical motion capture systems: optical systems based on passive markers and optical systems based on active markers. There are also optical systems that do not need markers since they apply previous knowledge about the shape of the human body and computer vision techniques to track the humans in the images captured by the cameras.

Passive markers are made of retroreflective materials which reflect the light (visible or infrared) emitted by LEDs mounted around the cameras. An example of a passive optical motion capture system is the *Vicon MX* system [Vicon-2011] from *Vicon*. The most basic *Vicon MX* architecture consists of a Gigabit Ethernet bridge where a maximum of eight cameras can be connected to a host PC with the *Vicon* software. In applications where more than eight cameras are required, several bridges can be interconnected. These cameras track the position of a group of reflective balls (see Figure 2.3.a) which are attached to the body of the human operator and reflect the strobe light emitted by the LED ring of each camera (see Figure 2.3.b). These cameras can register up to 240 fps with the highest resolution and up to 1000 fps with partial resolution. This system provides a positional accuracy of 0.1 mm and an angular accuracy of 0.15° for each marker. The tracked area varies from 0.5 m^3 to 1000 m^3 by increasing the number of cameras installed in the system. The system can track a maximum of 150 markers. Figure 2.3.c shows the application of a reflective optical system [MotionAnalysis-2011] in a framework for online imitation of human motion by the humanoid robot *HRP-2* [Montecillo-Puente-2010].

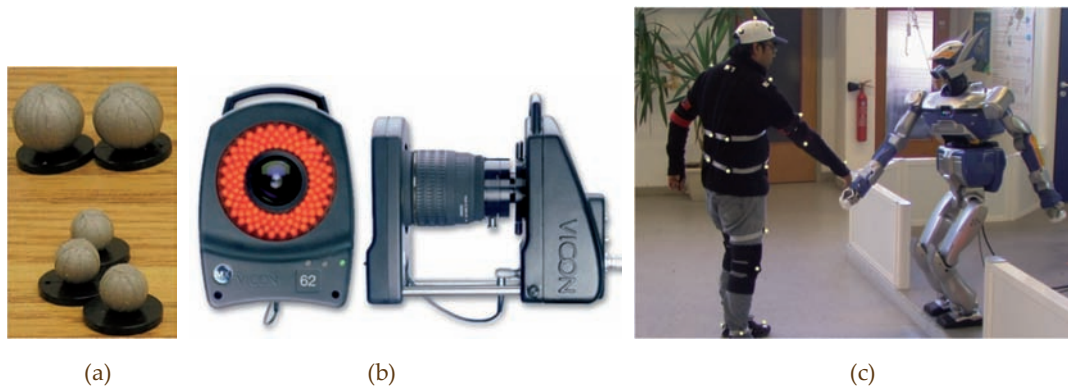


Figure 2.3. Optical motion capture systems based on passive markers: (a) passive markers and (b) camera of the *Vicon MX* system, Source: [Vicon-2011]; (c) online imitation of human motion by a humanoid robot, Source: [Montecillo-Puente-2010].

Active markers are composed of LEDs which emit light (visible or infrared) that is registered by the cameras installed in the environment. An example of an active optical motion capture system is the *Impulse* system [PhaseSpace-2011] from *PhaseSpace*. The main feature of this system is the use of LED markers (see Figure 2.4.a) that emit modulated patterns of pulses, which are used to uniquely identify each one. All the LED markers carried by a human operator are connected to a LED controller which not only lights the LEDs with batteries but also synchronizes wirelessly their modulated patterns with the server PC of the system. The cameras of the system (see Figure 2.4.b) which register the light patterns of the LEDs are also connected to the server PC in order to guarantee that all the components of the system are synchronized. This system can be composed of a maximum of 48 cameras and a maximum of 120 LED markers, which are tracked at a rate between 30 Hz and 480 Hz. Figure 2.4.c shows an active optical motion capture system presented in [Hanai-2009] for analysing the motion of a driver. It is composed of a GPS receiver for synchronisation, eight high intensity LEDs which are worn by the driver and several cameras installed inside the car to calculate the positions of the markers.

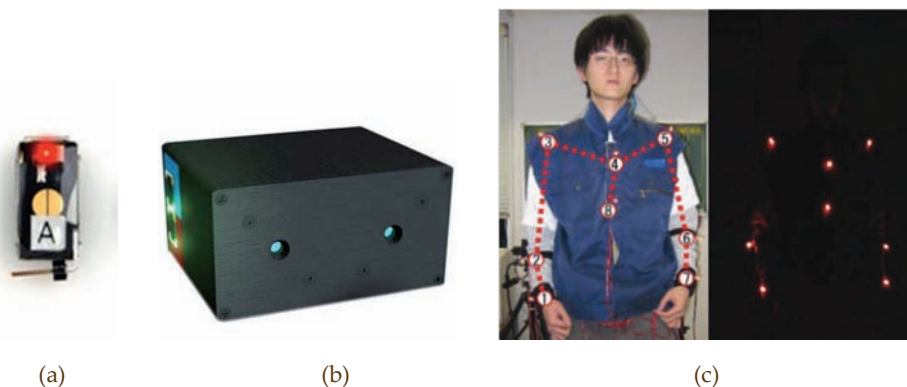


Figure 2.4. Optical motion capture systems based on active markers: (a) active marker and (b) camera of the *PhaseSpace Impulse* system, Source: [PhaseSpace-2011]; (c) physical motion analysis system in driving based on high intensity LEDs, Source: [Hanai-2009].

In the last years, markerless optical motion capture systems are becoming popular because of the comfort for the human operator of not carrying any special device. These systems only need a group of cameras installed in the environment and an initial calibration process where the limbs of the human operator are related to the bones of a skeleton in the images of all the cameras. After this calibration process, the motion

capture system applies segmentation and tracking algorithms to the different images of the cameras in order to separate the human operator's body from the background. Afterwards, machine learning algorithms are applied in order to match the different limbs of the operator with the corresponding parts of the skeleton [Shotton-2011]. Thereby, the system updates the positions and orientations of the bones of the skeleton and the human operator is completely localized. The *iPi Desktop Motion Capture* system [iPiSoft-2011] from *iPi Soft* and the *Xbox Kinect* system [Microsoft-2011] from *Microsoft* are two examples of markerless motion capture systems. The *iPi Desktop* system consists of a software suite (see Figure 2.5.a) that can use the images from up to 6 normal cameras for tracking people in an area of 7x7 m. The *Xbox Kinect* system combines an RGB camera with a range camera based on infrared structured light (see Figure 2.5.b) in order to track people in an area of 2x2 m. Figure 2.5.c depicts the markerless motion capture system applied in [Azad-2007] for action recognition and imitation of human motion from the images registered by a stereo pair installed in the head of the *ARMAR III* humanoid robot.



Figure 2.5. Markerless optical motion capture systems: (a) *iPi Desktop* software, Source: [iPiSoft-2011]; (b) camera of the *Xbox Kinect* system, Source: [Microsoft-2011]; (c) action recognition and imitation of human motion, Source: [Azad-2007].

The main advantage of passive optical motion capture systems is the possibility of easily increasing the level of detail of the motion registration by adding more markers inexpensively. Nevertheless, passive systems generally need a time-consuming process for data cleaning and filtering which has to be executed off-line. In fact, this data cleaning process can take from 1-2 minutes to 15-30 minutes per captured second depending on the complexity of the data initially registered by the motion capture system. There are passive systems which can provide real-time measurements but their accuracy is lower than passive non-real-time systems. Active optical motion capture systems have better accuracy because they do not have the correspondence problem (i.e. the confusion between different makers). However, active markers are bulkier since they usually include batteries or connections to a common controller. Markerless optical motion capture systems are simpler and easier to use due to the absence of markers. Nevertheless, markerless systems have smaller tracked areas and lower accuracy than optical motion capture systems based on markers. In fact, these markerless systems are usually used in computer games and human-computer interfaces but they cannot be employed in applications that require high accuracy and wide movements of the operator in a large room, such as industrial environments.

All the optical motion capture systems have common advantages and disadvantages over other motion capture technologies. On the one hand, optical systems based on markers are accurate (with sub-millimeter accuracy) and have high sampling rates due to the use of high-speed cameras. Optical motion capture systems are a mature technology, as well as magnetic systems, but they are not affected by metallic objects or electric devices. On the contrary, they have line of sight restrictions since the markers need to be seen by a minimum number of cameras in order to compute their location. In case of temporal occlusions of some markers, the accuracy of the optical systems is reduced or even the markers can be completely lost if the occlusion lasts too much time. In addition, the installation time of optical systems is usually greater than other motion capture systems due to the need of calibration of an important number of cameras. In fact, typical optical motion capture systems install between 6 and 24 cameras in the environment in order to reduce the occurrence of occlusions.

2.2.4 Inertial Motion Capture Systems

This kind of motion capture systems is composed of inertial sensors (i.e. accelerometers and gyroscopes) which are tied to each segment of the human operator's body to be tracked. The angular velocity registered by each gyroscope is used to compute the variation of the orientation of the sensor. The linear acceleration registered by each accelerometer represents the components of the gravity acceleration in the sensor's coordinate system. These acceleration components are used to compute the inclination (or tilt) of the sensor with respect to the ground plane. The measurements from the gyroscopes and the accelerometers are combined and an estimation of the orientation of each segment of the human operator's body is obtained. Finally, these orientation values obtained from the sensors are applied over a skeleton in the software of the motion capture system in order to obtain a real-time complete estimation of the position and orientation of all the bones of the human operator.

Inertial systems are widely spread nowadays because of the recent advances in sensor miniaturization. In particular, MEMS (Micro-Electro-Mechanical Systems) technology enables the integration of mechanical elements with electronics on a common silicon substrate and thus, small-sized compact inertial sensor manufacturing is possible. In fact, most current inertial motion capture systems are based on inertial measurement units (IMUs) that integrate several MEMS sensors. The typical IMU configuration consists of a small-sized box which combines three MEMS accelerometers and three MEMS gyroscopes. The accelerometers are placed orthogonally so that they measure linear acceleration in three axes. Similarly, the gyroscopes are placed orthogonally to measure the three components of the angular velocity in a pre-established coordinate system. The main problem of these inertial sensors is that they typically suffer drift (i.e. accumulated error). Since IMUs are based on the addition of relative measurements along time (i.e. dead reckoning), small errors in these measurements are slowly accumulated until they are noticeable. Most recent IMUs try to solve this problem by including additional sensors (such as GPS receivers or magnetometers), which give absolute position and/or orientation references for correcting the drift. The traditional application area of these sensor units is guidance as well as stabilization in Inertial Navigation Systems (INS) for military vehicles, watercraft, aircraft and spacecraft. However, these sensors are also becoming common in motion capture systems.

One example of an IMU applied in human motion capture is the *MTx* IMU [Xsens-2011a] from Xsens (see Figure 2.6.a). This sensor provides as output the 3D orientation of the segment body to which it is attached with an accuracy of 0.5° , a resolution of 0.05° and an update rate of 120 Hz. This orientation is obtained by fusing the measurements of the three gyroscopes, the three accelerometers and the three magnetometers of each IMU [Roetenberg-2006]. 17 *MTx* IMUs have been installed over a Lycra suit or a group of elastic straps (see Figure 2.6.b) in order to build the Xsens MVN motion capture suit [Xsens-2011b]. This inertial motion capture suit is able to track all the movements of the human operator who is wearing it. The measurements of the IMUs are sent wirelessly to a host PC where a software suite from Xsens process them and show them over a skeleton on real-time. Figure 2.6.c shows the application of an inertial motion capture system developed in [Miller-2004] for teleoperating the *NASA Robonaut* humanoid robot. In [Linden-2009], the inertial motion capture suit *IGS-190-M* [Animazoo-2011b] from Animazoo is used to develop a real-time teaching system for novice violinists (see Figure 2.6.d). First of all, the inertial suit registers the arms movements of several teachers in order to create a pattern for teaching. Afterwards, each student wears an inertial suit so that the teaching system can compare her/his arm movements with the learned pattern. When the teaching system detects that the student's movements do not follow the pattern, it provides the student with a vibrotactile feedback so that the erroneous movements can be corrected.

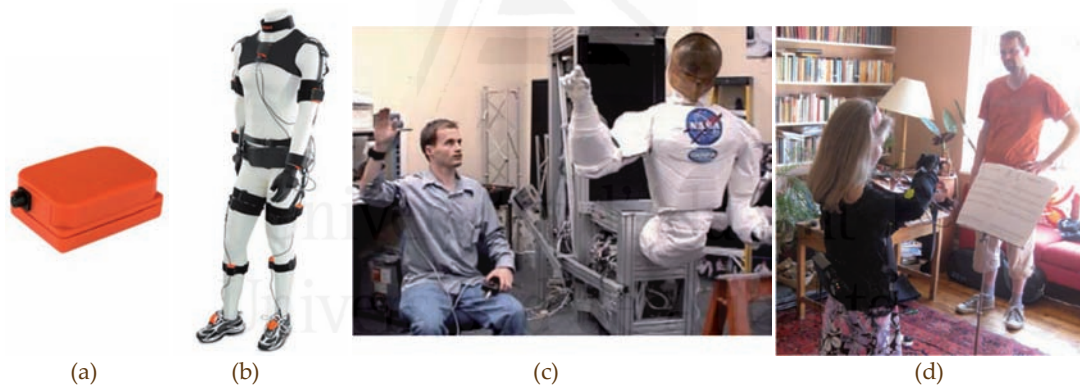


Figure 2.6. Inertial motion capture systems: (a) Xsens *MTx* IMU, Source: [Xsens-2011a]; (b) Xsens MVN motion capture suit; (c) Teleoperation system for the *Robonaut* robot, Source: [Miller-2004]; (d) real-time teaching system for novice violinists based on the Animazoo *IGS-190-M* inertial suit, Source: [Linden-2009].

The main advantages of inertial motion capture systems are their low latencies and their high update rates. They are self-contained and easy to install since no additional external devices (such as magnetic emitters or cameras) are needed to be placed in the tracked area. This feature permits to apply them in large rooms whose dimensions are only limited by the coverage of the wireless networks that connect the sensors with the host PC. For instance, this wireless coverage extends to several hundreds of square meters for most inertial systems. Inertial sensors do not have line of sight restrictions and are not negatively influenced by ferromagnetic materials. Nevertheless, the main disadvantage of inertial sensors is that they accumulate errors throughout the tracking time. As stated before, most modern inertial motion capture systems minimize this drift problem by integrating other sensors in the IMUs. Another problem of the use of inertial sensors is that they only return relative rotational measurements. Therefore, absolute positions can only be computed from estimation algorithms based on rigid

skeleton models of the human body and approximations of the kinematics of human gait.

2.2.5 Comparison of Human Tracking Systems

All the previously described motion capture systems are able to track the movements of a human operator but they have different features depending on the underlying technologies which make them differently appropriate for human interaction with robotic manipulators in industrial environments. Table 2.1 summarizes the main advantages and disadvantages of the different motion capture technologies.

Table 2.1. Comparison between the motion capture technologies.

Technology	Advantages	Disadvantages
Mechanical	<ul style="list-style-type: none"> - Robustness - Accuracy. - Real-time. - Low latency. 	<ul style="list-style-type: none"> - Uncomfortable for daily work. - Physical limitation of movements. - Only relative rotation data.
Magnetic	<ul style="list-style-type: none"> - Precise absolute position data. - No occlusions. - Simple installation: small sensors and magnetic source unit. 	<ul style="list-style-type: none"> - Magnetic distortions that reduce accuracy. - Small tracked area due to strength signal reduction with distance.
Optical-Passive	<ul style="list-style-type: none"> - Inexpensive accuracy increase by adding new markers. - Sub-millimeter position accuracy. - Small markers easy to wear. 	<ul style="list-style-type: none"> - Off-line data cleaning and filtering due to correspondence problem. - Lower accuracy than active optical systems. - Sensitive to occlusions. - Complex installation and calibration which reduce the tracked area.
Optical-Active	<ul style="list-style-type: none"> - No correspondence problem between markers. - Better accuracy than passive optical systems. - Sub-millimeter position accuracy 	<ul style="list-style-type: none"> - Bulkier markers than passive optical systems. - Sensitive to occlusions. - Complex installation and calibration which reduce the tracked area.
Optical-Markerless	<ul style="list-style-type: none"> - The human operator does not carry any marker or device. - No special cameras are needed. - Easy to install. 	<ul style="list-style-type: none"> - Sensitive to occlusions and to light changing conditions. - Generally not real-time since post-processing is needed. - Less accuracy than marker-based optical systems.
Inertial	<ul style="list-style-type: none"> - High rotational accuracy. - Real-time, robust and low latency. - Small sensors and easy installation in large rooms. - Low interference of magnetic distortions due to sensor fusion. - No occlusions. 	<ul style="list-style-type: none"> - Error accumulation due to measurement integration. Sensor fusion in IMUs is used to reduce it. - No absolute position data is available.

Mechanical motion capture systems are accurate but they are uncomfortable for daily work because the exoskeletons may limit the human operator's movements or

may collide with other objects in the environment while the human operator is moving. Magnetic motion capture systems are negatively influenced by magnetic fields present in industrial environments. The influence of constant magnetic distortions produced by fixed metallic objects (e.g. girders, reinforcing bars in the floor, etc.) can be removed by calibrating the magnetic system so that it takes them into account. Nevertheless, the influence of distortions generated by big metallic moving objects (such as robots) and by electrical devices (such as motors) cannot be eliminated because they are variable and impossible to include in the initial calibration of the magnetic system. Optical systems suffer from occlusions which reduce their accuracy. The occurrence of occlusions can be minimized by the installation of additional cameras in the tracked area but this installation may be difficult in cluttered industrial workplaces. Even if the installation of all the cameras of the system is possible, the continuous occlusions caused by the movements of the robots when they approach the human operator make them impractical for close human-robot interaction tasks.

Motion capture systems based on inertial sensors overcome these drawbacks. Inertial systems are more comfortable than mechanical systems since their sensors are small-sized, they do not restrain the freedom of movements of the human operator and they can be installed over normal clothes by using straps or over flexible Lycra suits. Inertial systems can track larger rooms than magnetic systems since they do not suffer from the strength reduction of magnetic fields with regard to distance. In addition, the influence of variable magnetic distortions in the precision of IMUs with magnetometers is substantially lower than in magnetic motion capture systems [Bachmann-2004] since the magnetometers of IMUs are only applied for correcting the drift of one of the rotation components but the main rotation values are obtained from the gyroscopes and the accelerometers. Furthermore, IMUs usually implement fusion algorithms based on filters [Roetenberg-2007a] [Lee-2009] which reduce the influence of these magnetic distortions by applying the measurements from the other sensors (gyroscopes and accelerometers) of the IMUs. Finally, inertial systems do not suffer from occlusions like optical systems and they do not need the installation of a complex infrastructure of devices (such as cameras in optical systems) in the tracked area. All the enumerated advantages of inertial systems over the other motion capture technologies have been considered in order to choose this technology as the most suitable for human tracking during human-robot interaction tasks in industrial environments. In the next section, the architecture of the proposed human tracking system is described and its accuracy is analyzed in detail.

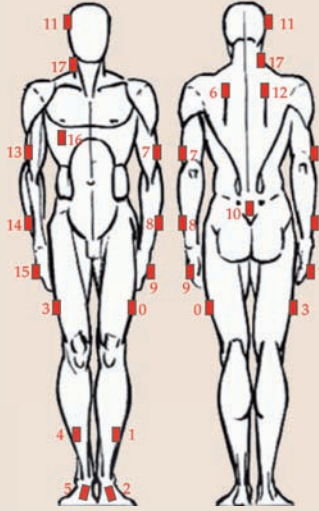
2.3 Architecture of the Human Tracking System

2.3.1 Inertial Tracking System

As mentioned above, inertial motion capture systems are the most suitable technology for complete human tracking in collaboration tasks with industrial manipulators. The current research employs the inertial motion capture system *GypsyGyro-18* (now known as *IGS-190-M*) [Animazoo-2011b] from Animazoo. This system is composed of a Lycra suit over which 18 IMUs are attached with Velcro patches. Each IMU is placed over a rigid segment of the human operator's body (as indicated in Table 2.2)

and provides the system with an orientation measurement of this segment at a maximum update rate of 120 Hz.

Table 2.2. Placement of IMUs in the *GypsyGyro-18* system.

IMUs Organization		Graphical Representation
Upper Left	Upper Right	
6 Left Collar Bone 7 Left Upper Arm 8 Left Forearm 9 Left Hand 10 Hips 11 Head	12 Right Collar Bone 13 Right Upper Arm 14 Right Forearm 15 Right Hand 16 Ribcage 17 Neck	
Lower Left	Lower Right	
0 Left Thigh 1 Left Shin 2 Left Foot	3 Right Thigh 4 Right Shin 5 Right Foot	

The 18 IMUs of the suit are organized in 4 groups depending on the part of the body where they are installed: upper left, upper right, lower left and lower right. All the IMUs of the same group are wired to a common connector which is plugged in a device called MPU (Main Processing Unit). The MPU provides the IMUs with power from a 9.6 V battery and recovers the measurements from the IMUs in order to send them wirelessly to a modem which is connected to the host PC (see Figure 2.7.a). This PC contains the software *GyroDemo* that applies the rotational measurements of the IMUs to a skeletal model of the human operator in order to determine the position and orientation of all the bones of the human operator's body. This skeleton is a hierarchical model which is composed of bones whose joints are represented as nodes (as shown in Figure 2.7.b). This hierarchical organization of the skeleton and the rotation measurements of each node of the skeleton are stored as BVH files [Meredith-2001].

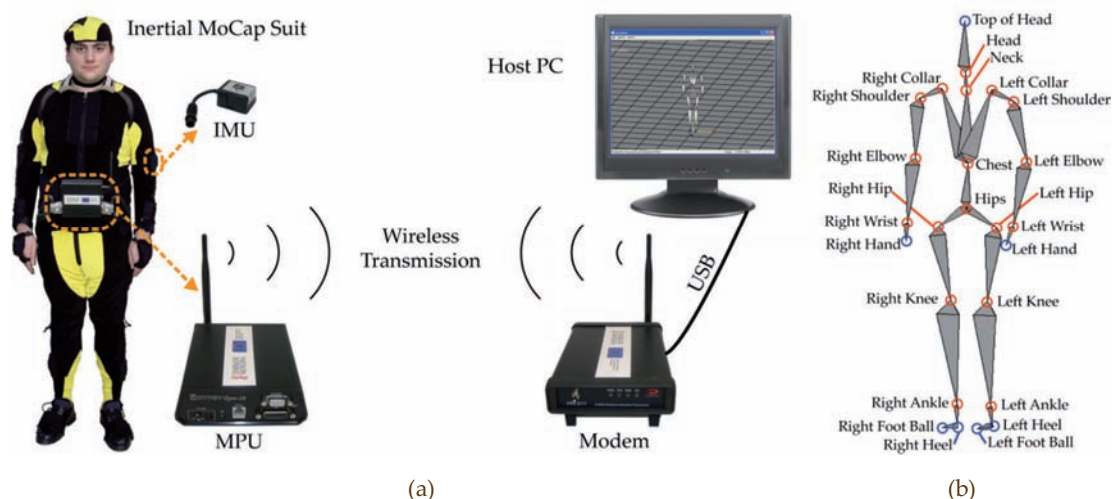


Figure 2.7. Components of the inertial motion capture system: (a) hardware architecture; (b) skeleton.

The rotation angle of each node of the skeleton is obtained from the relative orientation values registered by the two IMUs attached to the two adjacent bones of each joint. Furthermore, as the lengths of these bones in the skeleton correspond to the real dimensions of the segments of the human operator's body, the tracking system is able to determine not only the relative orientation between the bones but also their relative position by applying forward kinematics. In fact, the pose (i.e. position and orientation) of each node of the skeleton is represented as a relative transformation matrix from the parent node in the hierarchy of the skeleton. The global pose of the human is represented by the position and orientation of the Hips node of the skeleton because this is the root node of the skeleton's hierarchy and the rest of its nodes are represented by relative transformations with respect to it. The pose of this node is originally established as the position and orientation where the motion capture system is initialized. Afterwards, the software of the motion capture system applies a footstep extrapolation algorithm in order to determine when the human operator is walking and then increases the position of the skeleton with the amplitude of each footstep accordingly.

Therefore, the accuracy of this inertial motion capture system is limited by two elements. On the one hand, the accuracy of the relative poses of the bones of the skeleton is limited by the accuracy of the orientation measurements registered by the IMUs. On the other hand, the accuracy of the global pose of the skeleton is limited by the precision of the footstep extrapolation algorithm that updates the pose of the Hips node depending on the number and the amplitude of the steps the human operator takes. In the following sections, an analysis of these accuracy values is performed in order to evaluate the circumstances where this inertial motion capture system can be applied for human-robot interaction tasks in industrial environments.

2.3.1.1 Rotational Accuracy of the Inertial Tracking System

Since the orientation measurements registered by the IMUs of the inertial motion capture system are used to compute the rotation angles of the joints of the skeleton of the system, the rotational accuracy of the system is determined by the technical specifications of the IMUs. In particular, the *GypsyGyro-18* system applied in this research is composed of 18 *InertiaCube3* IMUs [Intersense-2011] from Intersense. Each of these IMUs integrates 3 gyroscopes, 3 accelerometers and 3 magnetometers which measure angular velocities, linear accelerations and magnetic field components along three perpendicular axes. The angular velocities measured by the gyroscopes are integrated to obtain the orientation of the operator's limbs as three ZXY Euler angles (yaw, roll and pitch, respectively). However, as stated before, gyroscopes quickly accumulate errors through time. For this reason, the gravitational acceleration obtained from the accelerometers is used to correct roll and pitch gyroscopic drift while the Earth magnetic field measured by the magnetometers is used to correct the drift in yaw. All these measurements are combined by a fusion algorithm described in [Foxlin-1998] so that a precise estimate of the IMU orientation is obtained. Thereby, the orientation measurements returned by these IMUs have a resolution of 0.03° and an accuracy of 1° in yaw and 0.25° in roll and pitch.

As previously stated, when IMUs based on magnetometers are applied, the magnetic distortions generated by the motors of the robot can influence negatively in the rotational accuracy of the inertial motion capture system which is used to track the

human operator who collaborates with the robot. This magnetic interference is substantially lower than in the case of magnetic human motion capture systems due to the fusion of all the sensor measurements of the IMU. However, this interference has to be analyzed and quantified in order to establish the limits of application of the inertial motion capture system in human-robot interaction tasks. The experimental setup implemented to perform this accuracy analysis is composed of two main elements: a *Mitsubishi PA-10* robotic manipulator [Bompos-2007] and an *InertiaCube3* IMU. The *Mitsubishi PA-10* is an anthropomorphic robot with 7 DOFs (see Figure 2.8.a), alternating rotating and pivoting joints: S1, S2, S3 (Shoulder), E1, E2 (Elbow), W1 and W2 (Wrist). This is the robotic manipulator that will be used in this research because of its high flexibility of movements thanks to the redundant joint S3 and its good payload relation since the manipulator weighs 40 Kg and is able to carry a payload of 10 Kg.

Although each joint of this manipulator contains an AC motor, there are only three different types of motors: a 250 W motor (for joints S1 and S2), a 210 W motor (for joints S3 and E1) and a 130 W motor (for joints E2, W1 and W2). In order to measure the influence of the magnetic fields generated by these motors in the accuracy of the inertial motion capture system, an IMU installed over a graduated rotating disk is used (see Figure 2.8.b). Different experiments have been performed in order to quantify the magnetic interferences of the motors of joints S1, E1 and E2, which correspond with each type of motor of the robot. These experiments consist on placing the IMU at different distances from each joint motor and registering the orientation measurements of this IMU while the motor turns at its maximum speed. The IMU remains static and the orientation measurement should always be $(0^\circ, 0^\circ, 0^\circ)$ since the IMU is always placed in the same orientation with the help of the graduated rotating disk.

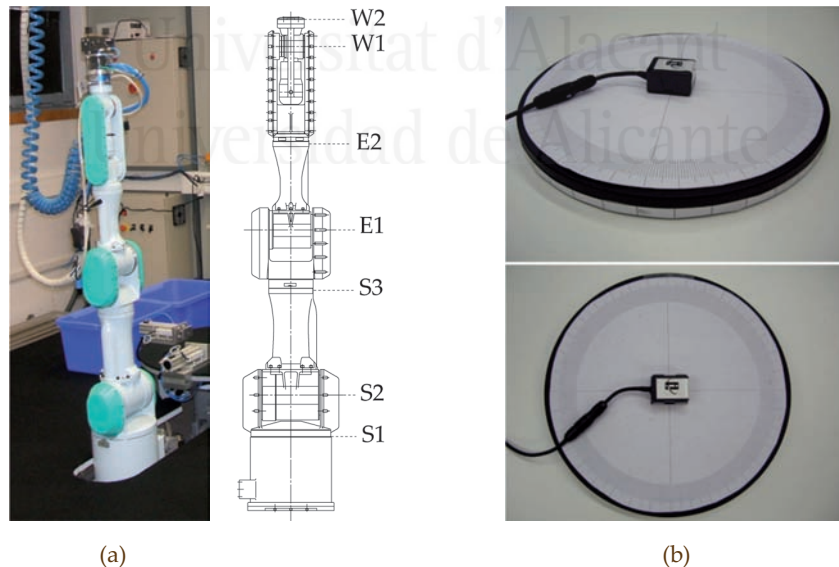


Figure 2.8. Experimental setup for rotational accuracy analysis of the inertial motion capture system: (a) Mitsubishi PA-10 robotic manipulator; (b) InertiaCube3 IMU over a graduated rotating disk.

For each type of joint motor, a table with the statistical evolution of the error (maximum, mean and standard deviation) in the yaw angle depending on the distance to the IMU is developed. The errors in the roll and pitch angles are not shown because they are not affected by the magnetic distortions. For each distance value between the joint motor and the IMU, two or three experiments have been performed. For instance,

Table 2.3 shows the error statistics of two experiments (S11a and S11b) where the IMU is placed 40 cm away from joint S1, two experiments (S12a and S12b) with a distance of 50 cm, two experiments (S13a and S13b) with a distance of 60 cm and two experiments (S14a and S14b) with distance of 70 cm. The results of these experiments indicate that the influence of the motor S1 over the IMU can be considered negligible for distances higher than 70 cm.

Table 2.3. Statistical evolution of error in yaw angle with regard to distance with motor S1.

	40 cm		50 cm		60 cm		70 cm	
	S11a	S11b	S12a	S12b	S13a	S13b	S14a	S14b
Maximum	2.03°	2.10°	0.23°	2.91°	0.00°	0.08°	0.00°	0.00°
Mean	1.11°	1.13°	0.18°	1.91°	0.00°	0.06°	0.00°	0.00°
Std. Dev.	0.93°	0.89°	0.10°	1.35°	0.00°	0.04°	0.00°	0.00°

Table 2.4 enumerates the evolution of the error in the yaw angle of the IMU depending on the distance to the joint E1. In this case, the influence of the magnetic field generated by the motor E1 is negligible for distances higher than 60 cm.

Table 2.4. Statistical evolution of error in yaw angle with regard to distance with motor E1.

	40 cm			50 cm			60 cm		
	E11a	E11b	E11c	E12a	E12b	E12c	E13a	E13b	E13c
Maximum	21.64°	25.89°	26.58°	4.52°	0.57°	0.00°	0.00°	0.00°	0.00°
Mean	17.04°	20.45°	19.75°	2.34°	0.45°	0.00°	0.00°	0.00°	0.00°
Std. Dev.	8.31°	9.95°	10.97°	1.94°	0.23°	0.00°	0.00°	0.00°	0.00°

Finally, Table 2.5 shows the evolution of the error in the yaw angle with regard to the distance to joint E2. The results of these experiments verify that the influence of the magnetic distortions from motor E2 is negligible for distances higher than 25 cm.

Table 2.5. Statistical evolution of error in yaw angle with regard to distance with motor E2.

	15 cm			20 cm			25 cm		
	E21a	E21b	E21c	E22a	E22b	E22c	E23a	E23b	E23c
Maximum	28.03°	3.56°	29.03°	5.87°	2.18°	0.60°	0.00°	0.00°	0.03°
Mean	17.67°	2.05°	22.20°	1.19°	1.81°	0.15°	0.00°	0.00°	0.02°
Std. Dev.	7.02°	1.62°	11.44°	2.22°	0.80°	0.13°	0.00°	0.00°	0.01°

By taking into account the results of the previous experiments, each motor of each joint can be covered by a sphere (see Figure 2.9.a), whose radius corresponds to the minimum distance from which the influence of the magnetic field over the yaw measurements of the IMUs can be considered negligible. Thereby, the PA-10 robotic manipulator is covered with a 3D surface composed by these spheres (see Figure 2.9.b). The inertial motion capture system should always be placed outside this surface in order to guarantee that the orientation measurements of the IMUs are not negatively influenced by the magnetic fields of the motors of the manipulator. In fact, the safety strategies that are described in Chapter 4 will take into account this surface as a minimum human-robot distance threshold so that the magnetic fields of the robot's motors do not alter the orientation measurements of the inertial motion capture system during the development of human-robot interaction tasks.

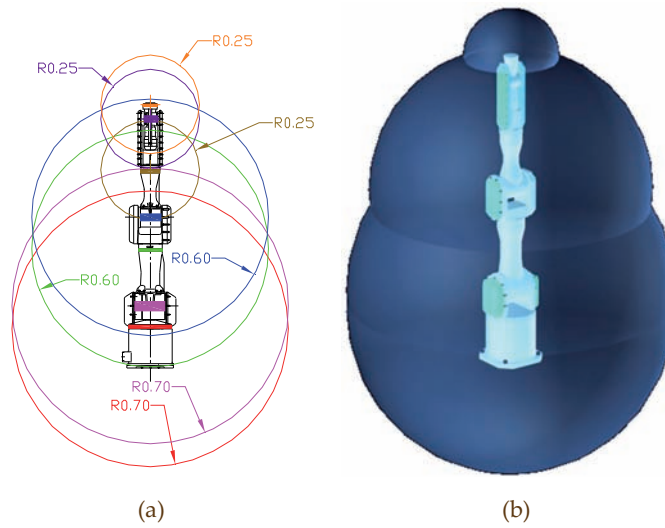


Figure 2.9. Areas of magnetic influence of the PA-10 over the inertial motion capture system: (a) spheres defined around the joint motors; (b) 3D surface around the manipulator.

2.3.1.2 Global Translational Accuracy of the Inertial Tracking System

As previously stated in [Section 2.3](#), the global pose of the human operator who is wearing the proposed inertial motion capture system is obtained by a footstep extrapolation algorithm. This algorithm detects when the human operator takes a step, then computes the amplitude of this step from the relative angles of her/his legs and finally updates the pose of the skeleton (i.e. Hips node) accordingly. In order to quantify the accuracy of this algorithm, a human operator has walked along three different linear paths (2, 3 and 4 meters long, respectively) over a tape measure so that the covered distance is equal between different experiments. Each trajectory has been repeated in 6 experiments. [Figure 2.10](#) depicts the error in the global position of the human operator in every experiment of these three paths.

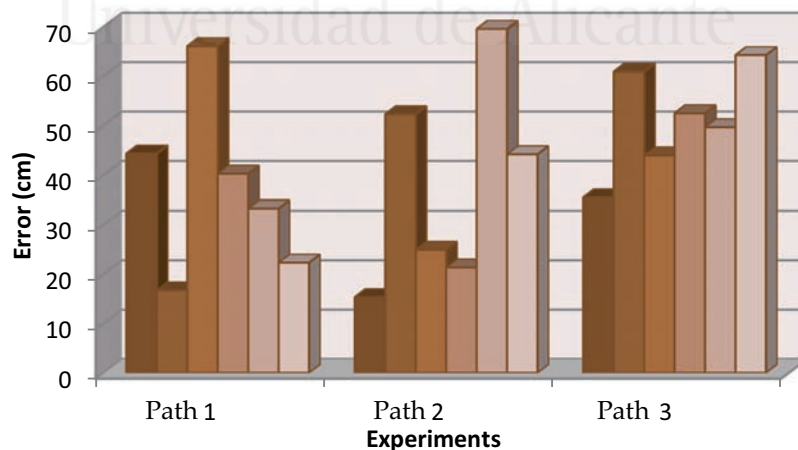


Figure 2.10. Global translational error of the motion capture system in three different paths.

[Table 2.6](#) enumerates the statistics of the error of the global position of the human operator obtained during the previous experiments. These data show that the maximum error in the estimation of the global position of the human is higher than 60 cm in the three paths. Thereby, this translational error cannot be neglected since it can represent more than 15% of the real measurement (e.g. 60 cm in a 4 m path). Furthermore,

the mean error varies from 36.73 cm in Path 1 (200 cm) to 51.09 cm in Path 3 (400 cm). Therefore, the error tends to increase when the covered distance increases. This result is logical since the errors in the global position of the human are caused by the footstep extrapolation algorithm when it fails to detect a footstep taken by the human. In this case, the skeleton is not translated to the new position and remains at the previous global position where it was before the undetected footstep. Thus, the probability of losing a footstep increases as the distance to be covered by the human operator is longer. Since this translational error is not negligible and it increases through distance and time, an additional localization system is needed to correct it. The next section describes in detail the proposed localization system, whose position measurements will be combined with the global position measurements from the inertial motion capture system.

Table 2.6. Statistical evolution of the global translational error of the motion capture system.

	Path 1 (200 cm)	Path 2 (300 cm)	Path 3 (400 cm)
Minimum	16.70 cm	15.33 cm	35.43 cm
Maximum	66.04 cm	69.54 cm	64.23 cm
Mean	36.73 cm	37.92 cm	51.09 cm
Std. Dev.	17.59 cm	20.97 cm	10.67 cm

2.3.2 Ultra-wideband Localization System

The accuracy analysis developed in Section 2.3.1.2 has shown that the global position measurements obtained by the inertial motion capture system contain high error values. These errors cannot be neglected in human-robot interaction tasks where a human operator has to collaborate with an industrial robotic manipulator. Therefore, an additional localization system is required to track with higher accuracy the global position of the human operator. However, this additional localization system will only have to determine the location of the Hips node of the skeleton of the human operator. The rest of nodes of the skeleton are positioned with respect to this node by the relative rotation measurements obtained by the IMUs of the inertial motion capture system.

Although the problem of outdoor localization of humans has been solved with GPS (i.e. Global Positioning System) technology, human localization in indoor environments is an open problem with no general optimal solution [Liu-2007]. In fact, there are several technologies for indoor localization with different advantages and disadvantages [Corrales-2006] [Torres-Solis-2010]. Three indoor localization technologies can be emphasized: ultrasound localization systems, optical localization systems and radio-frequency localization systems. Ultrasound localization systems (such as *AT&T Active Bat* [Harter-2002] and *MIT Cricket* [Priyantha-2005]) have high position accuracy (with errors of a few centimeters) but they require a complex installation because a high number of ultrasound nodes have to be placed in the tracked environment. Optical localization systems (such as *AT&T Active Badge* [Want-1992] and *Microsoft Easy Living* [Brumitt-2000]) use infrared emitting devices or groups of cameras to position the human operator in the environment but they suffer from line-of-sight restrictions which reduce their accuracy. Radio-frequency localization systems overcome these limitations of ultrasound and optical systems: they do not have line-of-sight restrictions and their installation is quite simple because it is composed by a small number of nodes installed in the environment. Radio-frequency systems are usually based on three different elec-

tromagnetic signals: Wi-Fi, RFID (Radio-Frequency Identification) and UWB (Ultra-wideband). Localization systems based on Wi-Fi can be easily installed by using previous network infrastructure but their accuracy (with errors between 1 and 3 m) is insufficient for human tracking in industrial environments. RFID systems are mainly used for object identification, their localization accuracy is not sufficient (with errors between 1 and 2 m) and the range of their antennas is limited (50 cm-3 m for passive systems and 10-100 m for active systems) [Liu-2007]. Because of these drawbacks, this research proposes the use of an Ultra-wideband localization system.

The main difference between traditional radio-frequency transmissions and UWB transmissions is that traditional transmissions send information by variations of the frequency/potency/phase of a sinusoidal wave (known as carrier); while UWB transmissions generate isolated pulses of small duration (tenths of a nanosecond) which occupy a wide range of frequencies (i.e. Ultra-wideband). Thanks to this operation, UWB technology has the following advantages over other radio-frequency localization technologies:

- *Immunity to multipath fading:* The multipath fading effect is caused by the reflection and refraction of an original radio signal over the objects in the environment so that several signals arrive at the receiver and cause interferences to each other (see Figure 2.11.a). This is one of the most important problems in traditional radio-frequency localization systems which reduce their accuracy (between 1 and 3 m). UWB technology overcomes this problem because UWB receivers are able to differentiate the original pulse from the reflected/refracted ones due to their small time duration. Therefore, UWB systems get better accuracy values (with errors smaller than 30 cm).
- *Immunity to radio-frequency interferences:* As described in the previous paragraph, UWB signals occupy a wide range of frequencies (3.1-10.6 GHz) which coincides with the band of frequencies of other radio-frequency signals (see Figure 2.11.b). In order to avoid interferences with coexisting technologies, the power of UWB signal is restricted to a maximum value of -41.3 dBm/MHz, which is substantially lower than the power values of the rest of wireless technologies.

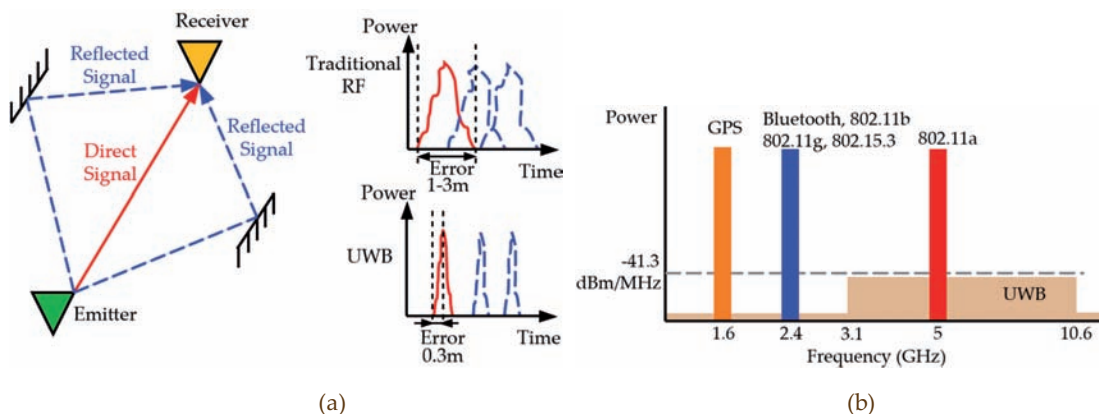


Figure 2.11. Advantages of the UWB technology: (a) immunity to multipath fading; (b) immunity to interferences.

The *Ubisense* system [Ubisense-2011] is the UWB localization system employed in this research to obtain the global position of the human operator in the workspace because of the previously described advantages over other localization systems. The *Ubisense* platform is composed by two different hardware devices: sensors and tags. Sensors (see Figure 2.12.a) are placed at fixed positions in the area to be tracked. Each sensor contains a transceiver (transmitter and receiver) of traditional radio-frequency (433 MHz UHF) and an array of four UWB receivers (5.8-7.2 GHz). Tags (see Figure 2.12.b) are smaller devices (5.8 x 9.2 x 1 cm) which are carried by the people or the objects to be localized. They contain two buttons, two LEDs and a buzzer for communication with the human operator.

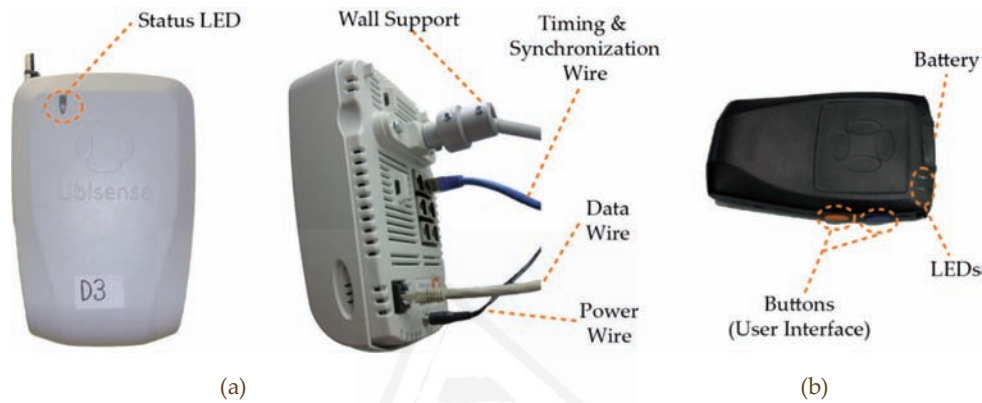


Figure 2.12. Hardware components of the *Ubisense* system: (a) sensor (front and side views); (b) tag.

Ubisense sensors are organized into cells which are composed by a minimum of four sensors and a maximum of seven sensors. Each cell can cover up to 400 m² but any indoor environment can be covered with an unlimited aggregation of cells. Within each cell, there is a master sensor which is in charge of the communication with the tags and the synchronization of the localization measurements from the rest of sensors (called slave sensors). In order to perform this measurement synchronization, all the slave sensors are connected with the master sensor through Ethernet timing wires (see Figure 2.13.a). In addition, all the sensors communicate with a controller PC through a switch to which they are connected with Ethernet data wires. This PC contains the *Ubisense* software platform which initializes the sensors, controls their functions, registers their measurements and executes user applications. Figure 2.13.b shows the physical distribution of the sensor cell in the laboratory used for this research.

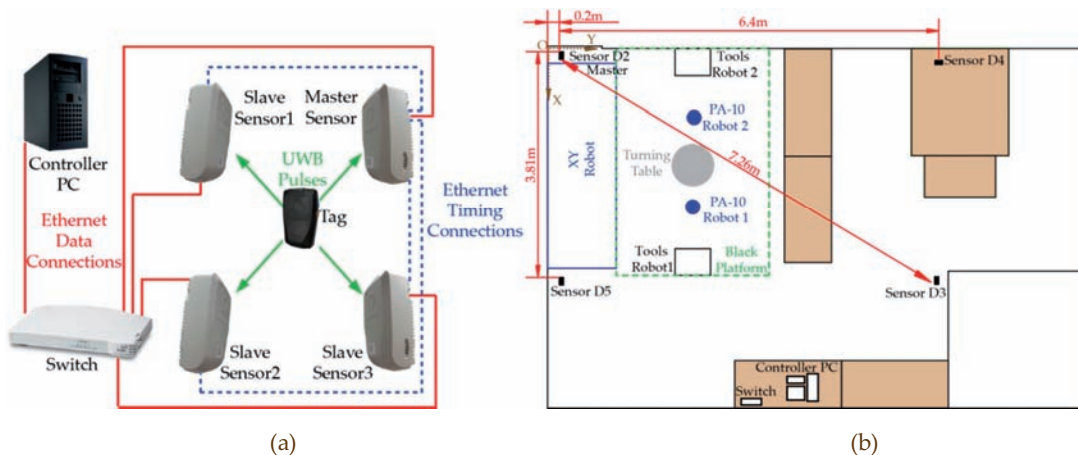


Figure 2.13. *Ubisense* hardware architecture: (a) interconnection of sensors; (b) installation in workplace.

The master sensor assigns a different time slot to each tag with a TDMA (Time Division Multiple Access) algorithm through the traditional radio-frequency band (433 MHz UHF). Thereby, several tags can share the same frequency spectrum without mutual interferences since they are only activated during the time slots assigned by the master.

When a tag is active during its time slot, it sends its identifier through the traditional radio-frequency band and then it sends a sequence of UWB pulses to the sensors. The sensors use a combination of TDOA (Time-Difference Of Arrival) and AOA (Angle Of Arrival) techniques in order to determine the location of this tag from the received UWB pulses. On the one hand, each sensor contains an array of four UWB receivers and it is able to determine precisely the angle of arrival of the received UWB signal (algorithm AOA). On the other hand, since all the sensors are synchronized by the master sensor, they are able to measure the time differences between the receptions of the same UWB pulse at each sensor (algorithm TDOA). These measurements are sent to the controller PC which combines them in order to obtain a precise estimate of the global position of the human operator in the coordinate system of the UWB system. Thereby, this UWB localization system is able to track the global position of the human operator with an accuracy of 15 cm and a maximum update rate of 10 Hz per tag. [Corrales-2007] describes in detail the calibration process of the system required to obtain these optimal accuracy and update rate values.

2.4 Conclusions

This chapter presents the necessity of tracking not only the global position of human operators who collaborate with industrial manipulators but also the position and orientation of all the segments of their bodies. Therefore, motion capture systems are required for the development of safe human-robot interaction tasks with industrial manipulators. Nevertheless, the wide variety of sensor technologies employed in motion capture systems involves a deep analysis of them in order to determine which one is the most suitable for the collaboration between humans and robotic manipulators.

The comparison between the different motion capture technologies (mechanical, magnetic, optical and inertial) comes to the conclusion that inertial motion capture systems fit better the requirements of industrial environments. They do not have line-of-sight restrictions like optical systems; they are more comfortable to wear than mechanical systems and they are not so influenced by magnetic distortions like magnetic systems. Thereby, this chapter proposes the use of an inertial motion capture suit composed by 18 IMUs. Each IMU contains three gyroscopes for determining the orientation (roll, pitch and yaw angles) of the body segment to which it is attached, three accelerometers for correcting the error accumulated in the roll and pitch angles and three magnetometers for correcting the error accumulated in the yaw angle. Since this system is going to be used near a robotic manipulator, several experiments have been performed in order to determine the minimum distance between the robot and the IMUs that makes their magnetometers not be influenced by the magnetic fields of the motors of the robot. This value will be used as a minimum distance threshold in safety strategies for the development of human-robot interaction tasks.

Finally, the accuracy of the global position of the human operator obtained by the inertial motion capture system has also been analyzed. This value is not directly computed from the relative orientation measurements of the IMUs, but it is obtained from a footstep extrapolation algorithm. This algorithm detects when the human operator takes a step and calculates the global displacement of her/his body from the length of her/his stride. The analysis of the results of this footstep extrapolation algorithm has shown that its accuracy is not sufficient for precise human tracking in human-robot interaction tasks. Thereby, this chapter proposes the inclusion of an additional localization system based on Ultra-wideband (UWB) signals. This localization technology is easy to install, insensitive to occlusions and robust to radio-frequency interferences. The proposed UWB system is composed of four sensors installed at fixed positions of the environment and a tag carried by the human operator. The global position measurements of this UWB system are combined with the measurements from the inertial motion capture system in order to get a precise estimation of the position and orientation of the full body of the human operator. The next chapter of this thesis describes deeply the algorithms which have been developed to implement this fusion of global position measurements from the two proposed tracking systems.



Universitat d'Alacant
Universidad de Alicante

Sensor Fusion

3 Sensor Fusion for Human Tracking

The previous chapter proposed the use of two systems for tracking human operators who collaborate with robotic manipulators: an inertial motion capture system and an UWB localization system. The inertial motion capture system is able to register over a skeleton the relative poses of all the bones which compose the body of the human. However, the global position of this skeleton in the environment may accumulate errors. Therefore, the global position measurements from the UWB localization system can be applied to correct these errors. This chapter proposes several strategies to implement the fusion between the global position measurements from both tracking systems. The first approach directly combines the measurements of both systems by recalculating their transformation matrix but it does not take into account their errors. Next, three novel fusion algorithms based on Bayesian Filtering techniques (Kalman and particle filters) are proposed and analyzed in different real experiments. They also include the transformation recalculation

3.1 Introduction	35
3.2 Review of Sensor Fusion Techniques	37
3.2.1 Sensor Fusion Algorithms based on Kalman Filters	39
3.2.2 Sensor Fusion Algorithms based on Particle Filters	44
3.3 First Approach for Sensor Fusion in Human Tracking	49
3.3.1 Coordinate Transformation	49
3.3.2 Transformation Recalculation Algorithm	50
3.4 Bayesian Filtering for Human Tracking	52
3.4.1 Kalman Filter Fusion Approach	52
3.4.2 Particle Filter Fusion Approach	54
3.4.3 Kalman-particle Combination Approach	57
3.5 Experimental Results	59
3.5.1 Comparison between Transformation Recalculation Algorithm and Sensor Fusion based on Kalman Filter.....	59
3.5.2 Comparison between Sensor Fusion Algorithms based on Bayesian Filtering	61
3.5.3 Application of Sensor Fusion in Human-Robot Interaction Tasks.....	66
3.6 Conclusions	70

and they solve the problem of tracking precisely the full body of human operators in human-robot interaction tasks. They constitute one of the main contributions of this thesis.

3.1 Introduction

The previous chapter proposed a tracking system for registering precisely all the movements of human operators who participate in human-robot interaction tasks with robotic manipulators. This tracking system is composed of two main components: an inertial motion capture system and an UWB localization system. The inertial motion capture system contains 18 IMUs which are attached to the limbs of the human operator. These IMUs register the relative orientation of each limb (roll, pitch and yaw angles) and apply them over a skeleton that represents the kinematic structure of the body of the human operator. Thereby, the inertial motion capture system is able to track on real-time the pose of each bone which composes the skeleton of the human operator.

Nevertheless, the measurements of the IMUs applied to this skeleton are relative orientations which do not contain global information about the position of the human operator in the environment. The global position of the human operator is computed by a footstep extrapolation algorithm that detects the moments when the human operator takes a step and calculates the global displacement generated by each step from the length of the corresponding stride. The main problem of this algorithm is that it does not always detect the steps taken by the human operator. When the algorithm fails to detect a step, the skeleton remains at the previous position and thus the global position of the human operator accumulates an error that can be larger than 60 cm, as shown in the accuracy analysis developed in the previous chapter. This is an important drawback for the development of human-robot interaction tasks because it reduces considerably the global position accuracy of the tracking system. In order to solve this problem, the use of an additional localization system based on UWB signals is proposed. This UWB system registers the global position of the human operator by computing the localization of a small tag which emits UWB pulses to four sensors installed at fixed places in the environment. In order to determine the best way to combine the measurements of both systems, their different features should be analyzed. Table 3.1 summarizes the main advantages and disadvantages of the two tracking systems.

Table 3.1. Comparison of the two tracking systems: inertial system and UWB system.

Tracking Systems	Advantages	Disadvantages
Inertial Motion Capture System	<ul style="list-style-type: none"> - High update rate (30-120 Hz). - Constant and stable sampling. - Localization of all the bones of the human body. - High accuracy of joint rotation measurements (less than 1°). - Minimal infrastructure for system installation 	<ul style="list-style-type: none"> - Low accuracy in the global position of the human (errors larger than 60 cm). - The coordinate system of the inertial system is not static since it is established at the initialization position of the system.
UWB Localization System	<ul style="list-style-type: none"> - Smaller errors in global position measurements (less than 15 cm). - Reduced infrastructure (4 sensors) for system installation. - The coordinate system of the UWB system is established at a fixed position of the workplace. 	<ul style="list-style-type: none"> - Low update rate (Maximum: 10 Hz, Normal: 4-7 Hz). - Variable sampling. - Only global position of the human but no information about the position of the bones of the body.

The comparison of Table 3.1 shows that both tracking systems have complementary features. The inertial motion capture system has high error values (larger than 60 cm) in the global position of the human operator while the UWB localization system obtains more accurate position measurements (with errors smaller than 15 cm). Nevertheless, the sampling rate of the UWB system (with a maximum value of 10 Hz) is not high enough to track quick human movements in industrial environments. Furthermore, this sampling rate is variable (usually between 4 and 7 Hz) because a software filter is applied by the UWB system to its position measurements in order to remove outliers. On the other hand, the inertial motion capture system supplies high data rates (30-120 Hz) which remain constant.

This complementarity between both tracking system guarantees that their combination will make the most of their advantages. The position measurements from the UWB system will be applied to correct the error accumulated by the inertial motion capture system due to the non-detection of steps during the execution of its footstep extrapolation algorithm. Nevertheless, the position measurements of the UWB system cannot be used alone because their sampling rate is very low and the position of the human operator will be updated with an excessive latency for the development of real-time tasks. Thereby, the global position measurements which are registered by the inertial motion capture system between each pair of UWB measurements will be used to reduce the latency of the UWB system. This way, the resulting system from the fusion of both tracking systems will have a higher sampling rate and a better accuracy than each system separately. In addition, since the coordinate system of the UWB is established statically at a fixed place of the environment, the measurements of the inertial system can be represented with regard to it so that they can be related with the positions of the robots in the environment for the development of human-robot interaction tasks. The rotational measurements obtained from the IMUs for each joint of the skeleton of the human operator will remain unchanged. They are accurate relative rotation transformations that will be used to determine the pose of each bone of the skeleton relatively to the global position of the root node (Hips node) of the skeleton.

Different techniques can be applied for the combination of the global position measurements of the root node of the skeleton registered by the two proposed tracking systems. In the next sections, previous research on sensor fusion techniques is analyzed and several fusion alternatives based on Bayesian filtering are proposed. These novel fusion algorithms constitute one of the main contributions of this thesis.

3.2 Review of Sensor Fusion Techniques

Pose (position and orientation) estimation by inertial sensors is a well-studied field with applications in: vehicle navigation, virtual reality (VR), augmented reality (AR), robotics and human motion capture. Positions and orientations are usually calculated by integrating accelerations and angular rates respectively returned by accelerometers and gyroscopes. As described in the previous chapter, this dead reckoning process introduces a small error (drift) which is accumulated through time and becomes considerable in a few seconds [El-Sheimy-2008]. Localization systems based on inertial sensors usually include additional sensors (e.g. GPS, ultrasound, magnetic, cameras, UWB and Wi-Fi) which return absolute pose measurements in order to correct the inertial drift. In these hybrid localization systems, Bayesian filtering techniques are generally used to estimate probabilistically the system's pose (state) from the noisy measurements of these sensors (observations) [Thrun-2005].

The Bayesian filtering theory defines the following elements in order to compute the probabilistic model that represents the time evolution of the state of the filter, which is periodically observed by the sensors:

- x_t : This is the state vector at time t . This state contains the elements to be estimated by the Bayesian filter and it depends on the application. In human tracking systems, this state usually contains the pose of the human operator.

- z_t : This is the observation vector at time t . It represents the way the sensors observe the state x_t and contains the measurements registered by the sensors at time t . In human tracking systems, this vector usually contains the poses registered by the inertial sensors and the other additional sensors (e.g. GPS, cameras, ultrasound, UWB, etc.).
- $X_t = \{x_0, x_1, \dots, x_t\} = \{X_{t-1}, x_t\}$: This is the list of states from time 0 to t .
- $Z_t = \{z_1, z_2, \dots, z_t\} = \{Z_{t-1}, z_t\}$: This is the list of observations from time 1 to t .

The main goal of Bayesian filtering is to determine the posterior density of the random variable x_t given the observations Z_t up to time t . This is also known as the belief $Bel(x_t)$ of state x_t and it represents the probability of the occurrence of state x_t (i.e. the human being at the pose established by this state) if the history of sensor measurements is Z_t :

$$Bel(x_t) = P(x_t | Z_t) = P(x_t | z_1, \dots, z_t) \quad (3.1)$$

The Bayesian Filtering approach applies a recursive solution in order to compute the probability density distribution of (3.1). First of all, the Bayes filter applies the total probability theorem to the prior probability distribution $P(x_t | Z_{t-1})$ in order to represent the probability of the state x_t with respect to the previous state x_{t-1} :

$$P(x_t | Z_{t-1}) = \int P(x_t | x_{t-1}, Z_{t-1}) P(x_{t-1} | Z_{t-1}) dx_{t-1} \quad (3.2)$$

The probability term $P(x_t | x_{t-1}, Z_{t-1})$ can be simplified to $P(x_t | x_{t-1})$ by assuming that the state transition is a Markov process where the state x_t only depends on the previous state x_{t-1} and is independent of previous observations Z_{t-1} . Equation (3.2) can be rewritten as follows by taking into account this simplification and the correspondence of belief in (3.1):

$$Bel^-(x_t) = P(x_t | Z_{t-1}) = \int P(x_t | x_{t-1}) Bel(x_{t-1}) dx_{t-1} \quad (3.3)$$

The belief $Bel^-(x_t)$ of (3.3) represents the prior probability of the state x_t before the measurement z_t . It is obtained from the belief of the previous state x_{t-1} and the probability $P(x_t | x_{t-1})$, which describes the *state evolution model* of the system. In other words, the probability $P(x_t | x_{t-1})$ describes how the system's state changes over time. In the case of human tracking systems, this conditional probability is the motion model which establishes where the human operator may be at time t , if she/he was previously at pose x_{t-1} . Equation (3.3) constitutes the first step of the Bayes filter, which is usually called *prediction step* because involves the computation of a prediction of the state x_t before incorporating the measurement z_t .

The second step of the Bayes filter is the *correction step* (also known as *observation update*), which tries to correct the previous prediction of the state x_t with the new measurement z_t . In order to do this, the Bayes rule is applied to (3.1):

$$P(x_t | Z_t) = \frac{P(z_t | x_t) P(x_t | Z_{t-1})}{P(z_t | Z_{t-1})} \quad (3.4)$$

Since the denominator of (3.4) is independent of the state, it can be simplified to a normalizing constant α_t that ensures that the posterior probability over the entire state space sums up to 1. By considering this normalizing factor and (3.3), the previous equation (3.4) can be expressed as follows:

$$Bel(x_t) = P(x_t | Z_t) = \alpha_t P(z_t | x_t) Bel^-(x_t) \quad (3.5)$$

This expression shows how the Bayes filter calculates the belief of the current state x_t during the correction step. To sum up, the belief $Bel(x_t)$ is proportional to the product of the belief $Bel^-(x_t)$ from the prediction step (i.e. prior probability of x_t before z_t) by the conditional probability $P(z_t | x_t)$, which represents the *observation model* of the system. In the case of human tracking systems, this observation model corresponds to the probability of the sensors registering the measurement z_t when the human operator is at pose x_t .

The dependence of $Bel(x_t)$ on $Bel^-(x_t)$ according to (3.5) and the dependence of $Bel^-(x_t)$ on $Bel(x_{t-1})$ according to (3.3) justify the recursive nature of the Bayesian filter since the current state x_t depends on the previous state x_{t-1} . Consequently, this recursive computation process of the posterior belief $Bel(x_t)$ requires an initial belief $Bel(x_0)$ at time $t = 0$ as boundary condition. In the case of human tracking systems, when the initial pose of the human operator is known, $Bel(x_0)$ should be a point mass distribution with 1 probability at the correct value of x_0 and 0 probability elsewhere. When the initial pose of the human operator is unknown, $Bel(x_0)$ should be a uniform distribution with equal probability for all the values of the domain of x_0 and sum up to 1.

The Bayes filter presents an abstract procedure to estimate recursively the state of a dynamic system but it does not implies a specific technique to compute the solution of this problem because it does not specify the representation of the belief $Bel(x_t)$. There are two main techniques to implement this abstract definition of a Bayes filter [Fox-2003]: Kalman filters and particle filters. Different adaptations of these general techniques have been proposed in previous work in order to combine measurements from several sensors. They are described in detail in the following sections.

3.2.1 Sensor Fusion Algorithms based on Kalman Filters

3.2.1.1 Description of the Kalman filter

Kalman filters [Kalman-1960] are the most widely used technique to implement Bayes filters [Simon-2001]. The standard Kalman filter approximates the beliefs of the Bayes filter by Gaussian distributions and represents the state transition model and the observation model as linear stochastic relations of the state.

The Kalman filter assumes that the true state x_t of the system evolves from the previous state x_{t-1} by a linear stochastic relation represented by a $n \times n$ matrix A_t (where n is the dimension of vector x_t) plus a Gaussian noise w_t :

$$x_t = A_t x_{t-1} + w_t \quad (3.6)$$

The matrix A_t represents the *state transition model* (also called *process model*) of the system and indicates the relation between the current state x_t and the previous state x_{t-1} . w_t is a Gaussian random vector that models the randomness in the state transition. It is called the *process noise* and is drawn from a multivariate normal distribution (i.e. Gaussian distribution) with a zero mean and a $n \times n$ covariance matrix Q_t :

$$w_t \sim \mathcal{N}(\mathbf{0}, Q_t) \quad (3.7)$$

The state transition probability $P(x_t | x_{t-1})$ from the Bayes filter is implemented by the Kalman filter as a multivariate normal distribution with the mean vector $A_t x_{t-1}$ (i.e. the state estimated by the state transition model of (3.6) without the process noise) and the covariance matrix Q_t :

$$P(x_t | x_{t-1}) = \mathcal{N}(A_t x_{t-1}, Q_t) = (2\pi)^{-\frac{n}{2}} \det(Q_t)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_t - A_t x_{t-1})^T Q_t^{-1} (x_t - A_t x_{t-1})\right) \quad (3.8)$$

Similarly to (3.6), the observation z_t at time t can be obtained from the true state x_t by a linear stochastic relation represented by a $m \times n$ matrix H_t (where m is the dimension of vector z_t) plus a Gaussian noise v_t :

$$z_t = H_t x_t + v_t \quad (3.9)$$

The matrix H_t represents the *observation model* (also called *measurement model*) of the system and indicates how the sensors map the true state x_t into the observed measurements z_t . v_t is a Gaussian random vector that models the errors in the measurements registered by the sensors. It is called the *observation noise* (or *measurement noise*) and is drawn from a multivariate normal distribution with a zero mean and a $m \times m$ covariance matrix R_t :

$$v_t \sim \mathcal{N}(\mathbf{0}, R_t) \quad (3.10)$$

The measurement probability $P(z_t | x_t)$ from the Bayes filter is implemented by the Kalman filter as a multivariate normal distribution with the mean vector $H_t x_t$ (i.e. the measurement obtained by the observation model of (3.9) without the measurement noise) and the covariance matrix R_t :

$$P(z_t | x_t) = \mathcal{N}(H_t x_t, R_t) = (2\pi)^{-\frac{m}{2}} \det(R_t)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(z_t - H_t x_t)^T R_t^{-1} (z_t - H_t x_t)\right) \quad (3.11)$$

The Kalman filter applies the probability distributions of (3.8) and (3.11) in the prediction and correction steps of the Bayes filter (equations (3.3) and (3.5), respectively) in order to determine the prior belief $Bel^-(x_t)$ and posterior belief $Bel(x_t)$ of the state x_t , respectively.

The prior belief $Bel^-(x_t)$ corresponds to a multivariate normal distribution with mean \hat{x}_t^- and covariance matrix P_t^- :

$$Bel^-(x_t) = \mathcal{N}(\hat{x}_t^-, P_t^-) = (2\pi)^{-\frac{n}{2}} \det(P_t^-)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_t - \hat{x}_t^-)^T P_t^- (x_t - \hat{x}_t^-)\right) \quad (3.12)$$

\hat{x}_t^- is a $n \times 1$ vector which represents the *a priori* state estimate at time t before incorporating the measurement z_t . P_t^- is a $n \times n$ matrix which represents the *a priori* error covariance of the state estimate \hat{x}_t^- and constitutes a measure of the accuracy of the state esti-

mate \hat{x}_t^- . Both elements are computed by the *prediction step* of the Kalman filter (i.e. Kalman filter's implementation of the prediction step of the Bayes filter) with the following expressions (see [Thrun-2005] for their detailed mathematical derivation):

$$\hat{x}_t^- = A_t \hat{x}_{t-1} \quad (3.13)$$

$$P_t^- = A_t P_{t-1} A_t^T + Q_t \quad (3.14)$$

As it is shown in (3.13) and (3.14), the prediction step of the Kalman filter predicts the state \hat{x}_t^- and covariance P_t^- estimates by applying the state evolution model A_t from (3.6) and the process error covariance Q_t to the previous estimates \hat{x}_{t-1} and P_{t-1} .

The posterior belief $Bel(x_t)$ corresponds to a multivariate normal distribution with mean \hat{x}_t and covariance matrix P_t :

$$Bel(x_t) = \mathcal{N}(\hat{x}_t, P_t) = (2\pi)^{-\frac{n}{2}} \det(P_t)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x_t - \hat{x}_t)^T P_t^{-1} (x_t - \hat{x}_t)\right) \quad (3.15)$$

\hat{x}_t is a $n \times 1$ vector which represents the *a posteriori* state estimate at time t after incorporating the measurement z_t . P_t is a $n \times n$ matrix which represents the *a posteriori* error covariance of the state estimate \hat{x}_t . They are computed by the *correction step* of the Kalman filter (i.e. Kalman filter's implementation of the correction step of the Bayes filter) with the following expressions (see [Thrun-2005] for their detailed mathematical derivation):

$$K_t = P_t^- H_t^T (H_t P_t^- H_t^T + R_t)^{-1} \quad (3.16)$$

$$\hat{x}_t = \hat{x}_t^- + K_t (z_t - \hat{z}_t) = \hat{x}_t^- + K_t (z_t - H_t \hat{x}_t^-) \quad (3.17)$$

$$P_t = (I - K_t H_t) P_t^- \quad (3.18)$$

The correction step of the Kalman filter adds the information of the observation z_t obtained by the sensors in order to correct the *a priori* estimates \hat{x}_t^- and P_t^- of the prediction step. Firstly, the correction step computes the *Kalman filter gain* K_t (see (3.16)), which is a $n \times m$ matrix that minimizes the *a posteriori* error covariance P_t . Secondly, the correction step obtains the *a posteriori* state estimate \hat{x}_t (see (3.17)) by adding to the *a priori* estimate \hat{x}_t^- the difference between the true observation z_t and the observation $\hat{z}_t = H_t \hat{x}_t^-$ estimated by the observation model from (3.9). Finally, the correction step obtains the error covariance P_t of the *a posteriori* state estimate by applying (3.18).

In (3.17), the Kalman filter gain K_t actuates as a scaling factor that gives different importance to the subtraction $z_t - \hat{z}_t$ (also called *residual* or *measurement innovation*) according to the values of P_t^- and R_t (see (3.16)). When the *a priori* covariance P_t^- is very small (i.e. practically null); K_t is very small and thus, the effect of the residual is very small since this means that the *a priori* estimate \hat{x}_t^- is good (i.e. with small error). On the contrary, when P_t^- is very large; K_t becomes approximately H_t^{-1} and thus, the correction step disregards the *a priori* estimate \hat{x}_t^- (since it has a large error) and uses $H_t^{-1} z_t$ (i.e. the state obtained from the measurements z_t) as the *a posteriori* estimate \hat{x}_t . When the measurement covariance R_t is very large; K_t is again small and thus, the correction step disregards the residual (since the error of the measurements is large) and the *a posteriori* estimate \hat{x}_t depends primarily on the *a priori* estimate \hat{x}_t^- . When the measurement covariance R_t is very small (i.e. practically null); K_t is approximately H_t^{-1} and thus, $H_t^{-1} z_t$ becomes the *a posteriori* estimate \hat{x}_t because the error of the measurements is very small and z_t can be trusted.

The previous paragraphs described how the standard Kalman filter obtains an optimal estimate of the state \hat{x}_t of the system and its corresponding error covariance P_t at time t by applying two steps: prediction and correction. The prediction step applies the state transition model to the previous state estimate \hat{x}_{t-1} in order to obtain a prior estimate \hat{x}_t^- of the state. Afterwards, the correction step applies the measurements z_t of the sensors to the prior estimate \hat{x}_t^- in order to correct it and compute a posterior estimate \hat{x}_t . Therefore, the Kalman filter is implemented as a recursive procedure which alternates both steps (prediction and correction) at each time step t and when a new measurement z_t is registered, as shown in Figure 3.1.

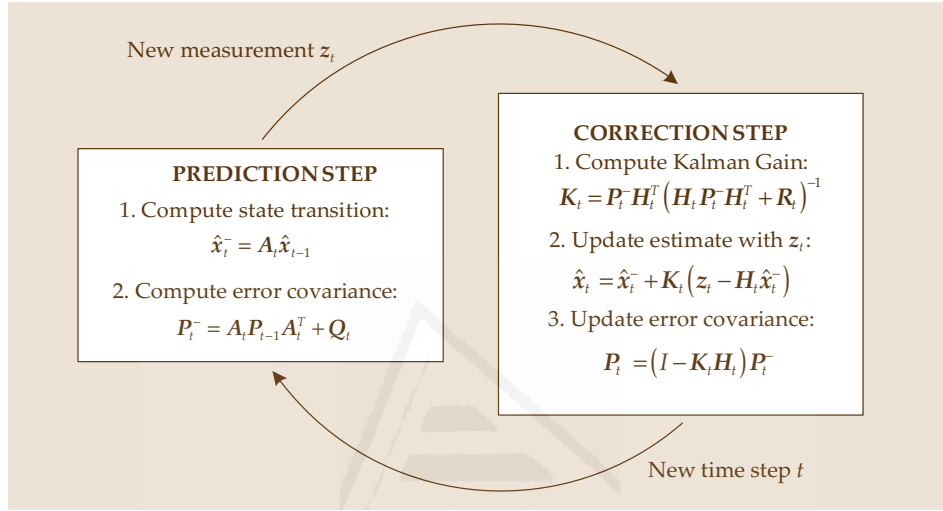


Figure 3.1. Diagram of the operation of the Kalman filter based on two steps (prediction and correction).

3.2.1.2 Previous Research on Kalman Filtering for Human Tracking

The standard Kalman filter described above can integrate the measurements from several sensors when they correspond to different components of the observation vector and they are independent. Nevertheless, when the measurements of the sensors represent the same components of the observation vector (e.g. multiple localization sensors which register the pose of a human), an adaptation of the standard Kalman filter is needed in order to integrate these measurements. In the field of human tracking systems, the standard Kalman filter has been typically adapted by two different techniques in order to incorporate measurements from multiple sensors: complementary Kalman filters and Kalman filters with multi-channel correction steps. Figure 3.2 depicts two general diagrams which represent how these two multi-sensor adaptations of the Kalman filter apply sensor measurements inside the filter.

A complementary Kalman filter is an easy way to integrate several sensors measurements in a Kalman filter because the internal structure of the filter is not changed. Complementary Kalman filters estimate sensors errors instead of direct measurements. They receive as input the differences between the sensors measurements (see Figure 3.2.a). Foxlin [Foxlin-1996] develops an IMU based on a complementary Kalman filter for head-tracking in virtual environments. It is composed by three orthogonal angular rate gyroscopes, a two-axis inclinometer and a two-axis compass. This system estimates errors in orientation (from the inclinometer and the compass) and angular rate (from the gyros). The complementary Kalman filter has also been used by Foxlin [Fox-

lin-2005] in a navigation system, called *NavShoe*, which tracks the location of a pedestrian from two IMUs mounted on his shoes and a GPS receiver (see Figure 3.3.a). Roetenberg et al. [Roetenberg-2007b] present a human motion capture system which uses the differences between an inertial tracking system and a magnetic tracker in position and orientation as the measurement updates for a complementary Kalman filter.

Caron et al. [Caron-2006] extend the definition of the Kalman filter to include measurements from multiple sensors (an IMU and a GPS receiver) in the correction step. This Kalman Filter has one measurement model for each sensor type which is weighted according to fuzzy context variables. These fuzzy variables represent sensors data reliability and are used to reject bad measurements. You and Neumann [You-2001] develop a similar solution where an extended Kalman filter with two independent correction channels (see Figure 3.2.b) combines position measurements from a camera and orientation measurements from three orthogonal rate gyroscopes in order to implement an Augmented Reality (AR) system [You-1999].

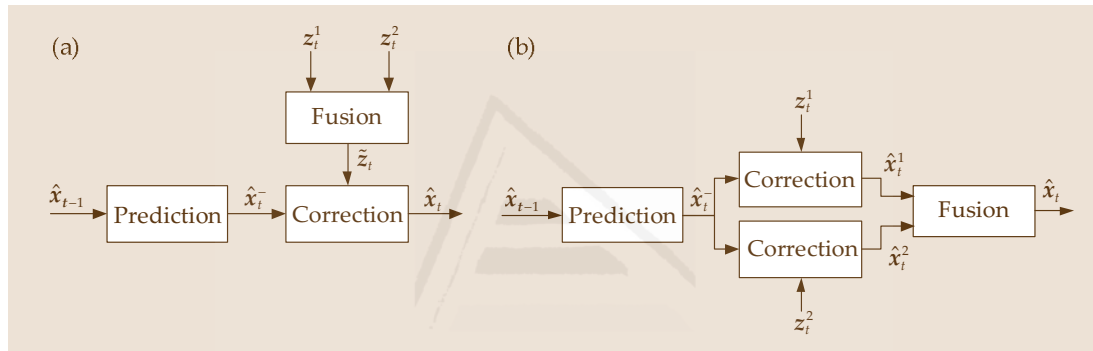


Figure 3.2. Multi-sensor adaptations of the Kalman filter: (a) complementary Kalman filter; (b) Kalman filter with multiple correction channels.

Ribo et al. [Ribo-2002] [Ribo-2004] present a wearable AR system that is mounted on a helmet (see Figure 3.3.b). It consists of a real-time 3D visualization subsystem (composed by a stereo see-through HMD) and a real-time tracking subsystem (composed by a camera and an IMU). Sensor fusion is accomplished by an extended Kalman filter. In the prediction step, inertial measurements are used to compute new state vector estimations in the temporal interval between two measurements from the vision-based tracker. In the correction step, vision measurements correct the state vector estimation from the prediction step.



Figure 3.3. Sensor fusion systems for human tracking based on Kalman filters: (a) navigation system *NavShoe*, Source: [Wan-2010]; (b) wearable Augmented Reality kit, Source: [Ribo-2002].

3.2.2 Sensor Fusion Algorithms based on Particle Filters

3.2.2.1 Description of the Particle Filter

Particle filters are sequential Monte-Carlo methods which are used to implement Bayes filters [Arulampalam-2002] [Thrun-2005]. They approximate the posterior belief $Bel(\mathbf{x}_t)$ of the state \mathbf{x}_t at time t by a set S_t of N_p samples or particles (see (3.19)). Each particle $\mathbf{x}_t^{[i]}$ corresponds to a concrete instantiation of the state at time t (i.e. a hypothesis about the value of the state) and it has associated a nonnegative normalized weight $w_t^{[i]}$ (called *importance factor*) which represents the likelihood of this state value:

$$Bel(\mathbf{x}_t) = P(\mathbf{x}_t | \mathbf{Z}_t) \sim S_t = \left\{ \langle \mathbf{x}_t^{[i]} | w_t^{[i]} \rangle \right\}_{i=1}^{N_p} \text{ and } \sum_{i=1}^{N_p} w_t^{[i]} = 1 \quad (3.19)$$

Therefore, the posterior probability distribution of a state \mathbf{x}_t at time t can be approximated by adding the weights of all the particles corresponding to this state (i.e. *Monte-Carlo approximation*):

$$Bel(\mathbf{x}_t) = P(\mathbf{x}_t | \mathbf{Z}_t) \approx \sum_{i=1}^{N_p} w_t^{[i]} \delta(\mathbf{x}_t - \mathbf{x}_t^{[i]}) \quad (3.20)$$

where $\delta(\mathbf{x}_t - \mathbf{x}_t^{[i]})$ represents the Dirac delta function. Similarly, the estimate $\hat{\mathbf{x}}_t$ of the state can be computed as the expectation value (i.e. weighted average) of all the particles at time t :

$$\hat{\mathbf{x}}_t = E[\mathbf{x}_t | \mathbf{Z}_t] = \sum_{i=1}^{N_p} w_t^{[i]} \mathbf{x}_t^{[i]} \quad (3.21)$$

The main advantage of this probability representation based on particles is that it can approximate virtually any sequence of probability distributions because it is not dependent on any particular parametric representation of the system (i.e. non-parametric approximation). Therefore, particle filters are not limited to Gaussian probability distributions with linear state transition and observation models like the standard Kalman filter. In fact, particle filters can be applied in tracking problems with non-Gaussian probability distributions (even multimodal) and with non-linear models.

Since the particle filter is an implementation of the Bayes filter, its main goal is to obtain an approximation of the posterior belief $Bel(\mathbf{x}_t)$ from the previous posterior belief $Bel(\mathbf{x}_{t-1})$. As these posterior beliefs are approximated by sets of particles (see (3.19)), the final goal of the particle filter is to compute a new set of particles $S_t = \left\{ \langle \mathbf{x}_t^{[i]} | w_t^{[i]} \rangle \right\}_{i=1}^{N_p}$ from the set of particle of the previous step $S_{t-1} = \left\{ \langle \mathbf{x}_{t-1}^{[i]} | w_{t-1}^{[i]} \rangle \right\}_{i=1}^{N_p}$.

In order to do this, the development of the particle filter is divided into two steps which are equivalent to the prediction and correction steps of the Bayes filter. The first step (called *sampling* or *prediction* step) involves the computation of the prior belief $Bel^-(\mathbf{x}_t)$. If the representation of $Bel(\mathbf{x}_{t-1})$ as a sum of particle weights (see (3.20)) is substituted in the general expression (3.3) of the prediction step of the Bayes filter, the following approximation for the prior belief $Bel^-(\mathbf{x}_t)$ is obtained:

$$Bel^-(\mathbf{x}_t) = P(\mathbf{x}_t | \mathbf{Z}_{t-1}) = \int P(\mathbf{x}_t | \mathbf{x}_{t-1}) Bel(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1} \approx \sum_{i=1}^{N_p} p(\mathbf{x}_t | \mathbf{x}_{t-1}^{[i]}) \cdot w_{t-1}^{[i]} \quad (3.22)$$

The term $\sum_{i=1}^{N_p} p(\mathbf{x}_t | \mathbf{x}_{t-1}^{[i]}) \cdot w_{t-1}^{[i]}$ in (3.22) corresponds to a Monte-Carlo integration and it can be represented as a new particle set S_t^- :

$$S_t^- = \left\{ \langle \mathbf{x}_t^{[i]} | w_{t-1}^{[i]} \rangle \right\}_{i=1}^{N_p} \quad \text{where } \mathbf{x}_t^{[i]} \sim P(\mathbf{x}_t | \mathbf{x}_{t-1}^{[i]}) \quad (3.23)$$

This particle set S_t^- is obtained by sampling N_p new samples $\mathbf{x}_t^{[i]}$ from the state transition distribution $P(\mathbf{x}_t | \mathbf{x}_{t-1}^{[i]})$ and keeping the importance weight $w_{t-1}^{[i]}$ of the previous step. Consequently, the $Bel^-(\mathbf{x}_t)$ can be approximated by S_t^- :

$$Bel^-(\mathbf{x}_t) = P(\mathbf{x}_t | \mathbf{Z}_{t-1}) \approx \sum_{i=1}^{N_p} w_{t-1}^{[i]} \delta(\mathbf{x}_t - \mathbf{x}_t^{[i]}) \quad (3.24)$$

After this prediction/sampling step, the particle filter has to incorporate the information of the measurement z_t in order to compute the posterior belief $Bel(\mathbf{x}_t)$. By substituting the Monte-Carlo expression (3.24) of the prior belief $Bel^-(\mathbf{x}_t)$ into (3.5), the following approximation of the posterior belief $Bel(\mathbf{x}_t)$ is obtained:

$$\begin{aligned} Bel(\mathbf{x}_t) &= P(\mathbf{x}_t | \mathbf{Z}_t) = \alpha_t P(z_t | \mathbf{x}_t) Bel^-(\mathbf{x}_t) \approx \alpha_t P(z_t | \mathbf{x}_t) \sum_{i=1}^{N_p} w_{t-1}^{[i]} \delta(\mathbf{x}_t - \mathbf{x}_t^{[i]}) \\ &= \alpha_t \sum_{i=1}^{N_p} P(z_t | \mathbf{x}_t^{[i]}) w_{t-1}^{[i]} \delta(\mathbf{x}_t - \mathbf{x}_t^{[i]}) = \alpha_t \sum_{i=1}^{N_p} \hat{w}_t^{[i]} \delta(\mathbf{x}_t - \mathbf{x}_t^{[i]}) \end{aligned} \quad (3.25)$$

The last term $\sum_{i=1}^{N_p} \hat{w}_t^{[i]} \delta(\mathbf{x}_t - \mathbf{x}_t^{[i]})$ in (3.25) corresponds to a set \hat{S}_t of particles with the same state values $\mathbf{x}_t^{[i]}$ as in S_t^- but with new importance weights $\hat{w}_t^{[i]}$ which are computed by multiplying the previous weights $w_{t-1}^{[i]}$ by the observation model $P(z_t | \mathbf{x}_t^{[i]})$:

$$\hat{S}_t = \left\{ \langle \mathbf{x}_t^{[i]} | \hat{w}_t^{[i]} \rangle \right\}_{i=1}^{N_p} \quad \text{where } \hat{w}_t^{[i]} = P(z_t | \mathbf{x}_t^{[i]}) \cdot w_{t-1}^{[i]} \quad (3.26)$$

Nevertheless, the weights of the particles in \hat{S}_t are not normalized and thus, they do not represent a real probability distribution but a proportional value. By normalizing these weights, the final S_t particle set, which represents the posterior belief $Bel(\mathbf{x}_t)$, is obtained:

$$Bel(\mathbf{x}_t) = P(\mathbf{x}_t | \mathbf{Z}_t) \sim S_t = \left\{ \langle \mathbf{x}_t^{[i]} | w_t^{[i]} \rangle \right\}_{i=1}^{N_p} \quad \text{where } w_t^{[i]} = \frac{\hat{w}_t^{[i]}}{\sum_{i=1}^{N_p} \hat{w}_t^{[i]}} \quad (3.27)$$

Equation (3.27) corresponds to the second step (called *correction* or *weight update* step) of the particle filter. This two-step implementation of the particle filter is known as *Sequential Importance Sampling* (SIS) and constitutes the basic framework for most particle filter algorithms because it is based on the general Monte-Carlo technique of *importance sampling* (see [Doucet-2011] for a more detailed theoretical description).

The main problem of SIS particle filters is that they provide estimates whose variance increases through time [Doucet-2000]. This fact involves that after a few iterations, most particles have negligible weight and the weight is only concentrated on a few particles. This phenomenon is called *degeneracy problem* and reduces the efficiency and precision of the particle filter because it implies that an increasing computational effort

is devoted to update particles whose weights are negligible and thus their contributions to the approximation of the posterior are almost null. This problem is alleviated by adding a third step to the particle filter: *resampling*. The resampling step consists of drawing N_p particles from the particle set S_t in such a way that the probability of drawing each particle $x_t^{[i]}$ is given by the posterior probability distribution $P(x_t | Z_t)$. In other words, the probability of keeping each particle $x_t^{[i]}$ in the new particle set is equivalent to its importance weight $w_t^{[i]}$. The new particle set \tilde{S}_t resulting from the resampling process will contain N_p new particles $\tilde{x}_t^{[i]}$ with the same weight $\tilde{w}_t^{[i]} = 1/N_p$:

$$Bel(x_t) = P(x_t | Z_t) \sim \tilde{S}_t = \left\{ \langle \tilde{x}_t^{[i]} | \tilde{w}_t^{[i]} = 1/N_p \rangle \right\}_{i=1}^{N_p} \quad \text{where } \tilde{x}_t^{[i]} \text{ is drawn from } S_t \quad (3.28)$$

Thereby, this resampling process changes the distribution of the particles by incorporating their importance weights. Whereas the particles $x_t^{[i]}$ of the set S_t are distributed according to $Bel^-(x_t)$, the particles $\tilde{x}_t^{[i]}$ of the set \tilde{S}_t are distributed according to $Bel(x_t)$. Consequently, the resulting set \tilde{S}_t usually contains duplicate particles which correspond to state values with high importance weights and does not usually contain particles with low importance weights. Several algorithms for implementing this resampling process are described in [Douc-2005] and [Hol-2006]. The resampling process is usually executed as the third step of the particle filter, after the prediction and weight update steps. This three-step particle filter is called *Sequential Importance Resampling* (SIR) or *bootstrap filter* [Gordon-1993]. Figure 3.4 depicts the evolution of the particle sets which are generated during the execution of the three steps (prediction, weight update and resampling) of the SIR particle filter. Table 3.2 enumerates the pseudo-code which is required to implement the SIR particle filter.

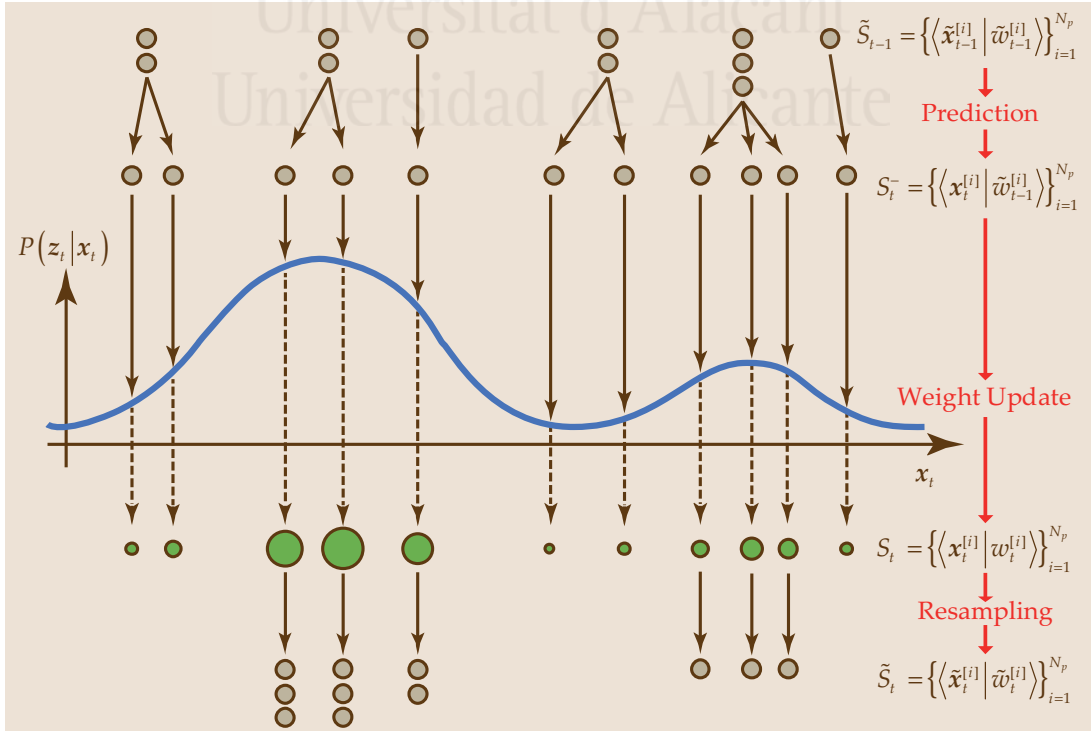


Figure 3.4. Evolution of the different particle sets generated during the execution of the SIR particle filter.

Table 3.2. Algorithm of the SIR particle filter.

Algorithm: SIR Particle Filter Input: $\tilde{S}_{t-1} = \left\{ \langle \tilde{x}_{t-1}^{[i]} \tilde{w}_{t-1}^{[i]} \rangle \right\}_{i=1}^{N_p}$, z_t Implementation: 01: $S_t = \tilde{S}_t = \emptyset$ 02: for $i=1$ to N_p 03: Sample $x_t^{[i]} \sim P(x_t \tilde{x}_{t-1}^{[i]})$ // STEP 1: State prediction with state transition model 04: $\hat{w}_t^{[i]} = P(z_t x_t^{[i]}) \cdot \tilde{w}_{t-1}^{[i]}$ // STEP 2a: Update weights with observation model 05: end for 06: Calculate total weight: $total = \sum_{i=1}^{N_p} \hat{w}_t^{[i]}$ 07: for $i=1$ to N_p 08: $w_t^{[i]} = \hat{w}_t^{[i]} / total$ // STEP 2b: Normalize importance weights 09: $S_t = S_t \cup \{ \langle x_t^{[i]} w_t^{[i]} \rangle \}$ // This is the solution particle set in a SIS particle filter 10: end for 11: for $i=1$ to N_p 12: Draw sample $\tilde{x}_t^{[i]}$ from S_t with probability $P(x_t Z_t)$ // STEP 3: Resampling 13: $\tilde{S}_t = \tilde{S}_t \cup \{ \langle \tilde{x}_t^{[i]} 1/N_p \rangle \}$ 14: end for Output: $\tilde{S}_t = \left\{ \langle \tilde{x}_t^{[i]} 1/N_p \rangle \right\}_{i=1}^{N_p}$

3.2.2.2 Previous Research on Particle Filtering for Human Tracking

Particle filters are also commonly used for sensor data fusion. When several sensors are included in a particle filter, different measurement models are defined for each sensor type. These measurement models represent the likelihood of a sensor measurement to be generated from a specific state value. Generally, particle weights are computed from the sensors likelihood and thus particles that better match sensor measurements will have higher weights. Ababsa and Mallem [Ababsa-2007] develop a camera pose tracking system for AR applications which combines IMU and vision measurements in a SIR (Sample Importance Resampling) particle filter. This filter has two particles weighting channels and a common estimation module. Jourdan et al. [Jourdan-2005] also define two different measurement models (for an IMU and UWB beacon ranges) whose product is used as particle weight of a SIR particle filter. Hightower and Borriello [Hightower-2004] not only establish two different measurement models (for an infrared badge system and an ultrasound time-of-flight badge system) but also adjust the number of particles with a Kullback-Leibler distance adaptive sampling method.

Tao and Hu [Tao-2008] present an arm motion tracking system for rehabilitation which combines the measurements of an inertial sensor attached to the wrist and a group of markers tracked by a camera (see Figure 3.5.a). This system implements a layered sampling particle filter which executes consecutively two SIR particle filters: the first particle filter integrates the measurements from the inertial sensor and the visual markers and the second particle filter applies an arm physical geometry constraint to reduce the particle degeneracy problem. Zhang et al. [Zhang-2009] develop a similar system for human upper limb motion capture based on IMUs (see Figure 3.5.b). Firstly,

they apply a SIR particle filter to combine the measurements from the accelerometers and the gyroscopes of the IMUs. Afterwards, a second particle filter is applied to the estimates from the first filter in order to correct them with a fusion of geometric constraints which model the limitations of movement in the elbow. In addition, the proposed system includes reflective markers which are tracked by an independent optical motion capture system whose measurements are used as ground-truth values.



Figure 3.5. Sensor fusion systems for human tracking based on particle filters: (a) arm tracking system for rehabilitation, Source: [Tao-2008]; (b) full body tracking with inertial sensors; Source: [Zhang-2009].

Kalman and particle filters can also be used together. Evennou and Marx [Evennou-2006] implement a regularized particle filter [Musso-2001] for the indoor localization of an autonomous mobile device by fusing measurements from two inertial sensors (a gyroscope and a biaxial accelerometer) and a Wi-Fi positioning system. During the prediction step of the particle filter, inertial measurements corrected with a Kalman filter are integrated in the mobile evolution law. During the correction step of the particle filter, Wi-Fi position measurements, which are obtained from signal strengths with a fingerprinting technique, are introduced in the computation of the weight of the particles. Schon et al. [Schon-2007] define a multi-rate particle filter which integrates measurements from two sensors with different sampling rates (fast and slow sensors). A set of parallel Kalman filters (one filter per particle) is applied every time a fast sensor measurement is received. Importance weight and resampling steps of the particle filter are executed only when a slow sensor measurement is received. Thereby, the sampling rate of the particle filter is reduced and the overall performance is increased. Rao-Blackwellized (or Marginalized) particle filters are another way to combine Kalman and particle filters. A Rao-Blackwellized particle filter partitions the state vector in two components: a linear Gaussian component which is computed with a standard Kalman filter and a non-linear non-Gaussian component which is computed with a sequential Monte Carlo technique (like particle filters). Giremus et al. [Giremus-2004] implement a Rao-Blackwellized particle filter in order to integrate inertial and GPS measurements in a navigation system.

In this thesis, four different fusion algorithms have been implemented. The first one is a simple scheme which introduces the principle of recalculating the transformation matrix between the tracking systems. This first approach is improved by three novel fusion algorithms based on a Kalman filter, a particle filter and a combination of a Kalman filter and a particle filter, respectively. All of them are described and analyzed in the next sections and constitute one of the main contributions of this thesis.

3.3 First Approach for Sensor Fusion in Human Tracking

3.3.1 Coordinate Transformation

The first step to combine the global position measurements of both tracking systems is to represent them in the same coordinate system. The position measurements from the UWB localization system are represented in the frame U . This is a static coordinate system which is established according to the fixed positions where the UWB sensors are installed. In the configuration employed in this research, the frame U is situated under the master sensor D2 of the UWB system (see Figure 2.13.b).

The global position measurements from the inertial motion capture system are represented in the frame G , which is established each time the human operator initializes the system. During this initialization process, the human operator keeps still for a few seconds and meanwhile the coordinate system of each IMU is instantiated at the current position of the sensor and its orientation is aligned with the direction of the Earth's North magnetic pole. The frame G corresponds to the coordinate system of the IMU situated over the hips of the human operator (i.e. the root node of the skeleton) but translated to the floor. This frame G is dynamic and is situated at the position of each initialization of the inertial system. Because of this fact, the frame U of the UWB system has been selected as the reference coordinate system for all the global position measurements since it allows comparing easily the global position of the human operator with the position of the robots in the environment. Therefore, all global position measurements from the system G should be transformed into the frame U . Figure 3.6 shows the spatial relation between the coordinate systems of the inertial motion capture system and the UWB localization system.

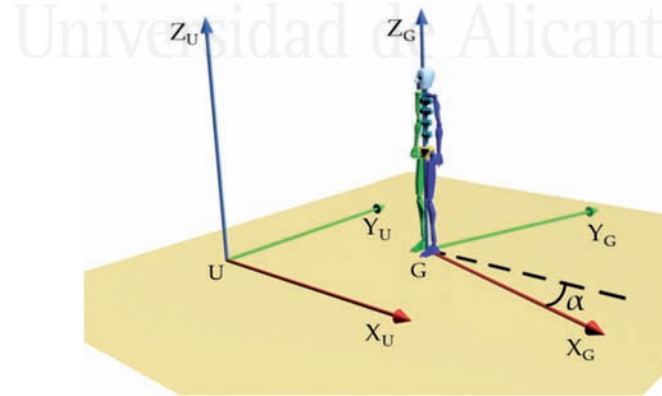


Figure 3.6. Coordinate systems of the tracking systems (UWB system and inertial motion capture system).

As shown in Figure 3.6, the XY planes of frames U and G are parallel to the plane of the floor of the environment. Therefore, between them there is only a translation (x_G^U, y_G^U, z_G^U) and a rotation about the Z axis by an angle α . A position vector \mathbf{p} defined in the reference frame of the inertial motion capture system as $\mathbf{p}^G = (x^G, y^G, z^G)$ can be represented in the UWB coordinate system $\mathbf{p}^U = (x^U, y^U, z^U)$ by applying the following equation:

$$\mathbf{p}^U = {}^U T_G \cdot \mathbf{p}^G = \text{Trans}(x_G^U, y_G^U, z_G^U) \cdot \text{Rot}(\mathbf{Z}^U, \alpha) \cdot \mathbf{p}^G \quad (3.29)$$

By expanding (3.29), the following expression for the transformation matrix ${}^U T_G$ is obtained:

$$\begin{bmatrix} x^U \\ y^U \\ z^U \\ 1 \end{bmatrix} = {}^U T_G \cdot \begin{bmatrix} x^G \\ y^G \\ z^G \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 & x_G^U \\ \sin(\alpha) & \cos(\alpha) & 0 & y_G^U \\ 0 & 0 & 1 & z_G^U \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x^G \\ y^G \\ z^G \\ 1 \end{bmatrix} \quad (3.30)$$

The parameter α in this transformation matrix is a known constant value which represents the angle between the Y axes of both frames. In particular, α corresponds to the angle between the Y axis of the static frame U and the vector towards the Earth's North magnetic pole because the Y axis of frame G is orientated in this direction during the initialization of the inertial system. Therefore, the only unknown variables of the transformation matrix ${}^U T_G$ are the three coordinates of the translation vector from frame U . They can be calculated from (3.30) by substituting two corresponding measurements from both tracking systems:

$$x_G^U = x^U - x^G \cos(\alpha) + y^G \sin(\alpha) \quad (3.31)$$

$$y_G^U = y^U - x^G \sin(\alpha) - y^G \cos(\alpha) \quad (3.32)$$

$$z_G^U = z^U - z^G \quad (3.33)$$

After obtaining the parameters of the transformation matrix ${}^U T_G$, all global position measurements from the inertial motion capture system will be transformed to the UWB frame by applying (3.30).

3.3.2 Transformation Recalculation Algorithm

The computation of the transformation matrix ${}^U T_G$ described in the previous section permits to develop an initial approach for a fusion algorithm. A simple approach involves using each measurement registered by any of the two tracking systems as an estimate of the global position of the human operator. The UWB measurements remain unchanged while the measurements from the inertial motion capture system are transformed by ${}^U T_G$. Nevertheless, if the transformation matrix ${}^U T_G$ is only computed at the beginning of the fusion algorithm, the translational errors accumulated by the inertial motion capture system (i.e. drift) are not corrected because each UWB measurement is not taken into consideration for the subsequent measurements from the inertial system. In order to solve this limitation, the transformation matrix has to be recalculated each time a new measurement from the UWB system is registered. Figure 3.7 presents the main steps of this first approach to define a basic fusion algorithm based on the recalculation of the transformation matrix ${}^U T_G$ and its application to the measurements of the inertial system.

Firstly, the fusion algorithm waits until the first two position measurements \mathbf{p}_{uwb}^U and $\mathbf{p}_{inertial}^G$ from the two tracking systems are registered and then, it calculates the first value of the transformation matrix ${}^U T_G$ by applying (3.31)-(3.33). Afterwards, whenever a new measurement $\mathbf{p}_{inertial}^G$ from the inertial system is registered, it is transformed into

$p_{inertial}^U$ in frame U by applying (3.30) and then, it is stored as an estimate of the current global position of the human operator. When a new measurement p_{uwb}^U from the UWB system is obtained, the system waits for another measurement from the inertial system in order to recalculate the transformation matrix with (3.31)-(3.33).

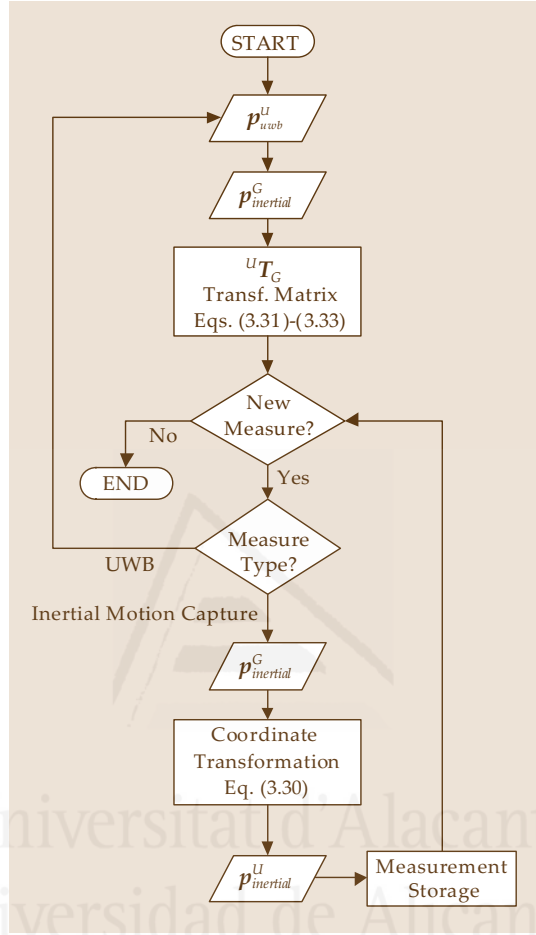


Figure 3.7. Diagram of the fusion algorithm based on transformation recalculation.

This recalculation process prevents the propagation of the drift from the inertial measurements which are registered before a UWB measurement towards the inertial measurements which are registered after this UWB measurement. In other words, each UWB measurement acts as a barrier for the inertial drift by triggering the recalculation of the transformation matrix which is applied to the subsequent inertial measurements. In addition, the fusion algorithm keeps the high update rate of the inertial system. Thereby, this algorithm is able to combine the advantages of both systems: the accuracy of the UWB localization system for correcting the inertial drift and the high sampling rate of the inertial system for registering quick movements of the human operator.

However, this algorithm has an important limitation because it does not take into account the translational error of the UWB system. In fact, the direct application of the UWB measurements in the transformation recalculation involves that they are always considered as perfect values. This simplification is too restrictive for a real-time human-robot interaction system because it generates discontinuities in the final fusion tra-

jectory of the human operator as a result of the errors between subsequent measurements. Section 3.5.1 presents several experiments where this drawback is analysed.

Therefore, this chapter proposes in the next section the development of three new fusion algorithms based on Bayesian Filtering [Corrales-2010a]. These fusion algorithms consider not only the drift of the inertial system but also the translational error of the UWB system. Although these Bayesian Filtering algorithms involve an improvement over the simple fusion algorithm described in this section, they also use the transformation recalculation principle to apply the corrections of the UWB measurements over the inertial measurements. The application of this principle is one of the contributions of this thesis.

3.4 Bayesian Filtering for Human Tracking

This section develops three fusion algorithms based on Bayesian Filtering. All of them use the transformation recalculation principle described in the previous section but they are based on three different techniques. The first proposed algorithm is based on a modified Kalman filter, the second proposed algorithm is based on a modified SIR particle filter and the third proposed algorithm implements a combination of the previous two algorithms. These algorithms are described in detail in the following sections.

3.4.1 Kalman Filter Fusion Approach

The first proposed fusion algorithm is based on the standard Kalman filter. In particular, the standard Kalman filter has been modified in order to incorporate sensor measurements from the two tracking systems by considering their complementary features: high sampling rate of the inertial motion capture system and location accuracy of the UWB system.

The state x_t which should be estimated by the Kalman filter developed for the work described in this thesis is composed by the coordinates $p = (x, y, z)$ of the global position of the human operator in the environment. The prediction step of this Kalman filter uses a state evolution model where the estimate of the state \hat{x}_t is obtained from the current measurement of the inertial motion capture system $z_{inertial}(t)$ plus a Gaussian noise $w_{inertial}(t)$:

$$\hat{x}_t = z_{inertial}(t) + w_{inertial}(t) \quad (3.34)$$

This model is implemented in the Kalman filter prediction step by using the following two equations:

$$\hat{x}_t = A z_{inertial}(t) \quad (3.35)$$

$$P_t = A P_{t-1} A^T + Q \quad (3.36)$$

The matrix A represents the state transition model of (3.34) and thus it is a 3×3 identity matrix (with ones on the main diagonal and zeros elsewhere) which uses the last measurement from the motion capture system as the current state estimate \hat{x}_t . P_t is the error covariance matrix which represents the accuracy of the state estimate \hat{x}_t . Q is the

3×3 covariance matrix of the error $w_{inertial}(t)$ of the inertial motion capture system. It is a diagonal matrix whose terms represent the variance of the position errors of the inertial system. The terms of this matrix are adjusted experimentally from a representative group of experiments so that the errors of the final state estimates are minimum.

The correction step of the filter is based on an observation model where the measurements from the UWB system $\hat{z}_{uwb}(t)$ are approximated by the last estimate of the prediction step \hat{x}_{t-1} plus a Gaussian noise $w_{uwb}(t)$:

$$\hat{z}_{uwb}(t) = \hat{x}_{t-1} + w_{uwb}(t) \quad (3.37)$$

This observation model is implemented in the Kalman filter correction step by using the following three equations:

$$K_t = P_{t-1} H^T (H P_{t-1} H^T + R)^{-1} \quad (3.38)$$

$$\hat{x}_t = \hat{x}_{t-1} + K_t (z_{uwb}(t) - H \hat{x}_{t-1}) \quad (3.39)$$

$$P_t = (I - K_t H) P_{t-1} \quad (3.40)$$

Position measurements from the UWB system are used as observations $z_{uwb}(t)$ in this step in order to correct the predicted position \hat{x}_{t-1} calculated from the motion capture system. The matrix H represents the measurement model of (3.37) and it is a 3×3 identity matrix because UWB measurements have the same dimension as the state. R is the 3×3 covariance matrix of the error of the error $w_{uwb}(t)$ of the UWB system. It is a diagonal matrix whose terms represent the variance of the position errors of the UWB system. The terms of this matrix are adjusted experimentally from a representative group of experiments so that the errors of the final state estimates are minimum.

These prediction and correction steps of the Kalman filter are implemented by the algorithm shown in Table 3.3. The main advantage of this algorithm over previous sensor fusion techniques based on Kalman filtering is that only one step of the algorithm (prediction or correction) is executed each time a measurement is obtained. The prediction step (equations (3.35) and (3.36)) is executed when a measurement from the inertial motion capture system is received (line 9 of Table 3.3) while the correction step (equations (3.38), (3.39) and (3.40)) is executed when UWB measurements are received (line 12 of Table 3.3). Thereby, the execution time of the algorithm is smaller than other fusion algorithms based on the Kalman filter ([Foxlin-1996], [Foxlin-2005], [Roetenberg-2007b], [Caron-2006] and [You-2001]) which complete both filter steps at each iteration. Furthermore, the direct use of the measurements from the inertial motion capture system in the prediction step involve a better approximation of the human motion than other theoretical transition state models which suppose a restricted motion based on constant velocities, constant accelerations or Wiener-process accelerations ([Li-2000], [Caron-2006]).

This algorithm reduces the drawbacks of both tracking systems by combining their complementary features. On one hand, the error accumulated by the inertial motion capture system is removed after each correction step execution because the transformation matrix uT_G is re-calculated (line 6 of Table 3.3) with the last state estimate obtained from an UWB measurement. This new matrix uT_G is used in the following pre-

diction step executions in order to represent the inertial system measurements in the UWB coordinate system (line 8 of Table 3.3) and thus the drift of previous inertial measurements is corrected. On the other hand, the low sampling rate of the UWB measurements does not reduce the latency of the system because the inertial system measurements registered between each pair of UWB measurements are used in the prediction step in order to obtain an estimate of the system's state (line 9 of Table 3.3). Thereby, this algorithm provides state estimates with a high sampling rate (30-120 Hz) which is suitable to track quick human movements.

Table 3.3. Fusion algorithm based on Kalman filter.

Algorithm: Sensor Fusion based on Kalman Filter**Input:** A, H, P_1, Q, R, α **Implementation:**

```

01:  $X = \emptyset$  // Set with the position estimates
02: Initialize  ${}^U T_G$  with the first two measurements [Eqs. (3.31), (3.32), (3.33)]
03: for each sensor measurement  $z(t)$ 
04:   if  $z(t)$  is from the inertial system [ $z_{inertial}(t)$ ]
05:     if  $z(t-1)$  is from the UWB system
06:       Recalculate  ${}^U T_G$  with  $z_{inertial}(t)$  and  $x_{t-1}$  [Eqs. (3.31), (3.32), (3.33)]
07:     end if
08:     Transform  $z_{inertial}(t)$  from frame  $G$  to frame  $U$  [Eq. (3.30)]
09:     [ $x_t, P_t$ ] = KalmanPrediction( $A, Q, z_{inertial}(t), P_{t-1}$ ) [Eqs. (3.35), (3.36)]
10:     Store  $x_t$  as state estimate for time  $t$ :  $X = X \cup \{x_t\}$ 
11:   else if  $z(t)$  is from the UWB system [ $z_{uwb}(t)$ ]
12:     [ $x_t, P_t$ ] = KalmanCorrection( $H, R, x_{t-1}, z_{uwb}(t), P_{t-1}$ ) [Eqs. (3.38), (3.39), (3.40)]
13:     Store  $x_t$  as state estimate for time  $t$ :  $X = X \cup \{x_t\}$ 
14:   end if
15: end for
Output:  $X$ 

```

3.4.2 Particle Filter Fusion Approach

Another widespread technique for implementing Bayesian filters is the particle filter. In this thesis, a fusion algorithm derived from a particle filter algorithm with a Sampling Importance Resampling (SIR) scheme [Gordon-1993] has been developed in order to compare it with the previous approach based on a modified Kalman filter.

For the human tracking application presented here, the prediction step of the SIR particle filter obtains a set S_t of N_p particles $x_t^{[i]}$ from the state transition distribution, which is modeled by adding a random Gaussian noise v_i of zero mean and covariance matrix Q to the position measurement from the motion capture system $z_{inertial}(t)$ (see (3.41)). The covariance matrix Q represents the error covariance of the motion capture system and its terms are computed as explained in the previous section.

$$S_t = \left\{ \left\langle x_t^{[i]} = z_{inertial}(t) + v_i \mid w_t^{[i]} = 1/N_p \right\rangle \right\}_{i=1}^{N_p} \quad \text{where } v_i \sim \mathcal{N}(0, Q) \quad (3.41)$$

The Gaussian error v_i in (3.41) is obtained from the Cholesky decomposition C of the covariance matrix $Q = C^T C$ and a normally distributed random vector $Z \sim \mathcal{N}(0, I)$:

$$v_i = C^T \cdot Z \sim (0, C^T \cdot C) = \mathcal{N}(0, Q) \quad (3.42)$$

In the importance weight update step, the weights $w_t^{[i]}$ of the particles are updated from the measurement model $p(z_{uwb}(t) | x_{t-1}^{[i]})$. This probability is calculated as a multi-variate normal distribution $\mathcal{N}(z_{uwb}(t), R)$ with the UWB measurement $z_{uwb}(t)$ as mean and R as covariance matrix. This normal distribution is evaluated for each particle $x_{t-1}^{[i]}$ of the prediction step (3.41). The covariance matrix R represents the error covariance of the UWB system and its terms are computed as explained in the previous section. Finally, the weights of all the particles will be normalized to ensure that they sum up to one.

$$w_t^{[i]} \sim \mathcal{N}(z_{uwb}(t), R) = \frac{1}{(2\pi)^{3/2} \det(R)^{1/2}} \cdot \exp\left(-\frac{1}{2}(x_{t-1}^{[i]} - z_{uwb}(t))^T R^{-1}(x_{t-1}^{[i]} - z_{uwb}(t))\right) \quad (3.43)$$

As described in Section 3.2.2.1, the SIS particle filter has to include an additional resampling step after the weight update in order to reduce the influence of the degeneracy problem. The resampling step reduces the degeneracy problem by defining a new particle set where state estimates with high weights are represented by several particles while state estimates with low weights are removed. The resampling algorithm draws with replacement N_p particles from the initial particle set S_t . The probability of drawing each particle is given by its importance factor. There are several algorithms to implement this drawing process [Douc-2005]. The most commonly used technique is the *multinomial resampling* which generates N_p random numbers $\{u_i\}_{i=1}^{N_p}$ in the interval $[0, 1)$ and then uses a binary search to find the index j (and the corresponding particle $x_t^{[j]}$) of the first cumulative probability distribution D_j from the normalized particle weights which is higher than u_i :

$$D_{j-1} < u_i \leq D_j \text{ where } D_j = \sum_{k=1}^j w_t^{[k]} \text{ and } D_0 = 0 \quad (3.44)$$

This algorithm has a computational cost of $O(N_p \log N_p)$ because it requires a binary search for each new particle. Instead of using this general resampling scheme, the *systematic resampling* algorithm [Carpenter-1999] has been implemented in this thesis because it has a better computational cost of $O(N_p)$ and it only needs to execute once the expensive operation of generating a random number [Hol-2006]. Table 3.4 contains the pseudo-code of the implemented resampling algorithm.

The implemented algorithm is described in detail in Table 3.5. It has a similar general structure than the Kalman filter algorithm described in Table 3.3: Motion capture measurements are used in the prediction step (line 9 of Table 3.5) and UWB measure-

ments are used in the importance factor calculation step (line 12 of Table 3.5). Just after registering a UWB measurement, the transformation matrix ${}^U T_G$ is recalculated (line 6 of Table 3.5) in order to correct the error accumulated by the previous measurements.

Table 3.4. Systematic Resampling Algorithm.

Algorithm: Systematic Resampling**Input:** $S_t = \{\langle x_t^{[i]} | w_t^{[i]} \rangle\}_{i=1}^{N_p}$ **Implementation:**

```

01:  $\tilde{S}_t = \emptyset$  // Initialize new set of resampled particles
02:  $j = 1$  // Initialize index
03:  $T = \text{Random\_Number}(0, 1/N_p)$  // Generate random number in interval  $[0, 1/N_p)$ 
04:  $D = w_t^{[1]}$  // Initialize cumulative probability distribution of weights
05: for  $i = 1$  to  $N_p$ 
06:   while  $T > D$  // Search for the first cumulative probability higher than  $T$ 
07:      $j = j + 1$  // Update index of particle
08:      $D = D + w_t^{[j]}$  // Update cumulative probability
09:   end
10:    $\tilde{S}_t = \tilde{S}_t \cup \{\langle x_t^{[j]} | 1/N_p \rangle\}$  // Add particle with index  $j$  to the new set of particles
11:    $T = T + 1/N_p$  // Update the random number
12: end for
Output:  $\tilde{S}_t = \{\langle x_t^{[i]} | 1/N_p \rangle\}_{i=1}^{N_p}$ 

```

Table 3.5. Fusion algorithm based on SIR particle filter.

Algorithm: Sensor Fusion based on SIR Particle Filter**Input:** N_p, Q, R, α **Implementation:**

```

01:  $X = \emptyset$  // Set with position estimates
02: Initialize  ${}^U T_G$  with the first two measurements [Eqs. (3.31), (3.32), (3.33)]
03: for each sensor measurement  $z(t)$ 
04:   if  $z(t)$  is from the inertial system [ $z_{inertial}(t)$ ]
05:     if  $z(t-1)$  is from the UWB system
06:       Recalculate  ${}^U T_G$  with  $z_{inertial}(t)$  and  $x_{t-1}$  [Eqs. (3.31), (3.32), (3.33)]
07:     end if
08:     Transform  $z_{inertial}(t)$  from frame  $G$  to frame  $U$  [Eq. (3.30)]
09:      $S_t = \text{PF\_Prediction}(z_{inertial}(t), Q)$  [Eqs. (3.41), (3.42)] // Update states
10:     Calculate mean  $x_t$  of particles in  $S_t$  and store it as estimate:  $X = X \cup \{x_t\}$ 
11:   else if  $z(t)$  is from the UWB system [ $z_{uwb}(t)$ ]
12:      $\{w_t^{[i]}\}_{i=1}^{N_p} = \text{PF\_ImportanceFactor}(S_{t-1}, z_{uwb}(t), R)$  [Eq. (3.43)] // Update weights
13:      $S_t = \{\langle x_{t-1}^{[i]} | w_t^{[i]} \rangle\}$  // Generate new particle set with new weights and previous states
14:      $S_t = \text{PF\_SystematicResampling}(S_t)$  [Table 3.4]
15:     Calculate mean  $x_t$  of particles in  $S_t$  and store it as estimate:  $X = X \cup \{x_t\}$ 
16:   end if
17: end for
Output:  $X$ 

```


The importance factor and resampling steps will only be executed with the frequency of UWB measurements. Thereby, the execution time of the algorithm is reduced with regard to previously developed particle filter algorithms ([Ababsa-2007], [Hightower-2004]), which have different particle weighing channels for each sensor and have always to perform the weighing and resampling steps.

3.4.3 Kalman-particle Combination Approach

The third fusion approach implemented in this thesis relies on combining the previous two Bayesian techniques: Kalman filter and particle filter. The Kalman filter is applied to measurements from the inertial motion capture system while the SIR particle filter is applied to UWB measurements. Previous state estimates $\hat{\mathbf{x}}_{t-1}$ are used in the prediction steps of these filters in order to interchange measurements between them and thus correct each other's errors.

The state evolution model of the prediction step in the Kalman filter is based on the previous state estimate $\hat{\mathbf{x}}_{t-1}$ plus a Gaussian noise $w_{KF}(t)$ with zero mean and covariance \mathbf{Q}_{KF} (which represents the dispersion of the difference between the current position and the previous one):

$$\hat{\mathbf{x}}_t^- = \hat{\mathbf{x}}_{t-1} + w_{KF}(t) \quad (3.45)$$

This prediction step is implemented by the following equations (where \mathbf{A} is a 3×3 identity matrix):

$$\hat{\mathbf{x}}_t^- = \mathbf{A}\hat{\mathbf{x}}_{t-1} \quad (3.46)$$

$$\mathbf{P}_t^- = \mathbf{A}\mathbf{P}_{t-1}\mathbf{A}^T + \mathbf{Q}_{KF} \quad (3.47)$$

The observation model of the correction step in the Kalman filter approximates the measurements of the inertial motion capture system from the state estimate of the prediction step $\hat{\mathbf{x}}_t^-$ plus a Gaussian noise $w_{inertial}(t)$ with zero mean and covariance \mathbf{R}_{KF} (which is equal to the covariance matrix \mathbf{Q} of the Kalman filter in Section 3.4.1):

$$\hat{\mathbf{z}}_{inertial}(t) = \hat{\mathbf{x}}_t^- + w_{inertial}(t) \quad (3.48)$$

This observation model is implemented in the Kalman filter correction step using the following three equations (where \mathbf{H} is a 3×3 identity matrix and $\mathbf{z}_{inertial}(t)$ is the motion capture system measurement):

$$\mathbf{K}_t = \mathbf{P}_t^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_t^- \mathbf{H}^T + \mathbf{R}_{KF})^{-1} \quad (3.49)$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_t^- + \mathbf{K}_t (\mathbf{z}_{inertial}(t) - \mathbf{H}\hat{\mathbf{x}}_t^-) \quad (3.50)$$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{H}) \mathbf{P}_t^- \quad (3.51)$$

Each time a UWB measurement is registered, the SIR particle filter is executed. In the prediction step of this particle filter, a set S_t of N_p particles $\mathbf{x}_t^{[i]}$ is obtained by adding a random Gaussian noise v_i with zero mean and covariance \mathbf{Q}_{PF} to the state estimate $\hat{\mathbf{x}}_{t-1}$ obtained with the Kalman filter from the previous inertial measurement:

$$S_t = \left\{ \langle x_t^{[i]} = \hat{x}_{t-1} + v_i \mid 1/N_p \rangle \right\}_{i=1}^{N_p} \text{ where } v_i \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{PF}) \quad (3.52)$$

The Gaussian error v_i in (3.52) is obtained from the Cholesky decomposition of \mathbf{Q}_{PF} as in (3.42). The importance factor calculation and resampling steps of this particle filter are implemented in a similar way to the previous particle filter algorithm described in Section 3.4.2. The weights $w_t^{[i]}$ of the particles computed in the prediction step are obtained from a multivariate normal distribution $\mathcal{N}(z_{uwb}(t), \mathbf{R}_{PF})$ (see (3.53)) while the resampling step is implemented by a systematic resampling approach (see Table 3.4). The covariance matrix \mathbf{R}_{PF} contains the variance of the errors of the UWB system and it is equal to the \mathbf{R} covariance matrix of the particle filter in Section 3.4.2.

$$w_t^{[i]} \sim \mathcal{N}(z_{uwb}(t), \mathbf{R}_{PF}) = \frac{1}{(2\pi)^{3/2} \det(\mathbf{R}_{PF})^{1/2}} \cdot \exp\left(-\frac{1}{2} \left(x_t^{[i]} - z_{uwb}(t)\right)^T \mathbf{R}_{PF}^{-1} \left(x_t^{[i]} - z_{uwb}(t)\right)\right) \quad (3.53)$$

The fusion algorithm obtained from this combination of Kalman and particle filters is described in Table 3.6:

Table 3.6. Fusion algorithm based on the combination of a Kalman filter and a SIR particle filter.

Algorithm: Sensor Fusion based on Kalman-Particle Filter	
Input: $A, H, P_1, Q_{KF}, R_{KF}, \alpha, N_p, Q_{PF}, R_{PF}$	
Implementation:	
01:	$X = \emptyset$ // Set with position estimates
02:	Initialize ${}^U T_G$ with the first two measurements [Eqs. (3.31), (3.32), (3.33)]
03:	for each sensor measurement $z(t)$
04:	if $z(t)$ is from the inertial system $[z_{inertial}(t)]$
05:	if $z(t-1)$ is from the UWB system
06:	Recalculate ${}^U T_G$ with $z_{inertial}(t)$ and x_{t-1} [Eqs. (3.31), (3.32), (3.33)]
07:	end if
08:	Transform $z_{inertial}(t)$ from frame G to frame U [Eq. (3.30)]
09:	$[x_t, P_t] = \text{KalmanPrediction}(A, Q_{KF}, x_{t-1}, P_{t-1})$ [Eqs. (3.46), (3.47)]
10:	$[x_t, P_t] = \text{KalmanCorrection}(H, R_{KF}, z_{inertial}(t), x_t, P_t)$ [Eqs. (3.49), (3.50), (3.51)]
11:	Store x_t as state estimate for time t : $X = X \cup \{x_t\}$
12:	else if $z(t)$ is from the UWB system $[z_{uwb}(t)]$
13:	$S_t = \text{PF_Prediction}(x_{t-1}, Q_{PF})$ [Eq. (3.52)]
14:	$\{w_t^{[i]}\}_{i=0}^{N_p} = \text{PF_ImportanceFactor}(S_t, z_{uwb}(t), R_{PF})$ [Eq. (3.53)]
15:	$S_t = \left\{ \langle x_t^{[i]} \mid w_t^{[i]} \rangle \right\}$ // Combine states from step 13 with weights of step 14
16:	$S_t = \text{PF_SystematicResampling}(S_t)$ [Table 3.4]
17:	Calculate mean x_t of particles in S_t and store it as estimate for time t : $X = X \cup \{x_t\}$
18:	end if
19:	end for
Output: X	

This algorithm improves the performance of the particle filter algorithm presented in Section 3.4.2 because its execution frequency is reduced. The high rate measurements from the motion capture systems are processed with the Kalman filter (lines 9-10 of Table 3.6) and thus the particle filter is only executed when slow measurements from the UWB system are obtained (lines 13-16 of Table 3.6). In a similar way to the fusion algorithms of Section 3.4.1 and Section 3.4.2, the transformation matrix uT_G is recalculated (line 6 of Table 3.6) after an UWB measurement is received in order to correct the error accumulated by the inertial system, as in the previous fusion algorithms.

3.5 Experimental Results

3.5.1 Comparison between Transformation Recalculation Algorithm and Sensor Fusion based on Kalman Filter

Some experiments have been performed to compare the accuracy of the transformation recalculation algorithm from Section 3.3.2 and the Kalman filter fusion algorithm from Section 3.4.1. A human operator wearing a *Gypsygyro-18* inertial motion capture suit and an *Ubisense* UWB tag (see Section 2.3 for a detailed description of these two tracking systems) has walked along two different pre-established paths: a linear trajectory (3m) and a rectangular trajectory (8m). Both trajectories are defined in the same XY plane with a constant height (coordinate Z) since the errors in the vertical component are negligible in comparison with the errors in the horizontal plane. These trajectories are marked on the floor so that the human operator can follow them with precision. The points of these marked trajectories are used as ground-truth values for computing the position errors of the tracking systems. Each trajectory has been covered fifteen times by the human operator. Therefore, a total of 30 experiments (15 for each trajectory) have been developed.

In Figure 3.8.a, original measurements from the UWB system and the inertial motion capture (MoCap) system in one experiment (no. 14) of the linear trajectory are represented in the same coordinate system. This plot shows the advantages and disadvantages of both tracking systems. On one hand, motion capture measurements have a higher sampling rate but they accumulate an error which increases through time. On the other hand, UWB measurements have a smaller sampling rate but do not accumulate errors through time.

Figure 3.8.b and Figure 3.8.c depict the position estimates obtained from the fusion algorithms. The trajectories obtained by the Kalman filter algorithm (see Figure 3.8.c) are more continuous than the trajectories from the transformation recalculation algorithm (see Figure 3.8.b). In addition, the position estimates of the Kalman filter have smaller errors than those of the transformation recalculation algorithm (see Figure 3.8.d).

In Figure 3.9.a, Figure 3.9.b and Figure 3.9.c, one experiment (no. 3) of the rectangular trajectory is represented and similar results are obtained. The estimates obtained from the Kalman filter are more precise than those from the transformation recalculation algorithm (see Figure 3.9.d).

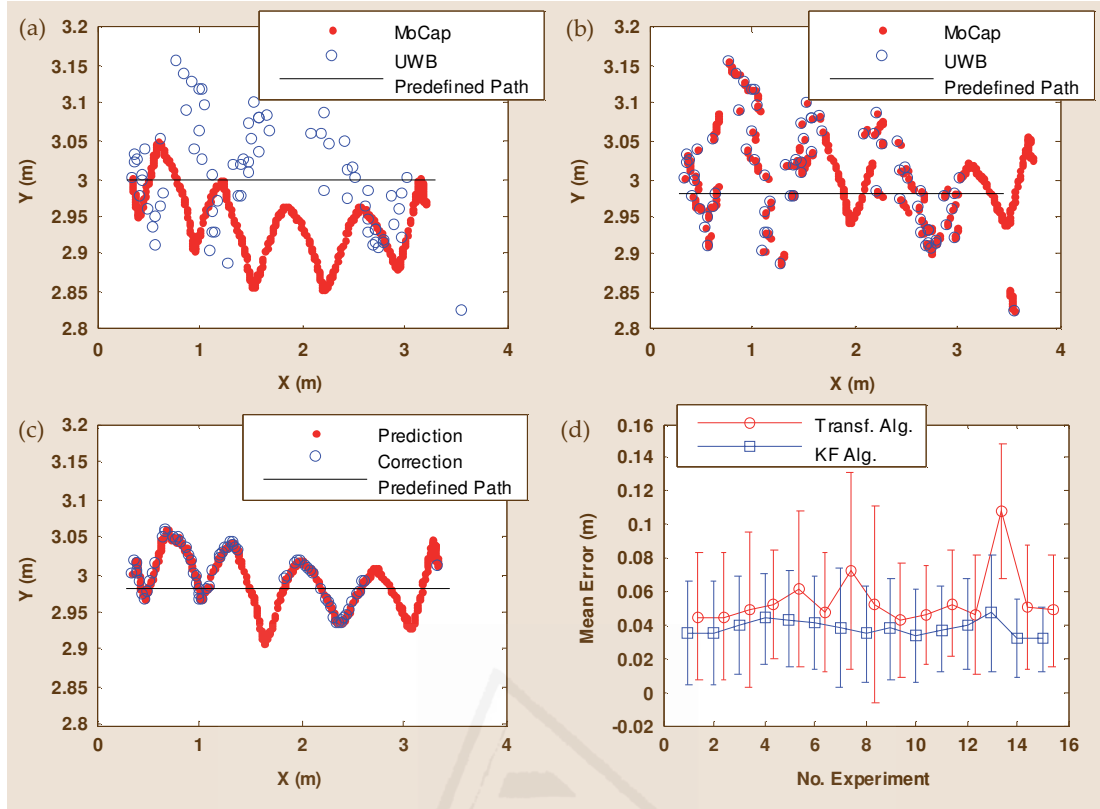


Figure 3.8. Linear trajectory experiments: (a) original position measurements in experiment no. 14; (b) position measurements obtained from the transformation recalculation algorithm in experiment no. 14; (c) position estimates obtained from the prediction and correction steps of the Kalman filter algorithm in experiment no. 14; (d) mean error and standard deviation of the position estimates from the two fusion algorithms in the 15 linear path experiments.

The higher accuracy of the Kalman filter estimates is due to the fact that this algorithm takes into account the errors of both tracking systems while the transformation recalculation algorithm only corrects the errors of the motion capture system. The transformation recalculation algorithm uses the UWB measurements as correct positions in order to calculate the transformation matrix and thus correct the motion capture errors. However, the errors of the UWB system are not corrected and cause discontinuities in the trajectories (as shown in Figure 3.8.b and Figure 3.9.b).

In conclusion, the Kalman filter fusion algorithm obtains more accurate position estimates and continuous trajectories than the transformation recalculation algorithm. In the following section, the Kalman filter fusion algorithm is compared with the other two approaches based on particle filters in order to verify which sensor fusion algorithm based on Bayes Filtering is the best solution for real-time human tracking in human-robot interaction tasks.

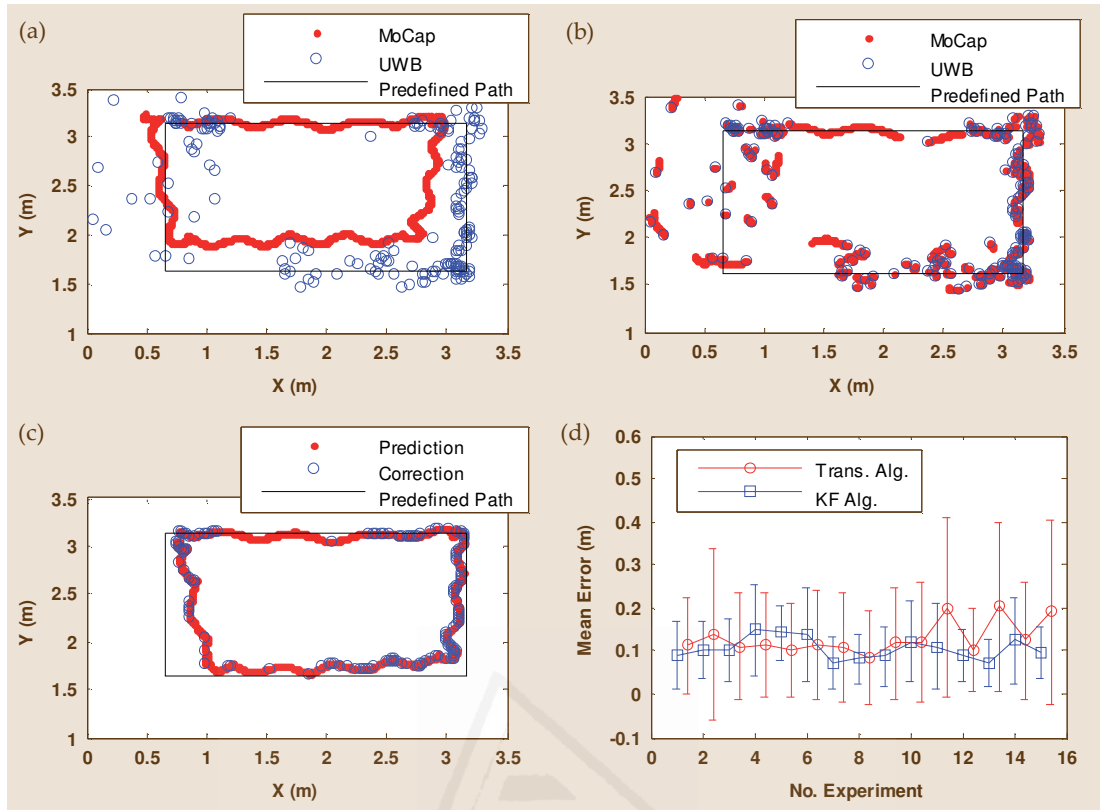


Figure 3.9. Rectangular trajectory experiments: (a) original position measurements from both tracking systems in experiment no. 3; (b) position measurements obtained from the transformation recalculation algorithm in experiment no. 3; (c) position estimates obtained from the prediction and correction steps of the Kalman filter algorithm in experiment no. 3; (d) mean error and standard deviation of the position estimates from the two fusion algorithms in the 15 rectangular path experiments.

3.5.2 Comparison between Sensor Fusion Algorithms based on Bayesian Filtering

In this section, the Kalman filter fusion algorithm described in Section 3.4.1, the particle filter fusion algorithm described in Section 3.4.2 and the Kalman-particle fusion algorithm described in Section 3.4.3 are applied over the trajectories of the previous section. As previously stated, these experiments consist on covering a linear path of 3 m and a rectangular path of 8 meters fifteen times each one. All the algorithms have been tested in the same computer: a PC with an Intel Pentium 4 2.8 GHz processor and 760 MB of RAM). In the case of the particle filter algorithms, a set of 100 particles is used.

The global position measurements obtained in one experiment (no. 10) of the linear path are depicted in Figure 3.10. Figure 3.10.a shows the original position measurements registered by the two tracking systems (inertial motion capture system and UWB system) in the same coordinate system. Figure 3.10.b, Figure 3.10.c and Figure 3.10.d depict the position estimates which are obtained in this experiment by applying the Kalman filter, the particle filter and the Kalman-particle filter fusion algorithms, respectively.

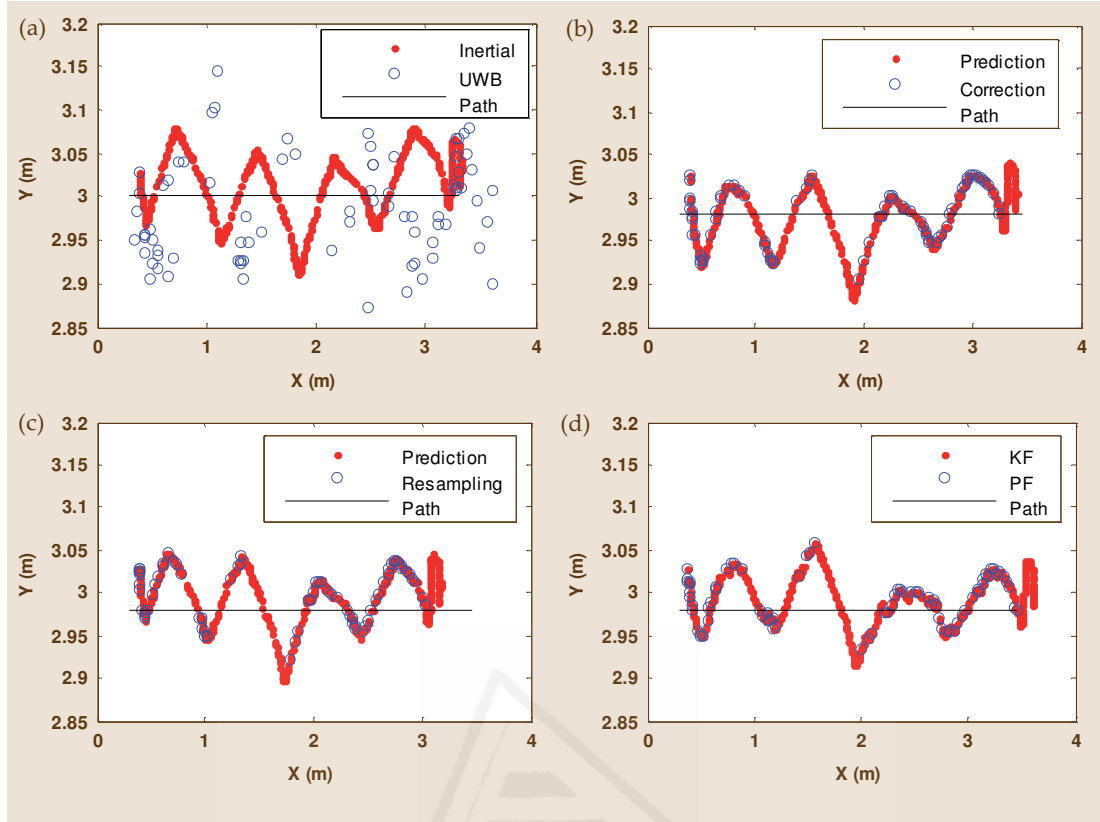


Figure 3.10. Linear trajectory experiments: (a) original position measurements in experiment no. 10; (b) position estimates from the Kalman filter algorithm in the prediction and correction steps; (c) position estimates from the particle filter algorithm in the prediction and resampling steps; (d) position estimates from the Kalman-particle filter algorithm in the Kalman filter (KF) and the particle filter (PF) steps.

Figure 3.11 shows the error statistics (error mean and standard deviation) of all the position estimates computed by the three fusion algorithms along the fifteen experiments of the linear path. These statistics are obtained by calculating the Euclidean distance between each position measurement and the corresponding ground-truth point in the linear path. The Kalman-particle filter algorithm has a global mean error ($0.037 \pm 0.029\text{m}$) very similar to the global error of the Kalman filter algorithm ($0.038 \pm 0.029\text{m}$). The particle filter algorithm has a slightly higher global error ($0.044 \pm 0.029\text{m}$).

Figure 3.12.a depicts the number of measurements of the inertial motion capture system and the number of measurements of the UWB localization system which are processed by the fusion algorithms at each experiment of the linear path. The number of measurements of the inertial system is always substantially higher than the number of measurements of the UWB system because of its higher sampling rate. Figure 3.12.b shows the execution time which is spent by the fusion algorithms to combine all the measurements of both tracking systems. The Kalman filter is the least expensive algorithm in terms of execution time with a global mean time of 0.077s. The Kalman-particle filter algorithm has an intermediate computational cost (global mean execution time: 1.137s) between the Kalman filter and the particle filter (global mean execution time: 1.523s).

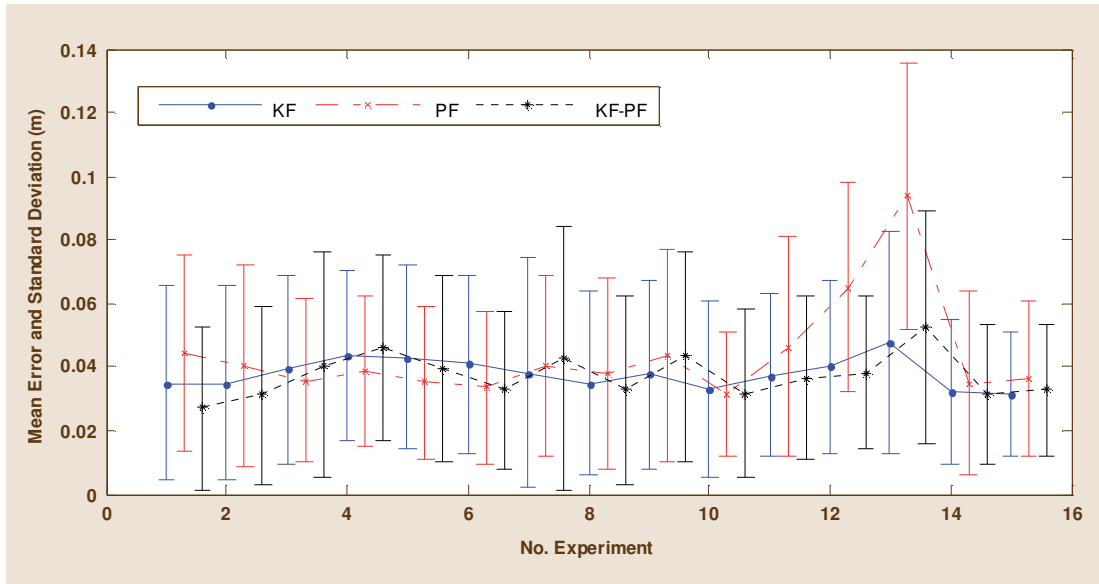


Figure 3.11. Mean error and standard deviation of the position estimates from the three fusion algorithms (KF: Kalman filter, PF: Particle filter, KF-PF: Kalman-particle filter) in the linear path experiments.

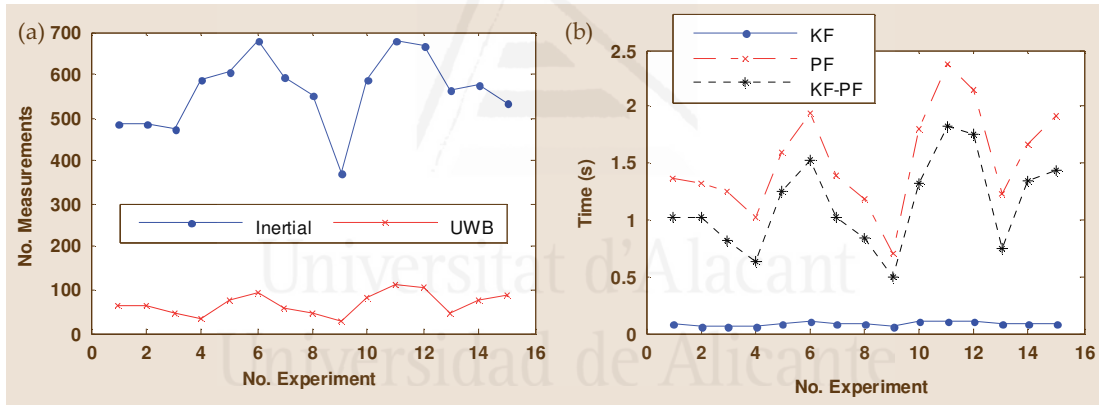


Figure 3.12. Computational cost of the linear path experiments: (a) number of measurements registered by the two tracking systems (inertial and UWB); (b) total execution time of the three fusion algorithms (KF: Kalman filter, PF: Particle filter, KF-PF: Kalman-particle filter).

Figure 3.13.a depicts the original global position measurements registered by the tracking system during one experiment (no. 7) of the rectangular path. The error accumulation of the inertial system is more evident in this trajectory because it is longer than the previous linear path (Figure 3.10.a). The differences in the accuracy of the position estimates obtained from the three fusion algorithms are also more important and they can be appreciated in Figure 3.13.b, Figure 3.13.c and Figure 3.13.d. The trajectory estimated by the Kalman-particle filter algorithm (Figure 3.13.d) is the most precise because it has a tighter fit to the real rectangular trajectory. The trajectory estimated by the particle filter algorithm (Figure 3.13.c) is the least precise because the estimated positions are farther from the real path.

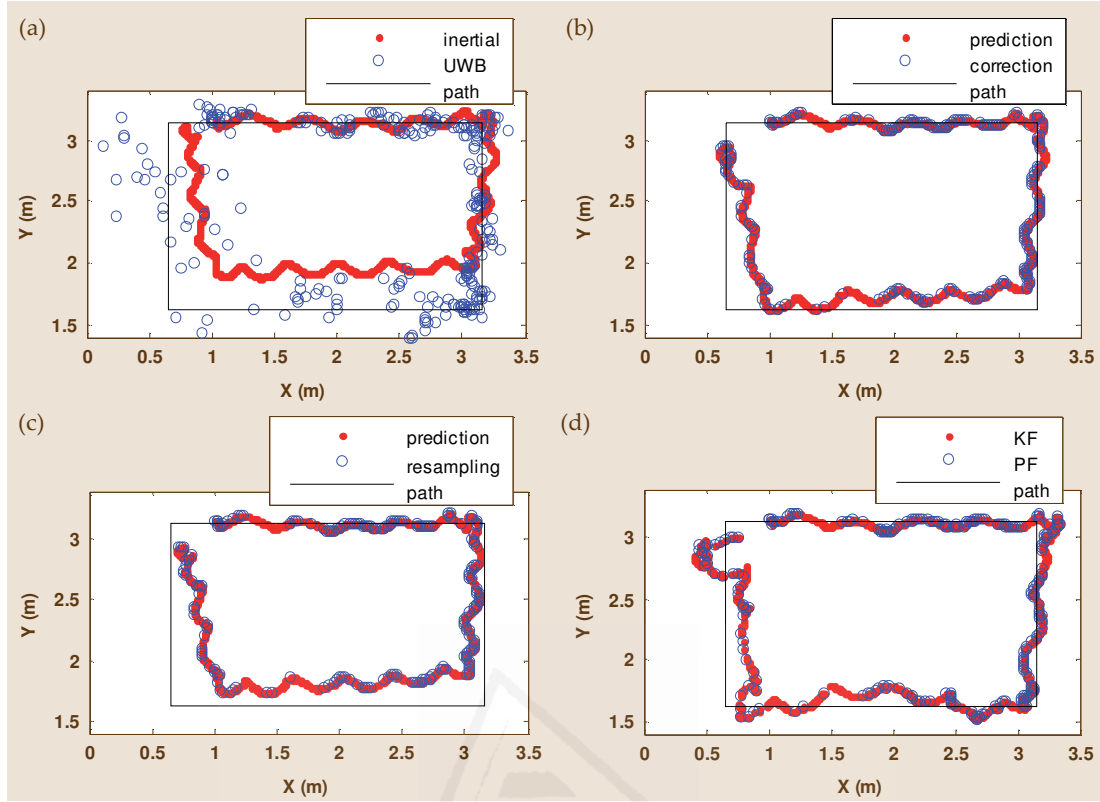


Figure 3.13. Rectangular trajectory experiments: (a) original position measurements in experiment no. 7; (b) position estimates from the Kalman filter algorithm in the prediction and correction steps; (c) position estimates from the particle filter algorithm in the prediction and resampling steps; (d) position estimates from the Kalman-particle filter algorithm in the Kalman filter (KF) and the particle filter (PF) steps.

A more in-depth description of the errors of the position estimates computed by the three fusion algorithms is shown in Figure 3.14. As it was stated before, the Kalman-particle filter algorithm obtains more accurate position estimates (global mean error: $0.063 \pm 0.055\text{m}$) than the Kalman filter (global mean error: $0.105 \pm 0.076\text{m}$) and the particle filter algorithms (global mean error: $0.184 \pm 0.127\text{m}$).

Figure 3.15.a shows the number of measurements which are combined by the fusion algorithms in each experiment of the rectangular trajectory. In this case, the number of processed measurements is higher because the path is longer. Figure 3.15.b depicts the execution time of the fusion algorithms to combine these measurements. On one hand, the number of measurements of the inertial system is the main factor that determines the different execution times of the same fusion algorithm at different experiments. On the other hand, when the execution times of the three fusion algorithms are compared, the same results than in the linear path experiments are obtained. The Kalman filter algorithm has a smaller computational cost (global mean execution time: 0.561s) than the Kalman-particle filter (global mean execution time: 4.358s) and the particle filter algorithms (global mean execution time: 5.947s).

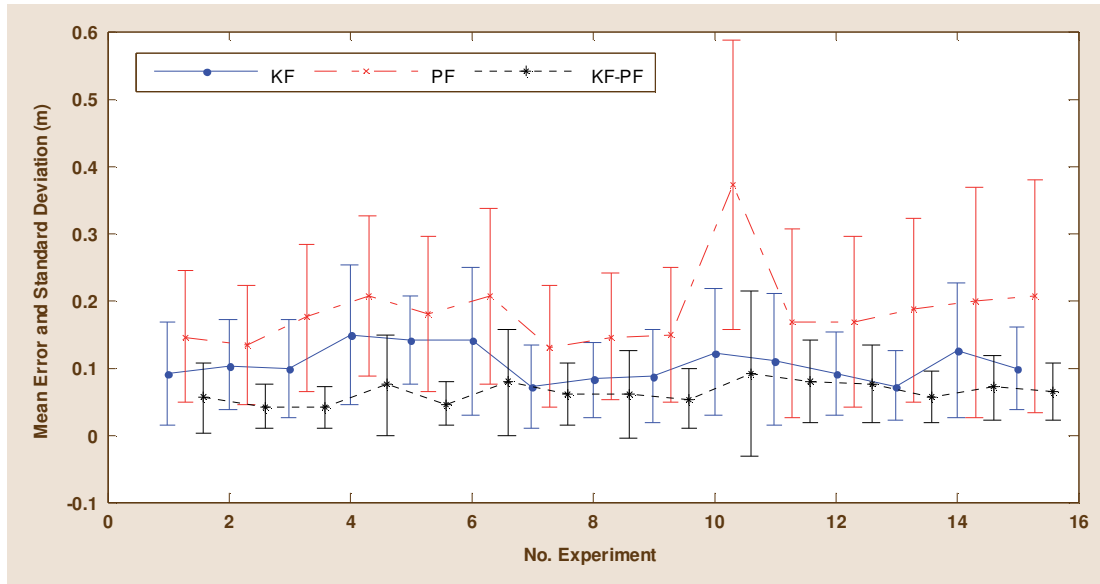


Figure 3.14. Mean error and standard deviation of the position estimates from the three fusion algorithms (KF: Kalman filter, PF: Particle filter, KF-PF: Kalman-particle filter) in the rectangular path experiments.

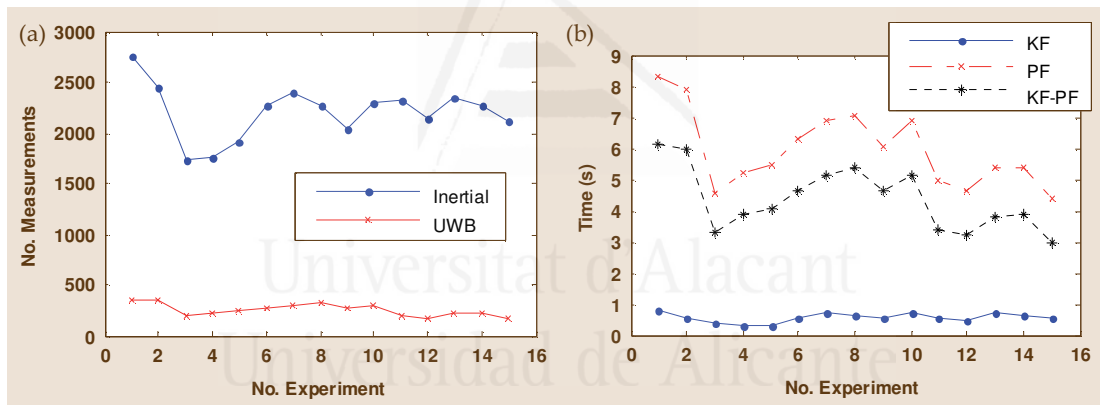


Figure 3.15. Computational cost of the rectangular path experiments: (a) number of measurements registered by the two tracking systems (inertial and UWB); (b) total execution time of the three fusion algorithms (KF: Kalman filter, PF: Particle filter, KF-PF: Kalman-particle filter).

The three fusion algorithms developed in this research can be compared according to the results of the experiments described above. The Kalman-particle filter combination algorithm obtains the most precise position estimates. Therefore, this algorithm is suitable for very precise human tracking applications. The better accuracy of this algorithm is based on its internal structure. The fast measurements of the inertial motion capture system can be easily represented with the linear Gaussian models of a Kalman filter while the slow measurements of the UWB system are better modeled with a non-linear non-Gaussian technique (like a particle filter) because of their high latency. The fusion algorithm based on a Kalman filter is the most efficient fusion algorithm because of its simple implementation. It has an error slightly higher than the Kalman-particle filter algorithm because it is not able to correct the errors of the inertial motion capture system as much as the Kalman-particle filter algorithm.

The fusion algorithm implemented for human tracking during the development of human-robot interaction tasks in this thesis is the Kalman filter fusion algorithm due to its excellent computational performance and its good accuracy. The human-robot interaction tasks developed in this thesis not only require the execution of a real-time human tracking process but also a robot control loop usually based on visual servoing and force control. The high computational cost of the Kalman-particle approach makes difficult to always guarantee real-time performance in this type of tasks where so many software elements and sensors coexist in the same controller PC. Thereby, the Kalman filter fusion algorithm becomes the optimal solution for the combination of measurements from both tracking systems in complex, computationally demanding human-robot interaction tasks. Nevertheless, in simple tasks where the control of the robot is less complex, the Kalman-particle filter combination algorithm will be preferred because of its high accuracy values.

3.5.3 Application of Sensor Fusion in Human-Robot Interaction Tasks

Two different human-robot interaction tasks have been performed in order to verify the suitability of the Kalman filter fusion algorithm for real-time human tracking. In both tasks, the experimental setup is the same and it is shown in [Figure 3.16](#). It consists of a human operator who wears a *GypsyGyro-18* inertial motion capture suit and an *Ubisense* UWB tag and a *Mitsubishi PA-10* robotic manipulator. In the environment, there are four UWB sensors for completing the localization of the UWB tag, as described in [Section 2.3](#). There are additional elements in the environment for the completion of the collaborative tasks, such as storage boxes for leaving parts and a turning table for installing the components which are manipulated by the robot.

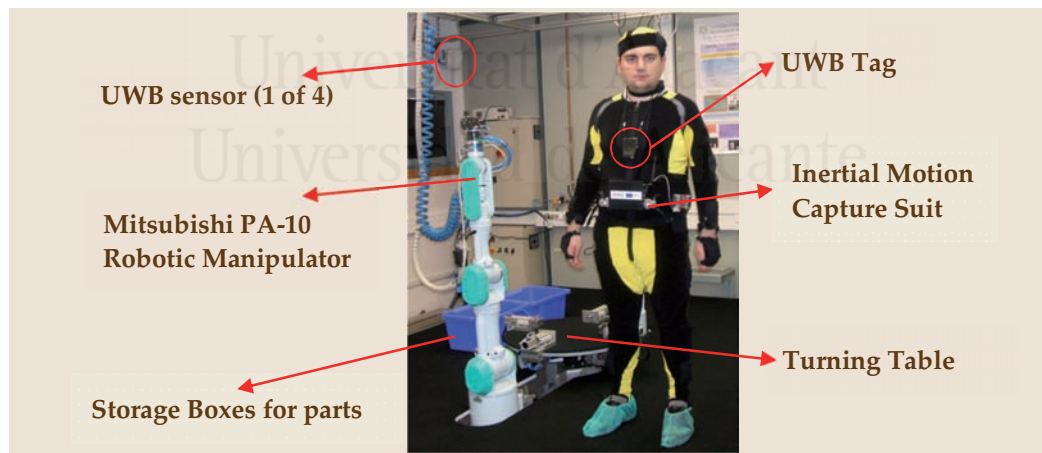


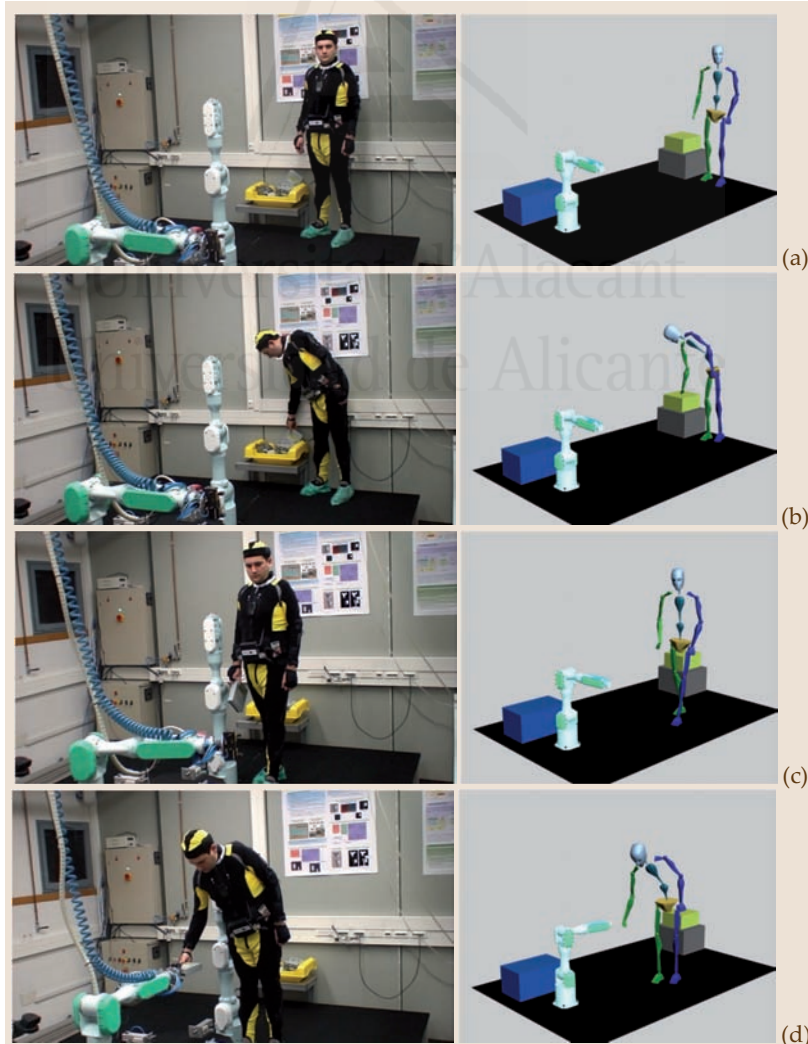
Figure 3.16. Experimental setup for the development of the proposed human-robot interaction tasks.

The global position estimates computed by the Kalman filter algorithm have been applied to the root node (Hips node) of the skeleton of the human body registered by the inertial motion capture system in order to track precisely the movements of all the limbs of the human operator. The following sections compare real images from the experiments with a 3D visualization of the human skeleton obtained by the inertial motion capture system and whose global position is updated with the fusion algorithm. Thereby, the correctness of the real-time human tracking is verified.

3.5.3.1 Human-Robot Interaction Task 1: Giving an Object to the Robot

The first experiment is a simple collaborative task [Corrales-2008a] where the human operator takes an object which is out of the robot's workspace and gives it to the robotic manipulator so that it can leave it inside a storage box. Figure 3.17 depicts the most important frames of the task and the complete sequence can be watched in Video 3.1 that is contained in the DVD attached to this thesis.

Firstly, the human operator chooses an object from the yellow storage box (a CD drive in this case) and takes it (Figure 3.17.a and Figure 3.17.b). Then, he carries the object along the workplace until he reaches the robot (Figure 3.17.c). Meanwhile, the robot controller obtains the global position of the human operator from the human tracking system and computes the distance between the end-effector of the robot and the global position of the human. When this distance is below a threshold (1.5 m in this experiment), the robot controller turns joint E1 of the robot until it faces the human. When the human stands still during a few seconds, the robot supposes that the object has been placed near its end-effector and it closes the pneumatic parallel jaw installed at its end-effector so that the object is grabbed by the robot (Figure 3.17.d). Finally, the robot carries the object to a blue storage box and leaves it inside (Figure 3.17.e and Figure 3.17.f).



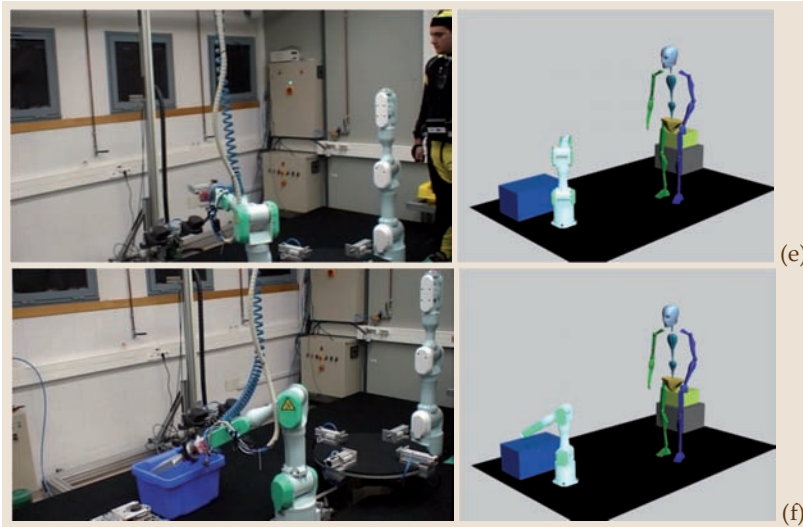


Figure 3.17. Frame sequence of a human-robot interaction task where a human operator gives an object to the robot. Each frame is composed by a photograph of the real task on the left and a 3D representation of the human operator's skeleton on the right.

Figure 3.17 shows not only real photographs of the developed task but also a 3D representation of the skeleton which integrates the global position estimates from the Kalman filter fusion algorithm and the rotational measurements from IMUs of the inertial motion capture system. The correspondence between the real photographs and this 3D model shows that limbs movements are very accurate and that the global localization of the human operator resulting from the fusion algorithm is appropriate. In addition, Video 3.1 verifies that the Kalman filter fusion approach can be used for real-time human tracking during the development of this human-robot interaction task.

3.5.3.2 Human-Robot Interaction Task 2: Cooperation in a Disassembly Task

In the second experiment, the robotic manipulator and the human operator collaborate in a disassembly task [Corrales-2010b]. In this task, the robotic manipulator has to remove two connectors from a metallic structure in order to leave them in a storage box. However, the collaboration of a human operator is needed because the unscrewing of the connectors is a difficult action to be performed by one robot. Figure 3.18 depicts the most important frames of the task and the complete sequence can be watched in Video 3.2 that is contained in the DVD attached to this thesis.

Firstly, the operator unscrews the first connector (Figure 3.18.a and Figure 3.18.b). When the operator has finished the unscrewing process, the robotic manipulator begins to move towards the structure by using a time-independent visual servoing technique [García-2007] [García-2009c] in order to grab the unscrewed connector (Figure 3.18.c). The end of the unscrewing process is determined by the tracking of the human operator. In particular, the robot does not begin its task until the distance between the end-effector of the robot and the operator is greater than a safety threshold (1m). While the robot is removing the first connector from the structure, the human operator unscrews the other connector (Figure 3.18.d). When the human operator has finished unscrewing the second connector, he leaves the robotic workspace (Figure 3.18.e). Finally, the manipulator places the first connector inside the storage box (Figure 3.18.f). The disassembly of the second connector by the robot has not been completed because it is a mere repetition of the previous visual servoing path tracking and it does not require

any new cooperation with the human operator who has already abandoned the workplace.

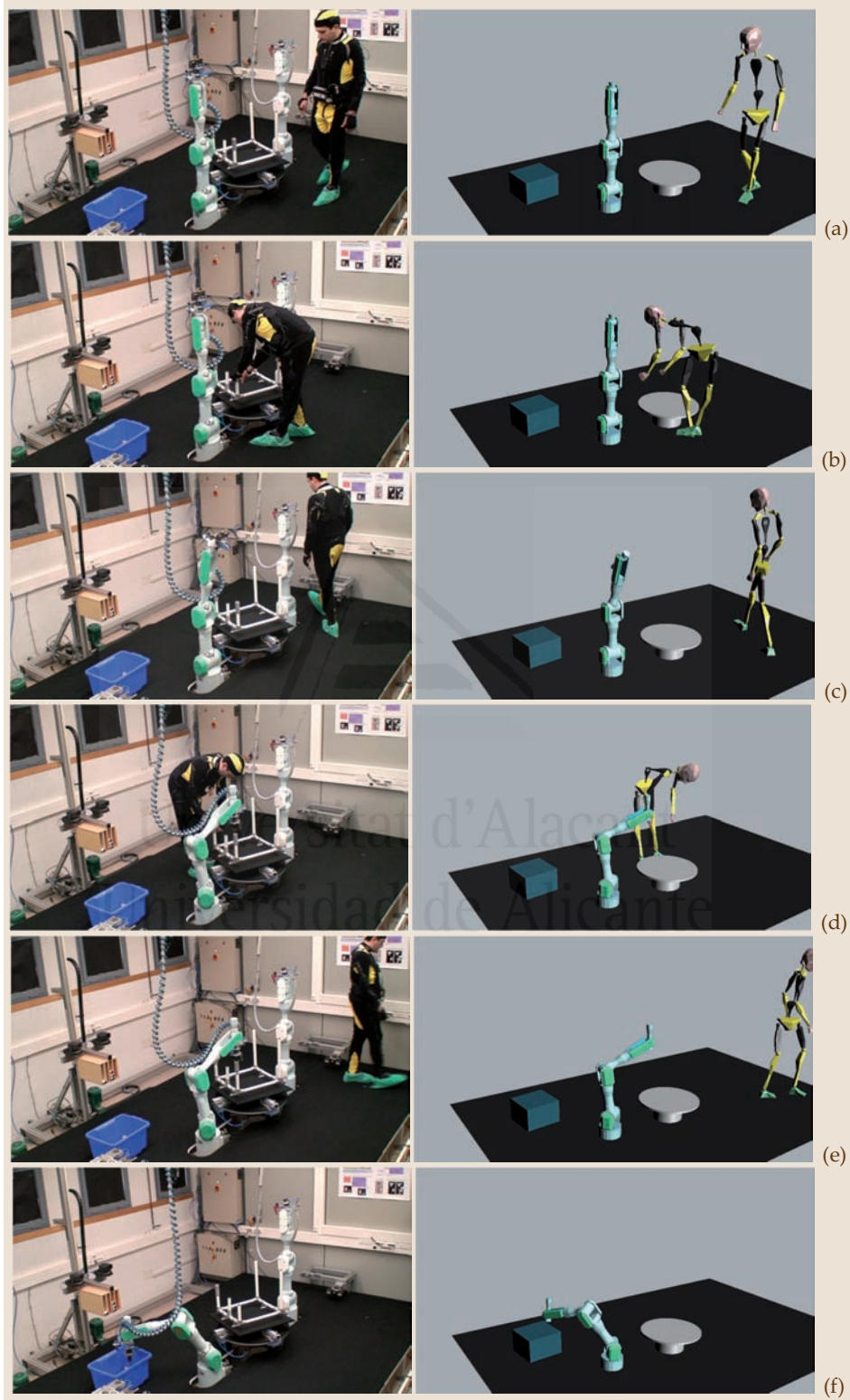


Figure 3.18. Frame sequence of a human-robot interaction task where a human operator collaborates with a robot in the disassembly of a metallic structure. Each frame is composed by a photograph of the real task on the left and a 3D representation of the human operator's skeleton on the right.

The correspondence between the real photographs of the experiment (left frames in Figure 3.18) and the 3D representation of the skeleton of the human (right frames in Figure 3.18) verifies that the global position estimates obtained by the Kalman filter fusion algorithm are suitable for real-time human tracking in this task. In addition, the correspondence between the pose (position and orientation) of each bone of this skeleton and the real pose of the limbs of the human in the photographs guarantees that the IMUs of the inertial motion capture system are able to track on real-time the movements of the complete body of the human operator.

3.6 Conclusions

In this chapter, three novel sensor fusion algorithms based on Bayesian Filtering have been developed and constitute one of the main contributions of this thesis. They combine position measurements from the two human tracking systems presented in Chapter 2 (an inertial motion capture system and a UWB localization system), which have different sampling rates and accuracy. The first algorithm is based on a Kalman filter where measurements from the motion capture system are used in the prediction step of the filter and UWB measurements are incorporated in the correction step. The second algorithm is based on a particle filter and follows a general structure very similar to the Kalman filter: motion capture measurements are used in the prediction step and UWB measurements are used in the importance factor calculation step. Both algorithms use motion capture measurements in the motion model to predict the next position of the human operator. UWB measurements are used in the observation model to correct the previously predicted positions. The third algorithm combines these two Bayesian techniques: motion capture measurements are processed by a Kalman filter while UWB measurements are processed by a particle filter.

In these three algorithms, the transformation matrix between the coordinate frames of both tracking systems is recalculated each time a UWB measurement is processed and thus next measurements from the motion capture system are corrected with the previous UWB position estimate. The application of this transformation matrix is one of the main innovations of the proposed algorithms because it removes the necessity of executing always consecutively the two steps (prediction and correction) of the filter as the traditional implementations of Bayesian filters. Thereby, the computational cost of the proposed algorithms is improved.

These fusion algorithms have been compared by performing several experiments where a human operator has followed two different paths: a linear trajectory and a rectangular trajectory. Several conclusions can be drawn from the obtained results. Firstly, the fusion of both tracking systems has been able to correct the drift which made impractical the use of the inertial motion capture system for the global localization of a human operator. Secondly, the errors of position estimates obtained by the Kalman filter and the Kalman-particle filter approaches are assumable (0.06-0.10m in a 8m trajectory) for the development of general human-robot interaction tasks. Thirdly, the development of several fusion algorithms enables the selection of one or the other solution depending on the efficiency and accuracy requirements of the application. Thereby, applications which require a quick response of the tracking system will use the Kalman filter algorithm while applications which require precise positioning will

implement the Kalman-particle solution. Finally, the Kalman filter algorithm has been selected as the fusion technique for the development of human-robot interaction tasks of this thesis because of its excellent computational performance and its good localization accuracy. It has been applied in two different human-robot interaction tasks in order to verify its suitability for the real-time human tracking.

To sum up, the proposed fusion algorithms solve the problem of combining the global position measurements of the used tracking systems (an inertial motion capture system and a UWB localization system) because they are able to obtain a real-time and precise estimate of the global location of the human operator in the environment. These global position estimates are applied on the skeleton of the inertial motion capture system so that not only the global location of the human operator is known but also the locations of all the bones of her/his skeleton. Nevertheless, this skeleton is composed of linear segments which only consider the lengths of the bones but not the rest of their dimensions. Since most human-robot interaction tasks need to compute the minimum distance between the human and the robot, this linear skeleton is not sufficient to obtain a precise distance value. The next chapter of this thesis presents a novel hierarchy of bounding volumes which cover the skeleton of the human operator in order to represent the dimensions of her/his body. This hierarchy of bounding volumes will be applied in the development of robot control strategies which guarantee the safety of human operators who collaborate with robotic manipulators in the development of cooperative tasks.

Safety

4 Safety Strategies for Human-Robot Interaction

The previous chapter presented three fusion algorithms which combine the global position measurements of two tracking systems (inertial motion capture system and UWB localization system) in order to register precisely all the movements of human operators who collaborate with robotic manipulators. These measurements are applied on a linear skeleton which represents the kinematic structure of the human body. However, this representation does not consider the dimensions of the surfaces of the human body, which can be important for the computation of the distance between humans and robots. In order to overcome this limitation, this chapter proposes a novel hierarchy of bounding volumes which not only cover the previous skeleton to model the human body's surface but they also have dynamic dimensions which change according to their velocity. This dynamic bounding volume hierarchy

4.1 Introduction	73
4.2 Review of Human and Robot Modeling in Cooperative Tasks	75
4.3 Hierarchy of Bounding Volumes	77
4.4 Dynamic Bounding Volume Algorithm	81
4.4.1 General Description of the Algorithm.....	81
4.4.2 Joint Angular Rate Estimation	83
4.4.3 Link Linear Velocity Computation.....	84
4.5 Safety Strategies.....	86
4.5.1 Human-Robot Distance Computation.....	86
4.5.2 Software Architecture	88
4.6 Experimental Results	90
4.6.1 Human-Robot Interaction Task 1: Changing a light bulb of a streetlamp	90
4.6.2 Human-Robot Interaction Task 2: Disassembly of a home appliance.....	94
4.6.3 Human-Robot Interaction Task 3: Assembly of a metallic structure.....	98
4.7 Conclusions	104

is one contribution of this thesis. It has been applied in the development of safety strategies which modify the behavior of robotic manipulators during the execution of cooperative tasks according to the human-robot distance.

4.1 Introduction

One of the main goals of the research performed in this thesis is the development of a human-robot interaction system which enables the collaboration between humans and robotic manipulators and which also guarantees the safety of human operators while this collaboration takes place. [Chapter 2](#) described the two tracking systems (an inertial motion capture system and a UWB localization system) which are employed in this thesis in order to localize precisely the human operators who participate in these cooperative tasks. [Chapter 3](#) explains the novel fusion algorithms based on Bayesian Filtering which have been implemented in order to combine the global position measurements of these two tracking systems. The global position estimates resulting from these fusion algorithms are applied on the root node (Hips node) of a skeleton (see [Figure 2.7.b](#)) which represents the kinematic structure of the human body. Furthermore, the rotational measurements from each IMU of the inertial motion capture system are applied as relative rotations to the corresponding joints of this skeleton so that every limb of the body of the human operator is localized precisely in the environment.

Although this skeleton model represents correctly the movements of the human operator; it does not take into consideration the shape and size of the surface of the human body, which is indispensable for collision avoidance. Therefore, it is necessary to create a model of the surfaces of the bodies of the human operator and the robot. This model should be precise by having a similar shape to the links but it should also be simple enough to compute collision detection on real-time. One of the main contributions of this thesis is the definition of a bounding volume hierarchical representation for the surfaces of the human and the robot which fulfills these two requirements: precision and efficiency. In particular, this bounding volume hierarchy is composed by three levels with different types of bounding volumes which correspond to different levels of precision according to the distance between the human operator and the robotic manipulator. Thereby, when the human operator is far enough from the manipulator, more efficient but also less precise bounding volumes are applied. When the human operator is near the manipulator, more precise but also less efficient bounding volumes are applied.

In addition, this thesis presents a novel dynamic bounding volume algorithm which determines the sizes of the bounding volumes of the third level of the hierarchy depending of the movements of the links which they cover. In particular, the radius of each bounding volume is proportional to the linear velocity of its corresponding link. Thus, the size of the bounding volume does not only model the shape of the link but it also represents the distance that the link will cover between each pair of sampling steps. Therefore, these dynamic bounding volumes are estimations of the motions that the links will perform in the next sampling step. This fact supposes an improvement with regard to previous collision detection techniques based on static bounding volumes. Static bounding volume algorithms only compute the collision detection at each sampling step without considering what may happen between the current sampling step and the next one.

The main goal of the development of these hierarchies is to simplify the computation of collision detection and minimum distance between their composing bounding volumes. In the case of human-robot interaction tasks, the collision between the bounding volumes of the human and the robot must be completely avoided because it may involve a physical injury to the human operator. Therefore, the development of safety strategies which guarantee the non-existence of human-robot collisions is based on the computation of the minimum distance between their bounding volumes. In fact, this thesis develops different safety strategies which modify the behavior of the robotic manipulator depending on the human-robot distance. When the minimum human-robot distance is below a pre-established threshold, the safety strategy stops the normal behavior of the robotic manipulator and executes a special behavior whose main goal is to avoid any kind of collision.

This chapter describes the hierarchy of dynamic bounding volumes which is used to represent the surfaces of the bodies of the human operator and the robot and how this hierarchy is applied in several safety strategies which guarantee the collision avoidance during the development of human-robot interaction tasks. In the following section, previous approaches for modeling the surfaces of 3D objects are analyzed in order to determine which one is the most suitable for human-robot interaction tasks.

4.2 Review of Human and Robot Modeling in Cooperative Tasks

Collision detection and minimum distance computation between 3D objects are two important techniques which are used in different applications [Jimenez-2001] [Kockara-2007] such as robotic motion planning, computer graphics, computational geometry and simulation. Both techniques can be considered as two different solutions for a common problem, the geometric proximity estimation [Steinbach-2006]. Proximity estimation techniques can be classified [Redon-2005] in discrete or continuous techniques depending on how the trajectories of the objects are represented. Discrete techniques sample the objects' trajectories at fixed time instants while continuous techniques take into account all the configurations of the objects along their complete trajectories. The most commonly used discrete techniques (feature tracking and bounding volume) and continuous techniques (swept-volume intersection and trajectory intersection) are described in the following paragraphs.

Feature tracking techniques apply the proximity queries over geometric features (such as vertices, edges or faces) that form the boundaries of the objects and require that the objects are modeled as convex polyhedrons. Several algorithms have been developed in the literature in order to track the closest features of the polyhedrons which represent the objects: Lin-Canny [Lin-1991], Voronoi-Clip (V-Clip) [Mirtich-1998], Gilbert-Johnson-Keerthi (GJK) [Ong-2001] and Hierarchical-Walk (H-Walk) [Guibas-2000]. These algorithms are suitable for objects which have regular geometric shapes because they can be modeled with a few polyhedral components. However, non-convex and irregular objects require polyhedrons composed by a large number of faces in order to approximate their surfaces. In these cases, feature tracking techniques become computationally unaffordable.

Bounding volume techniques define a group of bounding volumes that encapsulate the surface of the objects. Bounding volumes are geometric shapes which have simpler proximity queries than the complex objects they bound. Therefore, bounding volume techniques are more suitable for real-time applications than feature tracking techniques. The most commonly used bounding volumes are (see Figure 4.1) [Ericson-2005]: axis-aligned bounding boxes (AABBs) [Gino van den-1997], object-oriented bounding boxes (OOBBs) [Gottschalk-1996], swept-sphere volumes [Larsen-2000], discrete-orientation polytopes (k-DOPs) [Klosowski-1998] and spheres [Steinbach-2006] [Martinez-Salvador-2003].

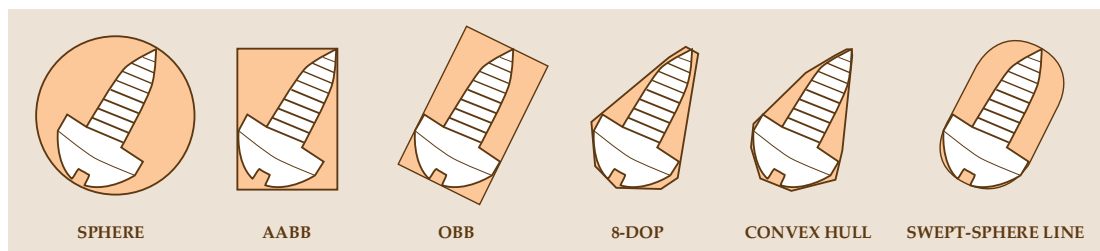


Figure 4.1. Most commonly used bounding volumes.

Swept volume intersection techniques [Cameron-1990] [Abdel-Malek-2005] [Maitech-1999] compute the volumes swept by the movements of the objects during the

whole time span and test them for overlap. These techniques are continuous because they consider all the possible configurations of the objects and no collision is missed out. However, the computation of these swept volumes is very complex and is computationally expensive.

Trajectory intersection techniques are based on the estimation of the trajectories of the objects and the detection of overlaps among them. The objects' trajectories are estimated by evaluating algebraic polynomials depending on time [Schweikard-1991] [Redon-2000] or by bisecting the objects' paths into segments [Schwarzer-2005]. The computational cost of these techniques becomes high when the objects' trajectories are too complex because the number of degrees of the polynomials and the number of path bisections have to be increased.

In robotics applications, proximity estimation techniques based on bounding volumes or a combination of another technique with a bounding volume technique are generally applied. For instance, in [Xu-2007b] a set of spheres is used to create a safety strip between the robot and the environment and the distances between these spheres are applied in an optimization method to determine the best configuration of the robot (see Figure 4.2.a). In [Martinez-Salvador-2003] a set of dynamic spheres is also applied for collision detection between robot arms and human operators (see Figure 4.2.b). These spheres are obtained from an approximation of the surface of the object by generalized cylinders and the number of spheres is dynamically changed depending on the possibility of a collision. In [Steinbach-2006] a sphere-tree hierarchy is defined in order to approximate the current configuration of the links of a humanoid robot and their range of motion. In [Balan-2006] an optimization method is implemented to select the direction of the robot's trajectory which minimizes the distance between the robot and its goal and which maximizes the distances between the sphere-based geometric models of the human and the robot.

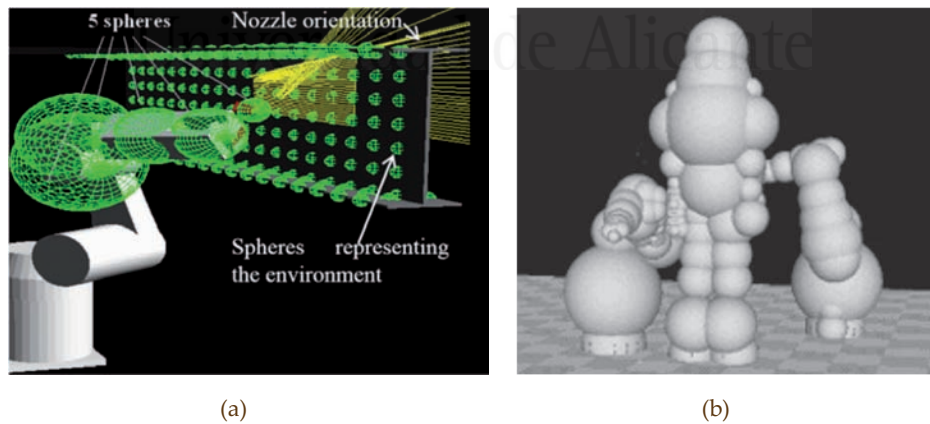


Figure 4.2. Bounding volumes in robotic applications: (a) safety strip of spheres that cover a robotic manipulator during a cleaning task of an I-beam of a bridge, Source: [Xu-2007b]; (b) spherical approximation of a person and two manipulators that collaborate in a task, Source: [Martinez-Salvador-2003].

As shown above, bounding volume techniques are preferred in robotic applications because they present a good relation between their computational complexity and the precision of the provided proximity data for real-time tasks. Feature tracking techniques are not convenient for human-robot interaction tasks because the human body requires complex polyhedrons with many faces to be modeled. Swept-volume

intersection and trajectory intersection techniques are also not used because the human body movements involve too complex trajectory representations. Thereby, the inclusion of the human operator in the robotic workspace increases the computational costs of these algorithms and makes them unfeasible for real-time proximity queries in human-robot interaction tasks. Because of all these drawbacks, representations based on bounding volumes which cover the surfaces of the human operator and the robots are the most appropriate solution for the development of safety strategies in human-robot interaction tasks. The following section describes in detail the hierarchy of bounding volumes which has been implemented in this thesis for this goal.

4.3 Hierarchy of Bounding Volumes

As it has been stated in [Section 4.1](#), the human tracking system (composed by the inertial motion capture system and the UWB localization system (as shown in [Figure 4.3.a](#)) applies the global position estimates obtained from the Kalman filter fusion algorithm and the rotational measurements from the IMUs of the inertial system over a skeleton (see [Figure 4.3.b](#)). This skeleton constitutes a simple representation of the kinematic structure of the human operator. However, this skeletal representation only considers the length of each link but it does not take into account the 3D dimensions of the link's surface. Therefore, an additional geometric representation is necessary to model the surface of the human's body and localize the human operator completely and precisely. This geometric representation has to be based on bounding volumes because it is the most efficient representation for real-time operation, as described in [Section 4.2](#). The selected bounding volume should fulfil two requirements: tight fitting to the links' surfaces and inexpensive distance computation. Previous similar human-robot interaction systems (such as [\[Balan-2006\]](#) and [\[Martinez-Salvador-2003\]](#)) implement sphere-based geometric representations to achieve this goal. These sphere-based representations have an inexpensive distance computation but they do not fit tightly to the links surfaces. A high number of spheres are needed to solve this problem and thus the final computational efficiency of this geometric representation is reduced.

In this chapter, a new bounding volume representation based on Sphere-Swept Lines (SSLs) is presented to overcome the limitations of previous spherical models. A SSL is a bounding volume obtained from the *Minkowski* sum of a sphere and a segment [\[Ericson-2005\]](#). SSLs have a better tight than spheres because they have a similar shape to the limbs. In addition, this representation can also be applied to the robotic manipulators which collaborate with human operators because their links usually have cylindrical shapes.

For instance, [Table 4.1](#) compares the volumes of the links of a *Mitsubishi PA-10* robotic manipulator which is used in this thesis with the volumes of bounding volumes based on spheres and SSLs. The geometric representation of the robot links based on spheres requires an average of 64% of empty volume more than the representation based on SSLs. As stated before, many previous approaches based on spheres use more than one sphere to cover each link more precisely. For instance, a minimum of 68 spheres are used for the human and 35 spheres for the robot in [\[Martinez-Salvador-2003\]](#) while 46 spheres are required for the human and 41 for the robot in [\[Balan-2006\]](#). The geometric representation proposed in this thesis requires only 18 SSLs for the hu-

man and 8 SSLs for the robot. This reduction in the number of bounding volumes involves a reduction in the number of pairwise distance tests: 144 tests in comparison with 2380 tests in [Martinez-Salvador-2003] and 1886 in [Balan-2006]. Thereby, the performance of minimum human-robot distance computation is improved with the SSL representation.

Table 4.1. Comparison of the space usage (volume) of the bounding volumes based on SSLs and spheres with regard to the real volumes of the links of a PA-10 robot.

Robot link	Real volume (cm ³)	SSL volume (cm ³)	Sphere volume (cm ³)	Ratio Sphere-SSL volume (%)
Base	7798.50	13550.42	15250.58	11.15
S1	2964.80	7241.83	16926.54	57.22
S2	4590.20	6410.59	31176.00	79.42
S3	3266.50	7519.02	15003.56	49.89
E1	1806.20	2579.58	16115.26	83.99
E2	1715.30	4071.64	18908.17	78.47
W1	720.08	808.64	7470.30	89.18

In addition, the distance computation between two SSLs has a low cost because it is calculated from the difference between the SSLs' radii and the distance between the inner segments of the SSLs [Schneider-2003]. The geometric representation of the human operator is computed by covering each of the links of the skeleton obtained from the tracking system with a SSL (see Figure 4.3.c). The location of each SSL is updated on real-time by matching the two ends of the inner segment of the SSL with the ends of the corresponding link.

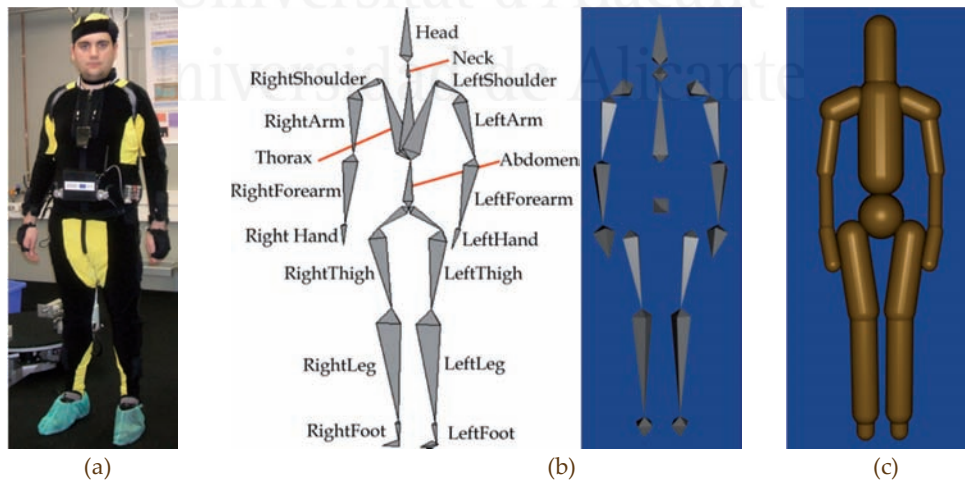


Figure 4.3. Model of the human operator: (a) Human operator with the tracking systems; (b) links/skeleton of the human; (c) SSL bounding volumes which cover the links of the human.

The robotic manipulator (see Figure 4.4.a) which collaborates with the human operator is modeled in a similar way. The positions of the links of the robot (see Figure 4.4.b) are computed by applying forward kinematics to the joint values registered by the robot controller. However, as in the case of the human operator, this procedure only obtains the locations of the ends of each link but it does not consider its surface. A SSL will cover each link in order to model all the dimensions of it (see Figure 4.4.c).

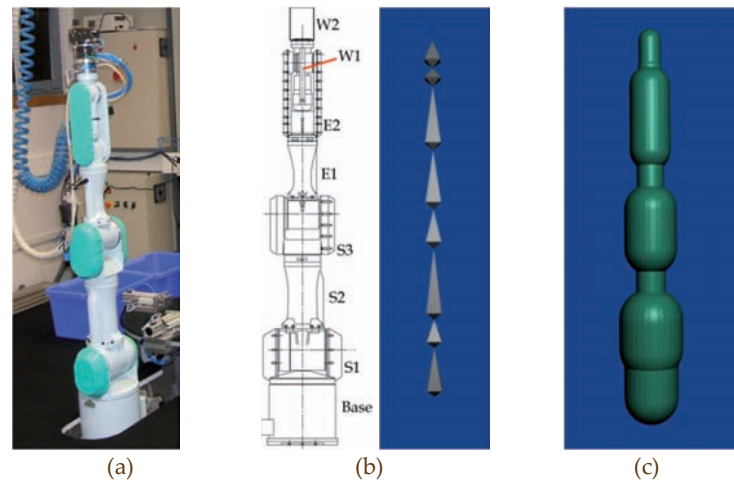


Figure 4.4. Model of the robotic manipulator: (a) PA-10 manipulator; (b) links/skeleton of the robot; (c) SSL bounding volumes which cover the links of the robot.

As stated before, this geometric representation consists of 18 SSLs for the human operator and 8 SSLs for the robotic manipulator. This fact implies that each time the minimum distance between the human and the robot is computed, 144 pairwise distance tests are required. In order to reduce this number of distance tests and increase the performance of the distance computation, a three-level hierarchy of bounding volumes has been developed [Corrales-2010c] [Corrales-2010d]. This hierarchy of bounding volumes is one of the contributions of this thesis. Figure 4.5 and Figure 4.7 depict the relations between the components of this hierarchy for the human operator and the robotic manipulator, respectively. Figure 4.6 and Figure 4.8 show the 3D representation of the bounding volumes which compose each level of this hierarchy for the human operator and the robotic manipulator, respectively.

The third level of this hierarchy is composed by the SSL bounding volumes described above (see Figure 4.6.c and Figure 4.8.c) and it will only be used when the human operator and the manipulator are working quite close to each other. A distance threshold will be established in order to determine whether the third level or the second level of this hierarchy have to be applied to obtain the minimum human-robot distance.

The second level of the bounding volume hierarchy is composed by AABBs (Axis-Aligned Bounding Boxes) which cover several SSLs of the third level that are related kinematically (see Figure 4.6.b and Figure 4.8.b). In particular, the second level of the bounding volume hierarchy for the human operator is composed by 5 AABBs (left lower limb, right lower limb, torso-head, left upper limb and right upper limb) and by 3 AABBs (base, arm and forearm) for the robot. Therefore, the number of pairwise distance tests is reduced to 15 for the second level of the hierarchy. The AABBs of this level are computed by looking for the maximum and minimum coordinates of the contained links and adding to them the maximum radius of the contained SSLs.

The first level of this hierarchy is composed by one global AABB which covers the body of the human or the robot (see Figure 4.6.a and Figure 4.8.a) and which is computed from the maximum and minimum coordinates of all the links. This global AABB will be used when the human and the robot are far from each other. In a similar way to the second and third levels, a distance threshold will be established in order to deter-

mine if the first or the second level of the bounding hierarchy is used to compute the human-robot distance.

The selection of the hierarchy level to be used depends on the distance between the human and the robot. When the distance between the human operator and the robot is high, a geometric representation based on AABBs (levels 1 or 2 of the hierarchy) is used. This representation increases the efficiency of the distance computation because the number of distance pairwise tests is reduced and AABB-AABB distance tests are less expensive than SSL-SSL distance tests. When the human-robot distance is small, the SSL bounding volumes are necessary to compute a good approximation of the distance because they are a more precise and detailed representation of the human and robot's links. However, as stated before, the SSL-based representation requires more pairwise distance tests. In order to reduce the number of SSL pairwise distance tests, a combination of the levels 2 and 3 of the hierarchy is implemented for the minimum distance computation. This minimum distance algorithm and the processing of the obtained value for modifying the robot behavior are explained in detail in [Section 4.5](#).

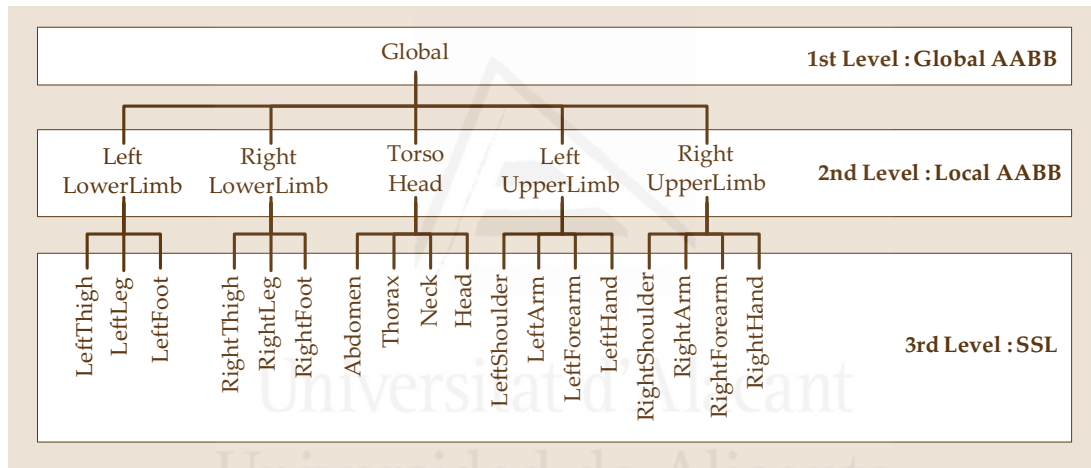


Figure 4.5. Hierarchy of bounding volumes which cover the body of the human operator.

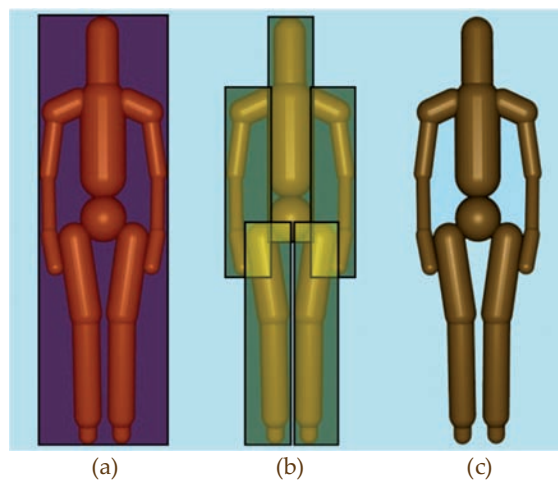


Figure 4.6. 3D representation of the hierarchy of bounding volumes for the human operator: (a) Global AABB (level 1); (b) local AABBs (level 2); (c) SSLs (level 3).

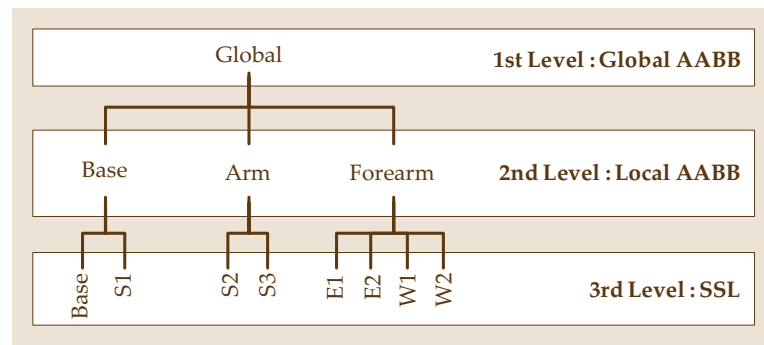


Figure 4.7. Hierarchy of bounding volumes which cover the structure of the robotic manipulator.

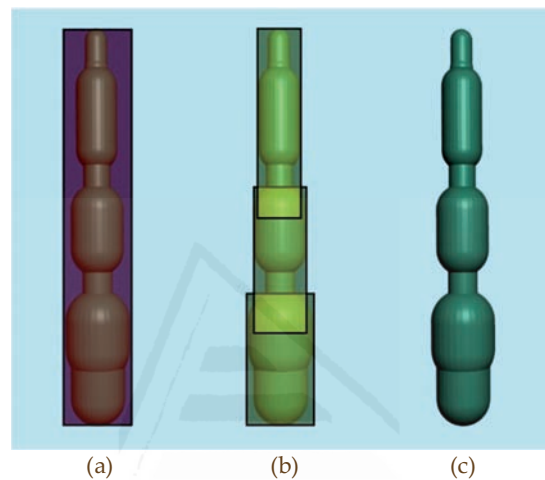


Figure 4.8. 3D representation of the hierarchy of bounding volumes for the robotic manipulator: (a) Global AABB (level 1); (b) local AABBs (level 2); (c) SSLs (level 3).

4.4 Dynamic Bounding Volume Algorithm

4.4.1 General Description of the Algorithm

The hierarchy of bounding volumes described in the previous section was initially implemented by a group of static bounding volumes. These bounding volumes had constant sizes which were pre-established beforehand and they did not change during the execution of the proximity queries. As stated in the previous sections, the main goal of the implementation of this hierarchy is the development of an efficient procedure to compute the minimum distance between the human operator and the robot that are cooperating in the same task.

Nevertheless, in the case of static bounding volumes, the precision of the collision avoidance algorithms depends directly on the frequency with which the human-robot distance is computed. In the case of the human tracking system developed in this thesis, this distance computation is performed each time a new position estimate of the human operator's global position is obtained by the fusion algorithms described in Chapter 3. Thereby, during the time interval between two consecutive measurements from the human tracking system, the safety strategy which controls the movements of the robot considers that the human is located at the pose (position and orientation) determined by the first measurement from the human tracking system. If the human operator is moving slowly near the robot, the possible displacement of the human be-

fore the second measurement of the tracking system can be negligible in most cases. Nevertheless, when the human operator is moving quickly near the robot, this possible displacement can be considerable (between 10-20 cm for update rates of 30 Hz) and should be considered.

In order to incorporate this possible displacement in the hierarchy of bounding volumes, this thesis proposes to define dynamic bounding volumes whose sizes do not only represent the static dimensions of the surface of the human/robot bodies but also estimations of possible future displacements provoked by the linear velocity of each link of their bodies. These dynamic bounding volumes are only applied in the third level of the hierarchy (i.e. SSLs) described in [Section 4.3](#) since in the other levels the human is too far away from the robot that the possible displacements are negligible in comparison with the real human-robot distance.

One of the main contributions of this thesis is the development of these dynamic bounding volumes whose sizes change depending on the linear velocity of the associated link. In particular, the radius $radius_i(t)$ of the SSL which encapsulates the link i of the skeleton has two components: a constant component $radius_surface_i$ which includes the maximum radius of the surface of the link and a variable component which is proportional to the linear velocity of the link at each time instant t :

$$radius_i(t) = radius_surface_i + |v_i(t)| \cdot \Delta t \quad (4.1)$$

The second term of (4.1) predicts the distance that the link will cover during the sampling period Δt . Therefore, the radius of the SSL does not only represent the geometry of the corresponding link but also an approximation of its movement between each pair of samples.

The velocities of all the links of the skeletons of the human operator and the robotic manipulator must be calculated in order to compute the radius of each SSL at each sampling instant. In order to compute these radii and then calculate the minimum distance between the human and the robot, the algorithm summarized in [Figure 4.9](#) is implemented and executed at a constant time interval Δt . Firstly, the measurements of the joint angles are obtained. The robot's joint angles are read from the robot's controller while the human's joint angles are obtained from an inertial motion capture suit which is worn by the human operator. Next, the angular rate of each joint is estimated from the variation of the joint angles. The linear velocity of each link is also calculated from the joint angular rates in order to update the radius of each bounding volume according to (4.1). The positions of the ends of each SSL are updated by applying forward kinematics. Finally, the minimum distance between the bounding volumes of the human and the bounding volumes of the robot is computed. This value is obtained by calculating the distance between each pair of SSLs and selecting the minimum one. In the following sections, the joint angular rate estimation step and the link linear velocity computation step of this algorithm are described in detail.

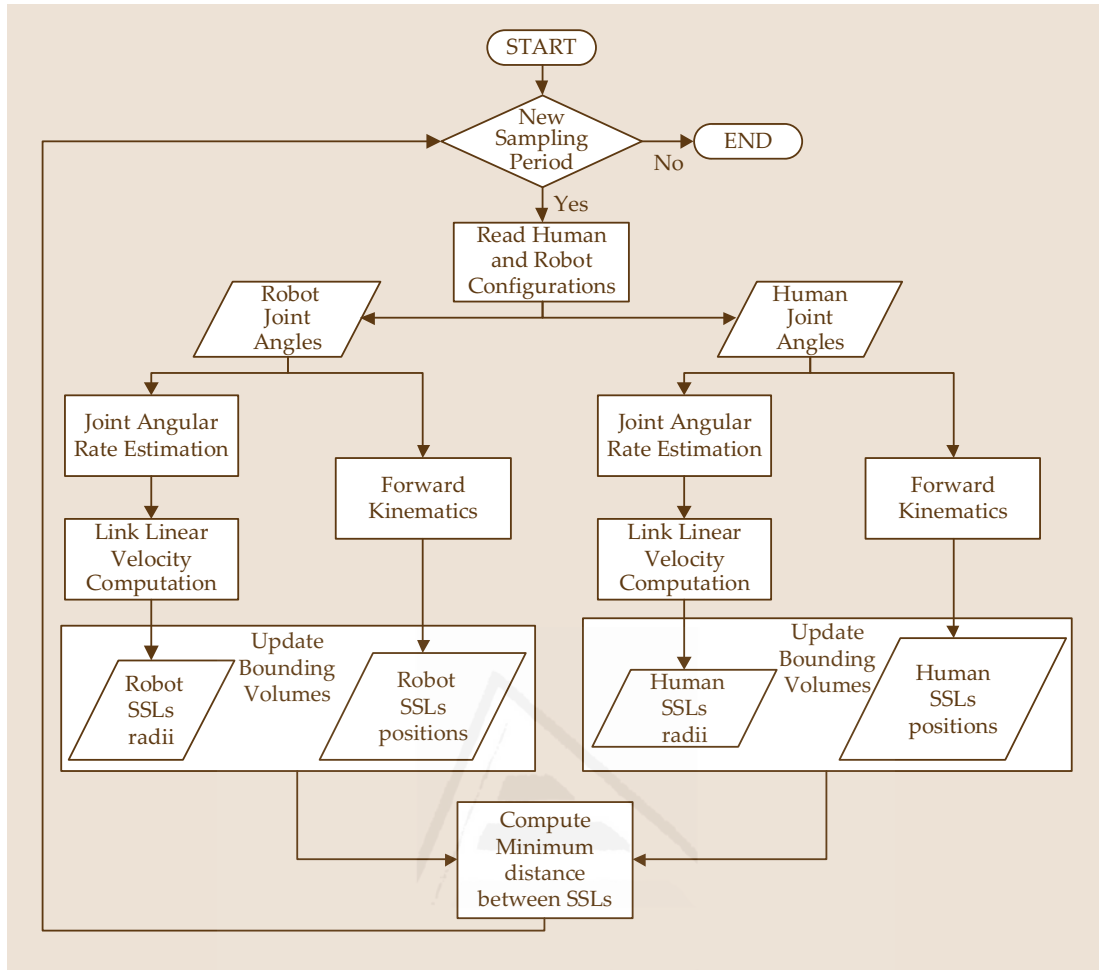


Figure 4.9. Diagram of the dynamic bounding volume algorithm based on the linear velocities of the links.

4.4.2 Joint Angular Rate Estimation

After reading the angle values of each joint of the human operator and the robot, the current angular rate of each joint should be computed. This value identifies the instantaneous velocity with which the joint angles change through time. A first approach for obtaining the angular rate of each joint i is the *finite difference method* (FDM) which uses the difference between the current joint angle $\alpha_i(t)$ and the previous one $\alpha_i(t-1)$:

$$\dot{\alpha}_i(t) = \frac{\alpha_i(t) - \alpha_i(t-1)}{\Delta t} \quad (4.2)$$

However, this method is not convenient when the sampling period Δt is small because the errors in the angle measurements are amplified and become significant [Janabi-Sharifi-2000].

Therefore, an adaptive windowing technique [Janabi-Sharifi-2000] has been implemented to overcome these limitations in the computation of the joint angular rates. This technique involves the calculation of the angular range in a window of previous angle measurements. The size of this window is selected adaptively depending on the magnitude of the velocity. On one hand, the window size should be small when the

velocity is high in order to maintain reliable estimates. On the other hand, when the velocity is low, the window should be large in order to obtain more precise estimates. In particular, the adaptive windowing technique uses a constant value ε_i which identifies the maximum allowable error between the current angle measurements α_i in the window and the estimates of these angles $\hat{\alpha}_i$ calculated from the angular rate in the window. Thus, the size n of the window is the maximum value $n = \max\{1, 2, 3, \dots\}$ that verifies the following equation:

$$|\alpha_i(t-i) - \hat{\alpha}_i(t-i)| \leq \varepsilon_i \quad \forall i \in \{1, 2, \dots, n\} \quad (4.3)$$

The previous angle estimate $\hat{\alpha}_i(t-i)$ is computed with the following equation:

$$\hat{\alpha}_i(t-i) = \alpha_i(t) - \frac{i(\alpha_i(t) - \alpha_i(t-n))}{n} \quad (4.4)$$

Once the size n of the window is established, the angular rate in this window can be obtained with the following equation:

$$\dot{\alpha}_i(t) = \frac{\alpha_i(t) - \alpha_i(t-n)}{n\Delta t} \quad (4.5)$$

A comparison between the angular rates obtained with the finite difference method of (4.2) and with the adaptive windowing method of (4.5) is shown in Figure 4.10. While the finite difference method presents small peaks due to the measurement noise, the adaptive windowing is able to smooth these peaks by estimating the angular rate in a dynamic window.

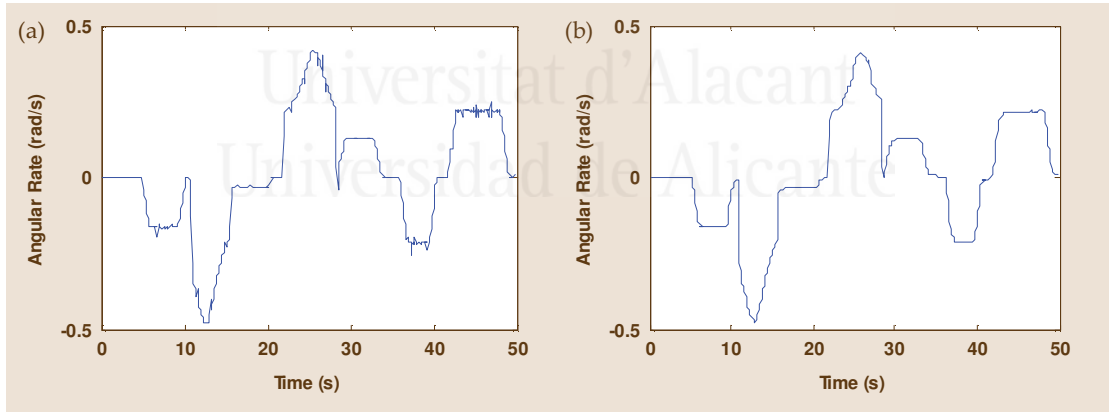


Figure 4.10. Evolution of the angular rate of the first joint of the robotic manipulator in a human-robot interaction task: (a) finite difference method; (b) adaptive windowing method (with $\varepsilon = 0.3$ rad).

4.4.3 Link Linear Velocity Computation

The next step of the dynamic bounding volume algorithm after the estimation of the angular rates of the joints is the calculation of the linear velocity of each link. First of all, the angular rate of each joint i (node of the skeleton) should be represented in the coordinate system attached to this joint.

However, the angular rates $\dot{\phi}_i(t) = [\dot{\alpha}_i(t), \dot{\beta}_i(t), \dot{\gamma}_i(t)]$ estimated in Section 4.4.2 are not represented in this coordinate system because they are obtained from Euler angles in ZXY order. Therefore, a transformation matrix $E_{ZXY}(\phi_i(t))$ between both coordinate

systems is needed. The columns of this matrix are obtained by representing the relative rotation axes of the Euler angles $[\hat{Z}_{Euler}, \hat{X}_{Euler}, \hat{Y}_{Euler}]$ with regard to the absolute axes of the coordinate system of joint i :

$$E_{ZXY}(\phi_i(t)) = [\hat{Z}_{Euler}, Rot_Z(\alpha_i(t)) \cdot \hat{X}_{Euler}, Rot_Z(\alpha_i(t)) \cdot Rot_X(\beta_i(t)) \cdot \hat{Y}_{Euler}] \quad (4.6)$$

The final form of the transformation matrix $E_{ZXY}(\phi_i(t))$ would be the following one:

$$E_{ZXY}(\phi_i(t)) = \begin{bmatrix} 0 & \cos(\alpha_i(t)) & -\sin(\alpha_i(t))\cos(\beta_i(t)) \\ 0 & \sin(\alpha_i(t)) & \cos(\alpha_i(t))\cos(\beta_i(t)) \\ 1 & 0 & \sin(\beta_i(t)) \end{bmatrix} \quad (4.7)$$

The angular rate of the joint i in its coordinate system is obtained by applying the transformation matrix $E_{ZXY}(\phi_i(t))$ to the Euler angular rate $\dot{\phi}_i(t)$:

$$\dot{\theta}_i(t) = E_{ZXY}(\phi_i(t)) \cdot \dot{\phi}_i(t) \quad (4.8)$$

As the joints of the skeleton are part of a kinematic chain, the angular velocity of the coordinate system of a joint i does not only depend on its angular rate but also it depends on the angular velocity of the previous joints (i.e. the predecessor nodes in the skeleton's hierarchy). Therefore, for calculating recursively the angular velocity of each coordinate system i , the angular velocity ${}^{i-1}\omega_{i-1}(t)$ of its parent $i-1$ in addition to its own angular rate $\dot{\theta}_i(t)$ are considered:

$${}^i\omega_i(t) = {}^iRot_{i-1}(t) \cdot {}^{i-1}\omega_{i-1}(t) + \dot{\theta}_i(t) \quad (4.9)$$

The matrix ${}^iRot_{i-1}(t)$ represents the rotation transformation between the coordinate system $i-1$ and the coordinate system i . It is obtained by multiplying the rotation matrices corresponding to the rotation angles of joint i :

$${}^iRot_{i-1}(t) = Rot_Z(\alpha_i(t)) \cdot Rot_X(\beta_i(t)) \cdot Rot_Y(\gamma_i(t)) \quad (4.10)$$

The linear velocity of each node of the skeleton is computed similarly. In particular, the linear velocity ${}^i v_i(t)$ of the coordinate system i is composed by the linear velocity ${}^{i-1} v_{i-1}(t)$ of its parent $i-1$ plus the linear velocity caused by the angular velocity of its parent $i-1$:

$${}^i v_i(t) = {}^iRot_{i-1}(t) \cdot ({}^{i-1} v_{i-1}(t) + {}^{i-1}\omega_{i-1}(t) \times {}^{i-1}P_i) \quad (4.11)$$

Where ${}^{i-1}P_i$ is the offset vector between the coordinate systems $i-1$ and i .

This linear velocity corresponds to the coordinate system i which is situated at the beginning of the link i . However, the size of the bounding volume which covers this link should be computed according to the maximum velocity of the link. Thus, the linear velocity ${}^i v_{i+1}(t)$ of the end of the link (where the child coordinate system $i+1$ is situated) needs to be calculated. This velocity not only contains the linear velocity of the beginning of the link (caused by the predecessors' movements) but also the linear velocity generated at the end of the link by the angular velocity of the joint:

$${}^i v_{i+1}(t) = {}^i v_i(t) + {}^i\omega_i(t) \times {}^i P_{i+1} \quad (4.12)$$

${}^i v_{i+1}(t)$ is used as the maximum linear velocity of link i and is applied in (4.1) in order to determine the radius of the corresponding SSL at each time step.

4.5 Safety Strategies

The main goal of the proposed hierarchy of dynamic bounding volumes is the efficient computation of the minimum distance between the human operator and the robots that cooperate in common tasks. This minimum distance is the main parameter employed for the development of safety strategies in human-robot interaction tasks. In this thesis, two different safety strategies have been proposed. Both strategies have the maximum priority in the execution sequence of the system's controller. Their normal behavior involves computing the minimum human-robot distance each time new measurements from the human tracking system are registered and verify if the calculated distance is below a pre-established safety threshold. If the human-robot distance is above the threshold, the safety strategy will not do anything more and the normal behavior of the robot continues. If the human-robot distance is below the threshold, the safety strategy stops the normal behavior of the robot and executes a special safety behavior. In the first safety strategy, this behavior makes the robot stop and remain still. In the second safety strategy, this behavior makes the robot go away from the human in a linear path so that the human-robot distance is equal to the safety threshold. In the following subsections, the implementations of these safety strategies are described. First of all, the human-robot distance algorithm is detailed. Secondly, the software architecture which constitutes the base of the implementation of the safety strategies is explained.

4.5.1 Human-Robot Distance Computation

As stated before, the main goal of the hierarchy of bounding volumes described in the previous sections is to represent the bodies of the human operator and the robot in such a way that the human-robot distance computation is precise and efficient for the development of cooperative tasks on real-time. The algorithm which has been implemented to compute the human-robot distance is shown in Table 4.2.

This algorithm receives as input parameters the two distance thresholds ($DIST_{1>2}$ and $DIST_{2>3}$) and the poses of the links of the human operator and the robot manipulator. The first threshold $DIST_{1>2}$ determines whether the minimum human-robot distance is obtained from the AABBs of the first level or from the AABBs of the second level. Similarly, the second threshold $DIST_{2>3}$ determines whether the minimum human-robot distance is obtained from the AABBs of the second level or from the SSLs of the third level. Firstly, this algorithm generates the two global AABBs of the first level (lines 1 and 2 of Table 4.2) by looking for the minimum and maximum coordinate values of all the links. Next, the minimum distance $mdist1$ between both global AABBs is calculated (line 3 of Table 4.2) and compared with the first threshold $DIST_{1>2}$. If $mdist1$ is bigger than the threshold, this distance value will be used as minimum human-robot distance (line 5 of Table 4.2) because they are so far from each other than the global AABB representation is sufficient.

Table 4.2. Minimum distance algorithm based on the three-level hierarchy of bounding volumes.

Algorithm: Minimum Human-Robot Distance

Input: Links poses of the human operator (from the human tracking system).
 Links poses of the robot manipulator (from the forward kinematics of the robot controller).
 Boolean flag to activate or deactivate the use of dynamic bounding volumes: *DynamicBV*
 Safety distance thresholds: $DIST_{1>2}$, $DIST_{2>3}$

Implementation:

```

01: Compute Global AABB (Level 1) for Human:  $AABB1_H$ 
02: Compute Global AABB (Level 1) for Robot:  $AABB1_R$ 
03:  $mdist1$  = MinimumDistance ( $AABB1_H$ ,  $AABB1_R$ )
04: if ( $mdist1 > DIST_{1>2}$ )
05:    $finalDist$  =  $mdist1$  //Distance in Level 1
06: else
07:   Compute Local AABBs (Level 2) for Human:  $AABB2_H[1..5]$ 
08:   Compute Local AABBs (Level 2) for Robot:  $AABB2_R[1..3]$ 
09:    $dist2[1..15]$  = PairwiseDistances( $AABB2_H[1..5]$ ,  $AABB2_R[1..3]$ )
10:    $dist2[1..15]$  = SortArrayInAscendingOrder( $dist2[1..15]$ )
11:    $mdist2$  = MinimumValue( $dist2[1..15]$ )
12:   if ( $mdist2 > DIST_{2>3}$ )
13:      $finalDist$  =  $mdist2$  // Distance in Level 2
14:   else
15:      $mdist3$  =  $FLOAT\_MAX\_VALUE$ 
16:     for each element  $i$  in  $dist2[1..15]$ 
17:       if ( $mdist3 < dist2[i]$ )
18:         break //Finish for loop in line 16
19:       end if
20:       Compute SSLs (Level 3) contained by  $AABB2_H[i]$ :  $SSL3_H[1..nLinksH_i]$ 
21:       Compute SSLs (Level 3) contained by  $AABB2_R[i]$ :  $SSL3_R[1..nLinksR_i]$ 
22:       if DynamicBV // If the computation of the dynamic BV is activated
23:         Update dimensions of SSLs by applying algorithm of Section 4.4
24:       end if
25:       for each element  $j$  in  $SSL3_H[1..nLinksH_i]$ 
26:         for each element  $k$  in  $SSL3_R[1..nLinksR_i]$ 
27:            $distSSL$  = MinimumDistance( $SSL3_H[j]$ ,  $SSL3_R[k]$ )
28:            $mdist3$  = MinimumValue( $distSSL$ ,  $mdist3$ )
29:         end for
30:       end for
31:     end for
32:      $finalDist$  =  $mdist3$  //Distance in Level 3
33:   end if
34: end if
35: Output:  $finalDist$ 

```

However, if the human-robot distance is smaller than the threshold $DIST_{1>2}$, the more precise AABB representation of the second level is required. First of all, the local AABBs of the second level are generated (lines 7 and 8 of Table 4.2) from the minimum and maximum coordinate values of the links contained in each AABB. As there are 5 AABBs for the human and 3 AABBs for the robot, they are stored in two arrays of these lengths: $AABB2_H[1..5]$ and $AABB2_R[1..3]$. All the possible distance values between each pair of AABBs are computed and stored in array $dist2[1..15]$ (line 9 of Table 4.2). This array is sorted in ascending order (line 10 of Table 4.2) and its minimum component is

used as the minimum distance value $mdist2$ for the second level of the bounding volume hierarchy (line 11 of Table 4.2). This value $mdist2$ is used as minimum human-robot distance if it is higher than the second threshold $DIST_{2>3}$ (line 13 of Table 4.2).

When $mdist2$ is smaller than $DIST_{2>3}$, the SSL representation is applied because AABB bounding volumes do not provide a sufficient level of detail for precise distance computation between very close links. In fact, the SSL bounding volumes have a better tight to the links than the AABB bounding volumes. In order to reduce the number of distance tests between each pair of SSLs, the distance values $dist2[1..15]$ between the AABBs of the second level are used as a lower threshold for the SSLs distance values. The SSLs are generated (lines 20 and 21 of Table 4.2) from the corresponding AABBs of the second level, which have been ordered according to their pairwise distances (line 10 of Table 4.2). Next, the dynamic bounding volume algorithm of Section 4.4 is applied in order to update the dimensions of each bounding volume according to the linear velocity of the link which is covered by it (line 23 of Table 4.2). After generating the SSLs contained by each pair of AABBs, the minimum distance $mdist3$ between them is computed (lines 27 and 28 of Table 4.2). If the SSL-SSL distance $mdist3$ becomes smaller than the distance $dist2[i]$ between the following pair of AABBs (line 17 of Table 4.2), the minimum distance search process will end (line 18 of Table 4.2) because the following SSLs will be contained by those AABBs and they will be further away. Thereby, in most cases, not all the 144 SSL-SSL tests are required.

The minimum distance value computed by the algorithm presented in Table 4.2 is used by the safety strategies in order to guarantee that there are no collisions between the human operator and the robot. In particular, the safety strategies verify that the human-robot distance is higher than a safety threshold. While this condition is fulfilled, the robot tracks its trajectory normally. Nevertheless, when the human-robot distance is lower than the safety threshold, the robot tracking process is temporarily stopped and a special safety behavior (different for each safety strategy) is activated. In order to implement these safety strategies, the software architecture described in the following section is developed.

4.5.2 Software Architecture

A multi-threaded software architecture has been developed in order to implement the human-robot interaction behavior described in the previous sections. It has been programmed as a C++ program which is executed in the controller PC. The robot controller, the motion capture system, the UWB localization system and the vision system are connected to this PC in order to avoid any synchronization problem between their measurements.

Three threads compose this software architecture (see Figure 4.11): the distance computation thread, the path tracking thread and the safety strategy thread. These three threads are executed simultaneously and share a common memory space where they interchange information. The distance computation thread obtains the links orientation measurements from the human tracking system (described in Chapter 2 and Chapter 3) and the joints angles of the robot from the robot controller. The positions of the links of the human and the robot are computed from these measurements through a forward kinematics algorithm and they are stored in the common memory space.

Finally, these links positions are used by the algorithm described in Table 4.2 in order to generate the hierarchy of bounding volumes and calculate the minimum distance between the human operator and the robot manipulator. The distance computation thread will update this human-robot distance value each time new measurements from the human tracking system and the robot controller are registered by the controller PC.

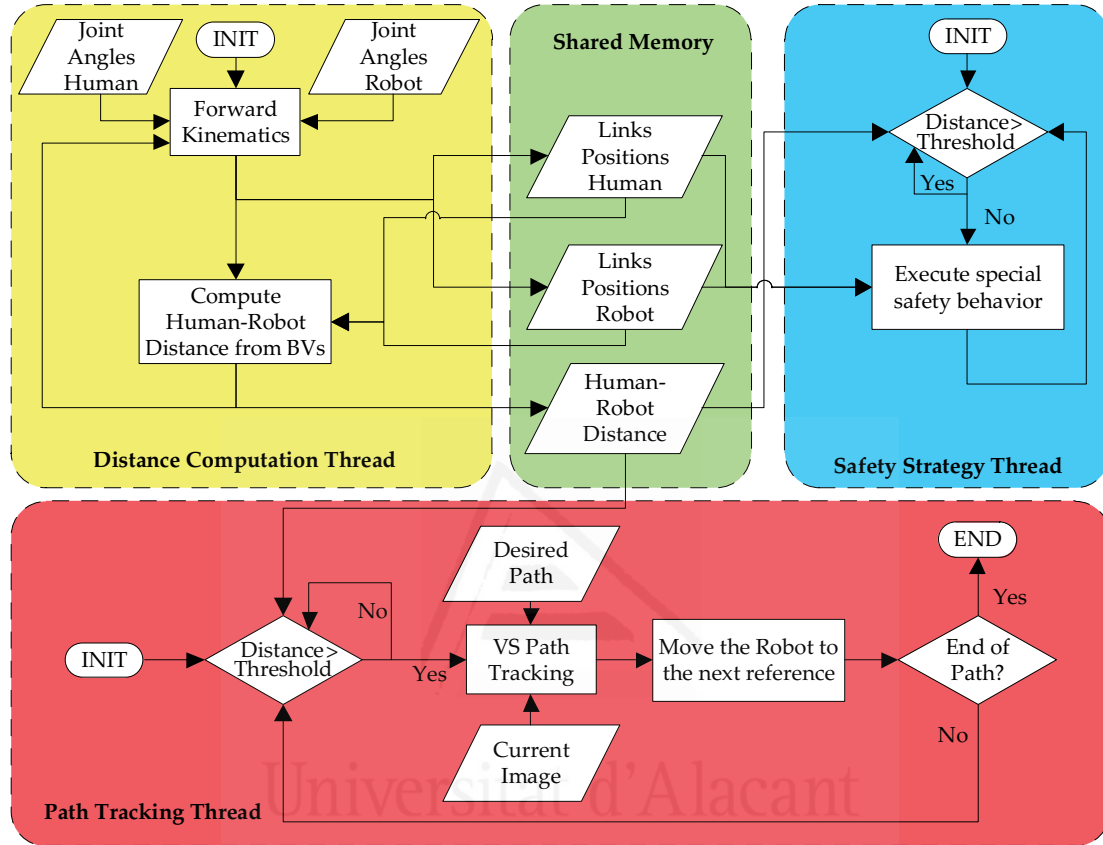


Figure 4.11. Components of the software architecture which implements the human-robot interaction behavior.

The minimum human-robot distance calculated by the distance computation thread is stored in the common memory space where it is checked by the other two threads. On the one hand, when this human-robot distance is greater than the safety threshold, the path tracking thread performs the normal behavior of the robot. This robot behavior is implemented as a visual servoing path tracking process which makes the robot follow a pre-established path according to the images that are registered by an eye-in-hand camera installed at its end-effector. In particular, a time-independent path tracker compares the current image from the eye-in-hand camera with the corresponding image of the desired path and calculates the robot velocity required to make them coincide. This path tracking control of the robot is explained in detail in [García-2009a] and [Pomares-2010] and it is not a contribution of this thesis. This tracking process ends when the desired path is completed and it is paused when the safety behavior is activated. On the other hand, when the human-robot distance is smaller than the safety threshold, the safety strategy thread executes the corresponding safety behavior. As stated before, two safety strategies with two different safety behaviors have been implemented. The first safety behavior stops the robot and keeps it still while the human-robot distance is below the safety threshold. The second behavior moves the robot

away from the human operator in a linear path so that the human-robot distance is always equal to the safety threshold. This escape trajectory is calculated from the line which joins the two closest links. While one of the safety behaviors is executed, the time-independent path tracking process is paused. The software architecture described in this section and the implemented safety strategies are two contributions of this thesis. In the following section, these safety strategies have been applied in real human-robot interaction tasks in order to verify their correct operation.

4.6 Experimental Results

Three different human-robot interaction tasks have been developed for the verification of the techniques described in this chapter. They are part of the research projects DPI2005-06222 ('Design, Implementation and Experimentation of Intelligent Manipulation Scenarios for Automatic Assembly and Disassembly Applications') and DPI2008-02647 ('Dexterous Manipulation through Haptic Perception and Visual Servoing by using an Articulated Structure over a Robotic Manipulator'). The first task involves changing a light bulb of a streetlamp; the second task is based on the disassembly of a home appliance and the third task is based on the assembly of a metallic structure. In all of them, there is a human operator who collaborates with one or two robotic manipulators in order to accomplish the task.

This human operator always wears an *Animazoo Gypsygyro-18* inertial motion capture suit and an *Ubisense* UWB tag. The measurements of both tracking systems are combined with the Kalman filter fusion algorithm described in [Section 3.4.1](#). The position estimates from this fusion process and the rotational measurements from the IMUs of the inertial motion capture suit are applied over the skeleton of the human body (see [Figure 4.3.b](#)). In addition, the joint values of the robots controllers are also applied over the skeletons of the robots (see [Figure 4.4.b](#)). By applying forward kinematics to these skeletons, the poses of the links of the human operator and the robots are updated at each execution step of the system's controller.

In order to guarantee the safety of the human operator during the development of these tasks not only the human tracking system described in [Chapter 2](#) and [Chapter 3](#) has been applied but also the hierarchy of bounding volumes and the safety strategies described in this chapter. Therefore, the updated links of the robots and the human are covered with the hierarchy of bounding volumes described in [Section 4.3](#). This hierarchy is used to compute the minimum human-robot distance through the algorithm of [Section 4.5.1](#) and finally is applied in the general architecture described in [Section 4.5.2](#) to modify the behavior of the robots with regard to this distance. These three tasks implements the safety behavior based on the escape linear trajectory from the human.

4.6.1 Human-Robot Interaction Task 1: Changing a light bulb of a streetlamp

The first developed human-robot interaction task involves the replacement of a blown light bulb in a streetlamp with a new one ([\[García-2009a\]](#) and [\[Pomares-2010\]](#)). The main elements which take part in this task are depicted in [Figure 4.12](#): the streetlamp whose light bulb needs to be replaced, the *Mitsubishi PA-10* robotic manipulator which replaces the light bulb, the human operator who brings the new light bulb and

the storage box where the blown light bulb is stored. The *Mitsubishi PA-10* robotic manipulator has three devices installed at its end-effector in order to perform the task: a three-fingered *Barrett Hand*, a *JR3* force sensor and an eye-in-hand *Photonfocus MV-D752-160-CL-8* camera. This camera is used to track four laser points projected on the floor which will be the extracted features for the visual servoing path tracking system which controls the normal trajectory of the robot. The human wears an *Animazoo Gypsygyro-18* inertial motion capture suit and an *Ubisense* UWB localization tag.

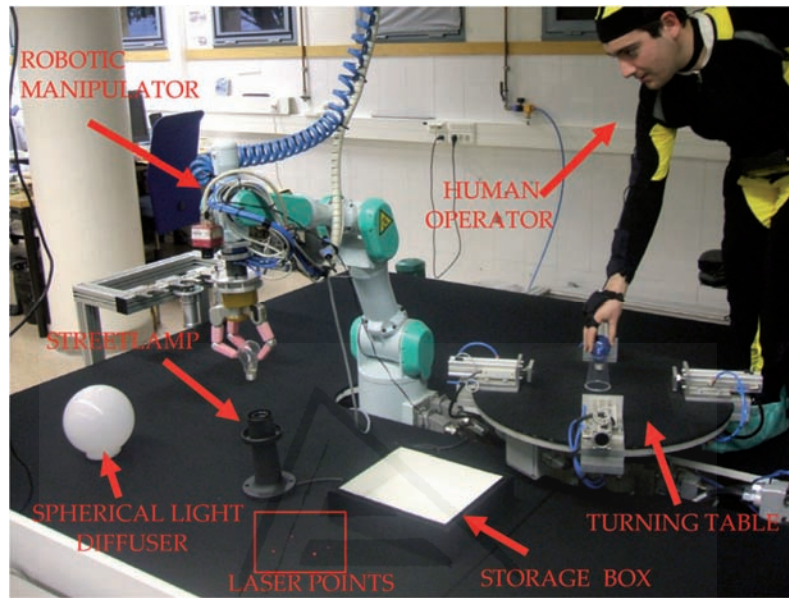


Figure 4.12. Experimental setup for the light bulb replacement task.

This task can be divided into the following phases which are shown in Figure 4.13:

1. *Disassembly of the streetlamp* (Figure 4.13.a): First of all, the robotic manipulator grasps the spherical light diffuser of the streetlamp with the three-fingered *Barrett* hand and turns it until it is released from the base. This moment is detected by a change in the force/torque pattern of the force sensor installed at the end-effector of the robot. This is an example of force control in a contact task. Then, the robot takes off the diffuser and leaves it over the floor.
2. *Transportation of the blown light bulb* (Figure 4.13.b): The robot unscrews the blown light bulb by analyzing again the force/torque pattern of the force sensor. Next, it uses the visual servoing path tracking technique in order to follow the trajectory between the streetlamp and the storage box. The visual servoing algorithm employs four laser points as image features to be followed by an eye-in-hand camera (installed at the end-effector). Finally, the robot leaves the blown light bulb inside the storage box.
3. *Collaboration of the human operator* (Figure 4.13.c): While the robotic manipulator is performing phase 2, the human operator enters the workspace in order to leave a new light bulb over the turning table. In this case, the human operator collaboration is needed since the storage box of new light bulbs is out of the robot's workspace. During this phase, the safety strategy computes constantly the distance between the human and the robot by applying the hierarchy of static bounding

volumes described in Section 4.3. When this distance is smaller than a safety threshold (1m in this task), the robot stops tracking the trajectory of phase 2 and activates a safety behavior which takes the robot away from the human in a linear path in order to keep the human-robot distance below the safety threshold.

4. *Installation of the new bulb and assembly of the streetlamp (Figure 4.13.d):* When the human operator goes away from the workspace, the safety behavior is deactivated and phase 2 is completed by leaving the bulb inside the storage box. Next, the robot takes the new bulb which is over the table and screws it in the streetlamp's base. Finally, the robot installs again the light diffuser. All these subtasks are performed by executing a visual servoing path tracker for controlling the robot's movements and a force control for determining the end of screwing activities.



Figure 4.13. Phases of the disassembly task of a streetlamp for light bulb replacement: (a) Disassembly of the streetlamp; (b) transportation of the blown light bulb; (c) collaboration of the human operator; (d) installation of the new light bulb.

The visual servoing path tracker and the force controller which determine the trajectory of the robot during the normal development of the task are not contributions of this thesis and they are explained in detail in [García-2009a], [Pomares-2010] and [García-2010]. This section is focused on phases 2 and 3 where the human operator collaborates in the task. Figure 4.14.a shows the desired 3D linear trajectory which is followed by the robotic manipulator during phase 2 if the safety behavior is never activated. This trajectory involves the transportation of the blown light bulb from the streetlamp to the storage box and it can be watched in Video 4.1 that is contained in the DVD attached to this thesis. Nevertheless, if the safety behavior is activated because the safety strategy detects that the human-robot distance is below the safety threshold (as shown in Figure 4.14.c, where the human-robot distance goes under 1 m) during phase 3, the

normal path tracking is stopped and the safety strategy makes the robot moves away from the human in a linear path. This safety behavior is able to keep the human-robot distance above the safety threshold while the human is too close to the robot (as shown in the time interval from $t=5$ to $t=10$ s of Figure 4.14.c). Nevertheless, this safety behavior makes the robot move away from the desired trajectory (as shown in Figure 4.14.d). When the human operator abandons the workplace, the visual servoing path tracker is activated again, it returns to the last visited position of the desired trajectory (see instant $t=10$ s of Figure 4.14.d) and ends the trajectory (as shown in Figure 4.14.b).

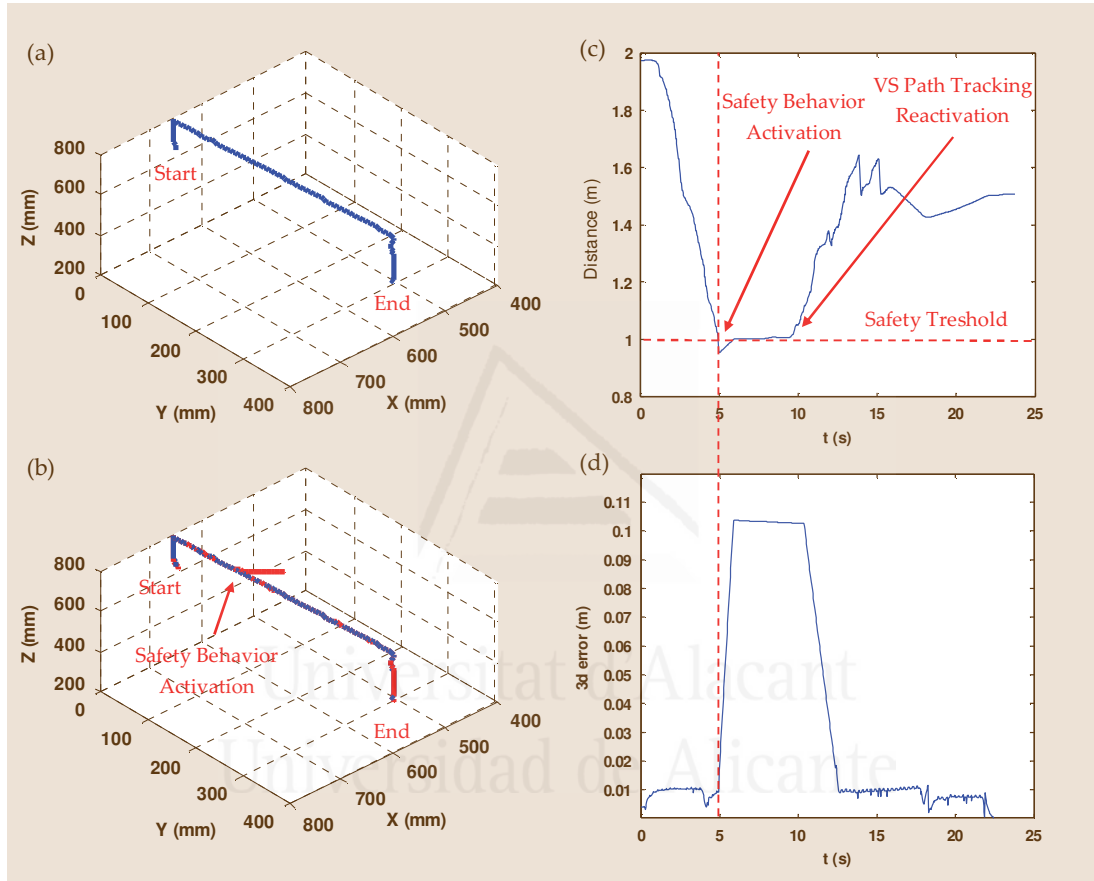


Figure 4.14. Evolution of the trajectories during the task: (a) Position of the robot's end-effector during the original trajectory; (b) position of the end-effector when the safety behavior is activated; (c) Minimum human-robot distance; (d) 3D error between the real position and the desired position of the end-effector.

Figure 4.15 depicts a group of frames of another execution of phases 2 and 3 of this task. This execution can be fully watched in Video 4.2 of the attached DVD. Firstly the robot begins to track the trajectory between the streetlamp and the storage box by applying visual servoing. This trajectory is marked as green points in Figure 4.15.a and Figure 4.15.b. When the human operator enters the workspace and approaches the turning table in order to leave the new light bulb, the safety strategy detects that the human-robot distance is below 1 m, stops the path tracking at the red point of the trajectory shown in Figure 4.15.c and activates the safety behavior so that the robot goes away from the human (as shown in Figure 4.15.d). When the human operator goes away from the workplace, the visual servoing path tracking is activated again. This path tracking moves the robot to the last visited position of the trajectory (as shown in Figure 4.15.e) and then visits the remaining points of the trajectory (as shown in Figure

4.15.f). Figure 4.15.a, Figure 4.15.b and Figure 4.15.f show on their right upper corners the skeletons registered by the human tracking system and the robot controller. Figure 4.15.c, Figure 4.15.d and Figure 4.15.e show the 3D error between the real position of the robot's end-effector and the desired position from the initially planned trajectory.

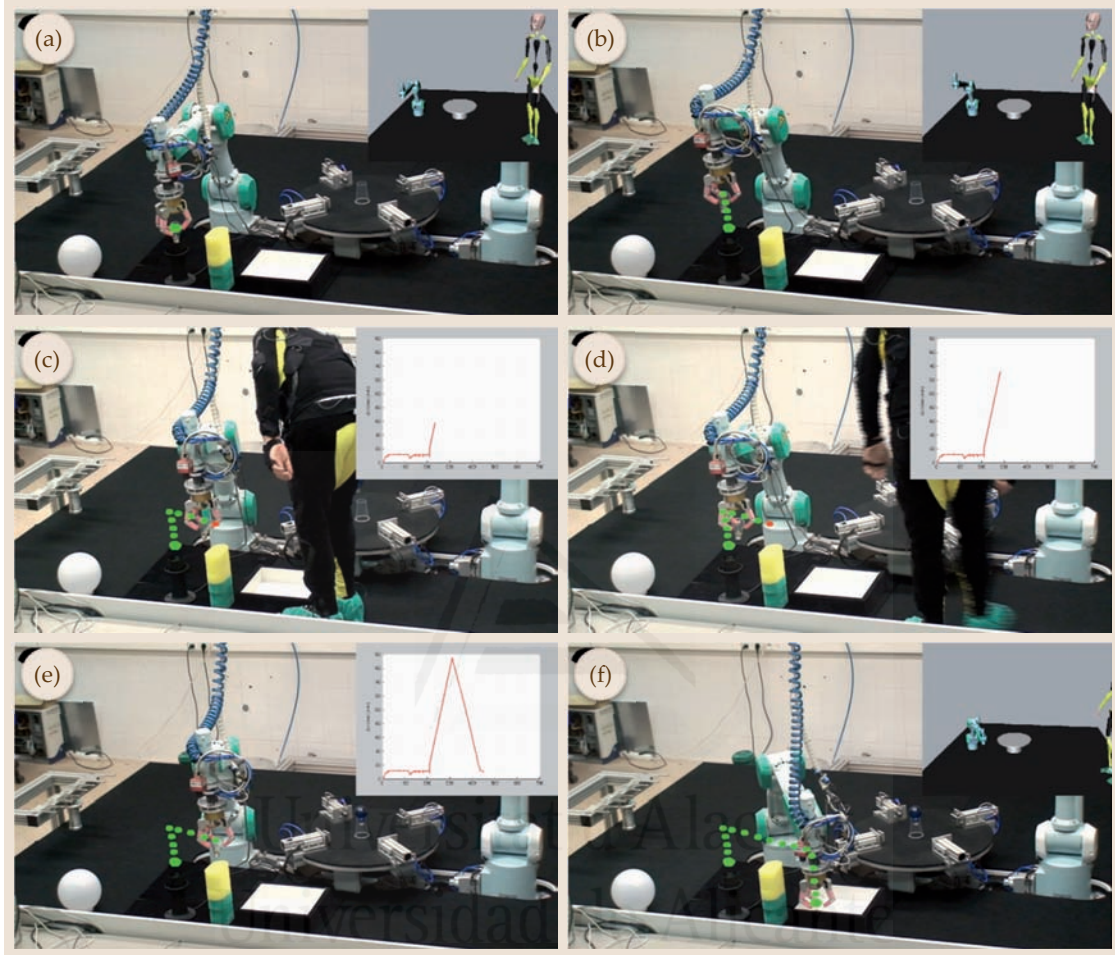


Figure 4.15. Sequence of frames of phases 2 and 3 of the task when the human enters the workplace and leaves the new light bulb over the turning table.

4.6.2 Human-Robot Interaction Task 2: Disassembly of a home appliance

The second human-robot interaction task proposed in this chapter aims to disassemble a home appliance ([Corrales-2010c] and [Corrales-2010d]). The object to be disassembled is a small fridge. The main elements which take part in this task are depicted in Figure 4.16: the fridge, the Mitsubishi PA-10 robotic manipulator which unscrews the fridge lid, the human operator who extracts the internal tray and the storage box where the different parts of the fridge are stored after they are disassembled. The Mitsubishi PA-10 robotic manipulator has three devices installed at its end-effector in order to perform the task: a screwdriver, a JR3 force sensor and an eye-in-hand Photon-focus MV-D752-160-CL-8 camera. The image trajectory is generated by using four laser points projected on the floor as extracted features for the visual servoing path tracking system. The human operator wears an Animazoo GypsyGyro-18 inertial motion capture suit and an Ubisense UWB tag.

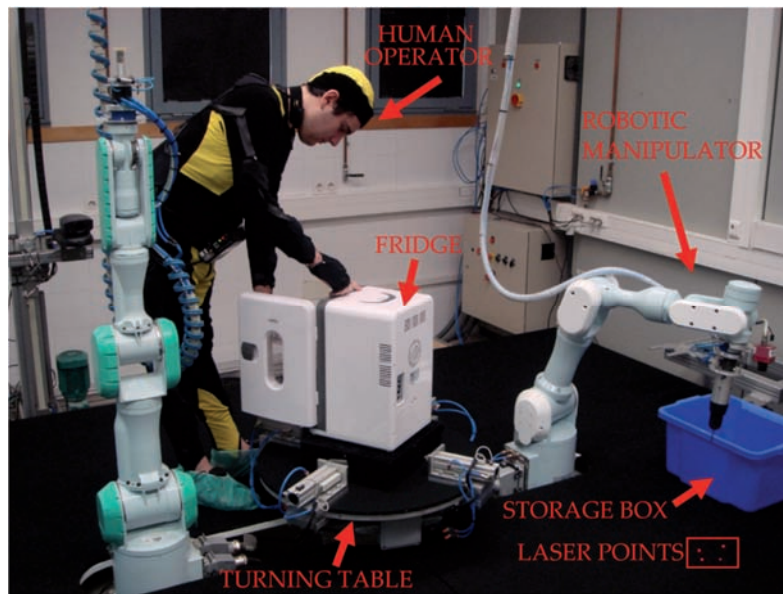


Figure 4.16. Experimental setup for the fridge disassembly task.

The disassembly task can be split in the following subtasks to be performed by the human operator and the robot:

- Robot:** The robotic manipulator has to remove the screws from the rear lid of the fridge. Firstly, the robotic manipulator goes from the storage box to the unscrew position by tracking a predefined path using the image path tracker based on visual servoing described in [Corrales-2010c]. Secondly, the robot unscrews the corresponding screw and then the robot is again guided to the storage box where the screw is left. This task is repeated until it removes all the screws.
- Human operator:** Meanwhile, the human operator has to empty the fridge's contents.

These two tasks described can be performed simultaneously. The safety system presented in this chapter is employed during the entire disassembly task. The sequence of the disassembly task is depicted in Figure 4.17. Each frame of Figure 4.17 is composed of a real photograph of the task and a 3D representation of the bounding volumes of the third level of the hierarchy (i.e. SSLs). The bounding volumes of the other levels (i.e. AABBs) are not shown for the sake of clarity because SSLs permit to see more easily that the configuration of the bounding volumes and the real configuration are the same. Nevertheless, the implemented distance algorithm computes the human-robot distance by applying all the levels of the bounding volume hierarchy (as described in Section 4.5.1) depending on the distance thresholds: $DIST_{1>2} = 2\text{m}$ and $DIST_{2>3} = 1\text{m}$. In this task, only static bounding volumes have been applied.

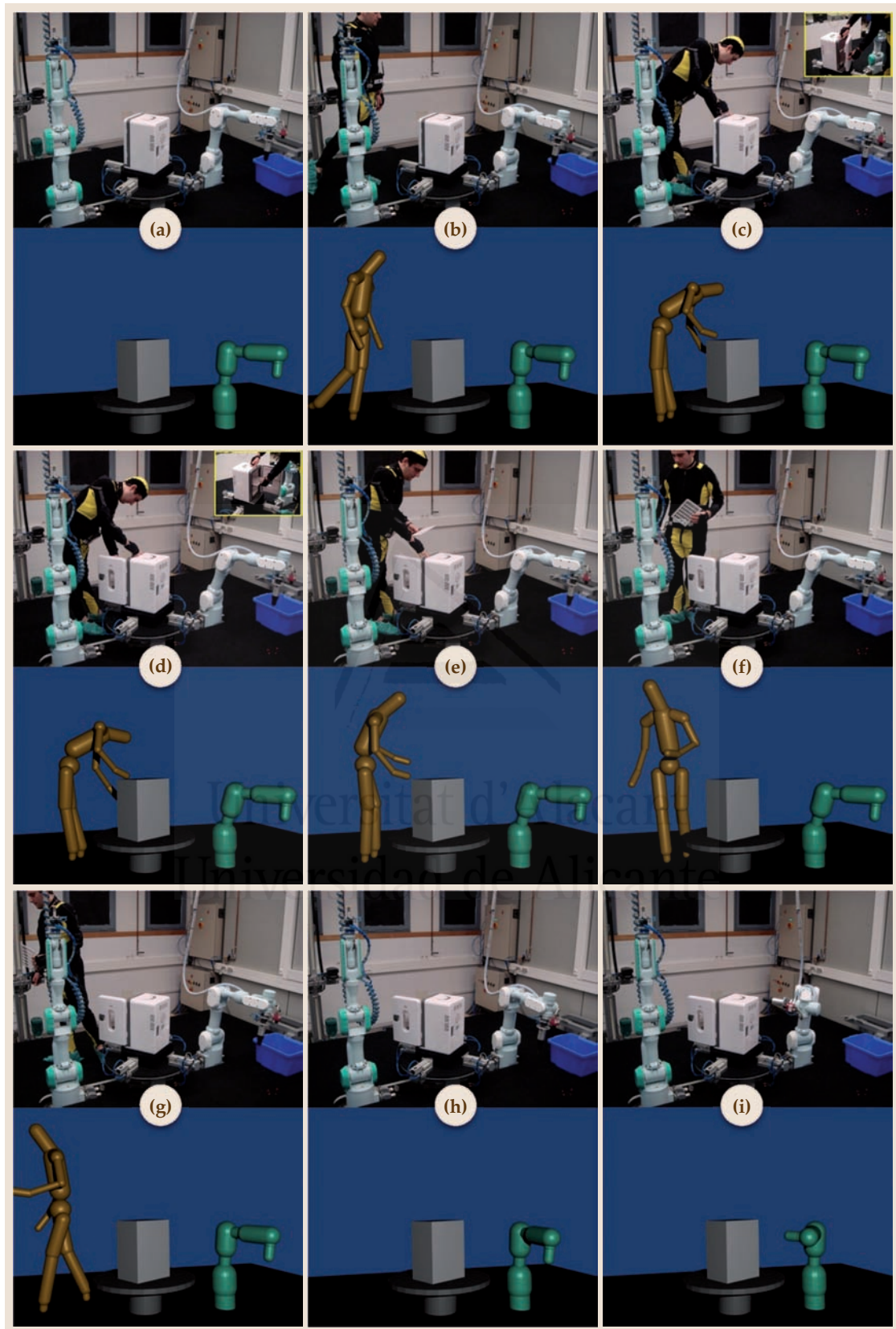


Figure 4.17. Sequence of frames of the fridge disassembly task. Each frame is composed by a photograph of the real experiment and a 3D representation of the bounding volumes of the third level of the hierarchy.

In this task, the image path tracker described in [Corrales-2010c] guides the robot from the storage box to the next screw (see Figure 4.17.a). During this tracking, the human approaches the fridge in order to empty its content (see Figure 4.17.b). The distance between the human operator and the robot falls under the safety threshold activating the safety strategy. The robot goes away from the human operator (see Figure 4.17.c and Figure 4.17.d) and stops until the distance exceeds the safety threshold again. Meanwhile, the human operator opens the fridge's door (see Figure 4.17.d) and takes the internal tray out of the fridge (see Figure 4.17.e and Figure 4.17.f) in order to carry it to a storage box which is out of the workspace. While the human operator is going away from the robot, the human-robot distance is again greater than the safety threshold and the visual servoing path tracking is re-activated (see Figure 4.17.g and Figure 4.17.h). Afterwards, the path is tracked correctly and the robot can arrive at the unscrew position by following the predefined path (see Figure 4.17.i).

Figure 4.18.a depicts the evolution of the computed human-robot distance along the development of the task. The temporal evolution of this plot is represented in iterations of the safety strategy of the system, which correspond to 33 ms each one since the safety strategy is executed with a frequency of 30 Hz. From iteration 102 to iteration 223 of the human-robot system execution, the human operator approaches the fridge and the human-robot distance is reduced until it goes below the safety threshold (1m in this case). Then, the safety behavior is activated and the robot maintains the safety distance between iterations 235 and 548. Afterwards, the human-robot distance is again greater than the safety threshold when the human takes away the fridge's contents to an external storage box. The normal behavior of the robot is taken up again.

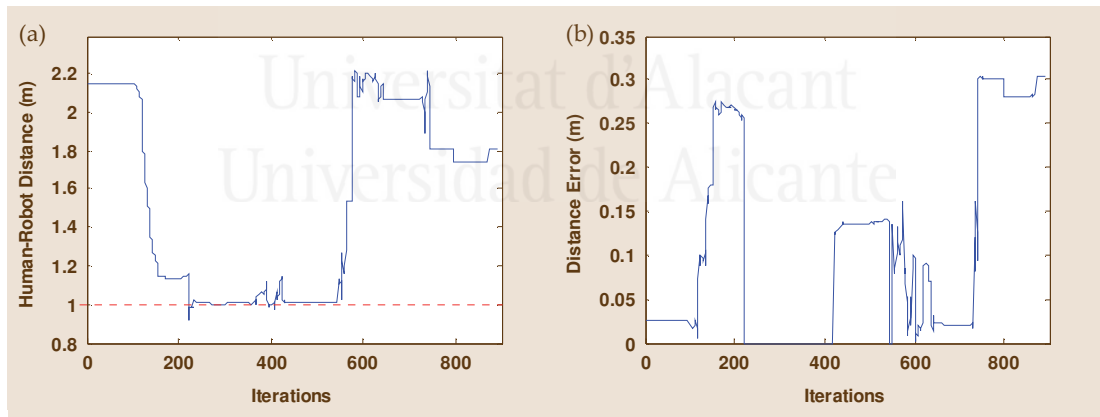


Figure 4.18. (a) Evolution of the human-robot distance; (b) Evolution of the distance error from the hierarchy of bounding volumes.

Figure 4.18.b depicts the error between the human-robot distances calculated from the hierarchy of bounding volumes (with the algorithm of Section 4.5.1) in comparison with the distances directly calculated from the SSL bounding volumes (i.e. by using always the bounding volumes of the third level of the hierarchy of Section 4.3). The SSL bounding volumes are used as ground-truth values in this case because they represent the better approximation of the human/robot surfaces in the three levels of the hierarchy of bounding volumes. This plot shows that there are distance errors of approximately 13 cm when applying the second level of the hierarchy of bounding volumes (i.e. local AABBs) and errors of 30 cm when applying the first level of the hierarchy of bounding volumes (i.e. global AABBs). These values are not important in com-

parison with the reduction of the computational cost of the distance computation. Furthermore, these errors are always positive since they are upper bounds of the human-robot distance and thus, the safety of the human is always guaranteed.

Figure 4.19 shows a histogram of the number of pairwise distance tests which are executed during the task. In the 76.8% of the executions of the distance algorithm, the number of performed pairwise tests is less than 20. Between 20 and 80 pairwise tests are required in 12% of cases. Finally, only in 11.2% of cases, the distance algorithm needs to perform more than 80 pairwise tests. These results prove that the distance algorithm presented in Section 4.5.1 reduces substantially the number of required pairwise distance tests in comparison with a simple bounding volume approach without any kind of hierarchy which will always need 144 tests for the SSLs.

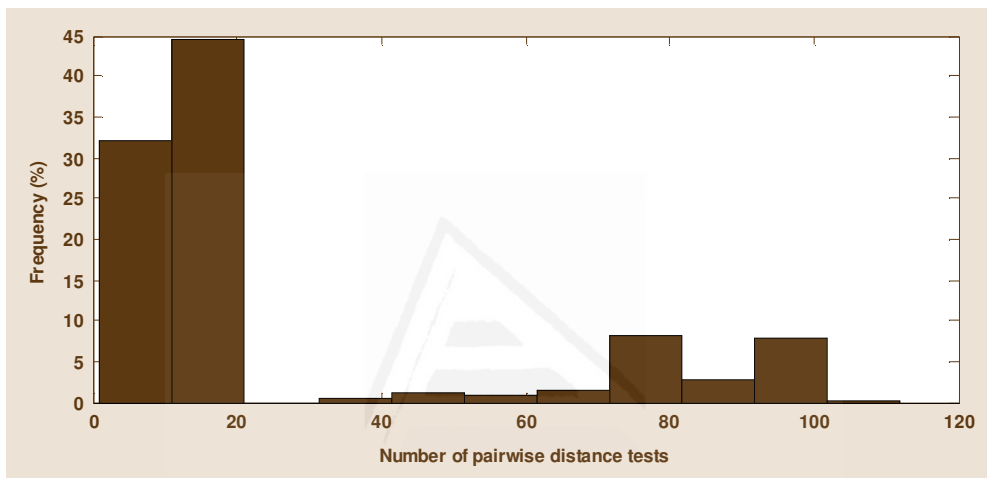


Figure 4.19. Histogram of the number of distance tests required for the minimum human-robot distance computation.

4.6.3 Human-Robot Interaction Task 3: Assembly of a metallic structure

The third human-robot interaction task developed in this chapter involves the assembly of a metallic structure where two robotic manipulators and a human operator cooperate [Corrales-2011]. The metallic structure to be assembled is composed of cylindrical tubes which are assembled between them by screwing metallic connectors of several shapes (see Figure 4.20.a). Figure 4.20.b depicts the experimental setup of this task: the metallic structure to be assembled whose base is fixed to the turning table, the two *Mitsubishi PA-10* robotic manipulators which handle the metallic tubes, the human operator who screws the connectors to the tubes and a storage box of tubes. The human collaboration is required to install the connector between the tubes because they are composed of two parts and they are very difficult to be handled by a robot. In addition, the use of a second manipulator is also needed in order to perform several actions of the assembly task in parallel. The robotic manipulator R1 has the following elements at its end-effector: a two-jaw pneumatic gripper for handling tubes, a *JR3* force for detecting the collision between tubes and connectors during tube insertion tasks and an eye-in-hand *Photonfocus MV-D752-160-CL-8* camera for tracking the path between the tube storage box and the turning table where the metallic structure is fixed. The robotic manipulator R2 has a three-fingered *Barrett* robotic hand which is used to hold the tubes which are screwed by the human operator. The human operator wears an *Anima-zoo GypsyGyro-18* inertial motion capture suit and an *Ubisense* UWB tag. This task is a

subtask of a more complex assembly task which is detailed in [Corrales-2008b], [Gil-2009] and [Pomares-2009]. In this section, only the part of this general task where the human operator collaborates with the robots is explained since the other parts of this general task do not belong to the scope of this thesis.

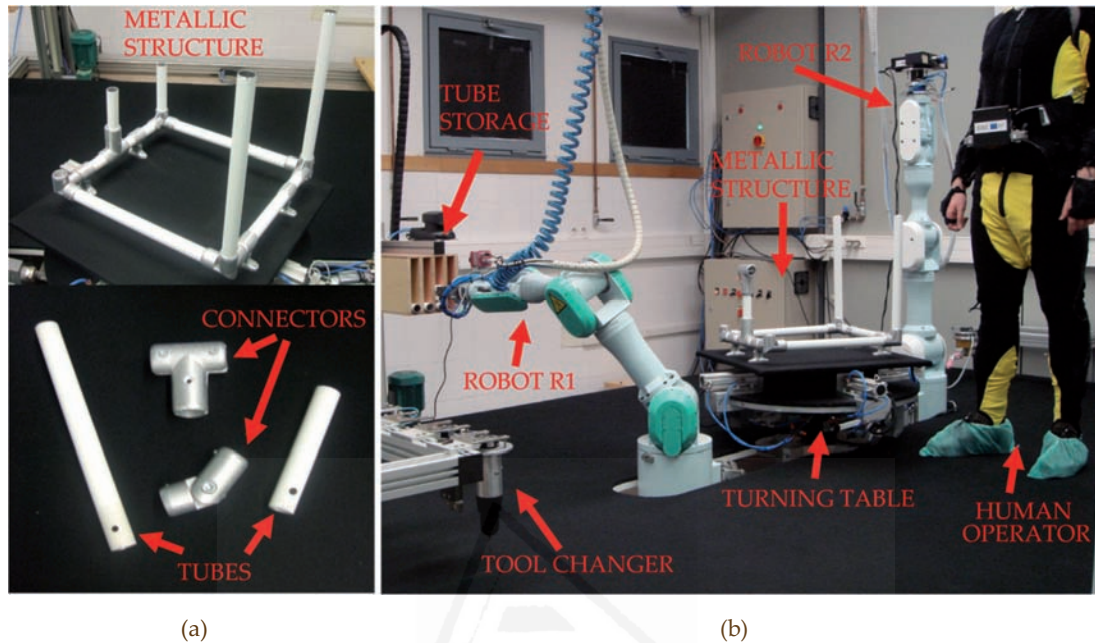


Figure 4.20. (a) components of the metallic structure which are used during the assembly task; (b) experimental setup of the metallic structure assembly task.

This task is organized into three different phases according to the different actions which are performed by the three agents (two robotic manipulators and a human operator) that take part in it. Figure 4.21 represents the different components which are assembled to the metallic structure during each phase of the task. The goal of the first manipulator is to insert a new tube (tube 2) in one connector of the structure while the goal of the second manipulator is to hold the tube (tube 1) where two T-connectors are screwed by the human operator.

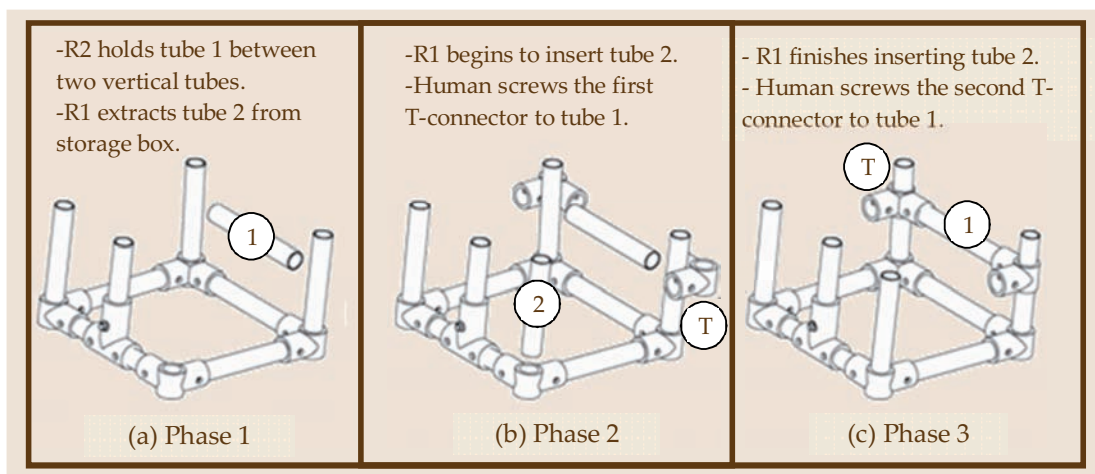


Figure 4.21. Components which are assembled to the metallic structure during the three phases of the assembly task.

Figure 4.22 represents not only photographs of the real experiment but also the corresponding dynamic bounding volumes from the third level of the hierarchy of bounding volumes. As in the previous section, only SSLs are represented for the sake of clarity but the human-robot distance employed by the safety strategy uses the three levels of the bounding volume hierarchy (as described in Section 4.5.1) and updates the dimensions of the dynamic bounding volumes of the third level according to the linear velocities of their contained links (as described in Section 4.4). The values of the distance thresholds which determine the applied level of the bounding volume hierarchy are the following ones: $DIST_{1>2} = 2\text{m}$ and $DIST_{2>3} = 1\text{m}$.

The three phases which composes the task can be watched in Video 4.3 in the attached DVD and they are explained in the following paragraphs:

1. *Tube handling* (Figure 4.21.a, Figure 4.22.a and Figure 4.22.b): One of the manipulators extracts one metallic tube from the storage box in order to insert it in one of the free connectors of the metallic structure (see Figure 4.22.a). The 3D representation of Figure 4.22.b shows how the dynamic bounding volumes of this manipulator grow in size while it is moving quickly. The other manipulator handles a tube with the Barrett Hand at a fixed position where the human operator will screw the connectors (see Figure 4.22.d). This manipulator will not move during all the execution of the task and thus the safety strategy will not be applied to it because it does not perform dangerous movements for the human. Meanwhile, the human operator prepares the connectors for the metallic structure outside the manipulators' workspace.
2. *First T-connector screwing* (Figure 4.21.b, Figure 4.22.c and Figure 4.22.d): The human operator enters in the manipulators' workspace in order to screw the first T-connector to the tube which is held by the second manipulator. The 3D representations of Figure 4.22.c and Figure 4.22.d show how the dynamic bounding volumes of the human operator change their sizes depending on their linear velocity. Meanwhile, the first manipulator continues approaching the metallic structure in order to insert the tube which has been extracted from the storage box in the previous phase (see Figure 4.22.c). It performs a visual servoing path tracking process in order to move towards the metallic structure. On the other hand, the safety strategy is executed constantly and computes the minimum distance between the dynamic bounding volumes of the human operator and the moving robot. When this human-robot distance is smaller than the safety threshold (0.5m in this case), the safety strategy stops the visual servoing path tracking of the first robot and activates a safety behavior which moves the robot away from the human in a linear path (see Figure 4.22.d) and tries to keep the human-robot distance above the safety threshold. While the human-robot distance is equal to the threshold, the robot keeps still.
3. *Second T-connector screwing* (Figure 4.21.c, Figure 4.22.e and Figure 4.22.f): After having screwed the first T-connector to the tube, the human operator picks up the other T-connector and walks around the workspace until reaching the other end of the tube (see Figure 4.22.e). The first manipulator resumes its motion when the human goes away and the distance between them is greater than the safety thresh-

old. This manipulator continues performing its task and inserts the tube it is carrying in the structure. Meanwhile, the human operator screws the second T-connector at the other end of the tube in order to fix it to the structure (see [Figure 4.22.f](#)). In this phase, the safety behavior is not activated because the distance threshold is not exceeded.

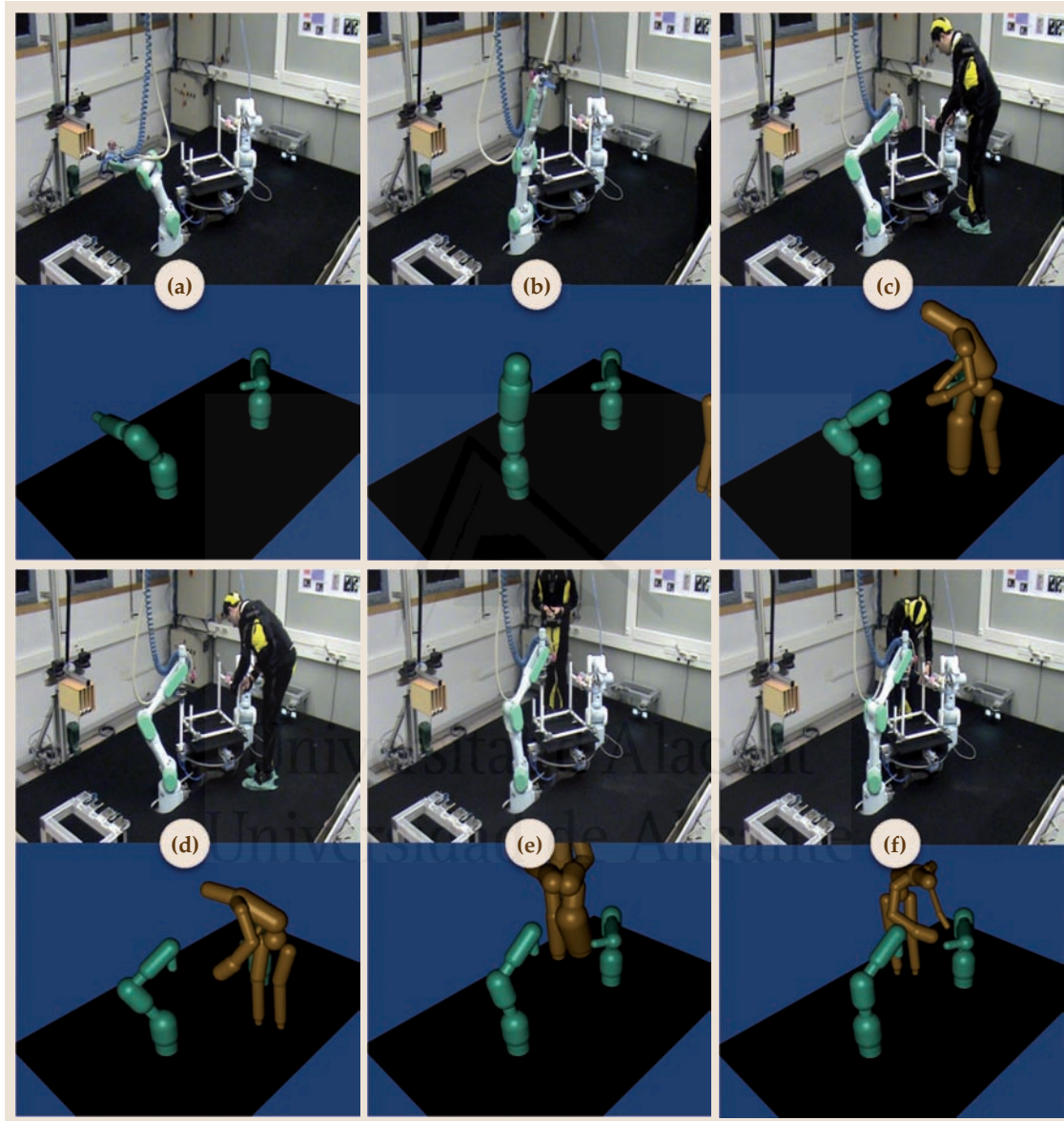


Figure 4.22. Sequence of frames of the metallic structure assembly task. Each frame is composed by a photograph of the real experiment and a 3D representation of the dynamic bounding volumes of the third level of the hierarchy.

In this task, the bounding volumes of the third level of the hierarchy are dynamic because the safety threshold is smaller (i.e. 0.5m) than in the previous human-robot interaction tasks (i.e. 1m) due to the proximity of the operation areas of the first robot and the human operator. Thereby, the velocities of the human operator and the robot are reflected on the bounding volumes and the safety of the human-robot interaction system is increased in order to avoid any collision between the human operator and the robot which is approaching the metallic structure. [Figure 4.23](#) and [Figure 4.24](#) de-

Figure 4.23 depicts a more detailed evolution of the radii of these dynamic bounding volumes. Only the changes of several bounding volumes have been represented for the sake of clarity. In particular, Figure 4.23 shows the changes in the radii of the three last links of the first robotic manipulator. The plots in Figure 4.24 show the changes in the radii of the left foot and the right hand of the human operator. These are the links with more radius variations because they are the terminal nodes of the skeleton with the maximum linear velocities.

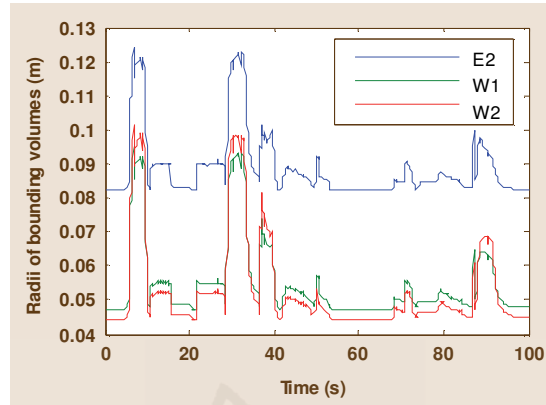


Figure 4.23. Evolution of the radii of the bounding volumes which cover the last three links (E2, W1 and W2) of the first robotic manipulator.

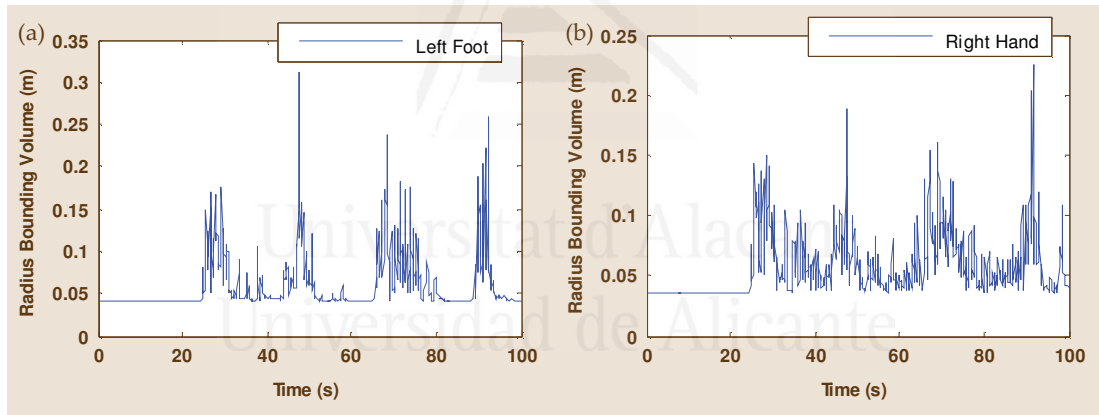


Figure 4.24. Evolution of the radii of the bounding volumes which cover (a) the left foot and (b) the right hand of the human operator.

Figure 4.25 depicts the evolution of the minimum human-robot distance obtained by the distance algorithm of Table 4.2 (including dynamic bounding volumes) for the assembly task. This plot shows how the human operator approaches the robot while the robot performs the time-independent visual servoing path tracking from iteration 1 to iteration 289. In iteration 290, the safety behavior starts and the robot controller pauses the path tracking. The safety behavior is executed from iteration 290 to iteration 449 and it tries to keep the human-robot distance above the safety threshold (0.5m). In iteration 450, the robot controller re-activates path tracking because the human-robot distance is again greater than the threshold when the human is going away from the workspace to take the second T-connector.

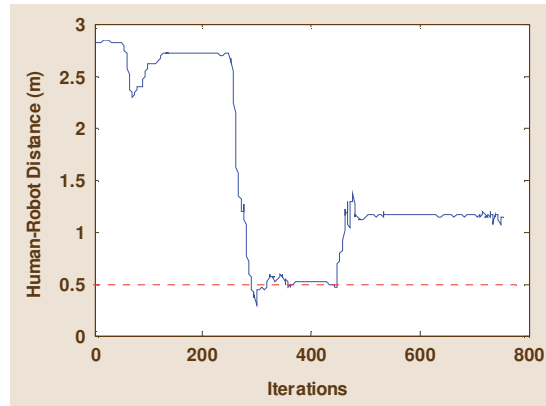


Figure 4.25. Evolution of the minimum human-robot distance during the metallic structure assembly task.

Figure 4.26.a depicts the error evolution of the distance obtained by the algorithm in Table 4.2 with regard to the distance values obtained from the SSL bounding volumes of the third level of the hierarchy, which are used as ground-truth values. This figure shows an assumable mean error of 4.6cm for distances greater than 1m. For distances smaller than 1m (between iterations 280 and 459), the error is null because the SSLs are used for the distance computation. The proposed distance algorithm in Table 4.2 obtains more precise distance values than simpler distance algorithms which do not consider the dimensions of the human and the robot bodies [García-2009a]. In particular, Figure 4.26.b shows the difference between the distance values obtained by the algorithm in Table 4.2 and the distance values computed by a algorithm which only considers the distance between the hips node of the human and the end-effector of the robot. The errors of this plot are in many cases higher than the safety threshold (0.5m). This result shows the necessity of considering the full structure of the human and robot bodies and using the proposed hierarchy of bounding volumes.

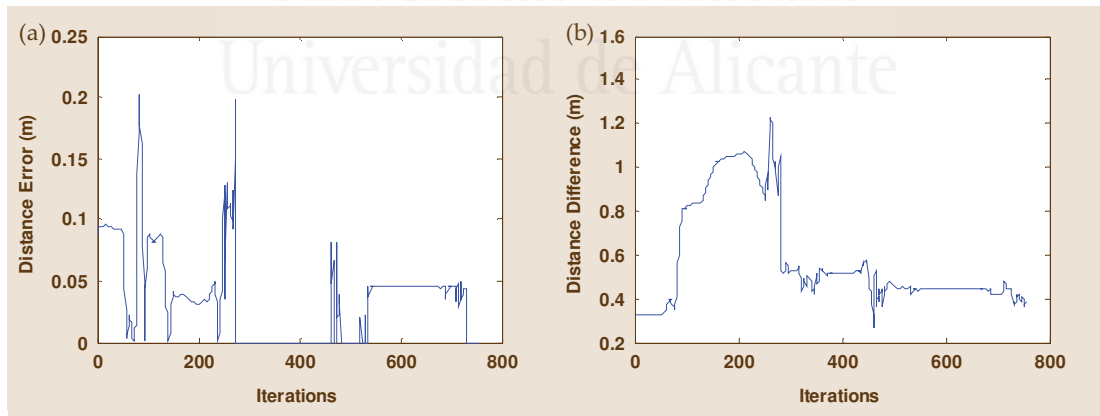


Figure 4.26. (a) Evolution of the distance error from the hierarchy of bounding volumes; (b) evolution of the difference between computing the distance with and without bounding volumes.

Figure 4.27 shows the histogram of distance tests which are performed for the distance computation during the disassembly task. In 64% of the executions of the distance algorithm, a reduced number of pairwise distance tests is required (between 1 and 16 tests) because the bounding volumes of the first and/or second levels of the hierarchy (AABBs) are used. In the remaining 36%, between 30 and 90 distance tests are executed for the third level of the hierarchy (i.e. dynamic SSLs). This fact demonstrates

that the hierarchy of bounding volumes involves a significant reduction of the computational cost of the distance computation with regard to a pairwise strategy where 144 distance tests would always be executed.

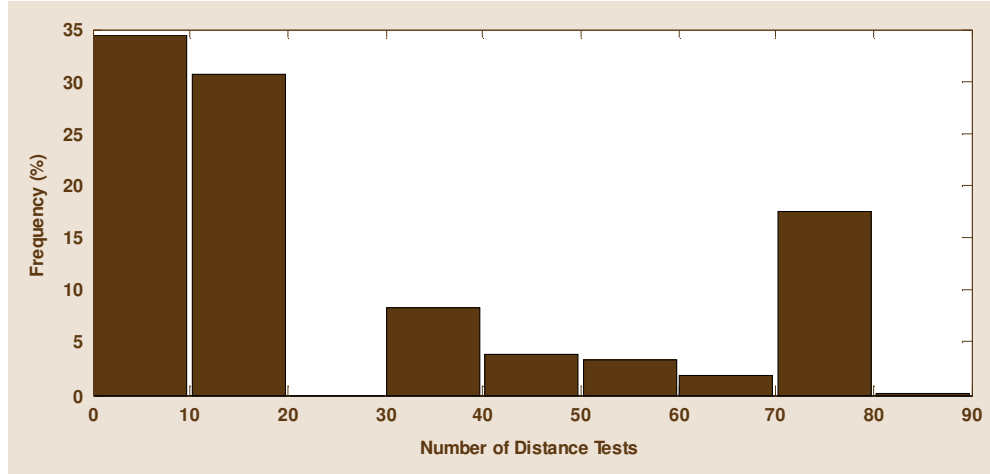


Figure 4.27. Histogram of the number of distance tests required for the minimum human-robot distance computation.

4.7 Conclusions

This chapter presents several contributions of this thesis. The first contribution of this chapter is a novel geometric representation of the human body and the structure of robotic manipulators for the development of safe human-robot interaction tasks in industrial environments. This representation is composed of two main elements for each agent (robot or human): a skeleton and a hierarchy of bounding volumes which cover the skeleton. The skeleton models the kinematic structure of the agent by a hierarchy of linear links and rotational joints. This hierarchy identifies the dependencies between the motion of each link and the other links which are connected to it through a kinematic chain. Forward kinematics is able to obtain the position of all the links of this skeleton from the rotation angles of the joints. These joint angles are obtained from the robotic controller in the case of the robot and from the inertial motion capture system presented in [Chapter 2](#) for the human operator. The global position of the robot's skeleton is fixed while the global position of the human's skeleton is updated with the sensor fusion algorithms described in [Chapter 3](#).

Bounding volumes are required to cover the surface of each link of these skeletons in order to model the dimensions of the bodies of the human and the robot. Typical sphere-based solutions are not convenient because a high number of small spheres is required in order to cover each link. This high number of spheres increases the computational cost of the proximity queries (collision detection and distance computation) between the robot's links and the human's limbs. The swept-sphere line representation described in this chapter does not have this problem because these bounding volumes adapt better to the cylindrical shape of the majority of the links in robotic manipulators and human beings (as demonstrated in [Table 4.1](#)). In this representation, only one bounding volume is enough to cover with sufficient precision the surface of the links of the robot and the human operator.

In addition, since the main goal of these bounding volumes is to compute efficiently the minimum human-robot distance, not only bounding volumes based on SSLs have been applied but a complete hierarchy of bounding volumes. This hierarchy is composed by three levels which are used depending on the value of the human-robot distance. The first level of this hierarchy is composed by two global AABBs which cover the complete bodies of the human and the robot. If the distance between these AABBs is bigger than a pre-established threshold ($DIST_{1>2}$), this distance value is used for the safety strategy since it is unnecessary to calculate a more detailed value. Otherwise, the second level of the hierarchy composed by local AABBs is applied to update the human-robot distance. This second level represents the main limbs of the human and the robot bodies. If this second approximation of the human-robot distance is again smaller than a second distance threshold ($DIST_{2>3}$), the third level based on SSLs is applied to compute the human-robot distance. This organization reduces significantly the number of pairwise distance tests required to compute the minimum human-robot distance, as demonstrated by the histograms of [Figure 4.19](#) and [Figure 4.27](#).

The second contribution of this chapter is the development of dynamic bounding volumes for the third level of the hierarchy. In particular, the radii of the SSLs change during the development of the human-robot task depending of the linear velocity of the link which is encapsulated by each bounding volume. Thereby, the radius of each bounding volume does not only include the maximum size of the encapsulated link, but also the estimation of the distance which can be covered by the link during each sampling period. This chapter describes in detail the novel algorithm which has been implemented in order to compute the dynamic radii of these bounding volumes. Firstly, this algorithm obtains the angular rates of the joints from joint angle samples with an adaptive windowing technique. Secondly, the angular and linear velocities of each link are calculated by considering the kinematic hierarchy of the skeleton. Finally, the radius of each bounding volume is updated according to the estimated linear velocity of the encapsulated link and the minimum distance between each pair of bounding volumes is computed.

Finally, the third contribution of this chapter is the application of the dynamic bounding volume algorithm and the hierarchy of bounding volumes in the implementation of safety strategies which modify the behavior of robotic manipulators which collaborate with humans in order to guarantee the avoidance of collisions. These safety strategies compute the minimum distance between the hierarchy of bounding volumes of the human operator and each robotic manipulator. This distance value is compared with a pre-established safety threshold and if the distance is below the threshold, a special safety behavior is activated. This safety behavior can involve stopping the normal path tracking of the robot or performing a linear path in order to go away from the human operator until the human-robot distance is again higher than the safety threshold. This second safety behavior has been applied in three different human-robot interaction tasks which have been developed in order to check the correct performance of the proposed human-robot safety system. In fact, the results of these real experiments verify that the proposed algorithms are suitable for the development of safe human-robot interaction tasks, which is one of the main goals of this thesis.

Dexterous Manipulation for In-Hand Object Reconfiguration

This chapter presents a novel dexterous manipulation planner for reconfiguring objects which are grasped by a multi-fingered hand during the development of human-robot interaction tasks. The proposed planner receives as input a current grasp of the object and computes the movements of the fingers of a robotic hand which are required to move the object to a desired configuration. It is divided into two components: a global planner and a local planner. The global planner generates the intermediate trajectory of the object while the local planner generates the small movements of the fingers which are needed to change between consecutive configurations of this trajectory. This local planner is based on a triangle mesh representation of the fingers and the object with rolling contacts. In addition, a finger readjustment technique based on tactile information has been added to the local planner in order to avoid undesired sliding. This dexterous manipulation planner constitutes one of the main contributions of this thesis.

5.1 Introduction	107
5.2 Review of Planning Techniques for Dexterous Manipulation	109
5.2.1 Manipulation Planning Strategies based on Graph Search Techniques	109
5.2.2 Manipulation Planning Strategies based on Hybrid Control Systems	112
5.2.3 Manipulation Planning Strategies based on Probabilistic Techniques	114
5.3 Representation of the Surfaces of the Hand and the Object	116
5.4 Geometric Manipulation Planner for In-Hand Manipulation	117
5.4.1 General Structure of the Geometric Manipulation Planner	117
5.4.2 Global Planner	119
5.4.3 Local Planner	120
5.5 Contact Evolution between Surface Primitives	122
5.5.1 Contact Evolution Graph	122
5.5.2 Contact Transitions Generated by Rolling	123
5.6 Finger Readjustment based on Tactile Information	125
5.7 Experimental Results	127
5.7.1 In-Hand Manipulation with Shadow Hand	127
5.7.2 In-Hand Manipulation with Barrett Hand	130
5.8 Conclusions	135

5.1 Introduction

As stated in [Chapter 1](#), one of the main goals of this thesis is the development of human-robot interaction tasks where human operators collaborate with robotic manipulators. Two requirements which are essential to perform these tasks are: the guarantee of the safety of human operators and the implementation of dexterous manipulation techniques which can be performed by robotic manipulators. The first requirement has been developed in this thesis by implementing safety strategies (as described in [Chapter 4](#)) which compute the minimum human-robot distance by taking into account the pose measurements registered by a multi-sensor human tracking system (as described in [Chapter 2](#) and [Chapter 3](#)) and the robot controller.

The second requirement is the implementation of special techniques to control the way robots manipulate the objects that take part in the task. The development of ma-

nipulation techniques for robotic manipulators enables the execution of more flexible human-robot interaction tasks where human operators and robotic manipulators exchange objects. In order to develop these tasks, the use of multi-fingered robotic hands is becoming more and more widespread because of its higher flexibility in comparison with traditional parallel jaw grippers. As described in [Chapter 1](#), one of the main restrictions which have to be considered for the development of manipulation tasks with multi-fingered robotic hands is their severe workspace limits [\[Han-1998\]](#) caused by the common use of coupled, under-actuated or passive joints [\[Melchiorri-2008\]](#). Due to this drawback, dexterous manipulation [\[Okamura-2000\]](#) is the most suitable manipulation technique for human-robot interaction tasks with multi-fingered robotic hands because it not only permits to change the configuration of the object (i.e. the position and orientation of the object) but also the configuration of the contact points (i.e. rolling and/or sliding). In fact, the possibility of changing the contact points reduces the disadvantages of kinematic limits of the fingers and thus, multi-fingered hands are able to perform more varied manipulation tasks than object manipulation tasks based on fixed contact points.

Nevertheless, dexterous manipulation is a complex problem since it involves taking into consideration an important number of constraints which restrict the number of possible solutions for the problem. The most important constraints which have to be considered are the following ones [\[Li-1989\]](#):

- *Constraints by finger kinematics:* The contact points with the object should be located inside the workspace of the fingers. Each finger is considered as a kinematic chain and each contact point should be a solution for the inverse kinematic problem of the corresponding finger.
- *Constraints for contact maintenance:* The finger should keep the contact with the object and contact breaking should only be allowed when finger gaiting strategies are applied. This constraint is generally represented by making equal the velocities of the finger and the object on the contact points.
- *Constraints for equilibrium:* The contact forces exerted by the fingers over the object through the contact points should balance the external forces (generally, the weight) which are applied over the object in order to keep equilibrium. This equilibrium condition is extended in some manipulation planners by developing force closure so that the fingers are able to balance any external force over the object.

All these constraints are taken into account in order to solve the object reconfiguration problem, which is the main goal of dexterous manipulation planners. The reconfiguration planning problem [\[Brock-2008\]](#) can be stated as follows: Starting from a given initial grasp of an object (defined by the configurations of the object and the fingers), find feasible trajectories (motions and contact forces) for the fingers to move the object to a desired final configuration.

This chapter presents a novel planner in order to solve the object reconfiguration problem in human-robot interaction tasks with multi-fingered hands. This planner is

based on the representation of the fingers and object surfaces as triangle meshes. Firstly, the proposed planner resolves the constraints of contact maintenance by computing the evolution of the contacting primitives (vertices, edges and faces) of these triangle meshes. Secondly, this planner applies the fingers' Jacobian matrix in order to compute the movements of the fingers which are required to generate the previously mentioned evolution of contacting primitives. Thereby, this step of the planner takes into account the constraints of the fingers kinematics. Finally, a novel finger readjustment technique is proposed in order to increment the contact forces over the object according to the information from tactile sensors installed over the fingers. These finger readjustments guarantee, with small corrections over the fingers' configurations previously computed by the Jacobian matrix, that the fingers do not lose contact with the object and thus, the equilibrium constraints are fulfilled and unstable grasps are avoided.

This dexterous manipulation planner based on the geometric representation of the contacting surfaces and the finger readjustments with tactile information is a contribution of this thesis. The following section describes previous research that has been developed to solve this manipulation planning problem. The subsequent sections will describe in detail the different steps of the proposed planner and will apply it on several experiments in order to verify its performance.

5.2 Review of Planning Techniques for Dexterous Manipulation

The planning strategies developed in the last years for the dexterous manipulation of objects by multi-fingered hands can be organized in three main types: strategies based on search algorithms over graphs, strategies based on control schemes for hybrid systems and strategies based on probabilistic motion planning techniques. In the following sections, the most important research works on these planning strategies are explained in detail.

5.2.1 Manipulation Planning Strategies based on Graph Search Techniques

The representation of the object reconfiguration problem as a path search over a graph is a common technique in many manipulation planning strategies. Nevertheless, the meaning of the nodes and edges of these graphs is different depending on the developed approach. In particular, two different graph representations are generally implemented. In the first graph representation, each node of the graph is equivalent to a specific grasp configuration which entails a pre-established contact relation between the fingers of the hand and the object. The edges of this graph represent the possible transitions between the nodes due to fingers' movements. In the second graph representation, each node of the graph is equivalent to a configuration of the object and its edges are transitions between these configurations. Thereby, in this graph representation, the nodes do not contain direct information about the fingers of the hand and the contact restrictions are verified by a lower level planner which changes the search over the graph accordingly.

In the first type of graph-based planners, the works by Trinkle and Hunter [Trinkle-1991] and Zhang et al. [Zhang-1996] can be emphasized. Trinkle and Hunter [Trinkle-1991] present a solution for the dexterous manipulation planning problem which considers the initial and final grasps as Contact Formations (CF). A CF is a description

of the grasp based on the fundamental elements (edges and vertices) of the surfaces of the object and the fingers which are in contact. The developed algorithm builds two CF trees from the initial and final grasps which are grown simultaneously until they have a common node/CF. Then, the path of CFs which connects the initial CF to the final CF through both trees constitutes the solution of the planning problem. The algorithm has been tested over a planar dexterous manipulation problem with five degrees of freedom.

Zhang et al. [Zhang-1996] develop a similar framework for planning dexterous manipulation based on the extension of the Contact Formations with the concept of Canonical Grasps (CGs). A CG is a set of pairs of topological features of the hand and of the object which are in contact during a grasp. The topological features of polyhedral objects are vertices, edges and planes while smooth objects are represented topologically by surface patches and curved edges. The proposed approach generates all the possible CGs which are relevant to the performed task and organizes them as nodes in a Grasp Transformation Graph (GTG). The edges of this graph represent the feasibility of the transition between each pair of CGs and are established analytically from kinematic constraints. Finally, for a given manipulation task, the manipulation planner associates the initial and final states of the task with two nodes of the GTG and searches a path between them in order to obtain a sequence of CGs which completes the task. This manipulation planner has been tested on the manipulation of a rectangular object with a three-fingered hand whose links have been simplified to segments as topological features for the algorithm. Figure 5.1.a depicts several of the intermediate CGs which have been calculated for a manipulation task which changes from a pinch grasp of the object towards a power grasp and Figure 5.1.b shows the corresponding CGs in the real experiment.

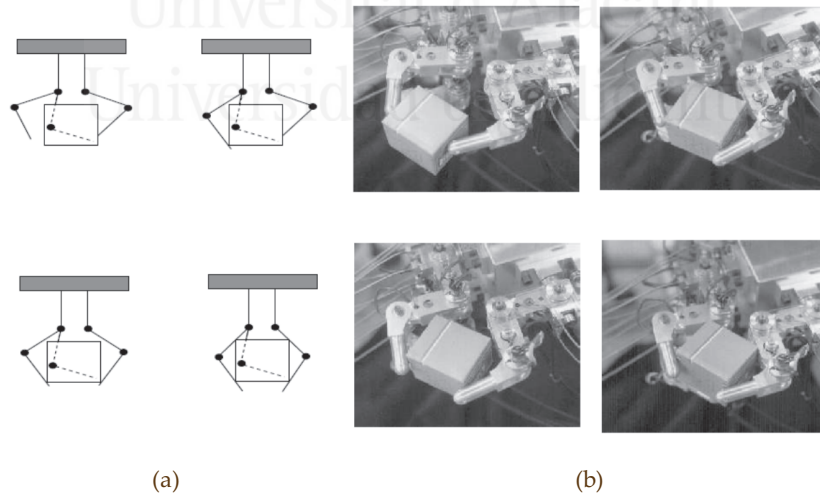


Figure 5.1. (a) sequence of CGs for manipulating a rectangular object; (b) the same CGs with a real hand, Source: [Zhang-1996].

In the second type of manipulation planners based on graph path searches, the graph is used to model the configuration space of the object. Cherif and Gupta [Cherif-1999] present a dexterous manipulation planner organized in two levels (global and local planner) where the first level performs a grid search algorithm that explores the configuration space of the object. The configuration space of the object (position and orientation) is divided into a collection of non-overlapping cells which form a 6-

dimensional discrete grid. First of all, the global planner finds the two cells where the initial and final configurations of the desired manipulation task are contained. Next, it executes an A* search algorithm in order to find a sequence of consecutive cells in the grid which connects the initial and final configurations of the object. At each iteration, the search algorithm determines the adjacent cells to the current one and chooses the one which minimizes a cost function. This cost function is defined as the distance between the current and the goal cells in the configuration space but it does not take into account any constraint due to the interaction between the hand and the object. Due to this fact, in the next step, the global planner invokes the local planner which uses Montana's equations [Montana-1988] in order to verify if the transition between both cells is feasible by the hand-object system. If the transition is possible, the global planner establishes as current cell the new one and repeats the process. If the transition is not possible, the global planner discards this cell and chooses the cell with the smallest cost among the remaining adjacent cells. Thereby, the global planner constructs iteratively a manipulation graph whose nodes correspond to the cells of the object configuration space and whose edges correspond to the transitions between configurations of the object in adjacent cells. This planner has been applied for the dexterous manipulation of an ellipsoidal object by three spherical fingertips. In [Cherif-2001], this method is extended for convex polyhedral objects and the local planner is restricted to a finger tracking strategy [Rus-1999] where only the contact points of a fingertip change while the other fingertips maintain the same contact points.

Vass et al. [Vass-1999] [Vass-2005] also present a two-level planner for the resolution of the dexterous manipulation problem. The global planner uses the same strategy as Cherif and Gupta in [Cherif-1999] and [Cherif-2001]. It divides the configuration space of the object in a cell grid and applies an A* search algorithm in order to form a path of intermediate configurations in different cells (subgoals) between the initial and final configurations. The local planner tries to find motion trajectories of the fingertips and the object between each pair of subgoals computed by the global planner. The main contribution of Vass et al. is the implementation of a simulated annealing algorithm in order to avoid local minima of the motion planning objective function. The proposed simulated annealing algorithm minimizes an energy function composed by the following criteria: no collision between the fingers, force equilibrium of the object, minimization of the absolute sum of the contact forces and minimization of the number of relative velocity changes. This algorithm has been applied on the manipulation of a sphere by four spherical fingertips, as shown in Figure 5.2.

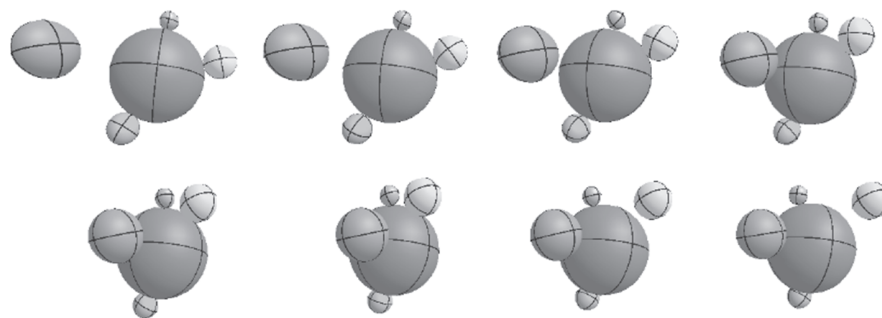


Figure 5.2. Manipulation of a sphere by four spherical fingers with finger relocation, Source: [Vass-2005].

5.2.2 Manipulation Planning Strategies based on Hybrid Control Systems

Another way for solving the dexterous manipulation planning problem is control theory. In this case, the current configuration of the object is seen as the current state of the system to be controlled and the aim of the planner is to define a control law which makes the system evolve from the current state towards the final configuration, which constitutes the desired state of the system and the reference for the control law. Nevertheless, classical control techniques cannot be directly applied to the dexterous manipulation planning since it is a complex problem which involves not only a continuous evolution of the system state (i.e. changes in the path of the object and the fingers) but also a discrete evolution (i.e. contact switch from rolling to sliding or vice-versa and finger gaiting). Due to this combination of continuous and discrete behaviors, dexterous manipulation systems should be represented as hybrid systems. Two main hybrid representations have arisen for dexterous manipulation planning: stratified systems and hybrid automata.

Stratified motion planning for robotic manipulation is presented by Goodwine in [Goodwine-1999]. The configuration space S_0 of a manipulation task of an object by a multi-fingered robotic hand (or by several robotic manipulators) can be organized in different strata according to the relative configurations between them. For instance, one stratum S_1 of the configuration space is composed by all the system configurations where finger 1 of the hand is in contact with the object. Another stratum S_2 is defined when finger 2 is in contact with the object. Similarly, when both fingers are simultaneously in contact with the object, another stratum S_{12} can be defined. In this case, this stratum S_{12} is obtained from the intersection of the previous strata (S_1 and S_2). This process can be repeated when more fingers contact with the object and thus the configuration space of the manipulation task is finally represented as a hierarchy of strata organized in different levels according to their dimension. Thereby, the strata of a lower level are obtained as substrata of the higher levels by means of the intersection of the strata of these higher levels.

The main difficulty of stratified systems is that different sets of equations of motion are applied on different strata since they represent different contact constraints between the fingers and the object. In addition, the goal configuration of the object cannot be usually reached by remaining in the same stratum. This fact involves that the motion planner based on these stratified systems should take into account transitions between different strata, i.e. finger gaiting. In order to solve this problem of transitions between different strata, the proposed manipulation planner combines all the strata and their transitions on a fictitious system (called bottom stratified extended system) where smooth planning strategies can be applied. This stratified motion planner has been applied to the manipulation of an egg-shaped object with four linear fingers (see Figure 5.3.a). In [Wie-2004], this motion planner is extended for non-smooth objects.

Harmati et al. [Harmati-2002] [Harmati-2004] improve the stratified motion planner proposed by Goodwine in [Goodwine-1999] by developing two new manipulation planning strategies: fitted stratified manipulation planning and fitted semi-stratified manipulation planning. The fitted stratified planner is based on the improvement of the computational performance of the original stratified planner by introducing a fictitious system (fitted system) that reduces the complexity of the algorithm by using nu-

merical computation. The fitted semi-stratified manipulation planning extends the fitted stratified planner with unconstrained finger relocations. Although these planners improved the previous stratified techniques, they present some limitations because the contact points are fixed (neither rolling nor sliding is possible) and the surfaces of the manipulated objects have to be smooth.

Another representation for hybrid control systems used by Xu et al. [Xu-2006] [Xu-2008] in dexterous manipulation is the hybrid automaton. The proposed manipulation planner based on the hybrid automaton divides the entire manipulation task into a sequence of submanipulations. The planning inside each submanipulation is continuous but the transitions between two consecutive submanipulations are discrete. Because of this fact, the hybrid automaton is defined by a collection of continuous state variables, a collection of discrete state variables and the evolution between them. The continuous state variables contain the position/orientation of the object and the positions of the fingers. The discrete state variables contain the type of submanipulation which is executed in each time interval by each finger. There are two types of submanipulations: manipulation mode (a free finger stays still while the other fingers manipulate the object) and substitution mode (a free finger moves on the object in order to substitute another finger whose movement is restricted).

The evolution of the continuous state variables of the hybrid automaton depends on the mode of submanipulation which is performed. The evolution of the discrete states is equivalent to the possible switches between modes of submanipulations along a manipulation task. Therefore, the dexterous manipulation planning problem is restated as the problem of driving the automaton from an initial state (initial object configuration) towards a final state (final object configuration) through a sequence of state transitions which are steered by a switching controller that decides the evolution of the discrete states and applies the appropriate evolution of the continuous states. This manipulation planner based on a hybrid automaton is applied on a planar manipulation task where a circle is rotated by three linear fingers (see Figure 5.3.b). This planner makes several assumptions which simplify the manipulation problem: no rolling and sliding motions of contacts are allowed, the object's boundary is a smooth surface and no collision between the fingers and the object is considered.

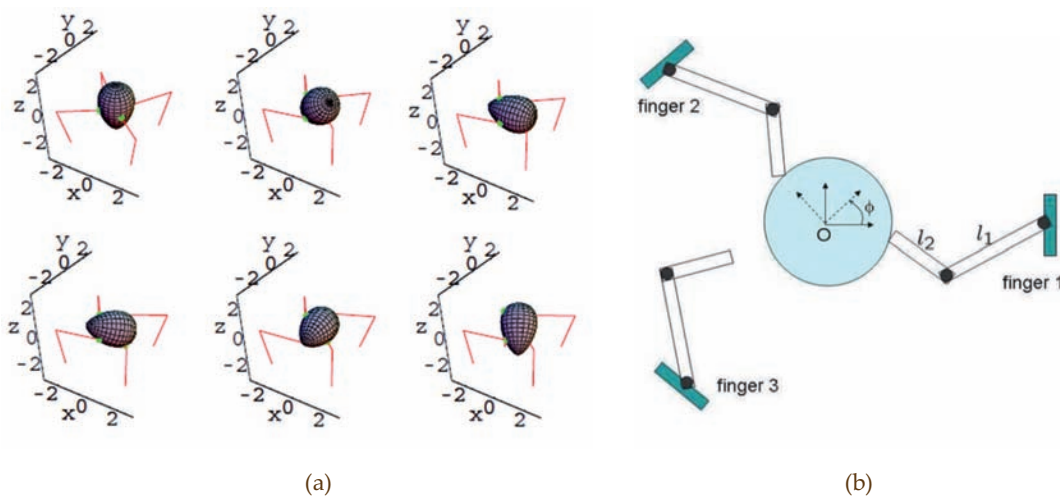


Figure 5.3. (a) Manipulation of an egg-shaped object with four fingers, Source: [Goodwine-1999]; (b) reorientation of a circle by three linear fingers, Source: [Xu-2008].

5.2.3 Manipulation Planning Strategies based on Probabilistic Techniques

Dexterous manipulation planning presents two main features which make it more difficult than other planning problems: the high-dimensionality of the configuration space (which is composed by the configuration space of the fingers, the configuration space of the object and their relative contact configurations) and the presence of non-holonomic constraints (such as the equality between the object and fingers velocities at the contact points). In order to overcome these difficulties, last research on dexterous manipulation planning has been focused on probabilistic techniques. The probabilistic motion planners are usually based on a randomized sampling process of the configuration space in order to find the intermediate positions of the robotic hand for completing the desired trajectory. This sampling process not only reduces the computational cost of other methods which need an exhaustive description of the configuration space but it also guarantees that the probability of not finding a trajectory if one exists approaches to zero when the number of samples is increased (probabilistic completeness). These two characteristics explain the suitability of using sampling-based probabilistic planners for the solution of the dexterous manipulation planning problem.

Although there is a great variety of probabilistic motion planning algorithms, the most commonly used in dexterous manipulation planning is RRT (Rapidly-exploring Random Tree) [LaValle-2001]. RRT is both an efficient data structure and a sampling scheme for quick searches on high-dimensional spaces. This probabilistic technique is based on the interactive construction of a tree whose nodes represent different configurations of the robotic system which are in the free space (i.e. they do not collide with obstacles) and whose edges represent transitions between these nodes which observe the non-holonomic constraints of the planning problem. Yashima et al. [Yashima-2003] [Yashima-2004] are the first researchers to apply a RRT-based planning method for dexterous manipulation. The proposed planner is divided into two levels: global planning level and local planning level. On the one hand, the global planner explores the object configuration space and constructs a RRT in order to arrive to the goal configuration. On the other hand, the local planner checks if the trajectories between the nodes (subgoal configurations) of the tree are feasible. The local planner is executed by the global planner in order to verify if it is possible to create a trajectory between two object configurations by taking into account the contact restrictions. It solves the inverse dynamic problem [Yashima-2002] by applying the Montana's equations [Montana-1988] for all combinations of contact modes (sliding and rolling) between each pair of subgoals of the trajectory. This dexterous manipulation planner has been applied in the reconfiguration of an ellipsoidal object by three planar fingertips (see Figure 5.4.a). The main problem of this solution is its reduced computational efficiency because of the complexity of the inverse dynamic problem. In addition, this planner does not consider finger gaiting or collision detection.

In order to overcome these limitations of the planning method described above, Saut et al. [Saut-2006] [Saut-2007] [Saut-2011] propose a planner which combines different probabilistic techniques. The proposed planner is divided into three phases: manipulation graph construction, path search in the graph and path transformation. The first phase generates a manipulation graph which samples random grasp configurations and connects them by manipulation actions. The second phase uses a graph

search technique to find a path in the manipulation graph which connects the initial and final grasp configurations of the manipulation task. Finally, the third phase simplifies the obtained path and divides it into a collection of feasible manipulation actions.

The goal of the first phase of the planner is to construct a manipulation graph which represents the topological structure of the configuration space of the hand-object system. This graph is composed by nodes which are random samples from the grasp space where all the fingers are in contact with the object and by edges which represent manipulation actions where the fingers can keep contact with the object (transfer paths) or they can break contact with it (regrasping paths). The manipulation graph is built by alternating randomly two steps: generation of nodes (i.e. random grasps) by using the Random Loop Generator (RLG) algorithm [Cortés-2002] and linkage of connected components of the graph by transfer and regrasping paths generated by a RRT-based planner. Once the graph is constructed, the second phase of the planner finds a path between the initial and goal grasp configurations by applying an A* graph search algorithm. Finally, the third phase of the planner simplifies the obtained path because the sampling process of the graph generation tends to create too complex and long paths. In order to solve this problem, the planner chooses randomly pairs of configurations inside the path and tries to link them by a linear path. Figure 5.4.b depicts a manipulation task of a sphere with four fingers implemented by this planner.

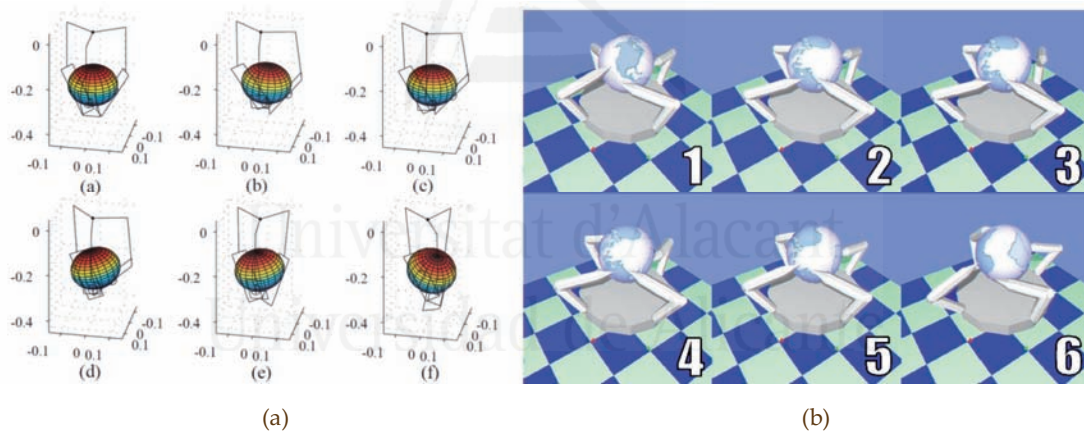


Figure 5.4. (a) Reconfiguration of an ellipsoidal object with three planar fingertips, Source: [Yashima-2003]; (b) reorientation of a sphere with a four-fingered hand including regrasping, Source: [Saut-2006].

Finally, Xu et al. [Xu-2007a] [Xu-2010] have extended their original finger gaits planning approach based on a hybrid controller (see Section 5.2.2) with a RRT planner. They keep the hybrid automaton representation for organizing the evolution equations of the system but they reformulate the planning problem in order to apply a RRT-based strategy. In particular, they define the domains of the discrete and continuous states of the hybrid automaton as the free space and the space of configurations where a task constraint (fingers' workspace, force-closure of the grasp or contact maintenance) is not satisfied is defined as the obstacle space. The finger gaits planning problem can be restated as a standard planning problem whose main goal is to find a path which connects the initial and final configurations through the free space of the system. Thus, a planning strategy based on an RRT can be applied to solve the problem. As in their previous works [Xu-2006] [Xu-2008], the authors have applied the developed manipulation planner in a reorientation task of a circular object by a 3-fingered planar hand.

5.3 Representation of the Surfaces of the Hand and the Object

The previous approaches presented in Section 5.2 solve the dexterous manipulation planning problem but all of them establish some restrictions about the surfaces of the object and the fingers. Some of them (such as [Trinkle-1991], [Zhang-1996] and [Xu-2008]) are only applied on planar objects. Other ones (such as [Yashima-2003], [Yashima-2004] and [Cherif-1999]) can only be applied to fingers and objects with parametric surfaces (such as ellipsoids and spheres) since they use Montana's equations for contact evolution computation. Others use sharp fingertips (such as [Saut-2006] and [Saut-2007]) in order to simplify their shape to a unique contact point.

This chapter proposes a novel dexterous manipulation planner which uses a more general representation of the surfaces of the fingers and the object so that it can be applied to any type of convex shape. In particular, both the surfaces of the object and the fingers are represented as triangle meshes. This surface representation has several important advantages:

- *No duplication of vertices:* This representation is usually based on a list of vertices and a list of faces. The list of vertices registers the 3D coordinates of every different vertex of every triangle. The list of faces registers for each triangle of the mesh the three indexes of its corresponding vertices in the list of vertices. The use of vertex indices avoids repeating the coordinates of vertices which are shared by several adjacent triangles. In addition, this representation also avoids the typical problem of inaccurate closure of convex shapes caused by the float precision of the vertices coordinates.
- *Adjustable level of detail:* The set of triangles which composes the surfaces of the object and the fingers can be changed in order to represent with more or with less detail their surfaces. On one hand, if a more detailed representation is needed, the current triangles can be subdivided in order to obtain a bigger number of smaller triangles which adapt better to the shape of the surface. On the other hand, if a simpler representation is needed, the vertices of the current triangles can be grouped in a smaller number of bigger triangles. This new representation approximates the surface worse, but the computational cost of managing it (not only in memory but also in execution time) is smaller. Thereby, this possibility of adjusting the level of detail of the modeled surfaces can be very useful in the development of manipulation tasks with different requirements. If small and precise movements are required, the level of detail should be increased but the corresponding growth of the computational cost can involve executing the dexterous manipulation planning off-line. If real-time and coarse movements are required, the level of detail should be reduced.
- *Interoperability:* The triangle mesh representation is a *de-facto* standard in computer graphics software. There are many software packages which can import and export different file formats of triangles meshes. In this research, the OBJ file format has been used to store the surfaces of the fingers and the objects. In addition, there are software packages which can apply filters over

this standard file types so that the resulting meshes can be cleaned, simplified or smoothed.

Figure 5.5 shows the triangle mesh representations of the two multi-fingered hands which have been applied in this thesis to develop dexterous manipulation tasks. Figure 5.5.a depicts the triangle mesh which covers the surface of the *Shadow* hand [Shadow-2011] while it is developing a manipulation task of a cube and Figure 5.5.b depicts the triangle mesh which covers the surface of the fingers of the *Barrett* hand [Barrett-2011] while it is grasping a sphere.

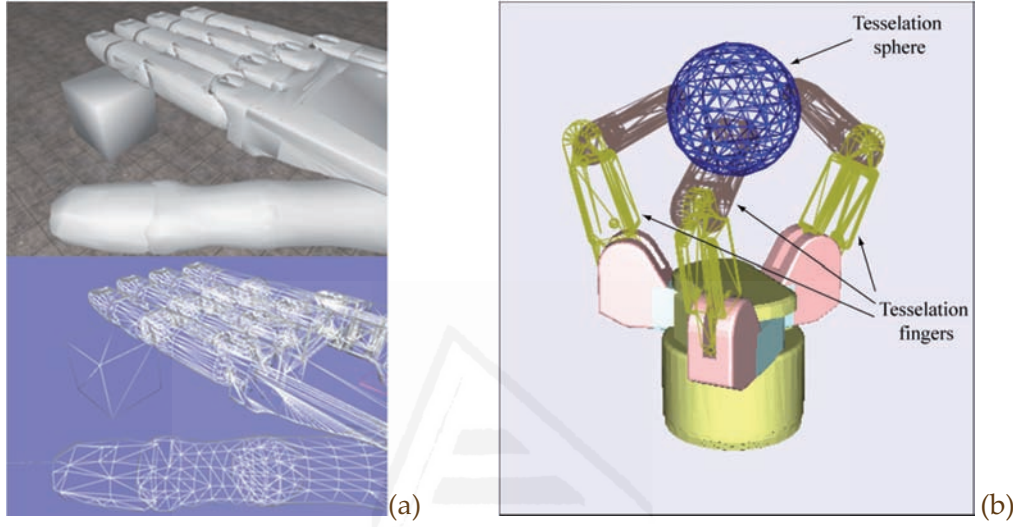


Figure 5.5. Triangle mesh representation of the used multi-fingered robotic hands: (a) *Shadow* hand; (b) *Barrett* hand.

5.4 Geometric Manipulation Planner for In-Hand Manipulation

5.4.1 General Structure of the Geometric Manipulation Planner

The aim of this chapter and one of the main contributions of this thesis is the design and implementation of a planner for the development of dexterous manipulation tasks of objects with multi-fingered hands. This manipulation planner uses the geometric model of the surfaces of the object and the fingers described in Section 5.3 in order to determine the evolution of the contacts between the fingers of the robotic hand and the manipulated object. The planner receives an input a given grasp (i.e. a specific configuration of the object restrained by several contacting fingers) and a desired final configuration of the object (i.e. position and orientation). From these input parameters, the planner has to compute the changes of the fingers joint angles which drive the object from the initial grasp towards the final desired configuration. During this process, the planner should take into account not only the maintenance of the contacts of all the fingers which are touching the object in the initial grasp (i.e. no breaking contacts are allowed) but also the kinematic restrictions of the fingers caused by the limits of their workspaces. The proposed planner has been organized in two main levels (or sub-planners) in order to solve this planning problem: the global planner and the local planner. This organization allows the development of a modular system where any of these components can be changed and improved without affecting the implementation of the other one. Figure 5.6 depicts a diagram with the main steps of these two levels.

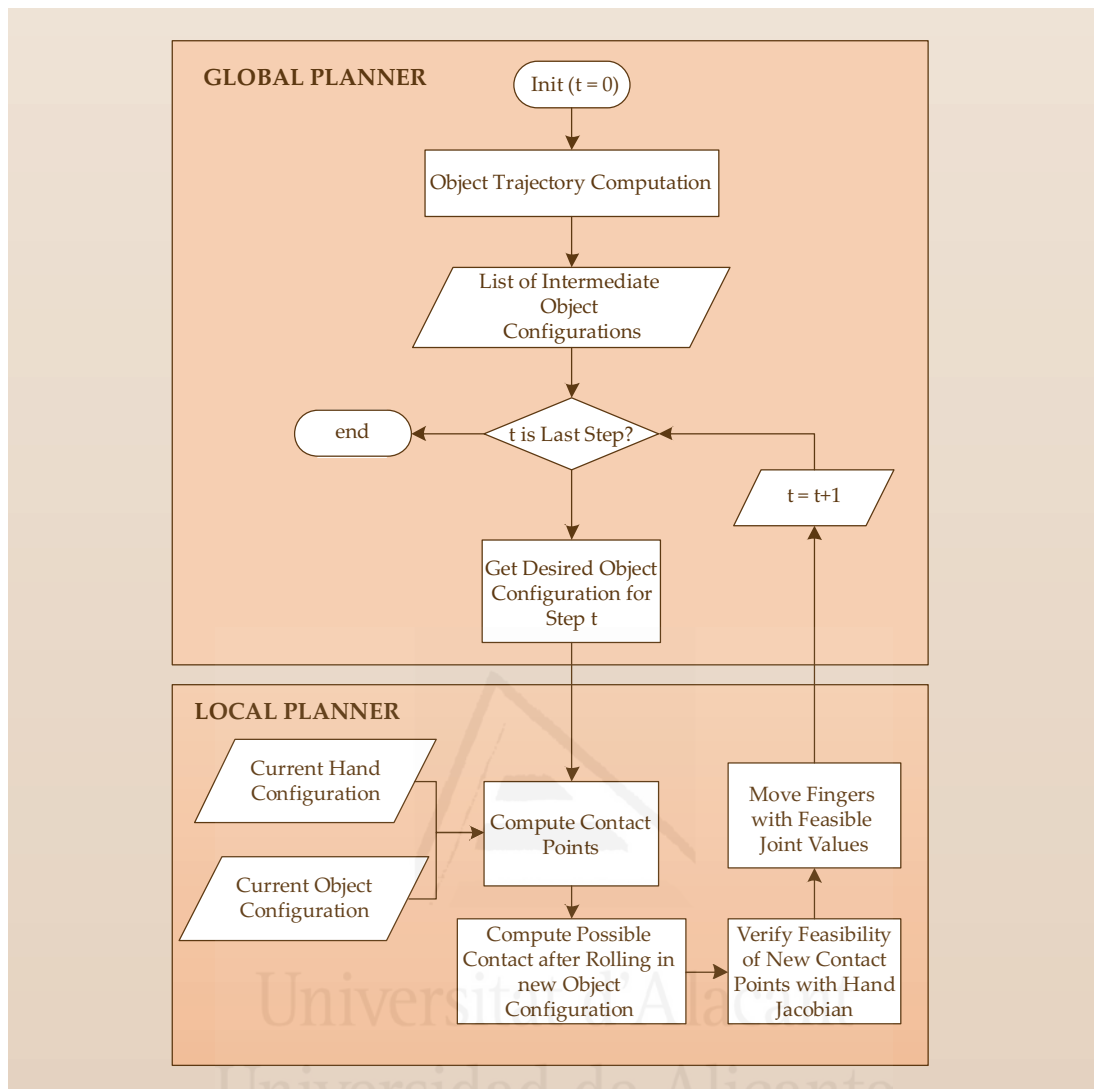


Figure 5.6. Diagram of the general structure of the developed geometric manipulation planner.

The main goal of the global planner is to obtain a group of intermediate object configurations which are able to join the initial state of the object with its final state by applying affine transformations (rotations and translations). Therefore, this group of object configurations represents the trajectory that should be followed by the object during the manipulation task.

Each intermediate object configuration obtained by the global planner in conjunction with the current configuration of the robotic hand (i.e. joint values of its fingers) are passed as input parameters to the local planner. The local planner is responsible for computing the movements of the fingers which have to be applied in order to move the object from the current configuration to the desired intermediate configuration. Firstly, it determines the current contact points between the object and the fingers and the corresponding pair of contacting primitives (i.e. vertices, edges and faces of the triangle meshes of the fingers' and object's surfaces). Secondly, it supposes that the object is in the desired intermediate configuration and it computes all the possible variations of the contact points by considering rolling between the object's and fingers' surfaces. Thirdly, it uses the pseudo-inverse of the Jacobian of each finger in order to verify

which contact configurations are kinematically feasible. Finally, the local planner moves the fingers towards the feasible configuration which involves a smaller joint rate so that the movements of the fingers are minimized. When these movements are executed, the global planner will recover the control of the planner in order to process the next intermediate object configuration.

5.4.2 Global Planner

As mentioned above, the global planner is in charge of the calculation of the trajectory which has to be followed by the manipulated object. The global planner will receive as input the initial and final configurations of the object, represented by the corresponding 4×4 homogenous transformation matrices of the coordinate frame associated with the origin of the triangle mesh of the object. It will also receive the number of steps N_s of the trajectory to be generated. In other words, N_s represents the number of intermediate configurations which have to be computed by the global planner between the initial configuration ${}^wT_{M_O}(0)$ of the object O (i.e. the transformation matrix of the frame M_O of the mesh of the object with respect to the frame of the world W at step 0) and the desired final configuration ${}^wT_{M_O}(N_s)$ at step N_s .

The developed planner implements a linear interpolation method to obtain the intermediate positions. Therefore, the position of the object ${}^w\mathbf{p}_{M_O}(t) = (x_{M_O}, y_{M_O}, z_{M_O})^T$ at each intermediate configuration ${}^wT_{M_O}(t)$ (i.e. the fourth column of the transformation matrix) is calculated by applying the following expression:

$${}^w\mathbf{p}_{M_O}(t) = (1-s) \cdot {}^w\mathbf{p}_{M_O}(0) + s \cdot {}^w\mathbf{p}_{M_O}(N_s) \quad \text{with } s = \frac{t}{N_s} \quad (5.1)$$

Where ${}^w\mathbf{p}_{M_O}(N_s)$ and ${}^w\mathbf{p}_{M_O}(0)$ are the 3D position vectors of the object at the initial and final configurations, respectively. In the case of the orientation of the object, a spherical interpolation method, known as SLERP (Spherical Linear Interpolation) [Shoemake-1985], is applied. Since this algorithm has to be applied to quaternions, the 3×3 rotation matrices integrated in the initial and final transformation matrices (${}^wT_{M_O}(0)$ and ${}^wT_{M_O}(N_s)$) are transformed into two quaternions (${}^w\mathbf{r}_{M_O}(0)$ and ${}^w\mathbf{r}_{M_O}(N_s)$). The SLERP algorithm applies the following equation in order to compute the quaternion ${}^w\mathbf{r}_{M_O}(t)$ corresponding to the intermediate configuration t :

$${}^w\mathbf{r}_{M_O}(t) = {}^w\mathbf{r}_{M_O}(0) \cdot \frac{\sin([1-s] \cdot \theta)}{\sin(\theta)} + {}^w\mathbf{r}_{M_O}(N_s) \cdot \frac{\sin(s \cdot \theta)}{\sin(\theta)} \quad \text{with } s = \frac{t}{N_s} \quad (5.2)$$

Where θ represents the angle between quaternions ${}^w\mathbf{r}_{M_O}(0)$ and ${}^w\mathbf{r}_{M_O}(N_s)$ and can be computed from their dot product:

$$\theta = \arccos({}^w\mathbf{r}_{M_O}(0) \cdot {}^w\mathbf{r}_{M_O}(N_s)) \quad (5.3)$$

The final expression of the 4×4 transformation matrix ${}^wT_{M_O}(t)$ corresponding to the step t of the object trajectory can be generated by combining the previous calculations:

$${}^wT_{M_O}(t) = \begin{bmatrix} {}^w\mathbf{R}_{M_O}(t) & {}^w\mathbf{p}_{M_O}(t) \\ \mathbf{0}_{1 \times 3} & 1 \end{bmatrix} \quad (5.4)$$

Where ${}^w p_{M_o}(t)$ is the column vector of the object position calculated from (5.1) and ${}^w R_{M_o}(t)$ is the 3×3 rotation matrix corresponding to the quaternion ${}^w r_{M_o}(t)$ calculated from (5.2). See [Watt-1992] and [Eberly-2004] for a detailed explanation of the algorithms which have been implemented in order to transform rotation matrices into quaternions and vice versa.

Equations (5.1), (5.2) and (5.4) are repeated for each step $t=0,1,\dots,N_s$ of the object trajectory until a list of intermediate configuration transformation matrices ${}^w T_{M_o}(t)$ is obtained. After having computed this list, the global planner will invoke the local planner with each object configuration of the list, as shown in Figure 5.6.

5.4.3 Local Planner

The local planner can be divided into three main phases: object-fingers contact configuration computation, generation of possible new contact configurations and verification/execution of the feasible finger movements. They are explained in the following sections.

5.4.3.1 Object-Fingers Contact Configuration Computation

First of all, the local planner has to determine the current relative contact configuration between the object and the fingers. Since the proposed dexterous manipulation planner has been implemented in two software simulators of a *Shadow* hand [Anycode-2011] and a *Barrett* hand [Corrales-2010e], it uses their collision detection functions (ODE [Smith-2010] and *JBullet* [JBullet-2011], respectively) to obtain the contact points between the triangle mesh of the object and the triangle mesh of each finger. Next, it determines which pair of faces of the object and finger surfaces correspond to each contact point by looking for the closest triangles in their triangle meshes. The relative contact configuration between each pair of triangles is determined by computing if they have a pair of touching primitives (vertex, edge or face) or if they are interpenetrating (see [Moller-1997] for the detailed algorithm). If there is no interpenetration between both triangles, the touching primitives are used to represent the initial contact configuration. If there is interpenetration (caused by the discrete time steps of the simulation), the local planner removes it by moving away the finger triangle from the object triangle until both triangles only touch on a pair of primitives.

Thereby, after this process, the local planner has an initial contact state for each contact point detected by the simulator. Each contact state j is characterized by a pair $(c_O^j, c_{F_i}^j)$ of primitives of the object O and the finger F_i which are touching, their global coordinates $({}^w p_{c_O^j}(t-1), {}^w p_{c_{F_i}^j}(t-1))$ in the simulation world W and their local coordinates $({}^{M_o} p_{c_O^j}(t-1), {}^{M_{F_i}} p_{c_{F_i}^j}(t-1))$ in the coordinate systems of the corresponding triangle meshes M_o and M_{F_i} at step $t-1$. The global coordinates of the primitives are used to determine the possible relative rolling movements between their surfaces while the local coordinates are used to determine which neighbor primitives will come into contact after these rolling movements.

5.4.3.2 Generation of Possible New Contact Configurations

After having calculated the current contact configuration between the object and the fingers, the local planner determines how each pair of contact primitives might

evolve when the object is situated in the next configuration of the trajectory. This configuration is represented by the transformation matrix ${}^W T_{M_o}(t)$ between frames M_o and W and is computed by the global planner, as explained in Section 5.4.2.

Firstly, the local planner supposes that the object is at the next configuration t of the trajectory and also considers that the relative contact configuration between the object and the finger has not changed. In other words, the contact primitives are the same but their global coordinates have changed. The new coordinates ${}^W p_{c_o^j}(t)$ of each contact primitive c_o^j of the object are obtained by applying the transformation matrix ${}^W T_{M_o}(t)$. The coordinates ${}^W p_{c_{F_i}^j}(t)$ of each contact primitive $c_{F_i}^j$ of finger F_i are obtained by taking into account that its coordinates ${}^{M_o} p_{c_{F_i}^j}(t)$ in the local frame M_o of the object are the same as in the previous step $t-1$ since the relative contact configuration remains equal:

$${}^W p_{c_{F_i}^j}(t) = {}^W T_{M_o}(t) \cdot {}^{M_o} p_{c_{F_i}^j}(t-1) = {}^W T_{M_o}(t) \cdot [{}^W T_{M_o}(t-1)]^{-1} \cdot {}^W p_{c_{F_i}^j}(t-1) \quad (5.5)$$

Therefore, the local planner obtains a new contact state at step t for each previous contact state at step $t-1$. Although these new contact states involve that the contact primitives are the same in steps $t-1$ and t , there could also be a relative movement between the surfaces of the object and the fingers so that new primitives come into contact. For this reason, the local planner also computes these new possible contact primitives provoked by the rolling between the contacting surfaces. This contact evolution has been represented by a contact evolution graph, which will be described in detail in Section 5.5. Thereby, for each current contact state at step $t-1$, a list of possible new contact states at step t is generated.

5.4.3.3 Verification and Execution of Feasible Finger Movements

Although the previous step of the local planner identifies a list of possible contact configurations between the object and the finger, not all of them can be performed since the fingers might not be able to reach the new contact points because of their kinematic limits. Because of this fact, each element of the previous list of possible contact states should be verified by applying the pseudo-inverse of the Jacobian J^+ of the corresponding finger to the displacement Δp of the contacting primitives from the initial contact state to the current possible contact state of the list. However, since the contact primitives may change during each contact point transition due to rolling, the Jacobian should be recomputed at each new contact point. In order to avoid this inefficient procedure, the Jacobian is always computed at a common point (the endpoint of the fingertip) and the displacements Δp are also computed from this point. Thereby, in each contact state, not only the rolling transformations are applied to the contacting primitives but also to the fingertip's endpoint.

The product between J^+ and Δp yields the required increment Δq of the joint values of the finger in order to move the finger to the desired contact position. This product is repeated iteratively (see Figure 5.7) until the position error (difference between the real fingertip position obtained by forward kinematics and the desired one) is below a pre-established threshold or a maximum number of iterations are executed. If during this process, the pseudo-inverse cannot be computed or the obtained joint rates

are too high, the contact point is not reachable by the finger and the corresponding contact state is removed from the list. Among all the resulting contact states which are kinematically feasible, the local planner chooses the contact state which requires a minimum variation of the joint values. Finally, the local planner moves the fingers by applying the computed new joint values and then the global planner takes control of the system in order to consider the next object configuration in the desired object trajectory. This procedure is repeated until the object arrives at the final desired configuration of the trajectory or when the list of possible new contact states is empty because neither of them is kinematically feasible.

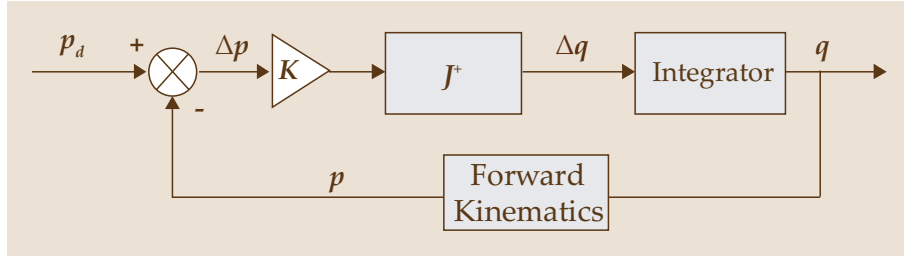


Figure 5.7. Control law of each fingertip to steer it towards the desired position.

5.5 Contact Evolution between Surface Primitives

5.5.1 Contact Evolution Graph

As stated in the previous section, the computation of new contact states generated by the rolling between the surfaces of the object and the fingers is obtained by applying a contact evolution graph, which is shown in Figure 5.8.

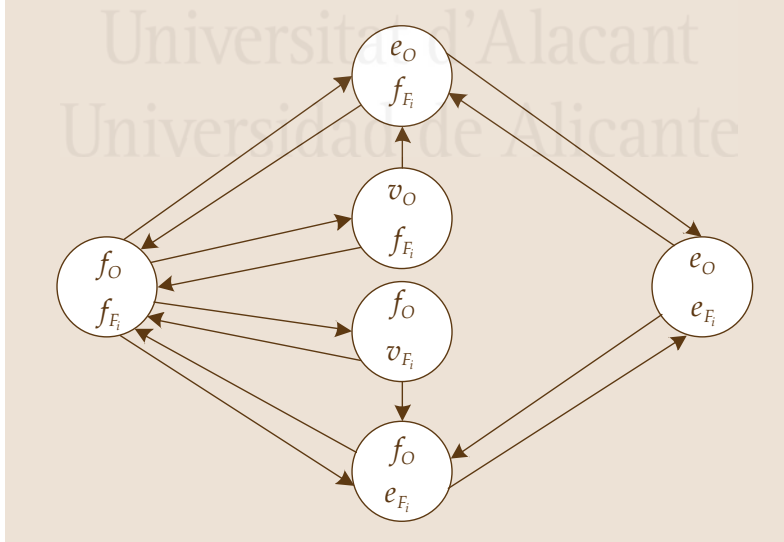


Figure 5.8. Contact evolution graph for new contact states due to rolling.

On one hand, each node of this graph models a contact state between one primitive of the object (f_o : face; e_o : edge; v_o : vertex) and one primitive of one finger (f_{Fi} : face; e_{Fi} : edge; v_{Fi} : vertex). Only non-degenerate pairs [Xiao-2001] of contacting primitives have been considered since contact between vertices and edges are not stable. On the other hand, each edge of the contact graph represents a transition between two consecutive contact states.

The initial contact state in the contact evolution graph at each step t is the contact state computed by the local planner with expression (5.5). This contact state represents the new possible contact configuration if no rolling has taken place. Then, the local planner generates a list of new contact states which could be created because of the rolling or rotation between the primitives of this initial state. This rotation ends when one of the primitives collides with a neighbor primitive of the initial state and thus a new contact state is created. For instance, if the initial contact state is composed by the pair (f_o, v_{F_i}) , the following contact states could be generated by the rotation of the finger surface around the vertex v_{F_i} until an adjacent edge e_{F_i} (i.e. new contact state (f_o, e_{F_i})) or an adjacent face f_{F_i} (i.e. new contact state (f_o, f_{F_i})) touches the object face f_o . The local planner does not only generate one contact transition for each initial contact state, but it generates all the possible contact transitions with every neighbor primitive. In the previous example, the local planner obtains a list of the edges and faces which contain the vertex v_{F_i} , creates all the new possible contact states for all of them and computes the coordinates of their primitives by supposing rolling transitions. The following section describes in detail how these rolling contact transitions are computed by the local planner.

5.5.2 Contact Transitions Generated by Rolling

As described above, the edges of the contact graph which connect two different nodes represent a contact transition (i.e. a change from an initial contact state towards a new final contact state). This transition is generated by the rolling between the two contact primitives of the initial contact state until one of them is substituted by the neighbor primitive of the final contact state. The contact graph contains 14 contact transitions generated by rolling. Only the two most representative transitions of this group of fourteen are described in detail in this section since the other ones are calculated in a similar way. In particular, the edge from state e_o / f_{F_i} to state f_o / f_{F_i} and the edge from state f_o / v_{F_i} to state f_o / f_{F_i} are explained next.

The transition from state e_o / f_{F_i} to state f_o / f_{F_i} involves that one face f_{F_i} (i.e. triangle) of finger F_i is initially touching the object through an edge e_o but the finger face f_{F_i} rotates around this edge until it collides with an adjacent face f_o of the object (see Figure 5.9). The global coordinates of the vertices of face f_o do not change with respect to the initial contact state since it is the finger which rotates around the object surface. Therefore, the local planner has to calculate the new global coordinates of the vertices of the finger face f_{F_i} .

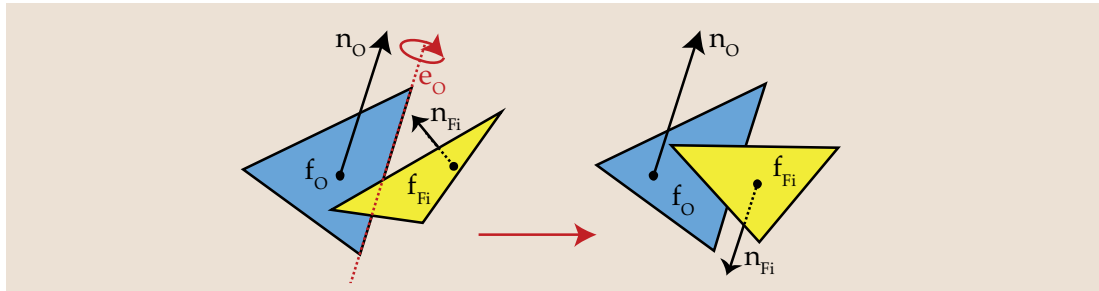


Figure 5.9. Contact transition from contract state e_o / f_{F_i} to contact state f_o / f_{F_i} .

Firstly, the local planner computes the angle α_{F_iO} which should be rotated by face f_{F_i} around edge e_O in order to make f_O and f_{F_i} lay in the same plane. This angle is equal to the supplementary angle between the normal vectors \mathbf{n}_O and \mathbf{n}_{F_i} of both faces since they should have opposite directions:

$$\alpha_{F_iO} = \pi - \arccos(\mathbf{n}_{F_i} \cdot \mathbf{n}_O) \quad (5.6)$$

Next, the local planner has to use the Rodrigues formula in order apply a rotation of α_{F_iO} around the normalized axis $\hat{\mathbf{e}}_O$ to the three vertices of the finger face ($f_{F_i}^{v_0}, f_{F_i}^{v_1}, f_{F_i}^{v_2}$). Nevertheless, since the rotation axis \mathbf{e}_O may not go through the origin, each vertex j of the finger face $f_{F_i}^{v_j}$ should be translated by the first vertex of the axis $\mathbf{e}_O^{v_0}$ in order to guarantee that the rotation is applied around the origin. Afterwards, this translation should be undone in order to place the vertex at its original location. The following expression summarizes this procedure:

$$\begin{aligned} \mathbf{f}_{F_i}^{v_j} = & \left(\mathbf{f}_{F_i}^{v_j} - \mathbf{e}_O^{v_0} \right) \cos(\alpha_{F_iO}) + \left[\hat{\mathbf{e}}_O \times \left(\mathbf{f}_{F_i}^{v_j} - \mathbf{e}_O^{v_0} \right) \right] \sin(\alpha_{F_iO}) \\ & + \hat{\mathbf{e}}_O \left[\hat{\mathbf{e}}_O \cdot \left(\mathbf{f}_{F_i}^{v_j} - \mathbf{e}_O^{v_0} \right) \right] (1 - \cos(\alpha_{F_iO})) + \mathbf{e}_O^{v_0} \end{aligned} \quad (5.7)$$

Finally, the local planner also applies this rotation transformation to the fingertip position in order to calculate its new location and multiply it by the pseudo-inverse of the Jacobian matrix, as described in [Section 5.4.3.3](#).

The transition from state f_O / v_{F_i} to state f_O / f_{F_i} involves that the finger rotates around the vertex v_{F_i} until the adjacent face f_{F_i} collides with the object face f_O (see [Figure 5.10](#)). As in the previous transition, the global coordinates of face f_O do not change but the local planner has to compute the new global coordinates of the vertices of the finger face f_{F_i} .

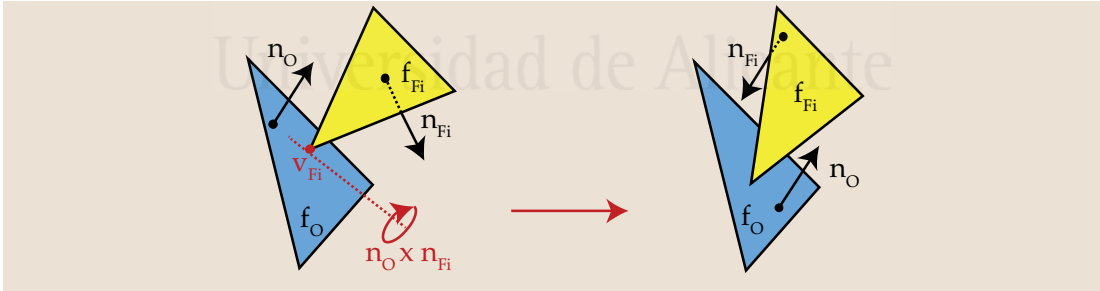


Figure 5.10. Contact transition from contract state f_O / v_{F_i} to contact state f_O / f_{F_i} .

In this case, the axis of rotation $\hat{\mathbf{a}}$ is the cross product between the normal vectors \mathbf{n}_O and \mathbf{n}_{F_i} and the angle of rotation α_{F_iO} is also computed as the supplementary angle between these normal vectors, as in (5.6). Finally, the Rodrigues formula is used but a translation by vertex v_{F_i} is applied in order to perform the rotation at the origin, as shown in the following expression:

$$\begin{aligned} \mathbf{f}_{F_i}^{v_j} = & \left(\mathbf{f}_{F_i}^{v_j} - \mathbf{v}_{F_i} \right) \cos(\alpha_{F_iO}) + \left[\hat{\mathbf{a}} \times \left(\mathbf{f}_{F_i}^{v_j} - \mathbf{v}_{F_i} \right) \right] \sin(\alpha_{F_iO}) \\ & + \hat{\mathbf{a}} \left[\hat{\mathbf{a}} \cdot \left(\mathbf{f}_{F_i}^{v_j} - \mathbf{v}_{F_i} \right) \right] (1 - \cos(\alpha_{F_iO})) + \mathbf{v}_{F_i} \end{aligned} \quad (5.8)$$

5.6 Finger Readjustment based on Tactile Information

The geometric planner presented in [Section 5.4](#) takes into consideration two constraints which are essential for dexterous manipulation: contact maintenance and finger kinematics. The contact maintenance constraint is verified by the contact evolution model described in [Section 5.5](#). This model is based on the continuous contact between adjacent primitives (i.e. vertices, edges and faces) of the triangle meshes representing the surfaces of the fingers and the object to be manipulated. The finger kinematics constraint is verified by the third phase of the local planner (see [Section 5.4.3.3](#)) which applies the pseudo-inverse of the Jacobian matrix of each finger in order to calculate the changes over the finger joint angles which are required to drive each finger to the new contact configuration.

The main limitation of this geometric planner is that it does not consider the contact forces which are applied from the fingers to the manipulated object. Thereby, the geometric planner can be applied for small in-hand manipulation tasks where the initial configuration of the object and the desired final configuration are very near each other. When both configurations are far away from each other, the forces applied by the fingers over the manipulated object tend to decrease after each execution step of the local planner. This reduction of the contact forces makes the grasp unstable and the object finally slips and the contacts between the fingers and the object are lost. In order to avoid this drawback and stop the manipulation task when the grasp can become unstable, the following finger readjustment algorithm is proposed in this thesis. It constitutes one of the contributions of this thesis in dexterous manipulation in conjunction with the geometric planner previously described.

The main goal of this finger readjustment algorithm is to guarantee that each contacting finger always applies a contact force higher than a pre-established safety threshold. The proposed finger readjustment algorithm has been added to the geometric planner presented in [Section 5.4](#). In particular, this algorithm is performed after each execution of the local planner and before giving the control again to the global planner. [Figure 5.11](#) depicts a general scheme of the main steps of this finger readjustment algorithm and its relation with the local and global levels of the geometric planner of [Figure 5.6](#).

First of all, the finger readjustment algorithm has to determine the contact force that the fingers are applying over the object. Generally, this function is implemented in multi-fingered robotic hands by the installation of arrays of tactile sensors which are able to register the pressure (i.e. force per unit area) applied to each cell (i.e. tactel) of these arrays. See [\[García-2009b\]](#) for a detailed review of the different types of tactile sensors applied in robotics. Therefore, the proposed algorithm obtains an array of pressure values for each finger. Next, it looks for the maximum pressure value of each tactile array and thus it determines the maximum pressure which is being applied by the corresponding finger to the object. After performing the required finger readjustments, the algorithm compares the maximum pressure value with a safety threshold. If the pressure is below this safety threshold, the manipulation algorithm stops since the contact between the object and the finger is very weak and further finger movements may make the grasp unstable. Otherwise, the manipulation continues.

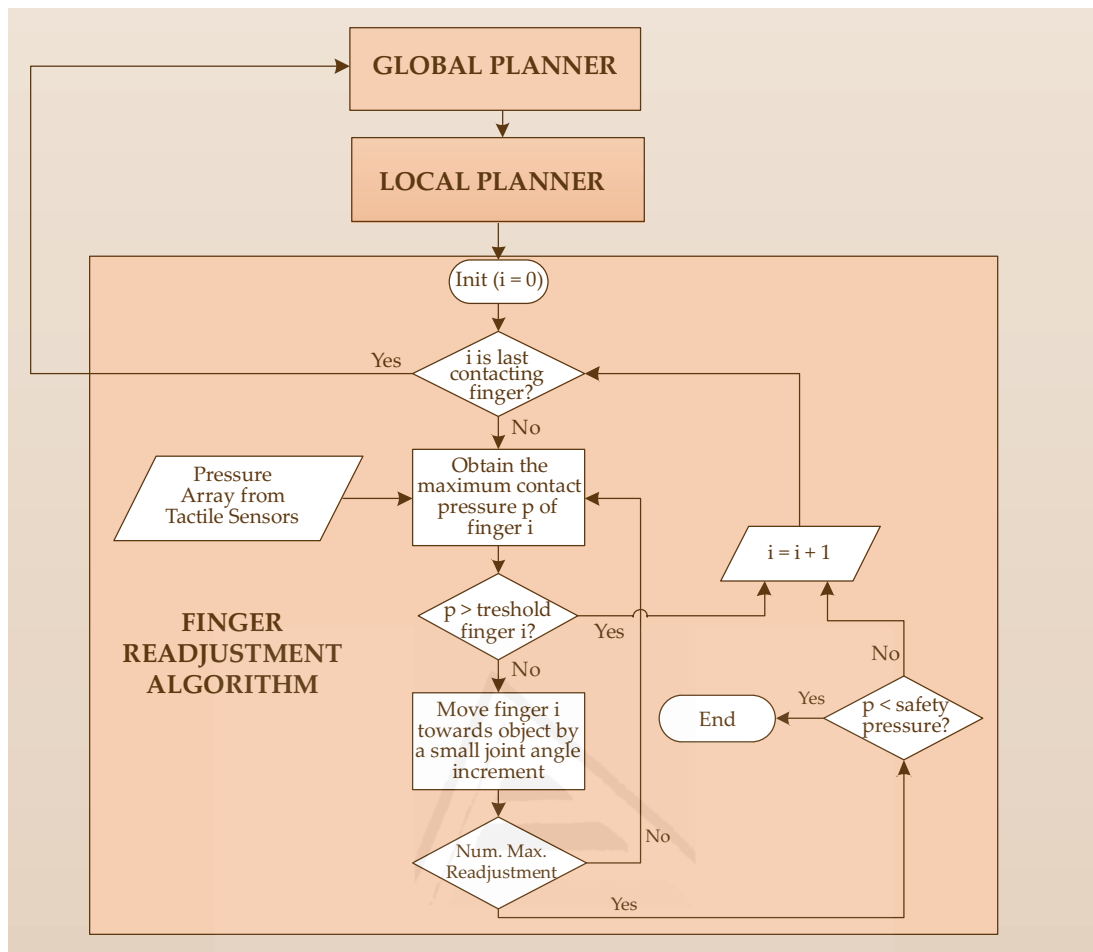


Figure 5.11. Diagram of the general structure of the developed dexterous manipulation planner after including the finger readjustment algorithm between the local and global levels of the geometric planner.

In each execution step of the finger readjustment algorithm, the finger pressure is compared with a readjustment threshold. This threshold is obtained empirically from previous manipulation tasks of the same object and represents an upper threshold for the pressure. Thereby, if the maximum pressure of the finger is above this threshold, no finger readjustment is needed since the applied pressure is sufficient. However, if the maximum pressure of the finger is below this threshold, the joint angles of the fingers are slightly incremented in order to increase the pressure over the object. This readjustment procedure is repeated iteratively until the pressure exceeds the pressure threshold of the finger or until a maximum number of readjustments are executed. Thereby, the algorithm verifies that the pressure of each finger is above the safety threshold and tries to maintain the pressure near the pressure threshold of each finger. When the readjustment process with one finger is finished (i.e. the applied pressure is above the pressure threshold of the finger or the maximum number of readjustments have been performed), the process is repeated for the next finger which is in contact with the object. This algorithm is not applied to the fingers which are not in contact with the object because they do not take part in the manipulation task. The maximum number of readjustments has been added to this algorithm in order to guarantee that the algorithm is not blocked in manipulation configurations where the pressure cannot be increased but it is above the safety pressure threshold.

5.7 Experimental Results

Two different groups of experiments have been developed in order to verify the performance of the dexterous manipulation planner described in this chapter. The first group of experiments uses a software simulator of a *Shadow* hand [Shadow-2011] while the second group of experiments uses a *Barrett* hand [Barrett-2011] with tactile sensors. These experiments are described in detail in the following sections.

5.7.1 In-Hand Manipulation with Shadow Hand

The geometric planner proposed in Section 5.4 has been implemented as a C++ program which communicates with the *Anycode Marilou* robotic simulator [Anycode-2011]. The 3D meshes of the *Shadow* robotic hand and the object to be manipulated are loaded in the simulator. This simulator is in charge of the computation of the contact points between the object and the fingers and the dynamic simulation of the interaction between them. This simulator uses the ODE (*Open Dynamics Engine*) library [Smith-2011] for computing the movements of rigid bodies which are in contact (i.e. collision detection between bodies and rigid body dynamics simulation).

Before applying the developed geometric planner, an initial grasp of the object needs to be configured since it will be the starting point of the planner. In order to do this, a graphic configuration program has been developed. This program allows the user to introduce the values of the joints of the fingers and the values of the position and orientation (heading, attitude and bank angles) of the object. This software is connected to the *Marilou* simulator so that the configurations of the hand and the object can be visualized on real-time while they are changed, as shown in Figure 5.12.

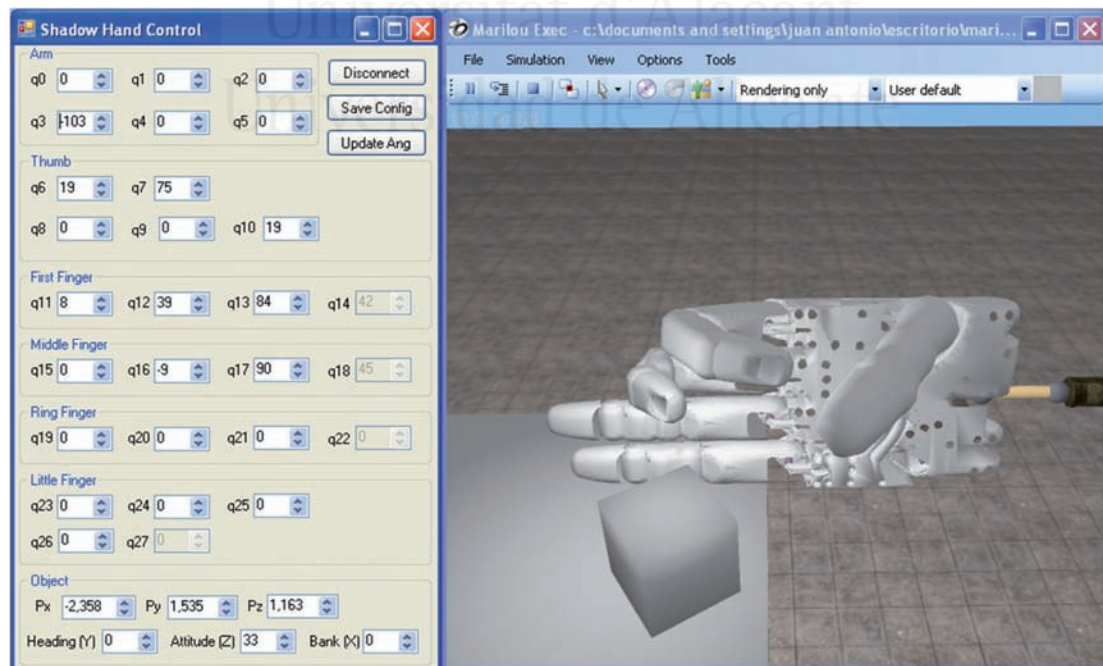


Figure 5.12. Simulator software of the *Shadow* hand with a window for updating joint values (left) and a panel with a 3D visualization of the hand and the object loaded in the environment (right).

After the initial grasp has been configured with the previous software, the C++ program of the geometric planner is executed. First of all, this program connects to the *Marilou* simulator in order to obtain all the vertices and faces of the meshes of the fingertips and the object. Next, the program asks the user for the changes of the object's position/orientation which are desired to be performed by the hand and the number of steps of the trajectory. The software applies the interpolation strategy of the global planner described in [Section 5.4.2](#) in order to obtain the desired trajectory of the object. Next, the local planner described in [Section 5.4.3](#) is executed for each step of the trajectory. In each step, the software gets the current object position/orientation and the current joint angles of the fingers from *Marilou*. Next, it updates the coordinates of the vertices of the meshes of the objects and fingertips. Then, the program asks *Marilou* for the contact points and computes the faces which are contacting. After all the pairs of contacting faces are registered, the algorithm uses the contact evolution graph of [Section 5.5](#) in order to obtain the next contact states and the corresponding fingertip positions. Finally, the program combines all the possible contact states and applies the pseudo-inverse of the Jacobian in order to compute the required joint values of the fingers. The program will select the joint values between all the contact configuration combinations which involve fewer changes with respect to the initial joint values and thus the joint rate is minimized. These joint values are sent to the *Marilou* simulator in order to move the fingers of the hand and the program changes to the next step of the trajectory where the local planner is executed again.

[Figure 5.13](#) depicts an example of three different trajectories where the planner has been applied to move a cube grasped by a *Shadow* robotic hand. The first trajectory ([Figure 5.13.a](#)) involves a displacement of the object by 20mm in the X axis (towards the palm). This trajectory can also be watched on [Video 5.1](#) in the attached DVD. The second trajectory ([Figure 5.13.b](#)) translates the object in the Y direction (towards the thumb) by 15mm. Finally, the third trajectory ([Figure 5.13.c](#)) contains a displacement of the object in the Z direction (upwards) by 20 mm.

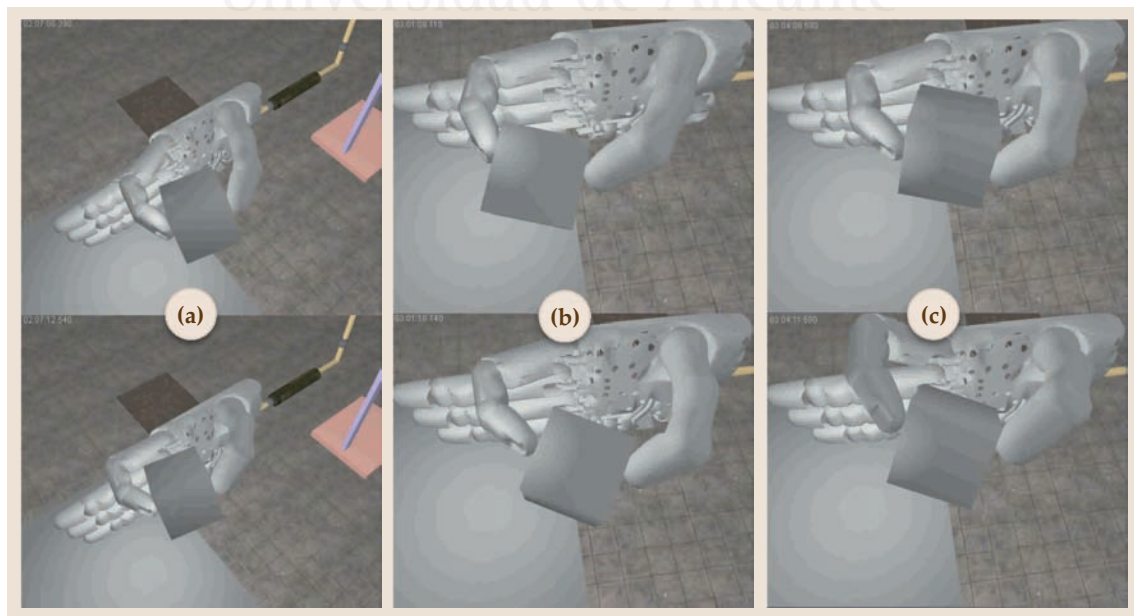


Figure 5.13. Simulation of translation manipulation tasks of a cube: (a) linear trajectory in the X axis; (b) linear trajectory in the Y axis; (c) linear trajectory in the Z axis. First frame is above and last frame is below.

Figure 5.14 depicts an example of two different trajectories where the planner has been applied to move two different objects grasped by a *Shadow* robotic hand. The first trajectory (Figure 5.14.a) rotates a hexagonal prism by 5° around the Y axis. The second trajectory (Figure 5.14.b) contains a rotation of an icosphere (i.e. a sphere generated by subdividing recursively each triangular face of an icosahedron into four triangles) around the Z axis by 5° . The trajectory of the hexagonal prism can be watched on Video 5.2 and the trajectory of the icosphere can be watched on Video 5.3, both can be found in the DVD attached with this thesis.

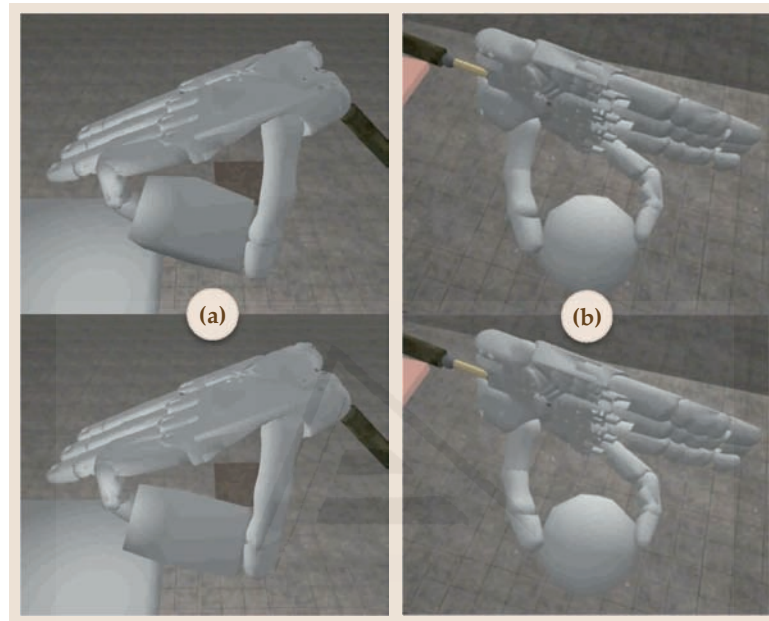


Figure 5.14. Simulation of rotation manipulation tasks: (a) rotation of a hexagonal prism around the Y axis; (b) rotation of an icosphere round the Z axis. First frame is above and last frame is below.

The last manipulation experiment has been repeated with different levels of detail in the definition of the icosphere in order to verify the evolution of the execution time with regard to the number of faces and vertices of the surface object. Table 5.1 shows the total execution time of the local planner in an Intel Core 2 Duo T7500 (2.2GHz) computer with 2GB of RAM. This table shows that the time execution of the local planner is practically identical with icospheres of up to 320 faces. Since most of the 3D models of the objects to be manipulated have a number of faces inferior to this value, the time execution of the local planner is limited. Nevertheless, if a complex object with a detailed mesh has to be manipulated, the geometric planner should be executed off-line and the generated trajectories should be applied afterwards.

Table 5.1. Evolution of execution time of the local planner depending on the complexity of the mesh.

Manipulated Object	Vertices	Faces	Time (s)
Icosphere - 1 subdivision	12	20	0.506
Icosphere - 2 subdivisions	42	80	0.534
Icosphere - 3 subdivisions	162	320	0.588
Icosphere - 4 subdivisions	642	1280	0.649
Icosphere - 5 subdivisions	2562	5120	1.907

The previous experiments show that the developed planner is able to complete the object trajectory indicated by the user by changing the configuration of the fingers' joints accordingly. Although this algorithm takes into consideration the geometry of the fingers and object surfaces and the kinematic limitations of the fingers, it does not take into account the contact forces which are applied by the fingers to the object. In fact, the experiments of this section have been limited to small displacements and rotations of the object because long trajectories usually make the object slip along the surface of the fingers and loose contact, making the grasp unstable. This problem is solved by extending the geometric planner with the finger readjustment algorithm presented in [Section 5.6](#). The next section describes several real experiments where this extension of the dexterous manipulation planner has been tested.

5.7.2 In-Hand Manipulation with Barrett Hand

In order to verify the performance of the finger readjustment algorithm of [Section 5.6](#), a real experiment setup composed by a three-fingered *Barrett* hand and 7 *PPS RoboTouch* tactile sensors (2 tactile sensors for each finger and 1 tactile sensor for the palm) have been employed (see [Figure 5.15.a](#)). The outer phalanxes of the fingers of the hand are covered with tactile sensors composed by an array of 22 tactels with an area of 36 mm² per tactel, the inner phalanxes are covered with tactile sensors composed by an array of 24 tactels with an area of 25 mm² per tactel and the palm is covered with a tactile sensor composed by an array of 24 tactels with an area of 100 mm² per tactel. All these tactels can register pressure values in the range 0-140 KPa with a frequency of 30 Hz and a sensitivity of 0.7 KPa. [Figure 5.15.b](#) shows a 3D representation of the pressure measurements registered by these sensors during a manipulation task.

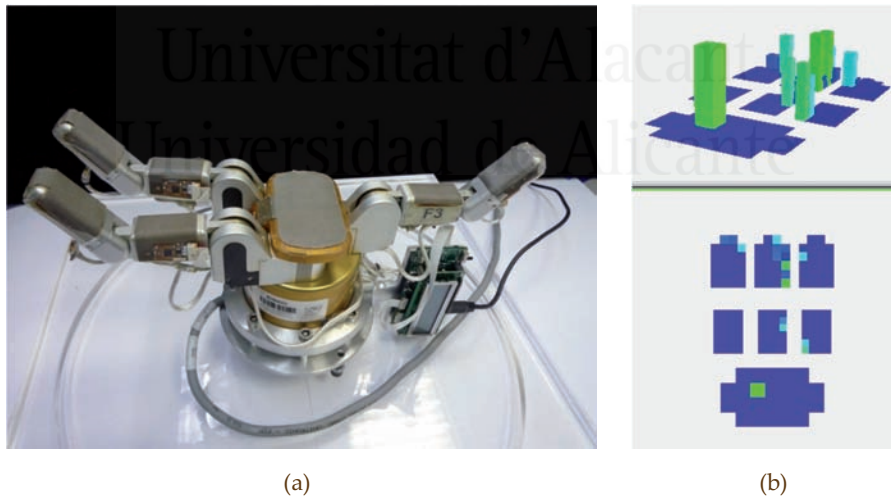


Figure 5.15. Experimental setup for dexterous manipulation planning: (a) *Barrett* hand with 7 tactile sensors; (b) 3D representation of the pressure measurements registered by the arrays of tactile sensors.

First of all, the geometric part of the dexterous manipulation planner must be executed in order to determine the movements of the fingers which are required to move the object from its initial configuration towards the final desired configuration. In order to do this, the geometric planner described in [Section 5.4](#) has been implemented in a software simulator of the *Barrett* hand. This software simulator is based on the platform *Easy Java Simulations* [[Esquembre-2004](#)] and contains the kinematic and dynamic models of the hand. In addition, collision detection and dynamic simulation functions

of the library *JBullet* [JBullet-2011] have been used to compute the contact points between the fingers and the object and to calculate the dynamic interaction between them, respectively. A description of this software simulator and the kinematic and dynamic models of the Barrett hand can be found in [Corrales-2010e] and Appendix A.

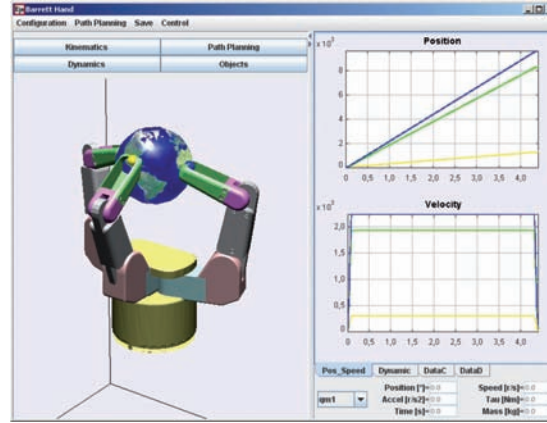


Figure 5.16. Software simulator of the Barrett hand where the geometric planner has been implemented.

This software simulator returns a list of finger joint angle increments for each step of the local planner which are executed by a C++ program. This program contains the control library of the *Barrett* hand and the library of the tactile sensors which registers the pressure measurements from all the tactile arrays. In this program, the finger readjustment algorithm of Section 5.6 has also been implemented. After the execution of the finger movements indicated by the local level of the geometric planner, the finger readjustment algorithm takes control, registers the pressure measurements from the tactile sensors and determines if it is necessary to close the fingers over the object in order to increase the applied pressure by following the steps described in Figure 5.11. If the applied pressure by one finger is too weak (i.e. it is below the safety pressure threshold), the manipulation task is definitely stopped due to the risk of grasp instability. Otherwise, the following intermediate configuration of the object computed by the global level of the geometric planner is considered.

Several real manipulation experiments have been developed in order to verify the performance of the proposed dexterous manipulation planner. The first object to be manipulated is a cylinder (i.e. a can). The cylinder is initially over a table and the *Barrett* hand is installed at the end effector of a *Mitsubishi PA-10* robotic manipulator (see Figure 5.17.a). Firstly, the robotic hand is closed automatically in order to grasp the cylinder from $t = 0$ s to $t = 10$ s, as shown in Figure 5.18.a, where the motor counts of the three fingers rise during this period. After this closing process of the fingers (see Figure 5.17.b), they apply different pressure values to the object: 12 KPa in the case of finger 1, 6 KPa in the case of finger 2 and 39 KPa in the case of finger 3 (as shown at $t = 10$ s in Figure 5.18.b). Next, the robotic manipulator moves the hand and the object up in order to separate it from the table from $t = 10$ s to $t = 18$ s and the dexterous manipulation planning begins to be executed from this initial grasp (see Figure 5.17.c). The local level of the planner executes consecutive increments of the joint angles which can be shown as linear slopes in Figure 5.18.a from $t = 18$ s to $t = 46$ s (see Figure 5.17.d-Figure 5.17.f). After each of these increments, the finger readjustment algorithm is executed and tries to keep the contact pressure of each finger above the next readjustment thresholds: 10KPa

for fingers 1 and 2 and 20 KPa for finger 3. For instance, several readjustments of fingers 1 and 2 are executed between $t = 18$ s and $t = 26$ s. They can be identified in [Figure 5.18.a](#) by horizontal lines where the motor counts do not change (i.e. the fingers push the object but their joint angles do not change since they are blocked by the object) and in [Figure 5.18.b](#), they can be identified by downward peaks of pressure which go below the pressure value of 10 KPa and subsequent upward peaks of pressure which go above this pressure value. The correspondence between the evolution of these pressure values and the movements of the fingers can be verified in the [Video 5.4](#) contained in the DVD attached to this thesis. This video also depicts the plot of the motor counts of the fingers and the 3D representation of the pressure arrays of the tactile sensors. After executing some readjustments of fingers 1 and 2 between $t = 18$ s and $t = 26$ s, the planner executes several increments of the finger joint angles from $t = 26$ s to $t = 34$ s without finger readjustments since all the pressure values are above the thresholds. At $t = 34$ s, a new readjustment of finger 3 is executed since its pressure is below 20 KPa. This readjustment is executed again after each increment of the local planner but the pressure of this finger decreases unyieldingly until it is below the safety pressure threshold (10 KPa for finger 3 and 5.0 KPa for fingers 1 and 2) and the manipulation task is definitively stopped. This safety requirement is identified in [Figure 5.17.f](#) because finger 3 is touching the object near the end of its fingertip and the grasp is near instability.

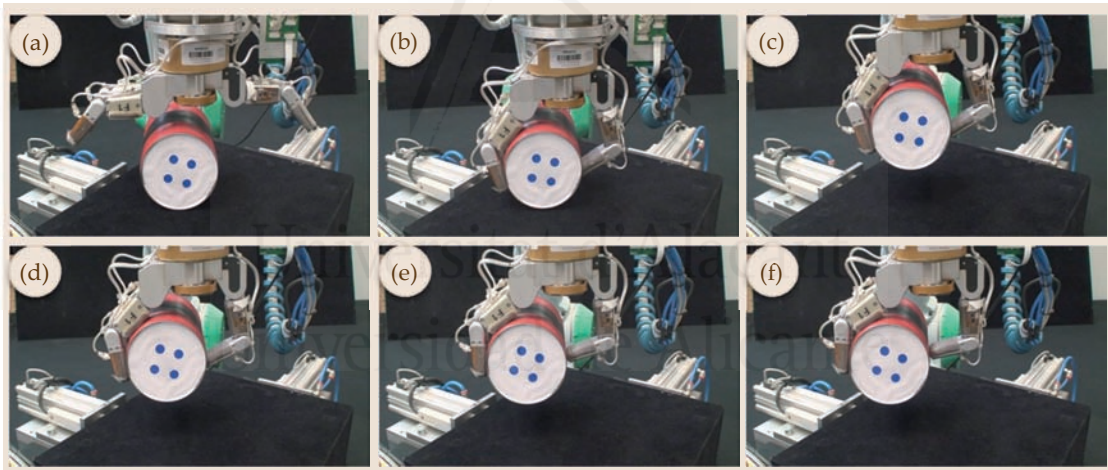


Figure 5.17. Sequence of frames of a manipulation task of a cylinder by applying the geometric planner with finger readjustments.

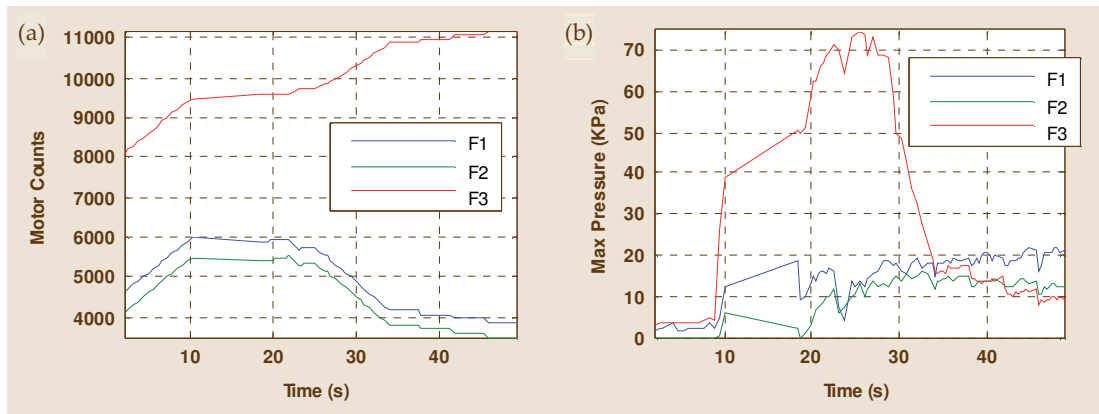


Figure 5.18. Evolution of the manipulation task of a cylinder: (a) Evolution of the motor counts of each finger; (b) evolution of the maximum contact pressure registered by the tactile sensors for each finger.

Two additional manipulation experiments have been performed with a sphere (i.e. a plastic ball). In both experiments the readjustment pressure thresholds are established to the following values: 15 KPa for fingers 1 and 2 and 30 KPa for finger 3). The safety pressure thresholds are established to the following values: 10 KPa for fingers 1 and 2 and 20 KPa for finger 3. Figure 5.19 depicts three frames of the linear trajectory of the object which is executed during the first manipulation task. Figure 5.20.a shows the evolution of the motor counts of the fingers (which are proportional to their joint angles) and Figure 5.20.b shows the evolution of the maximum pressure values registered by the tactile sensors for each finger. First of all, the fingers are closed in order to grasp the sphere. This phase can be identified by the slope in Figure 5.20.a between $t = 0$ s and $t = 21$ s.

Afterwards, the dexterous manipulation planner is executed from $t = 21$ s to $t = 41$ s. During this execution, after many of the finger joint angle increments performed by the local planner (which are identified by small slopes in Figure 5.20.a), readjustments of finger 2 are required since the maximum pressure applied by this finger is below 15 KPa. These finger readjustments can be identified by horizontal lines after a slope in Figure 5.20.a. These readjustments are sufficient to keep the pressure of finger 2 above the safety threshold from $t = 21$ s to $t = 37$ s. Nevertheless, at $t = 37$ s the pressure applied by finger 2 goes below the safety threshold (10 KPa) and after executing the maximum number of finger readjustments (this can be identified in Figure 5.20.a by the longest horizontal line at the end of the plot), the maximum contact pressure of finger 2 continues to be below this threshold. Thereby, the dexterous manipulation task is stopped. Further movements of this finger will provoke a loss of contact and the grasp will become unstable. The experiment can be watched on Video 5.5 of the attached DVD.

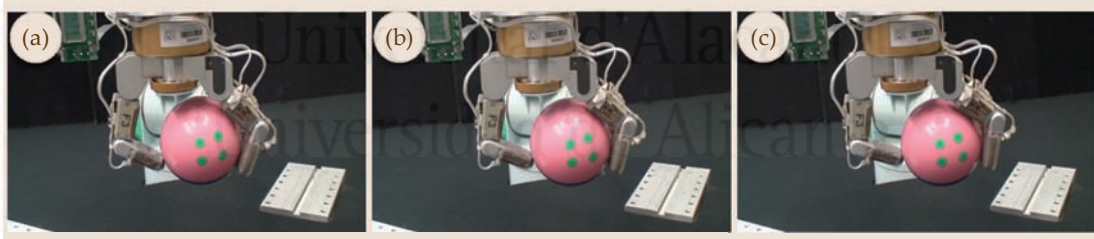


Figure 5.19. Sequence of frames of a manipulation task 1 of a sphere by applying the geometric planner with finger readjustments.

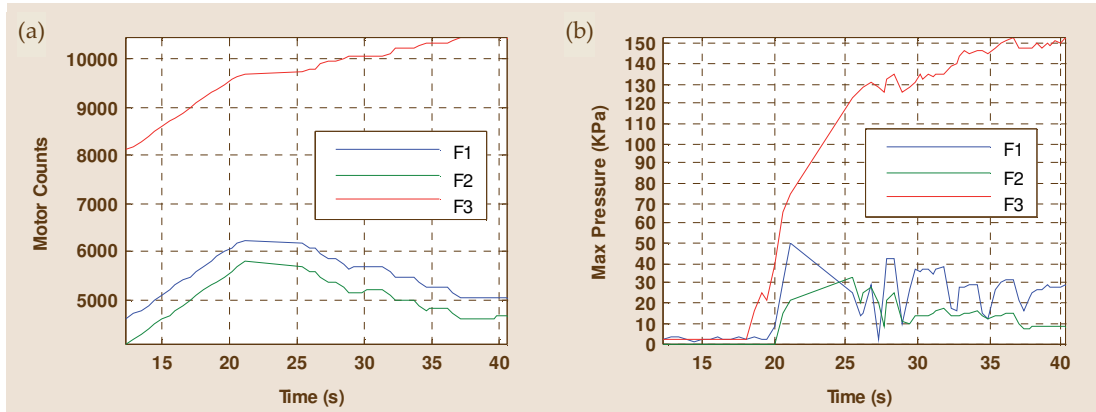


Figure 5.20. Evolution of the manipulation task 1 of a sphere: (a) Evolution of the motor counts of each finger; (b) evolution of the maximum contact pressure registered by the tactile sensors for each finger.

The second manipulation task of the sphere has the same pressure thresholds, but the final condition of the task is different. Whereas the first task is finished due to a weak pressure value of finger 2, the second task is finished because the fingers drive the object to the desired configuration of the object, established by the global planner. Figure 5.21 shows three configurations of this object trajectory: Figure 5.21.a is the initial configuration, Figure 5.21.b is an intermediate configuration and Figure 5.21.c is the final configuration. Firstly, as in the previous task, the fingers are closed in order to form the initial grasp of Figure 5.21.a. This can be identified in Figure 5.22.a by a slope in the evolution of the finger motor counts between $t = 0$ s and $t = 28$ s. Then, the dexterous manipulation planner is executed. Between $t = 28$ s and $t = 50$ s, several readjustments of fingers 1 and 2 are executed since their maximum contact pressure values are below 15 KPa. Between $t = 50$ s and $t = 73$ s several readjustments of finger 3 are executed since its contact pressure is below 30 KPa. These readjustments are able to keep the pressure of finger 3 near the readjustment threshold of 30KPa and above the safety threshold of 20 KPa. Thereby, the manipulation planner is able to complete the trajectory of the object without any risk of grasp instability. This manipulation task and the corresponding plots of motor counts, maximum pressure values and arrays of tactile measurements can be watched on Video 5.6 included in the DVD attached to this thesis.

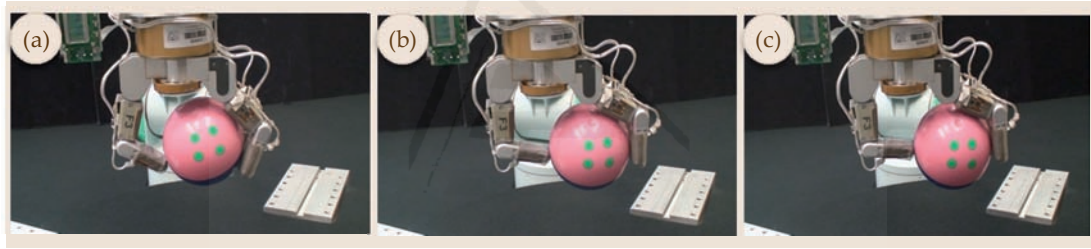


Figure 5.21. Sequence of frames of the manipulation task 2 of a sphere by applying the geometric planner with finger readjustments.

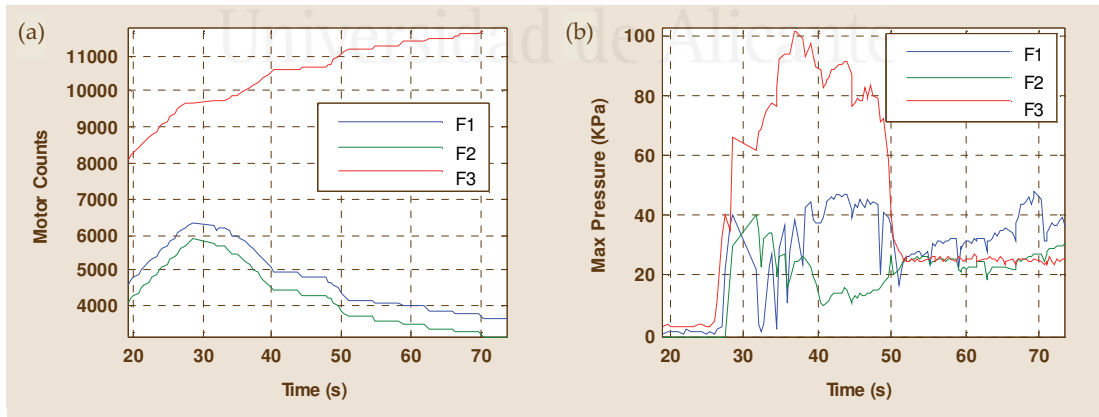


Figure 5.22. Evolution of the manipulation task 2 of a sphere: (a) Evolution of the motor counts of each finger; (b) evolution of the maximum contact pressure registered by the tactile sensors for each finger.

All these experiments have demonstrated that the execution of the finger readjustment algorithm after the local level of the geometric planner has achieved its two main goals. On one hand, the pressure exerted by the fingers is always above a safety threshold or the manipulation task is stopped to avoid grasp instability. On the other hand, this algorithm tries to keep this pressure near a readjustment threshold to avoid losing contact and thus manipulation tasks can perform longer object trajectories.

5.8 Conclusions

This chapter describes a novel planner for dexterous manipulation of objects by a multi-fingered hand. It is one of the main contributions of this thesis since it has several innovative features. Firstly, this planner uses a representation of the surfaces of the fingers and the object to be manipulated based on triangle meshes. This representation involves an important improvement in comparison with previous manipulation planners which need a parametric representation of these surfaces or a planar simplification of them.

Secondly, the developed planner is organized into two levels: a global planner and a local planner. The global planner computes the trajectory between the initial and final configurations of the object by interpolating not only the position but also the orientation of the object between both configurations. This list of intermediate configurations is used as input for the other level of the planner, the local planner. This local planner computes the evolution of the fingers' joints by taking into account their kinematics and the contact maintenance. The contact maintenance constraint is verified by the definition of a contact evolution model which represents how the contacting primitives (i.e. vertices, edges or faces of the triangle meshes) of the object and the fingers may change due to rolling between both surfaces. This contact model is another contribution of this thesis. After obtaining all the possible contact evolutions caused by the rolling between the surfaces, the local planner applies the pseudo-inverse of the Jacobian matrix of each finger in order to determine which changes over the finger joint angles are required to perform these contact changes. Thereby, all the contact changes which are not feasible by the fingers due to their kinematic restrictions are rejected.

Finally, this geometric planner has been extended with a finger readjustment algorithm which is executed after each step of the local planner and before the global planner. This geometric planner includes the management of the contact pressure which is exerted by each finger to the object in order to guarantee that the contact between the fingers and the object does not break. In order to do this, this algorithm defines two pressure thresholds for each finger: a readjustment threshold and a safety threshold. These thresholds are generated empirically by learning their values from previous manipulation tasks of the objects. If after executing a finger movement indicated by the local planner, the contact pressure of one of the fingers goes below the readjustment threshold; several readjustments of this finger are executed. These readjustments consist on moving gradually this finger towards the surface of the object. Thereby, the contact pressure is increased. These readjustments are stopped when the readjustment threshold is exceeded or after a maximum number of readjustments. In this last case, the finger readjustment algorithm verifies if the final pressure value is above the safety threshold. If it is above this threshold, the readjustment algorithm verifies the contact pressure of the following contacting finger. Otherwise, the manipulation task is stopped because the readjustment algorithm has not been able to increase the contact pressure of this finger and its contact may become unstable if further movements are performed. The geometric planner has been tested over a simulation of a *Shadow* hand and this planner together with the readjustment algorithm has been tested on several real manipulation tasks performed by a *Barrett* hand.

Conclusions

6 Conclusions

This last chapter of this thesis draws the most important conclusions from the research work developed along this thesis. In particular, this chapter analyzes the main contributions of this thesis and relates them with the main goal of this thesis which involves the development of human-robot interaction tasks. In addition, it enumerates all the scientific publications which have been written in the framework of this thesis. They are organized in different types: JCR journal papers, book chapters, non-JCR journal papers, international and

6.1 Conclusions	137
6.2 Publications	141
6.2.1 JCR Journal Papers.....	142
6.2.2 Book Chapters and non-JCR Journal Papers	144
6.2.3 International Conference Papers	145
6.2.4 National Conference Papers	148
6.3 Future Work	149

national conference papers. Finally, this chapter also proposes additional work which could be done in the future in order to continue the research lines of this thesis and thus, extend the possibilities of human-robot interaction.

6.1 Conclusions

As stated in [Chapter 1](#), the main goal of this thesis is to fulfill several requirements which are necessary for the development of human-robot interaction tasks where human operators collaborate with robotic manipulators in an unobtrusive and natural way. In fact, two main requirements have been accomplished in this thesis: the guarantee of the human safety during the execution of human-robot interaction tasks and the development of a manipulation planner which enables dexterous manipulation tasks performed by multi-fingered robotic hands installed on robotic manipulators that collaborate with humans.

The first requirement of human safety is implemented in this thesis in an innovative way in comparison with traditional safety solutions. Traditional safety systems get safety by completely isolating the workspace of the robot from the workspace of the human. Thereby, there is no real collaboration between them and only a transfer of products between them with conveyor belts is performed. This thesis goes in the direction marked by new human-robot safety standards which are trying to make the human and the robot share the same workspace. Nevertheless, this thesis goes one step further and proposes a close interaction where human and robots work over the same product simultaneously and/or interchange parts of this product. Obviously, in order to guarantee the safety of the human operator in this kind of tasks, it is necessary to track precisely the complete body of the human operator in order to determine if there is any possibility of collision with the robot. This thesis proposes in [Chapter 2](#) the use of motion capture systems in order to perform this full-body human tracking. Since motion capture systems are usually applied on computer animation, biomechanics and teleoperation, this thesis has developed a new comparative analysis of motion capture

technologies for human-robot interaction in industrial environments. This analysis comes to the conclusion that motion capture systems based on inertial sensors are a suitable solution because they are easy to install, real-time, have high rotational accuracy and do not suffer from occlusions. Nevertheless, their main disadvantage is the lack of direct global position measurements. A deep analysis of the inertial motion capture system employed in this thesis has verified this drawback and involves the use of an additional localization system (based on UWB pulses) which corrects the error accumulated by the global positions estimated from the inertial system.

The analysis of these two tracking systems shows that they have complementary features: the inertial motion capture system has accurate relative rotation measurements with a high sampling rate while the UWB localization system has accurate global position measurements with a low sampling rate. Previous sensor fusion algorithms do not take into account this complementariness and they consider both tracking systems as measurements from different sensors which are applied similarly in the fusion algorithm. For instance, Bayesian filtering techniques (e.g. Kalman filter and particle filters) use multiple independent correction channels or a direct combination of measurements which is applied over an only correction channel. Therefore, these solutions require that both steps (i.e. prediction and correction) of the filter are executed when one measurement is registered. Nevertheless, the fusion algorithms proposed in [Chapter 3](#) of this thesis only execute the prediction step of the filter when a measurement from the inertial motion capture system is registered and the correction step when a measurement from the UWB localization system is obtained. This procedure not only reduces the computational cost of the fusion algorithm (which is very important for real-time quick human tracking), but it also avoids implementing theoretical motion models in the prediction step which are based on restrictive assumptions (such as movements with constant velocities or constant accelerations) since it uses directly sensor measurements.

In order to correct the inertial position error through the UWB measurements without using consecutive executions of prediction and correction steps, the proposed algorithms are based on the recalculation of the transformation matrix between both tracking systems each time a new measurement of the UWB system is received. Thereby, all the measurements from the inertial motion capture system which are registered after a UWB measurement are corrected by applying to them the transformation matrix recalculated with this UWB measurement. In addition, the development of three different fusion algorithms (based on a Kalman filter, a particle filter and a combination of a Kalman filter and a particle filter) permits to apply the algorithm whose features better fit the requirements of the developed task. For instance, the Kalman filter fusion algorithm is ideal for complex tasks which need a very efficient fusion algorithm and the Kalman-particle filter fusion algorithm is ideal for precise tasks which need an accurate localization of the human operator. The Kalman filter algorithm has been successfully tested on several real human-robot interaction tasks where a real-time efficient tracking of the human operator is essential.

The rotational measurements from the inertial motion capture system and the global position estimates calculated by the fusion algorithm are applied on a skeleton which represents the structure of the body of the human operator who collaborates

with robotic manipulators. Nevertheless, this skeletal representation only considers the lengths of the limbs of the human body but it does not take into account their other dimensions which are very important in order to obtain precise human-robot distance values. This thesis proposes to cover each limb of the human body and each link of the robotic manipulator with a dynamic Swept-Sphere Line (SSL) bounding volume (i.e. capsules composed by a cylinder with two spherical caps). These bounding volumes adjust better to the shape of human limbs and robotic links than spheres. In fact, this representation needs less bounding volumes to cover their bodies than previous research works based on spheres. This reduction in the number of bounding volumes involves a reduction of the computational cost of the human-robot distance calculation since the number of distance tests between pairs of bounding volumes is reduced. Furthermore, the radii of these SSLs are increased proportionally to the linear velocity of the link which is covered by each SSL. This variation of the dimensions of the bounding volumes supposes an increase of the safety of the human-robot interaction task because it not only considers the size of the bodies of the human and the robot but also the maximum displacement which can be performed by their bodies during the period situated between two sampling steps of the human tracking system.

This thesis also proposes the organization of these bounding volumes on a hierarchy to reduce more the number of pairwise distance tests. This novel hierarchy of bounding volumes is composed by three levels. The first level contains global AABBs (Axis-Aligned Bounding Boxes) which cover with one box the complete body of the human operator and the robotic manipulator. The second level contains local AABBS which cover the main limbs and parts of the bodies of the human and the robot. Finally, the third level is composed by the SSLs described above. The use of one level or another is based on the level of detail which is needed for the human-robot distance. Firstly, the first level is used to obtain a coarse approximation of the distance. If this value is below a pre-established threshold, the second level of the hierarchy is applied because this means that a more detailed approximation of the distance is necessary. Similarly, if the distance value obtained by the second level is below a second pre-established threshold, the last level is required in order to calculate the best approximation of the human-robot distance. As it has been demonstrated in several experiments of [Chapter 4](#), this novel hierarchical organization reduces considerably the number of pairwise distance tests in comparison with traditional approaches based on a direct distance computation between bounding volumes.

After obtaining a suitable approximation of the minimum human-robot distance with the hierarchy of bounding volumes previously described, this distance value has to be analyzed by a safety strategy which determines how to alter the normal behavior of the robot in case of a risk of collision. This thesis has implemented a specialized software architecture which integrates the robot controller (usually based on visual servoing path tracking techniques and/or force control) with a safety strategy which is able to stop the normal behavior of the robot when the human-robot distance goes below a safety threshold. This safety behavior can be based on two different actions: to stop completely the robot while the human is too near or to move the robot away from the human in order to keep the safety distance. Three real human-robot interaction tasks have been developed in [Chapter 4](#) in order to verify the performance of the latter safety behavior. The results of these tasks demonstrate that the techniques developed

in this thesis have achieved their main goal, which involves making possible the direct safe collaboration of a human operator and several robotic manipulators in the same workspace and even in the assembly/disassembly of the same product.

The second main goal of this thesis is the development of dexterous manipulation tasks performed by multi-fingered robotic hands installed at the end-effector of robotic manipulators. These dexterous manipulation tasks based on in-hand movements of an object are necessary to perform more flexible human-robot interaction where the robot and the human operator may handle the same object and thus a reconfiguration of the object is necessary. Previous dexterous manipulation planners are based on simple representations of the objects and/or the fingers of the hand where their surfaces must be parametric (such as spheres or ellipsoids) or planar. These restrictions simplify the manipulation problem but cannot be applied in real situations where objects and fingers with complex shapes are used. In fact, most previous dexterous manipulation planners are only applied in software simulations of simple objects manipulated by linear fingers. This thesis overcomes this limitation by proposing a new dexterous manipulation planner which has been applied in [Chapter 5](#) in manipulation tasks with two real robotic hands: a five-fingered *Shadow* hand and a three-fingered *Barrett* hand. This dexterous manipulation planner is based on a general representation of the surfaces of the fingers and the objects as standard triangle meshes. This representation not only permits the interoperability with previous software so that models of objects and hands can be easily introduced in the planner but it also permits to adjust the level of detail of the surfaces depending on the requirements of the manipulation task. Thereby, the number of triangles of the surface can be increased when fine manipulation tasks are required and it can be reduced when quick real-time manipulation tasks are necessary.

The developed dexterous manipulation planner has been divided into two levels: global planner and local planner. This organization favors modularity since any of the two levels can be changed without having to redesign the overall planner. The global planner computes the intermediate configurations of the trajectory which drives the object from its initial configuration towards the desired final configuration. This trajectory computation is based on an interpolation of positions and orientations of the object. The local planner computes the movements which should be executed by the fingers in order to move the object between each pair of intermediate configurations of the trajectory generated by the global planner. This local planner is based on three main steps: the computation of the current contact state between the fingers and the object, the generation of all the possible contact changes which can take place from the current contact state and the computation of the finger joint angle increments which can originate these contact changes. The current contact state between each finger and the object (i.e. the pair of primitives of their surfaces which are in contact) is obtained by applying the collision detection functions of the software simulators where this planner has been implemented. The contact changes are computed from a new contact evolution model which is based on the triangle mesh representation of the surfaces of the fingers and the object. This contact model, which is one of the contributions of this thesis, is composed by a contact graph which identifies all the variations between neighbor contacting primitives (i.e. vertices, edges and faces) which can occur due to rolling between the triangle mesh surfaces of the each finger and the object. Finally, for

each new contact state generated from this graph, the local planner applies the pseudo-inverse matrix of the corresponding finger in order to determine if it is possible to perform this contact transition by a finite movement of the finger. Thereby, the local planner does not only consider the contact maintenance between the surfaces of the fingers and the object with the previous contact evolution model but it also applies the kinematic restrictions of the fingers so that contact transitions which require unfeasible finger movements are rejected.

The main limitation of the proposed dexterous manipulation planner is that it is initially based only on geometric information and it does not take into account the contact forces which are applied by the fingers to the object. This contact information is very important because the grasp of the manipulated object tends to become unstable after several in-hand manipulation movements if the contact forces are not considered. In fact, the geometric dexterous manipulation planner implemented in the simulator of the *Shadow* hand of this thesis cannot complete manipulation tasks which involve long trajectories of the object because, after several in-hand movements generated by the local planner, the grasp becomes unstable and some fingers lose contact with the object. This loss of contact occurs because the forces applied by each finger decrease gradually after each in-hand movement executed by the local planner. Due to this fact, this thesis proposes a new finger readjustment algorithm which is executed after each step of the local planner. Two are the main goals of this algorithm. On the one hand, it performs small readjustment movements of the fingers whose contact pressure is not sufficient. These readjustment movements consists of small increments of the finger joint angles which usually do not provoke any movement of the object but that involve an increase of the contact pressure applied to the object. On the other hand, the readjustment algorithm also stops the manipulation task when a too small contact pressure is detected on one contacting finger but it cannot be increased after several readjustment movements. Thereby, this algorithm guarantees that the stability of the grasp is kept during the execution of the manipulation task. This readjustment algorithm, in conjunction with the geometric planner, has been used in [Chapter 5](#) to complete several manipulation tasks of different objects with a *Barrett* hand equipped with tactile sensors. The execution of long trajectories of these objects during these tasks demonstrates that the proposed planner has achieved its main goal of performing dexterous manipulation based on in-hand movements.

6.2 Publications

The research developed in this thesis has been published in several types of scientific publications in order to share the proposed contributions with the scientific community. In particular, the developed publications can be divided into the following groups which are detailed in the next sections: 5 JCR journal papers (i.e. journals indexed by the *Journal Citation Reports* database of *Thomson Reuters*), 3 book chapters, 2 non-JCR journal papers, 6 international conference papers and 6 national (Spanish) conference papers.

6.2.1 JCR Journal Papers

- **“Safe human-robot interaction based on dynamic sphere-swept line bounding volumes”**. J. A. Corrales, F. A. Candelas and F. Torres, *Robotics and Computer-Integrated Manufacturing*, 27(1), pp. 177-185, Feb. 2011; ISSN: 0736-5845; Impact Factor: 1.687 (JCR 2009).

Abstract: This paper presents a geometric representation for human operators and robotic manipulators, which cooperate in the development of flexible tasks. The main goal of this representation is the implementation of real-time proximity queries, which are used by safety strategies for avoiding dangerous collisions between humans and robotic manipulators. This representation is composed of a set of bounding volumes based on swept-sphere line primitives, which encapsulate their links more precisely than previous sphere-based models. The radius of each bounding volume does not only represent the size of the encapsulated link, but it also includes an estimation of its motion. The radii of these dynamic bounding volumes are obtained from an algorithm which computes the linear velocity of each link. This algorithm has been implemented for the development of a safety strategy in a real human-robot interaction task.

- **“Safe human-robot cooperation based on an adaptive time-independent image path tracker”**. J. Pomares, F. A. Candelas, F. Torres, J. A. Corrales and G. J. García, *International Journal of Innovative Computing, Information and Control*, 6(9), pp. 3819-3842, Sept. 2010; ISSN: 1349-4198; Impact Factor: 2.932 (JCR 2009).

Abstract: This paper describes a multi-sensorial manufacturing system for performing complex disassembly tasks which require human-robot cooperation. This system is composed by two coordinated subsystems: a robotic image-based visual servoing path tracker and a safety system for human-robot interaction. The path tracker based on visual servoing solves the problems of previous computer vision systems proposed to guide robots in situations where the robot interacts with the workspace. On the one hand, this path tracker always completes the planned trajectory due to its time-independent behaviour. On the other hand, this path tracker is adaptive because the camera is calibrated online by using virtual visual servoing during the visual servoing task. The stability of this path tracker has been analyzed with the Lyapunov theory. Furthermore, in order to guarantee the human safety during the cooperation, a new collision avoidance system is presented. This safety system implements a multi-sensorial fusion approach to precisely determine the distance between all the links of the robot and all the limbs of the human operator during the task. Both subsystems have been evaluated in a real task which involves the change of a blown light bulb of a streetlamp.

- **“Sensor data integration for indoor human tracking”**. J. A. Corrales, F. A. Candelas and F. Torres, *Robotics and Autonomous Systems*, 58(8), pp. 931-939, Aug. 2010; ISSN: 0921-8890; Impact Factor: 1.361 (JCR 2009).

Abstract: A human tracking system based on the integration of the measurements from an inertial motion capture system and a UWB (Ultra-Wide Band) location system has been developed. On the one hand, the rotational measurements from the inertial system are used to track precisely all limbs of the body of the human. On the other hand, the translational measurements from both systems are combined by three different fusion algorithms (a Kalman filter, a particle filter and a combination of both) in order to obtain a precise global localization of the human in the environment. Several experiments have been performed to compare their accuracy and computational efficiency.

- **“Survey of Visual and Force/Tactile Control of Robots for Physical Interaction in Spain”.** G. J. García, J. A. Corrales, J. Pomares and F. Torres, *Sensors*, 9(12), pp. 9689-9733, Dec. 2009; ISSN: 1424-8220; Impact Factor: 1.821 (JCR 2009).

Abstract: Sensors provide robotic systems with the information required to perceive the changes that happen in unstructured environments and modify their actions accordingly. The robotic controllers which process and analyze this sensory information are usually based on three types of sensors (visual, force/torque and tactile) which identify the most widespread robotic control strategies: visual servoing control, force control and tactile control. This paper presents a detailed review on the sensor architectures, algorithmic techniques and applications which have been developed by Spanish researchers in order to implement these mono-sensor and multi-sensor controllers which combine several sensors.

- **“A cooperative robotic system based on multiple sensors to construct metallic structures”.** P. Gil, J. Pomares, S. T. Puente, F. A. Candelas, G. J. García, J. A. Corrales and F. Torres, *The International Journal of Advanced Manufacturing Technology*, 45(5-6), pp. 616-630, Nov. 2009; ISSN: 0268-3768; Impact Factor: 1.128 (JCR 2009).

Abstract: This paper describes a multisensorial robotic system to automatically construct metallic structures. Two robots must work cooperatively in the same workspace to perform the task. The robots are automatically guided using visual and force sensor information. A new time-independent visual force control system which guarantees the adequate robot behaviour during the construction of the structure is described. During the construction of the structure, a human operator works cooperatively with the robots in order to perform some tasks which cannot be automatically developed by the robots. To do so, a new human-robot cooperation approach is described in order to guarantee the human safety. The correct behaviour of the different subsystems proposed in the paper is demonstrated in Section 6 by the construction of a real structure composed of several metallic tubes and different types of pieces to join them.

6.2.2 Book Chapters and non-JCR Journal Papers

- **“Direct Visual Servoing to Track Trajectories in Human-Robot Cooperation”.** J. Pomares, J. A. Corrales, G. J. García and F. Torres, International Journal of Advanced Robotic Systems; ISSN: 1729-8806; Status: Under Review.

Abstract: This paper describes a dynamic image-based control system to guide two coupled robots. The first robot is a Mitsubishi PA-10 robotic manipulator which has a second mini-robot with 3 degrees of freedom (DOF) attached at its end-effector. The vision system used for the guidance of both robots is composed of a camera at the end-effector of the mini-robot. The paper presents a new method to perform the mini-robot guidance using dynamic control to track a previous generated image trajectory. The mini-robot performs the tracking in a workspace in cooperation with a human operator. Therefore, the proposed visual control is combined with virtual visual servoing to perform a safety behavior.

- **“Flexible Applications of Robotic Manipulators Based on Visual-Force Control and Cooperation with Humans”.** J. A. Corrales, G.J. García, F. Torres, J. Pomares and P. Gil, In: Manipulators: Theory, types and applications, G. Legnani and I. Fassi, Eds. Nova Science Publishers, 2011; Status: Under Review.

Abstract: Visual servoing, force control and human-robot interaction are important techniques for developing flexible tasks where human and robots can share the same workspace. These three components have been combined by the authors of this chapter for the development of several flexible assembly/disassembly tasks. This chapter does not only contain the theoretical foundations of these techniques previously developed by the authors but also a review of the main real applications where they have been applied.

- **“Kalman Filtering for Sensor Fusion in a Human Tracking System”.** J. A. Corrales, F. A. Candelas and F. Torres, In: Kalman Filter, V. Kordic, Ed. In-Tech: Vukovar, Croatia, May. 2010; pp. 60-72; ISBN: 978-953-307-094-0.

Abstract: This chapter proposes two different fusion algorithms in order to combine the localization measurements of an inertial motion capture system and a UWB localization system which are used to track human operators who collaborate with robotic manipulators. This first fusion algorithm transforms measurements from the two systems in the same coordinate system by recalculating the transformation matrix each time a new measurement from the UWB system is received. This approach relies heavily on the accuracy of the measurements from the UWB system because the transformation matrix recalculation assumes that the last UWB measurement is completely correct. Thus, errors in UWB measurements are not considered and only the translational errors of the motion capture system are corrected. Another fusion algorithm which takes into account UWB errors has been developed in order to overcome these drawbacks and obtain more continuous trajectories.

- **“Safe Cooperation between Human Operators and Visually Controlled Industrial Manipulators”**. J. A. Corrales, G.J. García, F. A. Candelas, J. Pomares and F. Torres, In: *Advances in Human-Robot Interaction*, A. Lazinica, Ed. In-Tech: Vukovar, Croatia, Feb. 2010; pp. 203-224; ISBN: 978-953-307-051-3.

Abstract: This chapter presents a safety system based on a hierarchy of bounding volumes for developing human-robot interaction tasks where human operator work near robotic manipulators. It is organized as follows. Section 2 describes the human tracking system which is used to localize precisely all the limbs of the human operator who collaborates with the robotic manipulator. Section 3 presents an introduction to visual servoing and shows how the robot is controlled by a time-independent path tracker based on it. Section 4 describes how the safety behaviour which avoids any collision between the human and the robot is implemented. This section presents the hierarchy of bounding volumes which is used to compute the human-robot distance and modify the movements of the robot accordingly. Section 5 enumerates the results obtained in the application of all of these techniques in a real human-robot interaction task where a fridge is disassembled. Finally, the last section presents the conclusions of the developed research.

- **“Using visual and force information in robot-robot cooperation to build metallic structures”**. J. Pomares, P. Gil, J. A. Corrales, G. J. García, S. T. Puente, and F. Torres, *Journal of Automation, Mobile Robotics & Intelligent Systems*, July 2009; pp. 96-101; ISSN: 1897-8649.

Abstract: In this paper, a cooperative robot-robot approach to construct metallic structures is presented. In order to develop this task, a visual-force control system is proposed. The visual information is composed of an eye-in-hand camera, and a time of flight 3D camera. Both robots are equipped by a force sensor at the end-effector. In order to allow a human cooperate with both robots, an inertial motion capture system and an indoor localization system are employed. This multisensorial approach allows the robots to cooperatively construct the metallic structure in a flexible way and sharing the workspace with a human operator.

6.2.3 International Conference Papers

- **“Geometric Planner for Dexterous Manipulation of Polyhedral Objects by Rolling Contacts with a Multi-fingered Hand”**, J. A. Corrales, V. Perdereau and F. Torres, In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Francisco, CA, USA, Sept. 2011; Status: Under Review.

Abstract: This paper presents a novel geometric planner for dexterous manipulation with a multi-fingered robotic hand. This planner receives as input an initial grasp of the object to be manipulated and a desired final configuration (position and orientation) for the object. The planner computes the movements of the fingers which are required to move the object from the initial grasp to this final configuration without breaking contacts. The planner is divided into two

components: global planner and local planner. The global planner generates the intermediate trajectory of the object between its initial and final configurations. The local planner computes the increments in the finger's joints which are needed to move the object between each pair of consecutive configurations in the trajectory. It uses a triangle mesh representation of the surfaces of the fingers and the object and a contact evolution graph in order to compute all the possible transitions between the contacting primitives of their surfaces. Both planners have been implemented as a program which communicates with a robotic simulator in order to test them in manipulation tasks.

- **“Modelling and Simulation of a Multi-fingered Robotic Hand for Grasping Tasks”**, J. A. Corrales, C. A. Jara and F. Torres, In: 11th International Conference on Control, Automation, Robotics and Vision, Singapore, Singapore, Dec. 2010; pp. 1577-1582; ISBN: 978-1-4244-7814-9.

Abstract: This paper develops the kinematic, dynamic and contact models of a three-fingered robotic hand (BarrettHand) in order to obtain a complete description of the system which is required for manipulation tasks. These models do not only take into account the mechanical coupling and the breakaway mechanism of the under-actuated robotic hand but they also obtain the force transmission from the hand to objects, which are represented as triangle meshes. The developed models have been implemented on a software simulator based on the Easy Java Simulations platform. Several experiments have been performed in order to verify the accuracy of the proposed models with regard to the real physic system.

- **“Visual Servoing Path Tracking for Safe Human-Robot Interaction”**, G. J. García, J. A. Corrales, J. Pomares, F. A. Candelas and F. Torres, In: IEEE International Conference on Mechatronics, Malaga, Spain, April 2009; pp. 672-678; ISBN: 978-1-4244-4195-2.

Abstract: When a human is introduced into a robotic cell, the robot controller must be aware of the human location in order to assure her/his physical integrity. This paper presents a pre-collision strategy which maintains a safety distance between a robot and a human who wears a tracking system composed of a motion capture suit and a UWB localization system. The system proposed is able to guide the robot using visual servoing through a previously defined path. The time-independent behaviour of this system enables the robot to completely track the desired trajectory, even in those cases when the robot goes away from the human during the tracking to guarantee her/his safety.

- **“Tracking based on Hue-Saturation Features with a Miniaturized Active Vision System”**, J. A. Corrales, P. Gil, F. A. Candelas and F. Torres, In: 40th International Symposium on Robotics, Barcelona, Spain, March 2009; pp. 107-112; ISBN: 978-84-920933-8-0.

Abstract: This paper presents a miniaturized active vision system for visual tracking. One of the main problems in visual tracking is the autonomy and

manageability of the system to be mounted on robotic structures, such as mobile and manipulator robots. The proposed active vision system has been built using a motorized platform characterized by its low price, lightness, small dimensions and wireless control. It is interesting for visual tracking applications where constraints of size and weight must be considered. In our mini active vision system, a tracking method based on CamShift has been implemented. The novelty of our tracker, in comparison with CamShift, is its ability to automatically combine a hue distance component and a saturation component from the HSV colour model in order to track objects in dynamic backgrounds with similar hue values.

- **“Intelligent Robotic Multisensorial System to Build Metallic Structures”,** P. Gil, J. Pomares, S.T. Puente, J. A. Corrales, G. J. García and F. Torres, In: IFAC International Workshop on Intelligent Manufacturing Systems, Szczecin, Poland, Oct. 2008; pp. 189-194; ISBN: 978-3-902661-40-1.

Abstract: This paper describes a multisensorial system employed in a robotic application developed to automatically construct metallic structures. The proposed system has the novelty of a high degree of flexibility with an intelligent multisensorial system. This sensorial system is composed of a visual-force control system, a time of flight 3D-camera, an inertial motion capture system and an indoor localization system. These two last sensors are used to avoid possible collisions between the human operator and the robots working in the same workspace.

- **“Hybrid Tracking of Human Operators using IMU/UWB Data Fusion by a Kalman Filter”,** J. A. Corrales, F. A. Candelas y F. Torres, In: 3rd ACM/IEEE International Conference on Human-Robot Interaction, Amsterdam, Netherlands, March. 2008; pp. 193-200; ISBN: 978-1-60558-017-3.

Abstract: The precise localization of human operators in robotic workplaces is an important requirement to be satisfied in order to develop human-robot interaction tasks. Human tracking provides not only safety for human operators, but also context information for intelligent human-robot collaboration. This paper evaluates an inertial motion capture system which registers full-body movements of a user in a robotic manipulator workplace. However, the presence of errors in the global translational measurements returned by this system has led to the need of using another localization system, based on Ultra-WideBand (UWB) technology. A Kalman filter fusion algorithm which combines the measurements of these systems is developed. This algorithm unifies the advantages of both technologies: high data rates from the motion capture system and global translational precision from the UWB localization system. The developed hybrid system not only tracks the movements of all limbs of the user as previous motion capture systems, but is also able to position precisely the user in the environment.

6.2.4 National Conference Papers

- **“Modelo Dinámico de una Mano Robótica con Mecanismo de Desacople Mecánico en los Dedos”**, J. A. Corrales, F. Torres and F. A. Candelas, In: XXXI Jornadas de Automática, Jaén, Spain, Sept. 2010; pp. 176-182; ISBN: 978-84-693-0715-1.

Abstract: This paper develops the dynamic model of a three-fingered robotic hand. In order to do so, it considers two special features of this kind of robotic devices: mechanical coupling between the joints of each finger of the hand and a decoupling mechanism (called breakaway) of these joints which is activated when the link is blocked by an object. On the one hand, the mechanical coupling between the joints is modeled by proportional relations between the joints of the fingers. On the other hand, the decoupling mechanism is modeled by an algorithm which computes the torque applied over the joint in order to determine the moment when the breakaway mechanism is activated. Then, a change of the previous proportional relations is performed. These particularities have been integrated in the dynamic model of the hand and they have been tested in several experiments.

- **“Modelado y Simulación de Sistemas Robóticos Complejos mediante Easy Java Simulations”**, C. A. Jara, J. A. Corrales, G. Lorenzo, S. T. Puente, F. A. Candelas and F. Torres, In: XXX Jornadas de Automática, Valladolid, Spain, Sept. 2009; pp. 348-354; ISBN: 13-978-84-692-2387-1.

Abstract: The main goal of this paper is focused on the presentation of a kinematic model and the simulation of two robotic systems: a three-fingered robotic hand and a multi-robot platform with a RRR robot coupled to an anthropomorphic PA-10 robot. In order to develop the simulations, Easy Java Simulations has been used. This is a software platform for the creation of animations and interactive applications in Java.

- **“La Tecnología RFID en el Contexto de la Robótica de Servicios: Breve Estado del Arte”**, J. A. Corrales, P. J. Sanz, F. Torres, F. A. Candelas and R. Marín, In: 2as Jornadas Científicas sobre RFID, Cuenca, Spain, Nov. 2008; pp. 1-7; ISBN: 978-84-612-7225-9.

Abstract: Whereas RFID technology has demonstrated its potential applications, in the last four or five years its combination with robotics begins to be spectacular. This combination enables the solution of applications which were impossible until now in different scenarios, from industrial environments to security systems. The current work tries to give an updated view, without being exhaustive, of a group of recent applications which combine both technologies: robotics and RFID. It divides the state of art into two big areas: navigation and tracking solutions and service robotics, including hospitals, museums, etc. Finally, it offers a prediction of the future of these technologies in a medium-term period.

- **“Diseño de una Mini-cámara Motorizada para el Seguimiento de Objetos”,** J. A. Corrales, P. Gil, F. A. Candelas and F. Torres, In: XXIX Jornadas de Automática, Tarragona, Spain, Sept. 2008; pp. 808-815; ISBN: 978-84-691-6883-7.

Abstract: This paper presents a miniaturized vision system based on a CMOS wireless camera with three motors which is able to track mobile targets. This system uses color features and histograms in order to detect the object in the image with a small computational cost. The novelty of this work lies in the flexibility of the designed system for applications where low weights and small dimensions are required in order not to restraint the movements of the robots where this system is installed.

- **“Captura de Movimiento y Localización en Interiores Aplicadas a Entornos Industriales”,** J. A. Corrales, F. A. Candelas and F. Torres, In: XXVIII Jornadas de Automática, Huelva, Spain, Sept. 2007; pp. 331-339; ISBN: 978-84-690-7497-8.

Abstract: The application of Ambient Intelligence in industrial environments improves the interaction between human operators and industrial machinery. In particular, the development of intelligent industrial environments enables the cooperation between human operators and industrial robots. However, in order to make this cooperation real, the system must localize precisely the human operator. Due to this reason, this paper proposes the use of a motion capture system. The analysis of its precision shows excessive errors in the global position measurements. In order to correct them, an additional localization system based on UWB technology is applied.

- **“Tecnologías en la Inteligencia Ambiental”,** J. A. Corrales, F. A. Candelas and F. Torres, In: XXVII Jornadas de Automática, Almería, Spain, Sept. 2006; pp. 326-332; ISBN: 978-84-689-9417-0.

Abstract: This paper presents the term “Ambient Intelligence” (AmI) and describes the different technologies which make possible its development: ubiquitous computing, ubiquitous communication and intelligent interfaces. We enumerate different localization techniques which are employed in intelligent environments to determine the localization of the user and offer the most suitable services. In order to better understand the possibilities of AmI, we also show its main areas of application.

6.3 Future Work

As described in [Section 6.1](#), the techniques developed along this thesis have fulfilled the two requirements established in the introduction of [Chapter 1](#) as its two main goals: the guarantee of the human safety during the development of human-robot interaction tasks and the development of dexterous manipulation tasks based on in-hand movements. Both objectives not only have been solved theoretically but also empirically by their application on several real experiments. Nevertheless, the proposed techniques can be extended in future work in order to develop more flexible and unobtrusive human-robot interaction tasks and more general manipulation tasks.

In particular, the human tracking system implemented in this thesis has only been applied to guarantee the safety of human operators with two simple safety behaviors. The first behavior is based on stopping completely the movements of the robotic manipulator while the human is near the robot and the second behavior is based on moving the robot away from the human in a linear path in order to keep the safety distance. Both safety behaviors are executed when the human is near the robot and require the stop of the normal behavior of the robot. Therefore, these behaviors do not suppose a real adaptation of the robotic task to the human movements but a safety strategy which stops temporarily the normal behavior of the robot for safety purposes. In order to develop more flexible human-robot interaction tasks, the information from the human tracking system cannot only be used to execute safety strategies but also to modify the normal behavior of the robot.

This adaptation of the behavior of the robot to the human operator's movements needs two different requirements to be completed: an interpretation of the actions which are performed by the human operator and a modification of the robot behavior depending on this interpretation. The first requirement involves using the localization data from the human tracking system not only to compute the minimum human-robot distance but also to extract actions from the movements of the human operator. For instance, this action interpretation system should analyze the relative movements of the human in order to determine when the human operator wants to give an object to the robot. This action could be recognized by identifying a small inclination of the trunk of the human operator and an extension of one arm towards the direction where the robot is situated. In addition, this action recognition could integrate the information from different sensors and not only from the sensors of the tracking system. For instance, in the previous example, the action of handing an object to the robot could be verified by a vision system which is able to segment the object which is carried by the human operator in her/his hand. After recognizing an action, the robot controller should be able to determine a modification of the current behavior of the robot in order to adapt it to this action. In the previous example, if the robot is performing a tracking process of a trajectory, it should modify this trajectory in order to move its end-effector towards the hand of the human operator so that it can grab the object handed by the human. Thereby, the robot controller is able to understand the intentions of the human operator and act accordingly. These techniques could be applied not only in the development of flexible industrial human-robot interaction tasks but also in the creation of robotic assistants in home environments.

The human-robot interaction tasks developed in this thesis have only been applied in a laboratory. Therefore, in future research, the safety strategies developed in this thesis should be tested in real industrial plants. In order to do this, an analysis of the requirements of the industry where this system is applied should be performed in order to particularize it. For instance, this analysis may establish the parts of the human operator's body which are necessary to be tracked because they are inside the robot's workspace. Thereby, the human tracking system can be simplified and its cost can be reduced and adapted to the real needs of the industry. This analysis should also determine the productive processes of the industry which can be really improved with the collaboration of human operators in order to apply this safety system only to them.

Finally, a study of the adaptation of the developed safety systems to the current safety standards of the industry should also be included in this analysis.

The other technique of this thesis that could be extended is the dexterous manipulation planner. As this planner is based on the continuous maintenance of the contact between the surfaces of the fingers and the object, it could be improved by adding the possibility of breaking contacts in a coordinated way and thus, new manipulation modes could be implemented (such as finger gaiting). These manipulation modes will increase the number of feasible manipulation tasks because the fingers could be repositioned over new contact points when they reach the limits of their workspaces. Another improvement of the dexterous manipulation planner could be achieved by implementing the global planner with a statistical technique which samples possible configurations of the object randomly. Thereby, intermediate configurations which are not feasible by the fingers could be substituted by other neighbor configurations.



Universitat d'Alacant
Universidad de Alicante

APPENDIX

Modeling of the Barrett Hand

This first appendix of this thesis includes a complete modeling of the *Barrett* hand. This is the robotic hand which has been used in the real manipulation tasks which have been performed in order to verify the correct execution of the dexterous manipulation planner proposed in this thesis. The execution of this planner requires a complete kinematic model of the multi-fingered robotic hand which is applied in the manipulation tasks because it is necessary to compute the pseudo-inverse of its Jacobian. This appendix not only details the kinematic model of

A.1 Introduction	153
A.2 Kinematic Model	154
A.2.1 Direct Kinematic Model	154
A.2.2 Inverse Kinematic Model	154
A.3 Dynamic Model	155
A.3.1 Inertia Matrix	156
A.3.2 Coriolis-Centrifugal Vector	157
A.3.3 Gravitational Vector	158
A.4 Breakaway Model	158

the *Barrett* hand with the coupling relations between its fingers joints but it also describes its dynamic model. This modeling also includes its decoupling mechanism (breakaway). These three components have been implemented in the software simulator of this hand.

A.1 Introduction

The *Barrett* Hand [Townsend-2000] is an example of a widely used robotic hand with under-actuated fingers. Under-actuated mechanical schemes have been widely applied in the design of many multi-fingered hands due to their compactness and their simplicity in control [Birglen-2008]. The joints of under-actuated fingers are generally coupled by transmission systems (such as tendons and gear trains) so that the number of required actuators is inferior to the number of degrees of freedom of the finger. In order to reduce the loss of dexterity of the fingers, many under-actuated fingers implement decoupling mechanisms (based on springs or clutches) which change the linkage relation between the joints when an object blocks one of the phalanxes. Thereby, under-actuated fingers are able to envelope the objects to be grasped and adapt to their shape.

The *Barrett* hand implements both mechanical techniques (joint coupling and decoupling). This hand is composed of three articulated fingers and a palm which integrates four servomotors and the control electronics. The two phalanxes of each finger are driven by one of the servomotors while the bases of fingers 1 and 2 rotate around the palm in a spread movement which is generated by the fourth servomotor. When the inner phalanx contacts an object with a specific force, a breakaway system based on a clutch decouples the outer phalanx from the inner phalanx so that it can perform an enveloping grasp of the object.

This appendix develops a complete description of this under-actuated hand, not only from a kinematic point of view, but also from a dynamic point of view and in both cases the breakaway system is taken into consideration.

A.2 Kinematic Model

A.2.1 Direct Kinematic Model

First of all, a coordinate system is assigned to each link of each finger k of the hand according to the standard Denavit-Hartenberg convention [Denavit-1955]. Figure A.1 shows all the frames assigned to each finger.

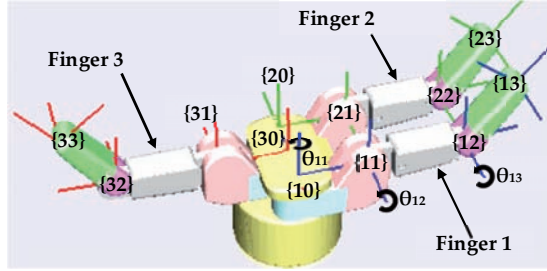


Figure A.1. Coordinate frames assigned to each finger of the Barrett hand.

Next, a set of Denavit-Hartenberg parameters are defined for each coordinate system (see Table A.1). The values for the dimension constants of the links are shown in Table A.2.

Table A.1. Denavit-Hartenberg parameters for each finger k where $r = [1, 1, 0]$ for $k = [1, 2, 3]$.

Frame	a_{ki}	α_{ki}	d_{ki}	θ_{ki}
$\{k1\}$	A_1	90°	0 mm	$r \cdot q_{k1}$
$\{k2\}$	A_2	0°	0 mm	$q_{k2} + \Phi_2$
$\{k3\}$	A_3	0°	0 mm	$q_{k3} + \Phi_3$

Table A.2. Dimension parameters of the links of the Barrett hand.

Parameters	Values
A_1, A_2, A_3	0.050, 0.070, 0.056 m
Φ_2, Φ_3	$2.46^\circ, 50^\circ$

Finally, the joint angles are related with the encoder counts of the motors by the following transmission equations:

$$\theta_{11} = -2/35 \cdot q_{M4} \quad (\text{A.1})$$

$$\theta_{12} = -2/35 \cdot q_{M4} \quad (\text{A.2})$$

$$\theta_{k2} = 1/125 \cdot q_{Mk} \quad (\text{A.3})$$

$$\theta_{k3} = 1/375 \cdot q_{Mk} \quad (\text{A.4})$$

Where $q_{Mk} = [0, 17500]$ with $k = [1, 2, 3]$ is the number of encoder counts of the motor of finger k . $q_{M4} = [0, 3150]$ is the number of encoder counts of the spread motor.

A.2.2 Inverse Kinematic Model

If the fingertip of finger k is located at the position ${}^{k0}\mathbf{p} = ({}^{k0}p_x, {}^{k0}p_y, {}^{k0}p_z)$, the corresponding joint values $(\theta_{k1}, \theta_{k2}, \theta_{k3})$ can be computed by considering the geometric relations between the links.

In particular, the angle θ_{k1} for fingers 1 and 2 is obtained with the following expression:

$$\theta_{k1} = \arctan({}^{k0}p_y / {}^{k0}p_x) \quad \text{for } k = 1, 2. \quad (\text{A.5})$$

In the case of finger 3, θ_{31} is always zero because it does not perform spread movements. After the computation of θ_{k1} , angles θ_{k2} and θ_{k3} can be calculated by taking into account that the two phalanxes of the finger are contained in a plane $k\Pi$, as shown in Figure A.2.

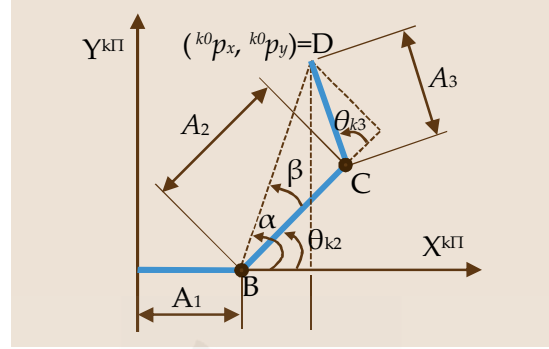


Figure A.2. Geometric relations of the links of a finger.

Thereby, by applying the law of cosines to the BCD triangle in Figure A.2, the following expression can be derived for the computation of the angle θ_{k3} :

$$\theta_{k3} = \arccos \left(\frac{\left(\sqrt{{}^{k0}p_x^2 + {}^{k0}p_y^2} - A_1 \right)^2 + {}^{k0}p_z^2 - A_2^2 - A_3^2}{2A_2A_3} \right) \quad (\text{A.6})$$

Finally, the angle θ_{k2} is obtained from the difference between the angles α and β in Figure A.2:

$$\theta_{k2} = \alpha - \beta = \arctan \left(\frac{{}^{k0}p_z}{\sqrt{{}^{k0}p_x^2 + {}^{k0}p_y^2} - A_1} \right) - \arctan \left(\frac{A_3 \sin(\theta_{k3})}{A_2 + A_3 \cos(\theta_{k3})} \right) \quad (\text{A.7})$$

A.3 Dynamic Model

For each finger k , its inverse dynamic model can be represented by the following equation:

$$\tau_k = D_k(\Theta_k) \ddot{\Theta}_k + H_k(\Theta_k, \dot{\Theta}_k) + C_k(\Theta_k) \quad (\text{A.8})$$

Where Θ_k is a $n_k \times 1$ vector with the joint values, τ_k is a $n_k \times 1$ vector with the joint torques, D_k is the $n_k \times n_k$ inertia matrix, H_k is the $n_k \times 1$ Coriolis-centrifugal vector and C_k is the $n_k \times 1$ gravitational vector. n_k is the number of joints of each finger k ($n_1 = n_2 = 3$ and $n_3 = 2$).

The Lagrange-Euler algorithm [Fu-1987] has been implemented in order to obtain the analytical representation of the matrices in (A.8), which are shown in the following subsections. The input parameters for this algorithm are the following features of each link j of each finger k : its mass m_j , its inertia tensor I_j computed at the frame $\{kj\}$ and the coordinates $p_j = (p_{jx}, p_{jy}, p_{jz})$ of its center of mass with respect to frame $\{kj\}$. The real val-

ues of these parameters for the *Barrett Hand* are shown in Table A.3). The gravity acceleration vector $\mathbf{g}=(g_x, g_y, g_z)$ at the base frame $\{k0\}$ of each finger is also required as input parameter.

Table A.3. Mass and inertia parameters of the links of the *Barrett Hand*.

Parameters	Values
m_1, m_2, m_3	0.16252678, 0.05795741, 0.03421781 Kg
$I_{1xx}, I_{1xy}, I_{1xz}$	0.0000658, 0.0000572, -0.0000011 Kg·m ²
$I_{1yy}, I_{1yz}, I_{1zz}$	0.0001310, -0.0000013, 0.0001808 Kg·m ²
$I_{2xx}, I_{2xy}, I_{2xz}$	0.0000047, -0.0000020, 0.0000030 Kg·m ²
$I_{2yy}, I_{2yz}, I_{2zz}$	0.0001208, -0.0000000, 0.0001218 Kg·m ²
$I_{3xx}, I_{3xy}, I_{3xz}$	0.00000298, -0.00000135, -0.00000032 Kg·m ²
$I_{3yy}, I_{3yz}, I_{3zz}$	0.00005836, -0.00000001, 0.00005732 Kg·m ²
p_{1x}, p_{1y}, p_{1z}	-0.01798334, -0.01569925, 0.00086502 m
p_{2x}, p_{2y}, p_{2z}	-0.03827982, 0.00107042, -0.00058628 m
p_{3x}, p_{3y}, p_{3z}	-0.03629026, 0.00182016, 0.00020041 m

A.3.1 Inertia Matrix

The inertia matrix D_k for fingers 1 and 2 ($k=1,2$) is a 3×3 symmetric matrix whose elements over its diagonal are the following ones:

$$D_k(1,1)= m_2 A_1^2 + m_3 A_1^2 - 2C_{23}I_{3xy}S_{23} + 2C_2m_2p_{2x}A_1 - I_{2xx}C_2^2 + I_{2yy}C_2^2 + C_{23}^2I_{3yy} - C_{23}^2I_{3xx} + A_2^2C_2^2m_2 + A_3^2C_{23}^2m_3 + A_2^2C_2^2m_3 + I_{1yy} + I_{3xx} - 2C_2I_{2xy}S_2 - 2S_2m_2p_{2y}A_1 + 2A_1m_2A_2C_2 + 2A_1m_3A_3C_{23} + 2A_1m_3A_2C_2 + 2A_1m_3p_{3x}C_{23} - 2A_1m_3p_{3y}S_{23} + 2A_1m_1p_{1x} - 2S_2m_2p_{2y}A_2C_2 + 2A_3C_{23}m_3A_2C_2 - 2A_3C_{23}m_3p_{3y}S_{23} + 2A_2C_2m_3p_{3x}C_{23} - 2A_2C_2m_3p_{3y}S_{23} + 2C_2^2m_2p_{2x}A_2 + 2A_3C_{23}^2m_3p_{3x} + I_{2xx} + A_1^2m_1 \quad (A.9)$$

$$D_k(1,2)= -I_{3yz}C_{23} - I_{3xz}S_{23} - I_{2yz}C_2 - I_{2xz}S_2 - A_3m_3p_{3z}S_{23} - A_2m_2p_{2z}S_2 - A_2m_3p_{3z}S_2 \quad (A.10)$$

$$D_k(1,3)= -I_{3yz}C_{23} - I_{3xz}S_{23} - A_3m_3p_{3z}S_{23} \quad (A.11)$$

$$D_k(2,2)= I_{2zz} + I_{3zz} + A_2^2m_2 + A_2^2m_3 + A_3^2m_3 + 2A_2m_2p_{2x} + 2A_3m_3p_{3x} + 2A_2A_3m_3C_3 + 2A_2m_3p_{3x}C_3 - 2A_2m_3p_{3y}S_3 \quad (A.12)$$

$$D_k(2,3)= I_{3zz} + A_3^2m_3 + 2A_3m_3p_{3x} + A_2A_3m_3C_3 + A_2m_3p_{3x}C_3 - A_2m_3p_{3y}S_3 \quad (A.13)$$

$$D_k(3,3)= m_3 \cdot A_3^2 + 2 \cdot m_3 \cdot p_{3x} \cdot A_3 + I_{3zz} \quad (A.14)$$

Where $C_i = \cos(\theta_{ki})$, $S_i = \sin(\theta_{ki})$, $C_{ij} = \cos(\theta_{ki} + \theta_{kj})$ and $S_{ij} = \sin(\theta_{ki} + \theta_{kj})$. The inertia matrix D_3 for finger 3 is a 2×2 symmetric matrix whose elements are:

$$D_3(1,1)= I_{3zz} + 2m_3p_{3x}A_3 + I_{2zz} + A_2^2m_2 + m_3A_2^2 + 2A_2m_2p_{2x} + m_3A_3^2 + 2m_3p_{3x}A_2C_3 - 2m_3p_{3y}A_2S_3 + 2m_3A_3C_3A_2 \quad (A.15)$$

$$D_3(1,2) = 2 \cdot m_3 \cdot p_{3x} \cdot A_3 - m_3 \cdot p_{3y} \cdot A_2 \cdot S_3 + m_3 \cdot p_{3x} \cdot A_2 \cdot C_3 + m_3 \cdot A_3^2 + I_{3zz} + m_3 \cdot A_3 \cdot C_3 \cdot A_2 2m_3 p_{3y} A_2 S_3 + 2m_3 A_3 C_3 A_2 \quad (A.16)$$

$$D_3(2,2) = I_{3zz} + 2 \cdot m_3 \cdot p_{3x} \cdot A_3 + m_3 \cdot A_3^2 \quad (A.17)$$

A.3.2 Coriolis-Centrifugal Vector

The Coriolis-centrifugal vector H_k for fingers 1 and 2 ($k=1,2$) is a 3×1 vector with the following elements:

$$\begin{aligned} H_k(1) = & -2q_{d1}q_{d2}C_{23}S_{23}I_{3yy} + 2q_{d1}q_{d2}C_2S_2I_{2xx} + 2q_{d2}q_{d3}I_{3yz}S_{23} + 2q_{d1}q_{d2}I_{3xy} - \\ & 4q_{d1}q_{d2}A_3C_{23}m_3p_{3x}S_{23} - 2q_{d2}q_{d3}A_3m_3p_{3z}C_{23} - 2q_{d1}q_{d3}A_2C_2A_3m_3S_{23} - 2q_{d1}q_{d3}A_2C_2m_3p_{3y}C_{23} - \\ & 2q_{d1}q_{d3}A_2C_2m_3p_{3x}S_{23} - 2q_{d1}q_{d2}A_1m_3p_{3x}S_{23} - 2q_{d1}q_{d2}A_3^2C_{23}m_3S_{23} + 2q_{d1}q_{d2}I_{2xy} - 2q_{d2}q_{d3}I_{3xz}C_{23} \\ & + 2q_{d1}q_{d2}C_{23}S_{23}I_{3xx} + 2q_{d1}q_{d3}C_{23}S_{23}I_{3xx} - 2q_{d1}q_{d3}C_{23}S_{23}I_{3yy} - 2q_{d1}q_{d2}C_2S_2I_{2yy} - q_{d3}^2A_3m_3p_{3z}C_{23} - \\ & q_{d2}^2A_3m_3p_{3z}C_{23} - q_{d2}^2m_3p_{3z}A_2C_2 + q_{d2}^2I_{2yz}S_2 + q_{d3}^2I_{3yz}S_{23} - q_{d3}^2I_{3xz}C_{23} + 2q_{d1}q_{d3}A_3m_3p_{3y} - \\ & 2q_{d1}q_{d2}A_1m_2p_{2x}S_2 - 2q_{d1}q_{d3}A_1A_3m_3S_{23} - 2q_{d1}q_{d2}A_1m_3A_2S_2 - 2q_{d1}q_{d2}A_2^2C_2m_3S_2 - \\ & 2q_{d1}q_{d2}A_1A_2m_2S_2 - 4q_{d1}q_{d2}A_2C_2^2m_2p_{2y} - 4q_{d1}q_{d3}A_3C_{23}^2m_3p_{3y} - 4q_{d1}q_{d2}A_3m_3p_{3y}C_{23}^2 - \\ & q_{d2}^2I_{3xz}C_{23} - 4q_{d1}q_{d2}C_2^2I_{2xy} - 4q_{d1}q_{d3}C_{23}^2I_{3xy} - 4q_{d1}q_{d2}C_{23}^2I_{3xy} + 2q_{d1}q_{d2}A_2m_2p_{2y} - \\ & 2q_{d1}q_{d2}A_1A_3m_3S_{23} + 2q_{d1}q_{d2}S_{23}m_3p_{3y}A_2S_2 - 4q_{d1}q_{d2}A_2C_2m_2p_{2x}S_2 - 2q_{d1}q_{d2}A_1m_3p_{3y}C_{23} + \\ & q_{d2}^2I_{3yz}S_{23} - 2q_{d1}q_{d2}A_2C_2A_3m_3S_{23} - 2q_{d1}q_{d2}A_2C_2m_3p_{3y}C_{23} - 2q_{d1}q_{d2}C_{23}m_3p_{3x}A_2S_2 - \\ & q_{d2}^2A_2m_2p_{2z}C_2 - 2q_{d1}q_{d2}A_2^2C_2m_2S_2 - 2q_{d1}q_{d2}A_1m_2p_{2y}C_2 + 2q_{d1}q_{d2}A_3m_3p_{3y} + 2q_{d1}q_{d3}I_{3xy} - \\ & 2q_{d1}q_{d2}A_3C_{23}m_3A_2S_2 - 4q_{d1}q_{d3}A_3C_{23}m_3p_{3x}S_{23} - 2q_{d1}q_{d3}A_1m_3p_{3x}S_{23} - 2q_{d1}q_{d3}A_1m_3p_{3y}C_{23} - \\ & q_{d2}^2I_{2xz}C_2 - 2q_{d1}q_{d2}A_2C_2m_3p_{3x}S_{23} - 2q_{d1}q_{d3}A_3^2C_{23}m_3S_{23} \end{aligned} \quad (A.18)$$

$$\begin{aligned} H_k(2) = & -2q_{d2}q_{d3}A_2C_2m_3p_{3x}S_{23} - 2q_{d2}q_{d3}A_2C_2m_3p_{3y}C_{23} - 2q_{d2}q_{d3}A_2C_2A_3m_3S_{23} + \\ & 2q_{d1}^2C_2^2A_2m_2p_{2y} + q_{d1}^2C_2m_2p_{2y}A_1 + q_{d1}^2S_2m_2p_{2x}A_1 + q_{d1}^2A_3^2S_{23}m_3C_{23} + q_{d1}^2A_3S_{23}m_3A_1 + \\ & q_{d1}^2A_2S_2m_3A_1 + q_{d1}^2A_2^2S_2m_3C_2 + 2q_{d1}^2C_{23}^2A_3m_3p_{3y} + q_{d1}^2C_{23}m_3p_{3y}A_1 + q_{d1}^2S_{23}m_3p_{3x}A_1 + \\ & q_{d1}^2A_2S_2m_2A_1 + q_{d1}^2A_2^2S_2m_2C_2 - q_{d3}^2A_2C_2A_3m_3S_{23} - q_{d3}^2A_2C_2m_3p_{3y}C_{23} - q_{d3}^2A_2C_2m_3p_{3x}S_{23} + \\ & 2q_{d1}^2C_2^2I_{2xy} + 2q_{d1}^2C_{23}^2I_{3xy} + 2q_{d2}q_{d3}A_2S_2A_3m_3C_{23} + 2q_{d2}q_{d3}A_2S_2m_3p_{3x}C_{23} - \\ & 2q_{d2}q_{d3}A_2S_2m_3p_{3y}S_{23} - q_{d1}^2A_3m_3p_{3y} - q_{d1}^2A_2m_2p_{2y} + q_{d1}^2S_{23}m_3p_{3x}A_2C_2 + q_{d1}^2A_2S_2A_3m_3C_{23} + \\ & q_{d1}^2A_2S_2m_3p_{3x}C_{23} + q_{d1}^2C_{23}m_3p_{3y}A_2C_2 + q_{d3}^2A_2S_2A_3m_3C_{23} + q_{d3}^2A_2S_2m_3p_{3x}C_{23} + \\ & q_{d1}^2A_3S_{23}m_3A_2C_2 - q_{d3}^2A_2S_2m_3p_{3y}S_{23} + 2q_{d1}^2A_3S_{23}m_3p_{3x}C_{23} - q_{d1}^2A_2S_2m_3p_{3y}S_{23} + \\ & 2q_{d1}^2S_2A_2m_2p_{2x}C_2 - q_{d1}^2I_{2xy} - q_{d1}^2I_{3xy} + q_{d1}^2S_2C_2I_{2yy} + q_{d1}^2C_{23}S_{23}I_{3yy} - q_{d1}^2S_2C_2I_{2xx} - \\ & q_{d1}^2C_{23}S_{23}I_{3xx} \end{aligned} \quad (A.19)$$

$$\begin{aligned} H_k(3) = & q_{d1}^2A_3^2S_{23}m_3C_{23} + q_{d2}^2A_3S_{23}m_3A_2C_2 + q_{d1}^2S_{23}m_3p_{3x}A_2C_2 + q_{d1}^2A_3S_{23}m_3A_2C_2 + \\ & q_{d1}^2A_3S_{23}m_3A_1 + q_{d1}^2C_{23}m_3p_{3y}A_2C_2 + 2q_{d1}^2A_3S_{23}m_3p_{3x}C_{23} + q_{d1}^2S_{23}m_3p_{3x}A_1 + \\ & q_{d1}^2C_{23}m_3p_{3y}A_1 + q_{d2}^2C_{23}m_3p_{3y}A_2C_2 + q_{d2}^2S_{23}m_3p_{3x}A_2C_2 + 2q_{d1}^2C_{23}^2I_{3xy} + q_{d2}^2S_{23}m_3p_{3y}A_2S_2 - \\ & q_{d2}^2C_{23}m_3p_{3x}A_2S_2 - q_{d2}^2A_3C_{23}m_3A_2S_2 + 2q_{d1}^2C_{23}^2A_3m_3p_{3y} - q_{d1}^2C_{23}S_{23}I_{3xx} + q_{d1}^2C_{23}S_{23}I_{3yy} - \\ & q_{d1}^2A_3m_3p_{3y} - q_{d1}^2I_{3xy} \end{aligned} \quad (A.20)$$

Where $q_{di} = \dot{\theta}_{ki}$. The Coriolis-centrifugal vector H_3 for finger 3 is a 2×1 vector with the following elements:

$$H_3(1) = -m_3A_2q_{d3}(q_{d3}+2q_{d2})(A_3S_3+p_{3y}C_3+p_{3x}S_3) \quad (A.21)$$

$$H_3(2) = q_{d2}^2m_3A_2(p_{3x}S_3+p_{3y}C_3+A_3S_3) \quad (A.22)$$

A.3.3 Gravitational Vector

The gravitational vector C_k for fingers 1 and 2 ($k=1,2$) is a 3×1 with the following elements:

$$\begin{aligned} C_k(1) = & p_{2y}g_y m_2 C_1 S_2 - p_{2y}g_x m_2 S_1 S_2 - p_{3x}g_y m_3 C_{23} C_1 + p_{3x}g_x m_3 C_{23} S_1 + p_{3y}g_y m_3 S_{23} C_1 - \\ & p_{3y}g_x m_3 S_{23} S_1 - p_{2x}g_y m_2 C_1 C_2 + p_{2x}g_x m_2 C_2 S_1 - g_y m_2 C_1 A_2 C_2 + g_x m_2 S_1 A_2 C_2 - g_y m_3 C_1 A_3 C_{23} - \\ & g_y m_3 C_1 A_2 C_2 + g_x m_3 S_1 A_3 C_{23} + g_x m_3 S_1 A_2 C_2 - A_1 g_y m_1 C_1 + A_1 g_x m_1 S_1 - p_{1x}g_y m_1 C_1 + \\ & p_{1x}g_x m_1 S_1 - p_{1z}g_x m_1 C_1 - p_{1z}g_y m_1 S_1 - p_{2z}g_x m_2 C_1 - p_{2z}g_y m_2 S_1 - p_{3z}g_x m_3 C_1 - p_{3z}g_y m_3 S_1 - \\ & g_y m_2 C_1 A_1 + g_x m_2 S_1 A_1 - g_y m_3 C_1 A_1 + g_x m_3 S_1 A_1 \end{aligned} \quad (A.23)$$

$$\begin{aligned} C_k(2) = & p_{2y}g_y m_2 C_1 S_2 - p_{2y}g_x m_2 S_1 S_2 - p_{3x}g_y m_3 C_{23} C_1 + p_{3x}g_x m_3 C_{23} S_1 + p_{3y}g_y m_3 S_{23} C_1 - \\ & p_{3y}g_x m_3 S_{23} S_1 - p_{2x}g_y m_2 C_1 C_2 + p_{2x}g_x m_2 C_2 S_1 - g_y m_2 C_1 A_2 C_2 + g_x m_2 S_1 A_2 C_2 - g_y m_3 C_1 A_3 C_{23} - \\ & g_y m_3 C_1 A_2 C_2 + g_x m_3 S_1 A_3 C_{23} + g_x m_3 S_1 A_2 C_2 - A_1 g_y m_1 C_1 + A_1 g_x m_1 S_1 - p_{1x}g_y m_1 C_1 + \\ & p_{1x}g_x m_1 S_1 - p_{1z}g_x m_1 C_1 - p_{1z}g_y m_1 S_1 - p_{2z}g_x m_2 C_1 - p_{2z}g_y m_2 S_1 - p_{3z}g_x m_3 C_1 - p_{3z}g_y m_3 S_1 - \\ & g_y m_2 C_1 A_1 + g_x m_2 S_1 A_1 - g_y m_3 C_1 A_1 + g_x m_3 S_1 A_1 \end{aligned} \quad (A.24)$$

$$\begin{aligned} C_k(3) = & m_3(p_{3y}g_z S_{23} + p_{3y}g_x C_{23} C_1 + p_{3y}g_y C_{23} S_1 + p_{3x}g_x S_{23} C_1 - p_{3x}g_z C_{23} + p_{3x}g_y S_{23} S_1 - g_z A_3 C_{23} \\ & + g_y S_1 A_3 S_{23} + g_x C_1 A_3 S_{23}) \end{aligned} \quad (A.25)$$

The gravitational vector C_3 for finger 3 is a 2×1 with the following elements:

$$\begin{aligned} C_3(1) = & -p_{2x}m_2 g_x S_2 - p_{2x}m_2 g_z C_2 - p_{2y}m_2 g_x C_2 + p_{2y}m_2 g_z S_2 - m_2 g_x A_2 S_2 - m_2 g_z A_2 C_2 - m_3 p_{3x} g_x S_{23} - \\ & m_3 p_{3x} g_z C_{23} - m_3 p_{3y} g_x C_{23} + m_3 p_{3y} g_z S_{23} - m_3 g_x A_3 S_{23} - m_3 g_x A_2 S_2 - m_3 g_z A_3 C_{23} - m_3 g_z A_2 C_2 \end{aligned} \quad (A.26)$$

$$C_3(2) = -m_3 p_{3x} g_x S_{23} - m_3 p_{3x} g_z C_{23} - m_3 p_{3y} g_x C_{23} + m_3 p_{3y} g_z S_{23} - m_3 g_x A_3 S_{23} - m_3 g_z A_3 C_{23} \quad (A.27)$$

A.4 Breakaway Model

As stated in the introduction section, breakaway systems enable under-actuated fingers to perform enveloping grasps. In the case of the *Barrett Hand*, this mechanism is implemented by a threaded shaft, a pair of Belleville spring washers and a spur gear [Townsend-2000]. In normal operation, the spur gear transmits the torque from the axis of the motor to the axis of the inner phalanx joint through the threaded shaft. However, if the inner phalanx contacts an object while the finger is closing and this object exerts a torque over the joint axis higher than a predefined threshold (τ_{break}); the spur gear is disengaged from the threaded shaft by the Belleville spring washers. Then, the motor torque is only transmitted to the outer phalanx while the inner phalanx remains mechanically jammed. The spur gear and the threaded shaft are reengaged when the finger is opening and passes through the joint position where the breakaway happened.

If an object contacts the inner phalanx at a point ${}^{k0}p_{c_i} = ({}^{k0}p_{c_i x}, {}^{k0}p_{c_i y}, {}^{k0}p_{c_i z})$ and applies a force ${}^{k0}f_{c_i}$ and a torque ${}^{k0}n_{c_i}$ on it, then the torque τ_{k2} which is transmitted to axis of joint $k2$ can be calculated by the following equation:

$$\tau_{k2} = \left({}^{k0}\hat{Z}_{k1} \times ({}^{k0}p_{c_i} - {}^{k0}p_{k1}) \right)^T {}^{k0}f_{c_i} + {}^{k0}\hat{Z}_{k1}^T {}^{k0}n_{c_i} \quad (A.28)$$

Where ${}^{k0}\hat{\mathbf{Z}}_{k1}$ are the coordinates of the Z axis of frame $\{k1\}$ with respect to frame $\{k0\}$. This vector corresponds to the third column of the transformation matrix ${}^{k0}\mathbf{T}_{k1}$, which is obtained by the direct kinematic model. ${}^{k0}\mathbf{p}_{k1}$ are the coordinates of the origin of frame $\{k1\}$ expressed in frame $\{k0\}$. This vector corresponds to the fourth column of the transformation matrix ${}^{k0}\mathbf{T}_{k1}$. If these relations are applied on (A.28), the following expression is obtained:

$$\begin{aligned} \tau_{k2} = & {}^{k0}f_{ciz} {}^{k0}p_{ciy} S(\theta_{k1}) - {}^{k0}f_{ciy} {}^{k0}p_{ciz} S(\theta_{k1}) + {}^{k0}f_{ciz} {}^{k0}p_{cix} C(\theta_{k1}) - \\ & {}^{k0}f_{cix} {}^{k0}p_{ciz} C(\theta_{k1}) - {}^{k0}f_{ciz} A_1 + {}^{k0}n_{cix} S(\theta_{k1}) - {}^{k0}n_{ciy} C(\theta_{k1}) \end{aligned} \quad (\text{A.29})$$

Therefore, the breakaway mechanism is activated when finger k is closing and the inner phalanx touches an object which generates a torque $\tau_{k2} > \tau_{break}$. The breakaway mechanism is deactivated when finger k is opening and joint $k3$ goes through the position q_{Mk_break} where the breakaway was activated. This model can be summarized by the pseudo-code algorithm presented in Table A.4.

Table A.4. Algorithm for the detection of the activation of the breakaway mechanism.

Algorithm: Breakaway Detection for finger k

Input: τ_{break} , q_{Mk} ;

Implementation:

```

01: BreakAway = false;
02: if finger  $k$  is closing
03:   if link  $k2$  touches an object
04:     Get force  $f_{ci}$  on contact point;
05:     Compute torque  $\tau_{k2}$  with equation (A.29);
06:     if  $\tau_{k2} > \tau_{break}$ 
07:       BreakAway = true;
08:        $q_{Mk\_break} = q_{Mk}$ ;
09:     end if
10:   end if
11: else
12:   if BreakAway == true
13:     if  $q_{Mk} = q_{Mk\_break}$ 
14:       BreakAway = false;
15:     end if
16:   end if
17: end if

```

Output: BreakAway

Finally, this breakaway detection model should be considered in the kinematic and dynamic models. In the case of the kinematic model, when the breakaway is activated, the angles θ_{k3} and θ_{k2} are decoupled and the relation in (A.4) is not verified. In particular, θ_{k2} remains fixed while the breakaway is activated and θ_{k3} moves freely. In addition, the transmission relation between the motor of the finger and the third joint is changed from (A.4) to the following expression:

$$\theta_{k3} = 4/375 \cdot q_{Mk} - 3/375 \cdot q_{Mk_break} \quad (\text{A.30})$$

In the case of the dynamic model, not only the decoupling between θ_{k3} and θ_{k2} should be considered, but also the fact that $\dot{\theta}_{k2}$ is null.

APPENDIX

Resumen

B. Resumen

Este apéndice contiene un resumen global de esta tesis doctoral, destacando los resultados obtenidos tanto en el campo de la seguridad en tareas de interacción humano-robot y como en el campo de la manipulación inteligente mediante manos robóticas con múltiples dedos. Además, se desarrolla un análisis detallado de estos resultados y se comparan con otras técnicas similares desarrolladas en investigaciones anteriores. Finalmente se presentan las conclusiones generales de esta tesis y se plantean líneas de trabajo futuro para la ampliación de la investigación desarrollada en esta tesis. Este anexo constituye una visión general y precisa de todas

B.1 Introducción	161
B.2 Resultados Obtenidos	164
B.2.1 Resultados en Estrategias de Seguridad para Interacción Humano-Robot	164
B.2.2 Resultados en Manipulación Robótica	167
B.3 Discusión de los Resultados.....	169
B.3.1 Discusión de los Resultados en Seguridad para Interacción Humano-Robot	169
B.3.2 Discusión de los Resultados en Manipulación Robótica	171
B.4 Conclusiones	172

las contribuciones de esta tesis y de su contenido. No obstante, en caso de que se desee profundizar en alguna de las contribuciones desarrolladas, se deberá consultar el capítulo correspondiente en el cuerpo de esta tesis

B.1 Introducción

Los robots manipuladores se utilizan cada vez más en entornos industriales [IFR-2008] debido a su precisión posicional, su repetitividad y durabilidad. Generalmente se aplican en tareas repetitivas de ensamblado/desensamblado donde todos los componentes siempre están en la misma posición y, por lo tanto, es suficiente con utilizar un control en posición del robot. Sin embargo, la aplicación de robots manipuladores que puedan reaccionar a cambios en el entorno podría mejorar la productividad de los procesos industriales actuales. De hecho, los robots manipuladores que son sensibles al contexto no necesitan estar totalmente aislados del resto de componentes de la línea de producción [Kulic-2004], reduciendo de este modo el coste del proceso de producción no sólo en términos de espacio (al eliminar las barreras de las células de trabajo actuales) sino también de tiempo (al permitir que se puedan ejecutar más tareas simultáneamente). Con el objetivo de desarrollar esta sensibilidad al contexto, los robots manipuladores están incorporando gradualmente sensores adicionales que registran información importante del entorno para adaptar su comportamiento a los cambios del entorno. Por ejemplo, se suelen instalar cámaras y sensores de fuerza [García-2009b] en el efector final del robot para determinar la trayectoria que debe seguir el robot según la evolución de las características visuales registradas por la cámara (técnicas de control visual) y las fuerzas y pares de contacto detectados por el sensor de fuerza (control de fuerza).

Esta sensibilidad al contexto de los robots manipuladores posibilita la participación de nuevos agentes durante la producción, como operadores humanos que pueden desarrollar aquellas tareas que los robots no pueden completar. De hecho, los robots y los humanos presentan características complementarias para el desarrollo de tareas comunes. Por una parte, el robot es capaz de realizar aquellas tareas que son peligrosas o extenuantes para el humano. Por otra parte, el humano es capaz de desarrollar aquellas tareas que requieren de inteligencia y destreza. Esta sinergia entre humanos y robots permite el desarrollo de tareas más flexibles y complejas que no pueden ser completadas individualmente por un robot o un humano [Green-2008]. Además, la aplicación de la interacción humano-robot en entornos industriales puede mejorar también la productividad del sistema porque el robot puede continuar su trabajo normal mientras que el operador humano está dentro de la celda de trabajo [Ebert-2005]. De este modo, las tareas industriales se pueden mejorar considerablemente al permitir que humanos y robots compartan el mismo espacio de trabajo. La transición de los estándares clásicos de seguridad industrial que consistían en aislar totalmente los robots de los operadores humanos mediante barreras físicas (como el estándar ISO-10218:1992 [ISO-1992]) hacia nuevos estándares (como el ISO-10218-1:2006 [ISO-2006] o el ISO-10218-2:2011 [ISO-2011], que todavía está en desarrollo) donde se permite la colaboración de humanos y robots en el mismo entorno de trabajo, demuestran el interés creciente de la industria en estos tipos de sistemas basados en la interacción humano-robot.

El principal objetivo de esta tesis es completar varios requisitos que son necesarios para transformar esta visión de humanos y robots trabajando juntos en una realidad. El primer requisito que se debe satisfacer es garantizar la seguridad del operador humano que colabora con el robot [Bicchi-2008]. Este requisito de seguridad es fundamental para los sistemas de interacción humano-robot donde humanos colaboran con robots industriales ya que estos robots suelen tener grandes dimensiones, suelen ser pesados y suelen trabajar a altas velocidades. Por lo tanto, cualquier colisión entre un humano y un robot de este tipo puede ser muy peligrosa para el humano y debería evitarse por completo.

El segundo requisito que se debe satisfacer es permitir la manipulación de objetos entre el operador humano y el robot manipulador [Edsinger-2007]. Una de las aplicaciones más comunes de los robots industriales es el ensamblado y desensamblado de productos. Este proceso de ensamblado/desensamblado normalmente conlleva la manipulación de distintos componentes para insertarlos o extraerlos de un lugar específico del producto. En tareas de ensamblado automático sin interacción humano-robot, estos componentes se colocan siempre en la misma posición y orientación de tal modo que es suficiente con aplicar un control en posición del extremo del robot. Cuando existe una interacción humano-robot en la tarea, la participación del humano introduce mucha incertidumbre sobre el objeto manipulado que requiere incrementar la flexibilidad de las tareas de manipulación del robot. Por ejemplo, cuando el humano entrega un componente al robot, este componente podría estar orientado de manera diferente a la configuración final en el producto. Esta situación obliga al robot a reorientar dicho componente. Otra situación típica que requiere un ajuste de la manipulación de un objeto se produce cuando el robot debe dar varios objetos a un humano pero el humano está situado cada vez en una posición diferente. En este caso, el robot debe tam-

bién orientar el objeto en la mejor configuración que permita al humano agarrarlo de la mejor manera posible.

Ambos requisitos son necesarios para el desarrollo de tareas de interacción humano-robot que incrementen la flexibilidad de los sistemas industriales compuestos por robots manipuladores. Esta tesis pretende desarrollar las técnicas necesarias para satisfacer ambos requisitos. Así, en el caso de garantizar la seguridad del humano, esta tesis propone centrarse en el desarrollo de estrategias de seguridad que eviten cualquier peligro de colisión. Ya que se pretende desarrollar tareas en las que el humano y el robot colaboren en el ensamblado de un mismo producto, estas estrategias de seguridad deberán ser capaces de registrar tanto los movimientos del robot como los movimientos del humano con el objetivo de calcular la distancia mínima que existe entre ellos y cambiar el comportamiento normal del robot en caso de que dicha distancia se encuentre por debajo de un determinado umbral de seguridad.

En el caso de la manipulación robótica de objetos, esta tesis se centra en el desarrollo de un planificador de movimientos de objetos generados mediante los dedos de una mano robótica. En los últimos años, las manos robóticas con múltiples dedos están sustituyendo poco a poco a las tradicionales pinzas robóticas [Yoshikawa-2010] gracias a dos importantes características: destreza (*dexterity*) y antropomorfismo (*anthropomorphism*) [Melchiorri-2008]. La destreza se corresponde con la capacidad de un efector final de realizar tareas de manera autónoma con una cierta complejidad mientras que el antropomorfismo se refiere a la semejanza de un efector final respecto a la mano humana en términos de tamaño, forma, consistencia y aspecto general. Ambas características aumentan la flexibilidad de las manos robóticas con múltiples dedos en comparación con las pinzas tradicionales. De hecho, las manos robóticas con múltiples dedos no sólo pueden realizar tareas de agarres de objetos sino que también pueden desarrollar tareas de manipulación que apliquen movimientos controlados a los objetos, tal como se desea realizar en esta tesis.

Las tareas de manipulación que pueden desenvolver las manos robóticas se pueden clasificar en tres tipos [Han-1998]:

- *Manipulación de objetos (Object manipulation)*: Este tipo de manipulación se basa en conseguir una determinada configuración (un conjunto posición y orientación) del objeto sin tener en cuenta la configuración de los contactos (posiciones y orientaciones relativas entre los puntos de contacto de las superficies del objeto y los dedos). Estas tareas de manipulación se implementan generalmente suponiendo que el objeto evoluciona hacia la configuración deseada manteniendo los puntos de contacto siempre fijos.
- *Ajuste del agarre (Grasp adjustment)*: Este tipo de manipulación pretende obtener una configuración de contacto determinada sin considerar la configuración del objeto. Estas tareas de manipulación se basan en cambiar los puntos de contacto con el objetivo de mejorar la calidad del agarre o generar nuevos agarres cuando los dedos alcanzan los límites cinemáticos. El principal objetivo de estas tareas es conseguir un agarre estable.

- *Manipulación diestra (Dexterous Manipulation)*: Este tipo de manipulación conlleva conseguir una configuración deseada tanto para el objeto como para los contactos. Éste es el tipo de tarea de manipulación más general ya que no sólo pretende mover el objeto a una determinada configuración sino también tiene en cuenta los cambios sobre los puntos de contacto basados en deslizamientos o rodadura.

En el caso de las tareas de interacción humano-robot, es necesario generalmente realizar tareas de manipulación que conllevan la reconfiguración de los objetos manipulados. Por ello, las técnicas basadas en el ajuste del agarre no son convenientes para estas tareas ya que no tienen en cuenta la configuración del objeto. Por otra parte, las limitaciones cinemáticas de las manos robóticas hacen que las técnicas basadas en la manipulación de objetos con puntos de contacto fijos sean difíciles de implementar ya que conllevan en muchos casos mover dichos puntos de contactos a zonas inalcanzables por los dedos [Han-1998]. Por ello, las técnicas de manipulación diestra [Okamura-2000] son la mejor solución para manipular objetos en tareas de interacción humano-robot ya que permiten que varíen los puntos de contacto. Esta posibilidad de cambiar los puntos de contacto reduce las limitaciones cinemáticas de los dedos de las manos robóticas ya que los puntos de contacto pueden variar con el objetivo de situarlos dentro de las zonas alcanzables por los dedos. Esta tesis se centra en el desarrollo de un planificador de movimiento de los dedos de una mano robótica con el objetivo de completar tareas de manipulación diestra de objetos.

B.2 Resultados Obtenidos

Esta sección se centra en enumerar y describir de manera detallada todas las contribuciones que se han desarrollado a lo largo de esta tesis doctoral. Tal como se ha descrito en la sección anterior, esta tesis pretende satisfacer dos requisitos para el desarrollo de tareas de interacción humano-robot: garantizar la seguridad del humano durante la ejecución de las tareas de interacción humano-robot y desarrollar tareas de manipulación de objetos mediante manos robóticas. Las contribuciones desarrolladas se organizan en estos dos grandes campos de investigación de esta tesis doctoral.

B.2.1 Resultados en Estrategias de Seguridad para Interacción Humano-Robot

En el campo de la seguridad humana en tareas de interacción humano-robot, esta tesis ha desarrollado las siguientes contribuciones:

- *Análisis de los sistemas de seguimiento de humanos para la interacción humano-robot en aplicaciones industriales*: Tal como se ha descrito en la [Sección B.1](#), la seguridad del humano es el primer requisito que se tiene que cumplir en los sistemas de interacción humano-robot. Como es necesario desarrollar estrategias que eviten la colisión entre el humano y el robot, es necesario localizar el humano durante el desarrollo de la tarea. Sin embargo, en tareas de interacción humano-robot donde los humanos y los robots colaboran directamente sobre los mismos objetos, no es suficiente con aplicar sistemas de localización global que sólo registran una posición 3D global del humano en el entorno. En estos ca-

sos, es necesario realizar un seguimiento de todas las partes del cuerpo humano mediante el uso de sistemas de captura de movimiento. Esta tesis ha desarrollado una comparativa de las ventajas e inconvenientes (véase la [Tabla 2.1](#)) de las diferentes tecnologías de captura de movimiento (sistemas mecánicos, sistemas magnéticos, sistemas inerciales y sistemas ópticos) con el objetivo de aplicarlas en entornos industriales donde colaboren robots y humanos en un mismo espacio de trabajo.

- *Desarrollo de un sistema de seguimiento de movimiento para el cuerpo humano:* El análisis previo ha permitido determinar que los sistemas de captura de movimiento basados en sensores inerciales son los más adecuados para tareas de interacción humano-robot en aplicaciones industriales. En particular, los sistemas inerciales presentan una mayor precisión rotacional de todas las articulaciones del cuerpo humano, una frecuencia de muestreo alta, una fácil instalación y no sufren de oclusiones como los sistemas ópticos. Estas ventajas han provocado la utilización de un sistema inercial de captura de movimiento basado en 18 sensores en esta tesis. Este sistema obtiene medidas rotacionales precisas para cada articulación del cuerpo humano y éstas son aplicadas sobre un esqueleto que representa la estructura completa del cuerpo del operador humano. Sin embargo, un análisis desarrollado para determinar la precisión de la posición global de dicho esqueleto en el entorno (véase la [Tabla 2.6](#)) ha permitido comprobar que dicha posición acumula un error que es importante para el desarrollo de tareas de interacción humano-robot. Por ello, esta tesis propone el uso de un sistema de localización adicional basado en señales UWB (*Ultra-wideband*, señales de banda ultra-ancha). Este sistema proporciona valores de precisión en el posicionamiento global del humano mucho mejores. La combinación de ambos sistemas permite obtener un sistema de seguimiento del cuerpo humano que combina las ventajas de ambos sistemas (véase la [Tabla 3.1](#)): la precisión rotacional y la alta frecuencia de muestreo del sistema inercial de captura de movimiento con la precisión de posicionamiento global del sistema UWB.
- *Tres algoritmos de fusión sensorial basados en el filtro de Bayes:* Las medidas de posición global obtenidas de los dos sistemas anteriores (sistema inercial y sistema UWB) deben combinarse para obtener una estimación precisa de la posición global del humano en el entorno con una frecuencia de muestreo alta. Esta tesis propone tres nuevos algoritmos basados en el filtro de Bayes para implementar esta fusión sensorial. El primer algoritmo se basa en un filtro de Kalman (véase la [Tabla 3.3](#)), el segundo algoritmo se basa en un filtro de partículas (véase la [Tabla 3.5](#)) y el tercer algoritmo se basa en una combinación de un filtro de Kalman y un filtro de partículas (véase la [Tabla 3.6](#)). Estos tres algoritmos se fundamentan en un principio común que consiste en el recálculo de la matriz de transformación existente entre los sistemas de coordenadas de ambos sistemas cada vez que se registra una medida del sistema UWB (véase la [Figura 3.7](#)). Este principio permite reducir el coste computacional del algoritmo con respecto a sistemas de fusión sensorial similares ([Foxlin-1996], [Foxlin-2005], [Roetenberg-2007b], [Caron-2006] y [You-2001] para sistemas

basados en filtros de Kalman; [Ababsa-2007] y [Hightower-2004] para sistemas basados en filtros de partículas) al evitar tener que ejecutar los dos pasos de los filtros (predicción y corrección) en cada iteración del algoritmo. Este principio permite también compatibilizar las frecuencias de muestreo diferentes de ambos sistemas. Además, el hecho de haber desarrollado tres algoritmos con características diferentes permite aplicarlos en situaciones distintas dependiendo de los requisitos que se tengan que cumplir en cada momento, especialmente en lo referente a tiempos de cómputo y a precisión de las estimaciones obtenidas.

- *Diseño e implementación de una jerarquía de volúmenes dinámicos para calcular la distancia humano-robot:* El esqueleto del humano obtenido a partir del sistema de seguimiento descrito anteriormente sólo contiene información sobre la longitud de los huesos del humano pero no modela el resto de dimensiones de la superficie de cada parte del cuerpo humano. Por ello, esta tesis propone cubrir el esqueleto anterior con un grupo de volúmenes envolventes (*bounding volumes*) con el objetivo de calcular de manera eficiente la distancia mínima existente entre el humano y el robot. Se propone utilizar volúmenes basados en SSLs (*Swept-Sphere Lines*) ya que permiten un mejor ajuste a las superficies del humano y el robot (véase la [Tabla 4.1](#)) que las propuestas anteriores basadas en esferas ([[Martinez-Salvador-2003](#)] y [[Balan-2006](#)]). Para ello, no sólo se cubre el esqueleto del humano con estos volúmenes sino también la estructura del robot. Para reducir el número de tests de distancia que se tienen que calcular entre cada par de volúmenes, esta tesis propone la organización de dichos volúmenes envolventes en una jerarquía de tres niveles (véanse la [Figura 4.5](#) y la [Figura 4.6](#) para el caso del humano y la [Figura 4.7](#) y la [Figura 4.8](#) para el caso del robot). El primer nivel está compuesto por AABBs (*Axis-Aligned Bounding Boxes*) que cubren totalmente el cuerpo del humano y el robot con una única caja. El segundo nivel está compuesto por AABBs que cubren los principales miembros del cuerpo del humano (brazos, piernas y tronco) y los principales componentes del brazo robótico utilizado (base, brazo y antebrazo). Finalmente, el tercer nivel está compuesto por SSLs (*Swept-Sphere Lines*) que cubren cada uno de los huesos que componen el esqueleto del humano y el esqueleto del robot. Además, los radios de las cápsulas que forman los volúmenes SSLs son incrementados de manera proporcional a la velocidad lineal del hueso que cubren, utilizando el algoritmo mostrado en la [Figura 4.9](#). Con esta evolución dinámica de los volúmenes se aumenta la seguridad del sistema ya que las dimensiones de los volúmenes SSLs no sólo representan las dimensiones de la superficie de las partes del cuerpo que cubren sino que también representan una aproximación de su desplazamiento cuando el robot y el humano desarrollan movimientos rápidos cerca el uno del otro.
- *Diseño e implementación de estrategias de seguridad para tareas de interacción humano-robot:* El algoritmo que calcula la distancia mínima entre el humano y el robot haciendo uso de la jerarquía de volúmenes anterior (véase la [Tabla 4.2](#)) se ha aplicado en esta tesis para desarrollar estrategias de seguridad en tareas de interacción humano-robot. Estas estrategias calculan la distancia humano-

robot en cada iteración del controlador del robot (o cuando se reciben medidas del sistema de seguimiento del humano) y verifican que dicha distancia esté siempre por encima de un determinado umbral de seguridad (véase la arquitectura del sistema en la [Figura 4.11](#)). Si dicha distancia se sitúa por debajo del umbral, el comportamiento normal del robot se detiene y se activa un comportamiento especial de seguridad cuyo objetivo es evitar cualquier tipo de colisión.

B.2.2 Resultados en Manipulación Robótica

En el campo de la manipulación robótica, esta tesis ha desarrollado las siguientes las siguientes contribuciones:

- *Modelado de las superficies de los dedos y el objeto a manipular como mallas de triángulos:* Los planificadores de manipulación diestra utilizados en investigaciones anteriores se aplican generalmente sobre objetos simples porque están basados en una representación muy específica de sus superficies. Por ejemplo, algunos de ellos (como [\[Trinkle-1991\]](#), [\[Zhang-1996\]](#) y [\[Xu-2008\]](#)) sólo se aplican a objeto planos. Otros necesitan un modelo paramétrico de las superficies (como [\[Yashima-2003\]](#), [\[Yashima-2004\]](#) y [\[Cherif-1999\]](#)) porque tienen que aplicar las ecuaciones de Montana [\[Montana-1988\]](#) para calcular la evolución de los puntos de contacto. Otros utilizan modelos lineales de los dedos (como [\[Saut-2006\]](#) y [\[Saut-2007\]](#)) para asegurar que sólo existe un único punto de contacto entre cada dedo y la superficie del objeto. Otros requieren dedos esféricos o elipsoidales (como [\[Cherif-1999\]](#)). Esta tesis propone un modelo más flexible donde las superficies de los dedos y del objeto a manipular se representan mediante mallas de triángulos (véase la [Figura 5.5](#)). Este modelo no sólo permite representar formas más variadas y complejas sino que también permite ajustar la precisión de las superficies mediante el aumento o la reducción del número de triángulos de la malla. Así, el coste computacional de analizar estas superficies puede ser adaptado a las necesidades de cada tarea de manipulación (es decir, al nivel de detalle en la evolución del contacto entre las superficies y a las restricciones temporales de ejecución).
- *Diseño de un modelo de evolución del contacto debido a la rodadura entre superficies de mallas de triángulos:* Esta tesis propone un nuevo modelo de evolución del contacto entre mallas de triángulos. En primer lugar, se propone un grafo de evolución del contacto (véase la [Figura 5.8](#)) para representar la transición entre pares adyacentes de primitivas que están en contacto (por ejemplo, vértices, aristas y caras de las mallas de triángulos). Cada nodo de este triángulo está asociado a una configuración de contacto específica identificada por dos primitivas en contacto (una perteneciente a la superficie del dedo y otra perteneciente a la superficie del objeto). Cada arista entre dos nodos del grafo se asocia a una transición de una configuración de contacto hacia otra configuración de contacto causada por la rodadura de una de las primitivas del nodo inicial hasta que entra en contacto con una nueva primitiva adyacente. Por ejemplo, la [Figura 5.9](#) muestra la transición de un nodo en el que coinciden una cara del dedo con una arista del objeto hacia un nuevo nodo en el que dicha cara del

dedo coincide con una cara adyacente a la arista del objeto. Esta transición se produce por una rotación de la cara del dedo alrededor de la arista del objeto hasta que dicha cara entra en contacto con una cara adyacente del objeto, generando una nueva configuración de contacto.

- *Diseño e implementación de un planificador geométrico para la manipulación diestra de objetos mediante movimientos internos de la mano:* Esta tesis desarrolla un nuevo planificador para la manipulación diestra de objetos basado en el modelo de contacto descrito en el párrafo anterior. Este planificador recibe como entrada un agarre inicial del objeto y una configuración (posición y orientación) final del objeto y genera como salida los movimientos necesarios de los dedos de una mano robótica para llevar el objeto a la configuración deseada. Este planificador se ha implementado en dos niveles (véase la [Figura 5.6](#)): planificador global y planificador local. El planificador global genera un conjunto de configuraciones intermedias del objeto realizando una interpolación no sólo de la posición sino también de la orientación entre las configuraciones inicial y final. El planificador local recibe como entrada dos configuraciones intermedias consecutivas del objeto obtenidas por el planificador global y calcula los movimientos necesarios de los dedos para generar la transición entre dichas configuraciones. Para ello, el planificador local implementa los siguientes pasos: primero calcula la configuración de contacto actual; después supone que el objeto se encuentra en la siguiente configuración intermedia y calcula todas las posibles transiciones de contacto que pueden llevar a dicha configuración intermedia a partir de la configuración anterior según lo indicado por el grafo de evolución de contactos de la [Figura 5.8](#); finalmente, aplica la pseudoinversa de la matriz Jacobiana de cada dedo para determinar qué incrementos hay que aplicar a los ángulos de las articulaciones de los dedos para generar dichas transiciones (véase el esquema de la [Figura 5.7](#)). El planificador local elegirá aquella transición que implique una variación mínima de los ángulos de las articulaciones y que sea realizable cinemáticamente.
- *Técnica de reajuste de los movimientos de los dedos para mantener la estabilidad del agarre durante el desarrollo de tareas de manipulación diestra:* La principal limitación del planificador descrito en el apartado anterior es que sólo tiene en cuenta la información geométrica de la superficie del objeto y los dedos pero no considera las fuerzas de contacto. Esta limitación hace que sea muy difícil aplicar el planificador para tareas de manipulación que conlleven una trayectoria larga del objeto ya que sucesivos movimientos de los dedos generados por el planificador local reducen drásticamente las fuerzas de contacto aplicadas hasta que el agarre es inestable. Para resolver este problema, esta tesis propone utilizar la información de presión de sensores táctiles instalados en los dedos para realizar pequeños reajustes de los movimientos de los dedos que aseguren que se mantiene una cierta presión de contacto. Esta técnica de reajuste de los dedos se realiza después de cada ejecución del planificador local y antes de pasar el control al planificador global (véase la [Figura 5.11](#)). Esta técnica verifica si la máxima presión de contacto de cada dedo está por encima de un umbral. Si se cumple esta condición, no hay ningún reajuste. Pero si la presión es-

tá por debajo del umbral, este algoritmo realiza pequeños movimientos de los dedos que van aumentando progresivamente la presión ejercida sobre el objeto. Este procedimiento de reajuste se repite iterativamente hasta que se consigue superar el umbral de presión o hasta que se supera un determinado número máximo de reajustes (establecido para evitar bucles infinitos). En el caso de que el reajuste finalice debido a que se ha superado el número máximo de reajustes, se comprobará si la presión del dedo en cuestión está por debajo de un umbral de seguridad (distinto e inferior al umbral de reajuste anterior). Si está por debajo de dicho umbral de seguridad, la tarea de manipulación se detendrá ya que el contacto entre el dedo y el objeto es muy débil y puede provocar que el agarre se vuelva inestable. Si, por el contrario, la presión es mayor que el umbral de seguridad, se devolverá el control al planificador global para que calcule la siguiente configuración intermedia del objeto. Después, dicha posición intermedia se pasará al planificador local para que calcule y aplique los movimientos necesarios de los dedos de la mano. Finalmente, se volverá a aplicar este algoritmo de reajuste. Este proceso se irá repitiendo hasta que se lleve a la configuración final o se detendrá antes cuando el agarre pueda llegar a ser inestable debido a una presión de contacto muy baja, tal como se ha descrito anteriormente.

B.3 Discusión de los Resultados

En la sección anterior se han enumerado y descrito las distintas aportaciones de esta tesis y sus fundamentos teóricos. En las siguientes secciones se realizará un análisis de los experimentos donde estas aportaciones han sido verificadas y se enumerarán las principales ventajas extraídas a partir del análisis de los resultados de dichos experimentos.

B.3.1 Discusión de los Resultados en Seguridad para Interacción Humano-Robot

La primera contribución de esta tesis cuyo correcto funcionamiento se debe verificar es la correspondiente a los tres algoritmos de fusión sensorial basados en técnicas bayesianas (filtro de Kalman, filtro de partículas y combinación de filtro de Kalman y de partículas) que se utilizan para combinar las medidas de posición global del sistema inercial de captura de movimiento y del sistema UWB. Para ello, el humano ha completado dos trayectorias distintas (una lineal y otra rectangular), 15 veces cada una. La [Figura 3.10](#) muestra una comparativa de las estimaciones de posición obtenidas por los tres algoritmos en una de las ejecuciones de la trayectoria lineal mientras que la [Figura 3.13](#) muestra la misma comparativa pero para una ejecución de la trayectoria rectangular. Los resultados de precisión de los experimentos de ambas trayectorias (véase la [Figura 3.11](#) y la [Figura 3.14](#)) demuestran que las estimaciones realizadas por el tercer algoritmo basado en la combinación de un filtro de Kalman y uno de partículas son las que tienen un menor error. Por otra parte, los resultados de tiempo de ejecución (véase la [Figura 3.12.b](#) y la [Figura 3.15.b](#)) demuestran que el algoritmo basado en el filtro de Kalman es el más eficiente computacionalmente. Por ello, dependiendo de los requisitos de la tarea desarrollada se podrá utilizar un algoritmo u otro. Así, en el caso de sistemas de interacción sencillos donde es muy importante obtener la mayor precisión

posicional posible, se utilizará el tercer algoritmo basado en la combinación de un filtro de Kalman y un filtro de partículas. Por otra parte, en el caso de sistemas complejos que se basan en algoritmos de control del robot que deben combinar medidas de distintos sensores (por ejemplo, control visual con cámaras y control de fuerza con sensores de fuerza/par) será conveniente aplicar el algoritmo más eficiente (en este caso, el primer algoritmo basado en un filtro de Kalman) para garantizar la ejecución en tiempo real de la tarea de interacción humano-robot. En los experimentos desarrollados en esta tesis, se ha utilizado el algoritmo de fusión sensorial basado en el filtro de Kalman debido a su eficiencia computacional y a la necesidad de garantizar la respuesta en tiempo real del sistema de seguridad. De hecho, este algoritmo se ha utilizado satisfactoriamente para desarrollar dos tareas de interacción humano-robot mostradas en la [Figura 3.17](#) y la [Figura 3.18](#).

Otra de las contribuciones importantes de esta tesis que debe ser verificada son las estrategias de seguridad para la interacción humano-robot basadas en la jerarquía de volúmenes dinámicos. Se han desarrollado tres tareas reales en las que un operador humano coopera con varios robots manipuladores para comprobar que las estrategias de seguridad implementadas son capaces de garantizar la seguridad del operador humano. En las tres tareas se ha implementado el mismo comportamiento de seguridad, consistente en alejar el robot en línea recta del humano con el objetivo de mantener una distancia mínima de seguridad. La primera tarea consiste en cambiar una bombilla de una lámpara. Mientras el robot manipulador desenrosca la bombilla fundida y la lleva a un contenedor de reciclaje, el humano entra en el entorno de trabajo para dejar sobre una mesa giratoria una nueva bombilla que deberá ser instalada por el robot en la lámpara (véase la [Figura 4.13](#)). Al acercarse el humano a la mesa, la distancia humano-robot calculada por la estrategia de seguridad a partir de los volúmenes envolventes llega a ser inferior al umbral de seguridad de 1 m, provocando la activación del comportamiento de seguridad que aleja al robot de su trayectoria normal para mantener la distancia de seguridad por encima de dicho umbral (véanse las gráficas de la [Figura 4.14](#) y los fotogramas de la [Figura 4.15](#)). La segunda tarea consiste en el desensamblado de un pequeño electrodoméstico (nevera portátil). En esta tarea, mientras que el robot desatornilla la tapa posterior de la nevera; el humano abre la puerta delantera para extraer el contenido de su interior. Durante esta interacción, la distancia baja del umbral de seguridad de 1 metro (véase la [Figura 4.18.a](#)) y obliga al robot a alejarse del humano para mantener esta distancia (véase la [Figura 4.17](#)). La tercera tarea consiste en el ensamblado de una estructura metálica. En esta tarea colaboran dos robots con el humano. Uno de los robots se encarga de extraer los tubos que componen la estructura de un almacén y de colocarlos en los conectores vacíos de la estructura. El segundo robot se encarga de mantener un tubo en una determinada posición para que sus extremos sean fijados mediante dos conectores por el humano. De nuevo, cuando el humano se acerca a los robots y la distancia humano-robot baja por debajo del umbral de seguridad (0.5 m para este caso) tal como se muestra en la [Figura 4.25](#), se activa el comportamiento de seguridad para el robot que está en movimiento intentado colocar un nuevo tubo (tal como se muestra en la [Figura 4.22](#)). Estos experimentos no sólo han permitido comprobar la correcta activación del comportamiento de seguridad sino que también han permitido verificar la reducción del coste computacional del cálculo de la distancia humano-robot al utilizar la jerarquía de volúmenes envolventes y reducir el

número de tests de distancia entre pares de volúmenes (tal como se muestra en los histogramas de la [Figura 4.19](#) y la [Figura 4.27](#)).

B.3.2 Discusión de los Resultados en Manipulación Robótica

Al igual que con las contribuciones en el campo de la seguridad en tareas de interacción humano-robot, también se han desarrollado varios experimentos para comprobar el correcto funcionamiento de las técnicas correspondientes a la manipulación robótica. En concreto, se han utilizado dos manos robóticas reales: la mano *Shadow* [[Shadow-2011](#)] de 5 dedos y la mano de *Barrett* [[Barrett-2011](#)] de 3 dedos.

Para comprobar el funcionamiento del planificador geométrico mostrado en la [Figura 5.6](#) (sin tener en cuenta el algoritmo de reajuste) se ha utilizado un simulador robótico software [[Anycode-2011](#)] al que se ha incorporado el modelo cinemático de la mano *Shadow* y un modelo 3D basado en mallas de triángulos de su superficie (véase la [Figura 5.12](#)). Este simulador utiliza las funciones de la librería ODE [[Smith-2010](#)] para obtener los pares de caras de las mallas del objeto y del dedo que están en contacto y para simular la interacción dinámica entre los cuerpos del objeto y de los dedos. Por otra parte, se ha desarrollado un programa en C++ en el que se ha implementado el planificador geométrico para la manipulación diestra de objetos descrito en esta tesis. Este programa recibirá los siguientes parámetros de entrada: el agarre actual del objeto (que se utilizará como configuración inicial para el planificador) y la configuración deseada del objeto (que será introducida por el usuario). Después aplicará el planificador global y el planificador local descritos en la Sección [B.2.2](#) para calcular los movimientos necesarios de los dedos de la mano *Shadow* que muevan el objeto hacia la configuración deseada indicada por el usuario. Estos movimientos serán aplicados en el simulador para comprobar la evolución de la tarea de manipulación. La [Figura 5.13](#) muestra tres diferentes tareas de manipulación aplicadas a un cubo con el objetivo de trasladarlo en distintas direcciones. La [Figura 5.14](#) muestra la utilización de este planificador para realizar dos tareas en las que se cambia la orientación de dos objetos diferentes (un prisma hexagonal y una esfera). Estas tareas han demostrado que el planificador desarrollado es capaz de realizar correctamente las tareas de manipulación planteadas. Sin embargo, la limitación de sólo poder implementar tareas de manipulación donde las configuraciones inicial y final del objeto estén muy cercanas también demuestra que es necesario ampliar el planificador geométrico con información de las fuerzas de contacto aplicadas por los dedos mediante al algoritmo de reajuste de los dedos mostrado en la [Figura 5.11](#).

Para comprobar las ventajas de aplicar conjuntamente el planificador geométrico con el algoritmo de reajuste de los dedos se ha utilizado una mano de *Barrett* equipada con 7 sensores táctiles (véase la [Figura 5.15](#)). El planificador geométrico se ha implementado en un simulador de la mano de *Barrett* basado en *Easy Java Simulations* [[Corrales-2010e](#)] y que utiliza la librería *JBullet* [[JBullet-2011](#)]. Este software devuelve una lista de incrementos de los ángulos de las articulaciones de los dedos para cada paso del planificador local. Esta lista contiene los movimientos calculados por el planificador local para mover el objeto entre cada par de configuraciones intermedias calculadas por el planificador global. El algoritmo de reajuste de los dedos (véase la [Figura 5.11](#)) se ha implementado como un programa en C++ que incorpora las librerías de control de la mano de *Barrett* y las librerías para registrar la información de presión de los sen-

sores táctiles. Después de ejecutar los movimientos de los dedos calculados por el planificador local, el algoritmo de reajuste de los dedos registra las medidas de presión de los sensores táctiles y determina si es necesario aplicar pequeños reajustes de los dedos para incrementar la presión aplicada sobre el objeto. Después de aplicar estos reajustes, el algoritmo de reajuste también verificará que las presiones de todos los dedos están por encima de los umbrales de seguridad para continuar con la siguiente configuración intermedia del objeto. En caso contrario, se parará la tarea de manipulación debido al riesgo de inestabilidad en el agarre.

Este algoritmo de reajuste se ha aplicado en un experimento de manipulación de un cilindro y en dos experimentos de manipulación de una esfera. En la manipulación del cilindro (véase la [Figura 5.17](#)) se ha constatado que el algoritmo de reajuste se ha ejecutado siempre que alguna de las presiones de los dedos estaba por debajo de los umbrales de reajuste y ha terminado definitivamente la manipulación cuando la presión del dedo 3 se ha mantenido por debajo del umbral de seguridad después de realizar el número máximo de reajustes (véase la [Figura 5.18](#)). La primera manipulación de la esfera (véase la [Figura 5.19](#)) muestra un resultado similar al del cilindro, en el que se aplican distintos reajustes hasta que la tarea se detiene definitivamente cuando la presión del dedo 2 se queda por debajo del umbral de seguridad (véase la [Figura 5.20](#)). Finalmente, la segunda manipulación de la esfera (véase la [Figura 5.21](#)) se ejecuta por completo hasta llegar a la configuración final deseada de la esfera ya que, en este caso, el algoritmo de reajuste consigue mantener siempre la presión por encima de los umbrales de seguridad (véase la [Figura 5.22](#)). Todos estos experimentos conllevan unas trayectorias de los objetos mucho más largas que las trayectorias desarrolladas con el simulador de la mano *Shadow*. Este hecho demuestra que la inclusión del algoritmo de reajuste ha mejorado las prestaciones del planificador geométrico original al tener en cuenta la información de las presiones de contacto ejercidas por los dedos sobre el objeto.

B.4 Conclusiones

El análisis realizado en el apartado anterior sobre los resultados de los distintos experimentos implementados en esta tesis ha permitido verificar la idoneidad de las técnicas desarrolladas para cumplir los dos principales requisitos establecidos inicialmente: la seguridad del humano en tareas de interacción humano-robot y el desarrollo de tareas de manipulación diestra por parte del robot mediante el uso de manos robóticas. Por una parte, las distintas tareas reales de ensamblado y desensamblado donde han colaborado un humano con varios robots manipuladores han demostrado que las estrategias de seguridad implementadas han sido capaces de activar un comportamiento de seguridad del robot cuando la distancia humano-robot estaba por debajo de un determinado umbral de seguridad. Por otra parte, las distintas tareas de manipulación desarrolladas con la mano de *Barrett* han demostrado que el planificador geométrico de manipulación diseñado, junto con la aplicación del algoritmo de reajuste de los dedos, ha sido capaz de mover el objeto agarrado por la mano hasta una determinada configuración final o de parar dicha manipulación en algún punto intermedio de la trayectoria para garantizar que el agarre no se vuelve inestable. No obstante, la investigación desarrollada en esta tesis se puede ampliar en trabajos futuros con el objetivo de desa-

rollar tareas de interacción humano-robot más flexibles y tareas de manipulación más generales.

En concreto, el sistema de seguimiento del operador humano implementado en esta tesis sólo se ha utilizado para garantizar la seguridad del humano mediante dos comportamientos de seguridad basados en la distancia humano-robot. El primer comportamiento de seguridad consiste en detener totalmente los movimientos del robot mientras que la distancia humano-robot esté por debajo del umbral de seguridad. El segundo comportamiento se basa en alejar el robot del humano en línea recta hasta que la distancia humano-robot se encuentre por encima del umbral de seguridad. Ambos comportamientos de seguridad requieren parar el funcionamiento normal del robot. Por lo tanto, no suponen una adaptación real del comportamiento del robot a los movimientos humanos sino que constituyen una estrategia temporal de seguridad. Para desarrollar tareas de interacción humano-robot más flexibles, se propone utilizar la información procedente del sistema de seguimiento del humano no sólo para asegurar una distancia humano-robot segura sino también para interpretar las acciones que está llevando a cabo el humano y cambiar el funcionamiento del robot de la manera más adecuada para responder a esas acciones del humano. Por ejemplo, si el humano se inclina hacia el robot y extiende el brazo, el robot debería interpretar que el humano desea entregarle un objeto. Además, el robot debería tener un comportamiento predefinido asociado a dicha acción, como por ejemplo, abrir la mano robótica para coger el objeto entregado por el humano. De este modo, la interacción humano-robot se volvería más intuitiva para el humano y la cooperación entre ambos mejoraría considerablemente.

Otro aspecto a desarrollar en trabajos futuros es la aplicación de las estrategias de seguridad desarrolladas en entornos industriales reales ya que hasta ahora sólo se han aplicado en un laboratorio. Para llevar a cabo esta implementación en plantas industriales reales, se deberá desarrollar un análisis para adecuar el sistema de seguridad actual a la industria concreta donde se vaya a aplicar. Este análisis podría incluir un estudio de la adecuación de las técnicas de seguridad desarrolladas a los estándares de seguridad actuales de la industria y un estudio de los procesos de producción de la industria donde la inclusión del humano pueda mejorar su productividad.

La otra técnica que se podría ampliar en esta tesis sería el planificador de manipulación diestra. Este planificador se podría mejorar añadiéndole dos nuevas características. Por una parte, el planificador actual se fundamenta en el mantenimiento constante de los contactos entre el objeto y los dedos. Para mejorarlo, se podría incorporar la posibilidad de romper puntos de contacto, separando un determinado dedo del objeto para colocarlo en otra parte de la superficie del objeto y para sustituirlo por otro dedo adyacente. La inclusión de estos nuevos modos de manipulación aumentaría la flexibilidad de la manipulación ya que permitiría mover los dedos siempre hacia zonas de contacto dentro de sus límites cinemáticos. Por otra parte, también se podría cambiar el actual planificador global basado en una interpolación de la posición y orientación del objeto por una técnica estadística que muestree configuraciones del objeto. De este modo, aquellas configuraciones intermedias no realizables por los dedos debido a limitaciones cinemáticas podrían ser sustituidas por otras configuraciones cercanas generadas aleatoriamente.

References

- [Ababsa-2007] Ababsa, F. E.; Mallem, M., "Hybrid Three-Dimensional Camera Pose Estimation Using Particle Filter Sensor Fusion". *Advanced Robotics*, 21(1) pp. 165-181, 2007.
- [Abdel-Malek-2005] Abdel-Malek, K.; Yang, J. Z., "Sweeping of an object held by a robotic end-effector". *Robotics and Computer-Integrated Manufacturing*, 21(2) pp. 159-173, Apr 2005.
- [Andriacchi-2000] Andriacchi, T. P.; Alexander, E. J., "Studies of human locomotion: past, present and future". *Journal of Biomechanics*, 33(10) pp. 1217-1224, Oct. 2000.
- [Animazoo-2011a] Animazoo, "IGS-190-M Motion Capture System". [Online: 2011a, April]. Available: <http://www.animazoo.com/motion-capture-systems/igs-190-m-motion-capture-system/>.
- [Animazoo-2011b] Animazoo, "Gypsy-7 Motion Capture System". [Online: 2011b, April]. Available: <http://www.animazoo.com/motion-capture-systems/gypsy-7-motion-capture-system/>.
- [Anycode-2011] Anycode, "Marilou Robotics Studio". [Online: 2011, May]. Available: <http://www.anycode.com/>
- [Arulampalam-2002] Arulampalam, M. S.; Maskell, S.; Gordon, N.; Clapp, T., "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking". *IEEE Transactions on Signal Processin*, 50(2) pp. 174-188, Feb. 2002.
- [Ascension-2011] Ascension, "MotionStar Motion Tracking System". [Online: 2011, April]. Available: <http://www.ascension-tech.com/>.
- [Azad-2007] Azad, P.; Asfour, T.; Dillmann, R., "Toward an Unified Representation for Imitation of Human Motion on Humanoids", In *IEEE International Conference on Robotics and Automation*, Roma, Italy, 10-14 April 2007; pp. 2558-2563.
- [Bachmann-2004] Bachmann, E. R.; Yun, X.; Peterson, C. W., "An Investigation of the Effects of Magnetic Variations on Inertial/Magnetic Orientation Sensors", In *IEEE International Conference on Robotics and Automation (ICRA 2004)*, New Orleans, LA, USA, 26 April - 1 May 2004; pp. 1115-1122.
- [Balan-2006] Balan, L.; Bone, G. M., "Real-time 3D collision avoidance method for safe human and robot coexistence", In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, October 2006; pp. 276-282.
- [Barrett-2011] Barrett, "Barrett Hand". [Online: 2011, May]. Available: <http://www.barrett.com/>.

- [Bicchi-2008] Bicchi, A.; Peshkin, M. A.; Colgate, J. E., "Safety for Physical Human-Robot Interaction". In Springer Handbook of Robotics, B. Siciliano and O. Khatib, Eds.; Springer: Heidelberg, Germany, 2008; pp. 1335-1348.
- [Birglen-2008] Birglen, L.; Laliberté, T.; Gosselin, C., "Underactuated Robotic Hands". Springer: Berlin, Germany, 2008.
- [Bompos-2007] Bompos, N. A.; Artemiadis, P. K.; Oikonomopoulos, A. S.; Kyriakopoulos, K. J., "Modeling, full identification and control of the mitsubishi PA-10 robot arm", In IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2007), ETH Zürich, Switzerland, 4-7 Sept. 2007; pp. 1-6.
- [Brock-2008] Brock, O.; Kuffner, J.; Xiaco, J., "Motion for Manipulation Tasks". In Handbook of Robotics, B. Siciliano and O. Khatib, Eds.; Springer: Heidelberg, Germany, 2008; pp. 615-645.
- [Brumitt-2000] Brumitt, B.; Meyers, B.; Krumm, J.; Kern, A.; Shafer, S., "EasyLiving: Technologies for Intelligent Environments". In Second International Symposium on HandHeld and Ubiquitous Computing (HUC 2000), P. Thomas and H.-W. Gellersen, Eds.; Springer: Heidelberg, Germany, 2000; Vol. 1927/2000, pp. 97-119.
- [Cameron-1990] Cameron, S., "Collision detection by 4-dimensional intersection testing". IEEE Transactions on Robotics and Automation, 6(3) pp. 291-302, Jun 1990.
- [Caron-2006] Caron, F.; Duflos, E.; Pomorski, D.; Vanheeghe, P., "GPS/IMU Data Fusion Using Multisensor Kalman Filtering: Introduction of Contextual Aspects". Information Fusion, 7(2) pp. 221-230, 2006.
- [Carpenter-1999] Carpenter, J.; Clifford, P.; Fearnhead, P., "Improved particle filter for nonlinear problems". IEE Proceedings Radar Sonar and Navigation, 146(1) pp. 2-7, Jan. 1999.
- [Corrales-2006] Corrales, J. A.; Torres, F.; Candelas, F. A., "Tecnologías en la Inteligencia Ambiental", In XXVII Jornadas de Automática, Almería, Spain, 2006; pp. 326-332.
- [Corrales-2007] Corrales, J. A., "Localización en Entornos Industriales Inteligentes"; Research Report University of Alicante: Alicante, July, 2007; pp. 56-70.
- [Corrales-2008a] Corrales, J. A.; Candelas, F. A.; Torres, F., "Hybrid Tracking of Human Operators using IMU/UWB Data Fusion by a Kalman Filter", In 3rd ACM/IEEE International Conference on Human Robot Interaction, Amsterdam, Netherlands, 12-15 March 2008a; pp. 193-200.
- [Corrales-2008b] Corrales, J. A.; García, G. J.; Gil, P.; Pomares, J.; Puente, S. T.; Torres, F., "Intelligent Robotic Multisensorial System to Build Metallic Structures", In 9th IFAC Workshop on Intelligent Manufacturing Systems (IM S'08), Szczecin, Poland, 2008b; pp. 189-194.

- [Corrales-2010a] Corrales, J. A.; Candelas, F. A.; Torres, F., "Sensor data integration for indoor human tracking". *Robotics and Autonomous Systems*, 58(8) pp. 931-939, Aug. 2010a.
- [Corrales-2010b] Corrales, J. A.; Candelas, F. A.; Torres, F., "Kalman Filtering for Sensor Fusion in a Human Tracking System". In *Kalman Filter*, V. Kordic, Ed. Intech: Vukovar, Croatia, 2010b; pp. 59-72.
- [Corrales-2010c] Corrales, J. A.; García, G. J.; Candelas, F. A.; Pomares, J.; Torres, F., "Safe Cooperation between Human operators and Visually Controlled Industrial Manipulators". In *Human-Robot Interaction*, D. Chugo, Ed. INTECH: Vukovar, Croatia, 2010c; pp. 203-224.
- [Corrales-2010d] Corrales, J. A.; Torres, F.; Candelas, F. A., "Safe Human-Robot Interaction based on a Hierarchy of Bounding Volumes", In *IEEE ICRA 2010 Workshop on Multimodal Human-Robot Interfaces*, Anchorage, Alaska, May 2010d; pp. 74-79.
- [Corrales-2010e] Corrales, J. A.; Jara, C. A.; Torres, F., "Modelling and Simulation of a Multi-fingered Robotic Hand for Grasping Tasks", In *11th International Conference on Control, Automation, Robotics and Vision (ICARCV 2010)*, Singapore, Singapore, 5-8 Dec. 2010e; pp. 1577-1582.
- [Corrales-2011] Corrales, J. A.; Candelas, F. A.; Torres, F., "Safe human-robot interaction based on dynamic sphere-swept line bounding volumes". *Robotics and Computer-Integrated Manufacturing*, 2011(1) pp. 177-185, Feb. 2011.
- [Cortés-2002] Cortés, J.; Siméon, T.; Laumond, J. P., "A Random Loop Generator for Planning the Motions of Closed Kinematic Chains using PRM Methods", In *IEEE International Conference on Robotics and Automation (ICRA'02)*, Washington, DC, USA, 11-15 May 2002; pp. 2141-2146.
- [Cherif-1999] Cherif, M.; Gupta, K. K., "Planning Quasi-Static Fingertip Manipulations for Reconfiguring Objects". *IEEE Transactions on Robotics and Automation*, 15(5) pp. 837-848, 1999.
- [Cherif-2001] Cherif, M.; Gupta, K. K., "Global Planning for Dexterous Reorientation of Rigid Objects: Finger Tracking with Rolling and Sliding ". *The International Journal of Robotics Research*, 20(1) pp. 57-84, 2001.
- [Denavit-1955] Denavit, J.; Hartenberg, R. S., "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices". *ASME Journal of Applied Mechanics*, 22 pp. 215-221, 1955.
- [Douc-2005] Douc, R.; Cappe, O.; Moulines, E., "Comparison of Resampling Schemes for Particle Filtering", In *4th International Symposium on Image and Signal Processing and Analysis (ISPA 2005)*, Zagreb, Croatia, 15-17 Sept. 2005; pp. 64-69.
- [Doucet-2000] Doucet, A.; Godsill, S.; Andrieu, C., "On sequential Monte Carlo sampling methods for Bayesian filtering". *Statistics and Computing*, 10(3) pp. 197-208, July 2000.

- [Doucet-2011] Doucet, A.; Johansen, A. M., "A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later". In *The Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovski, Eds.; Oxford University Press: Oxford, UK, 2011; pp. 656-704.
- [Eberly-2004] Eberly, D. H., "Game Physics". Elsevier: San Francisco, CA, USA, 2004.
- [Ebert-2005] Ebert, D.; Komuro, T.; Namiki, A.; Ishikawa, M., "Safe human-robot-coexistence: emergency-stop using a high-speed vision-chip", In *IEEE International Conference on Intelligent Robots and Systems (IROS 2005)*, Alberta, Canada, 2-6 Aug. 2005; pp. 2923 - 2928
- [Edsinger-2007] Edsinger, A.; Kemp, C. C., "Human-Robot Interaction for Cooperative Manipulation: Handing Objects to One Another", In *IEEE International Conference on Robot and Human Interactive Communication (RO-MAN 2007)*, Jeju, Korea, 26-29 Aug. 2007; pp. 1167-1172.
- [El-Sheimy-2008] El-Sheimy, N.; Hou, H.; Niu, X., "Analysis and Modeling of Inertial Sensors Using Allan Variance". *IEEE Transactions on Instrumentation and Measurement*, 57(1) pp. 140-149, Jan. 2008.
- [Ericson-2005] Ericson, C., "Real-time collision detection". Elsevier: San Francisco, CA, USA, 2005.
- [Esquembre-2004] Esquembre, F., "Easy Java Simulations: a software tool to create scientific simulations in Java". *Computer Physics Communications*, 156(2) pp. 199-204, Jan. 2004.
- [Esteves-2005] Esteves, C.; Arechavaleta, G.; Laumond, J.-P., "Motion planning for human-robot interaction in manipulation tasks", In *IEEE International Conference on Mechatronics and Automation*, Niagara Falls, Ontario, Canada, 29 Jul. - 1 Aug. 2005; pp. 1766-1771.
- [Evennou-2006] Evennou, F.; Marx, F., "Advanced Integration of WIFI and Inertial Navigation Systems for Indoor Mobile Positioning". *EURASIP Journal on Applied Signal Processing*, 2006(1) pp. 164-164, 2006.
- [Field-2011] Field, M.; Pan, Z.; Stirling, D.; Naghdy, F., "Human motion capture sensors and analysis in robotics". *Industrial Robot: An International Journal*, 38(2) pp. 163-171, 2011.
- [Fox-2003] Fox, V.; Hightower, J.; Lin, L.; Schulz, D.; Borriello, G., "Bayesian filtering for location estimation". *IEEE Pervasive Computing*, 2(3) pp. 24-33, 2003.
- [Foxlin-1996] Foxlin, E., "Inertial Head-Tracker Sensor Fusion by a Complementary Separate-Bias Kalman Filter", In *IEEE Virtual Reality Annual International Symposium*, Santa Clara, California, 1996; pp. 185-194.
- [Foxlin-1998] Foxlin, E.; Harrington, M.; Altshuler, Y., "Miniature 6-DOF Inertial System for Tracking HMDs", In *Helmet and Head-Mounted Displays III*, SPIE Aerosense98 Conference, Orlando, FL, USA, 13 April 1998; pp. 214-228.
- [Foxlin-2005] Foxlin, E., "Pedestrian Tracking with Shoe-Mounted Inertial Sensors". *IEEE Computer Graphics and Applications*, 25(6) pp. 38-46, 2005.

- [Fu-1987] Fu, K. S.; Gonzalez, R. C.; Lee, C. S. G., "Robotics: Control, Sensing, Vision and Intelligence". McGraw-Hill: New York, USA, 1987.
- [García-2007] García, G. J.; Pomares, J.; Torres, F., "A new time-independent image path tracker to guide robots using visual servoing", In IEEE Conference on Emerging Technologies and Factory Automation (ETFA 2007), Patras, Greece, 25-28 Sept. 2007; pp. 957-964.
- [García-2009a] García, G. J.; Corrales, J. A.; Pomares, J.; Candelas, F. A.; Torres, F., "Visual Servoing Path Tracking for Safe Human-Robot Interaction", In IEEE International Conference on Mechatronics, Málaga, Spain, 14-17 April 2009a; pp. 1-6.
- [García-2009b] García, G. J.; Corrales, J. A.; Pomares, J.; Torres, F., "Survey of Visual and Force/Tactile Control of Robots for Physical Interaction in Spain". *Sensors*, 9(12) pp. 9689-9733, Dec. 2009b.
- [García-2009c] García, G. J.; Pomares, J.; Torres, F., "Automatic robotic tasks in unstructured environments using an image path tracker". *Control Engineering Practice*, 17(5) pp. 597-608, May 2009c.
- [García-2010] García, G. J. "Control Visual-Fuerza Autocalibrado para Seguimiento de trayectorias en tareas cooperativas robóticas". Thesis, Universidad de Alicante, Alicante, 2010.
- [Gil-2009] Gil, P.; Pomares, J.; Puente, S. T.; Candelas, F. A.; García, G. J.; Corrales, J. A.; Torres, F., "A cooperative robotic system based on multiple sensors to construct metallic structures". *The International Journal of Advanced Manufacturing Technology*, 45(5-6) pp. 616-630, 2009.
- [Gino van den-1997] Gino van den, B., "Efficient collision detection of complex deformable models using AABB trees". *Journal of Graphics Tools*, 2(4) pp. 1-13, 1997.
- [Giremus-2004] Giremus, A.; Doucet, A.; Calmettes, V.; Tourneret, J. Y., "A Rao-Blackwellized particle filter for INS/GPS integration", In IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04), Montreal, Quebec, Canada, 2004; pp. 964-967.
- [Goodwine-1999] Goodwine, B., "Stratified Motion Planning with Application to Robotic Finger Gaiting", In 14th IFAC World Congress, Beijing, China, 1999.
- [Gordon-1993] Gordon, N. J.; Salmond, D. J.; Smith, A. F. M., "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". *IEE Proceedings F Radar and Signal Processing*, 140(2) pp. 107-113, April 1993.
- [Gottschalk-1996] Gottschalk, S.; Lin, M. C.; Manocha, D., "OBBTree: A hierarchical structure for rapid interference detection", In ACM SIGGRAPH Conference on Computer Graphics, New Orleans, LA, USA, October 1996; pp. 171-180.

- [Gray-2005] Gray, J.; Breazeal, C.; Berlin, M.; Brooks, A.; Lieberman, J., "Action parsing and goal inference using self as simulator", In IEEE International Workshop on Robots and Human Interactive Communication (ROMAN 2005), Nashville, TN, USA, 13-15 Aug. 2005; pp. 202-209.
- [Green-2008] Green, S. A.; Billingham, M.; Chen, X.; Chase, G., "Human-Robot Collaboration: A Literature Review and Augmented Reality Approach in Design". International Journal of Advanced Robotic Systems, 5(1) pp. 1-18, Nov. 2008.
- [Guibas-2000] Guibas, L. J.; Hsu, D.; Zhang, L., "A hierarchical method for real-time distance computation among moving convex bodies". Computational Geometry-Theory and Applications, 15(1-3) pp. 51-68, Feb 2000.
- [Han-1998] Han, L.; Trinkle, J. C., "Dextrous Manipulation by Rolling and Finger Gaiting", In IEEE International Conference on Robotics and Automation, Leuven, Belgium, 16 May 1998-20 May 1998 1998; pp. 730-735.
- [Hanai-2009] Hanai, M.; Sato, R.; Fukuma, S.; Mori, S.-i.; Shimozawa, T.; Hayashi, N., "Physical Motion Analysis System in Driving using Image Processing", In International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS 2009), Kanazawa, Japan, 7-9 Dec. 2009; pp. 123-126.
- [Harmati-2002] Harmati, I.; Lantos, B.; Payandeh, S., "On Fitted Stratified and Semi-Stratified Geometric Manipulation Planning with Fingertip Relocations". The International Journal of Robotics Research, 21(5-6) pp. 489-510, 2002.
- [Harmati-2004] Harmati, I. "Stratified Motion and Manipulation Planning Algorithms". Thesis, Budapest University of Technology and Economics, Budapest, Hungary, 2004.
- [Harter-2002] Harter, A.; Hopper, A.; Steggles, P.; Ward, A.; Webster, P., "The Anatomy of a Context-Aware Application". Wireless Networks, 8(2-3) pp. 187-197, March 2002.
- [Heinzmann-2003] Heinzmann, J.; Zelinsky, A., "Quantitative Safety Guarantees for Physical Human-Robot Interaction". International Journal of Robotics Research, 22(7-8) pp. 479-504, July 2003.
- [Hightower-2004] Hightower, J.; Borriello, G., "Particle Filters for Location Estimation in Ubiquitous Computing: A Case Study", In Sixth International Conference on Ubiquitous Computing (UbiComp 2004), Nottingham, England, 2004; pp. 88-106.
- [Hol-2006] Hol, J. D.; Schon, T. B.; Gustafsson, F., "On Resampling Algorithms for Particle Filters", In Nonlinear Statistical Signal Processing Workshop, Cambridge, UK, 13-15 Sept. 2006; pp. 79-82.
- [IFR-2008] IFR, "World Robotics Industrial Robots". IFR Statistical Department: Frankfurt, Germany, 2008.

- [Ijspeert-2002] Ijspeert, A. J.; Nakanishi, J.; Schaal, S., "Movement imitation with nonlinear dynamical systems in humanoid robots", In IEEE International Conference on Robotics and Automation (ICRA 2002), Washington, DC, USA, 11-15 May 2002; pp. 1398-1403.
- [Intersense-2011] Intersense, "InertiaCube3". [Online: 2011, April]. Available: <http://www.intersense.com/pages/18/11/>.
- [iPiSoft-2011] iPiSoft, "iPi Desktop Motion Capture System". [Online: 2011, April]. Available: <http://www.ipisoft.com/products.php>.
- [ISO-1992] ISO, "ISO 10218:1992 - Manipulation industrial robots -- Safety". In International Organization for Standardization: Geneva, Switzerland, 1992.
- [ISO-2006] ISO, "ISO 10218-1 - Robots and robotic devices -- Safety requirements -- Part 1: Industrial robots". In International Organization for Standardization: Geneva, Switzerland, 2006.
- [ISO-2011] ISO, "ISO 10218-2 - Robots and robotic devices -- Safety requirements -- Part 2: Industrial robot systems and integration". In International Organization for Standardization: Geneva, Switzerland, 2011.
- [Izani-2004] Izani, M.; Eshaq, A. R.; Zainuddin, N.; Razak, A., "A Study on Practical Approach of Using Motion Capture and Keyframe Animation Techniques", In Eighth International Conference on Information Visualisation (IV'04), London, UK, 14-16 Jul. 2004; pp. 849-852.
- [Janabi-Sharifi-2000] Janabi-Sharifi, F.; Hayward, V.; Chen, C. S. J., "Discrete-time adaptive windowing for velocity estimation". IEEE Transactions on Control Systems Technology, 8(6) pp. 1003-1009, 2000.
- [JBullet-2011] JBullet, "Java Port of Bullet Physics Library (JBullet)". [Online: 2011, May]. Available: <http://jbullet.advel.cz/>.
- [Jimenez-2001] Jimenez, P.; Thomas, F.; Torras, C., "3D collision detection: A survey". Computers and Graphics, 25(2) pp. 269-285, 2001.
- [Jourdan-2005] Jourdan, D. B.; Deyst, J. J., Jr.; Win, M. Z.; Roy, N., "Monte Carlo localization in dense multipath environments using UWB ranging", In IEEE International Conference on Ultra-Wideband (ICU 2005), Zurich, Switzerland, 2005; pp. 314-319.
- [Kalman-1960] Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems". Transactions of the ASME-Journal of Basic Engineering, 82(D) pp. 35-45, 1960.
- [Klosowski-1998] Klosowski, J. T.; Held, M.; Mitchell, J. S. B.; Sowizral, H.; Zikan, K., "Efficient collision detection using bounding volume hierarchies of k-DOPs". IEEE Transactions on Visualization and Computer Graphics, 4(1) pp. 21-36, 1998.
- [Kockara-2007] Kockara, S.; Halic, T.; Iqbal, K.; Bayrak, C.; Rowe, R., "Collision detection: A survey", In IEEE International Conference on Systems, Man and Cybernetics, Montreal, QC, Canada, October 2007; pp. 4046-4051.

- [Kulic-2004] Kulic, D.; Croft, E., "Safe Planning for Human-Robot Interaction", In IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, 26 Apr. - 1 May 2004; pp. 1882-1887.
- [Lafvas-2007] Lafvas, M., "Robots- In industry and beyond". In ISO Focus, International Organization for Standardization: Geneva, Switzerland, 2007; Vol. 2007, pp. 34-36.
- [Larsen-2000] Larsen, E.; Gottschalk, S.; Lin, M. C.; Manocha, D., "Fast distance queries with rectangular swept sphere volumes", In IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, April 2000; pp. 3719-3726.
- [LaValle-2001] LaValle, S. M.; Kuffner, J. J., "Rapidly-Exploring Random Trees: Progress and Prospects". In Algorithmic and Computational Robotics: New Directions, B. R. Donald, K. M. Lynch and D. Rus, Eds.; A. K. Peters, Ltd.: Natick, MA, USA, 2001; pp. 293-308.
- [Lee-2009] Lee, J. K.; Park, E. J., "Minimum-Order Kalman Filter with Vector Selector for Accurate Estimation of Human Body Orientation". IEEE Transactions on Robotics, 25(5) pp. 1196-1201, Oct. 2009.
- [Li-2000] Li, X. R.; Jilkov, V. P., "A Survey of Maneuvering Target Tracking: Dynamic Models", In SPIE Conference on Signal and Data Processing of Small Targets, Orlando, FL, USA, April 2000; pp. 1-24.
- [Li-1989] Li, Z.; Canny, J. F.; Sastry, S. S., "On Motion Planning for Dexterous Manipulation, Part I: The Problem Formulation", In IEEE International Conference on Robotics and Automation, Scottsdale, Arizona, USA, 14-19 May 1989; pp. 775-780.
- [Lim-1999] Lim, H.-o.; Tanie, K., "Collision-Tolerant Control of Human-Friendly Robot with Viscoelastic Trunk". IEEE/ASME Transactions on Mechatronics, 4(4) pp. 417-427, 1999.
- [Lin-1991] Lin, M. C.; Canny, J. F., "A fast algorithm for incremental distance calculation", In IEEE International Conference on Robotics and Automation, Sacramento, Ca, USA, April 1991; pp. 1008-1014.
- [Linden-2009] Linden, J. v. d.; Schoonderwaldt, E.; Bird, J., "Towards a real-time system for teaching novices correct violin bowing technique", In IEEE International Workshop on Haptic Audio visual Environments and Games (HAVE 2009), Lecco, Italy, 7-8 Nov. 2009; pp. 81-86.
- [Liu-2007] Liu, H.; Darabi, H.; Banerjee, P.; Liu, J., "Survey of Wireless Indoor Positioning Techniques and Systems". IEEE Transactions on Systems, Man and Cybernetics-Part C: Applications and Reviews, 37(6) pp. 1067-1080, Nov. 2007.
- [Lu-2005] Lu, S.; Chung, J. H., "Collision detection enabled weighted path planning: a wrist and base force/torque sensors approach", In International Conference on Advanced Robotics (ICAR 2005), Seattle, WA, USA, 18-20 July 2005; pp. 165-170.

- [Luca-2000] Luca, A. D., "Feedforward/Feedback Laws for the Control of Flexible Robots", In IEEE International Conference on Robotics and Automation (ICRA 2000), San Francisco, CA, USA, 24-28 April 2000; pp. 233-240.
- [Maiteh-1999] Maiteh, B. Y.; Blackmore, D.; Abdel-Malek, L.; Leu, M. C., "Swept-volume computation for machining simulation and Virtual Reality application". *Journal of Materials Processing & Manufacturing Science*, 7(4) pp. 380-390, April 1999.
- [Martinez-Salvador-2003] Martinez-Salvador, B.; Perez-Francisco, M.; Del Pobil, A. P., "Collision detection between robot arms and people". *Journal of Intelligent & Robotic Systems*, 38(1) pp. 105-119, Sep 2003.
- [Melchiorri-2008] Melchiorri, C.; Kaneko, M., "Robot Hands". In *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds.; Springer: Heidelberg, Germany, 2008.
- [Meredith-2001] Meredith, M.; Maddock, S., "Motion Capture File Formats Explained"; Research Report University of Sheffield: Sheffield, UK, 2001; pp. 1-36.
- [Microsoft-2011] Microsoft, "Xbox Kinect Motion Capture System". [Online: 2011, April]. Available: <http://research.microsoft.com/en-us/projects/vrkinect/>.
- [Miller-2004] Miller, N.; Jenkins, O. C.; Kallmann, M.; Mataric, M. J., "Motion capture from inertial sensing for untethered humanoid teleoperation", In 4th IEEE/RAS International Conference on Humanoid Robots, Santa Monica, Los Angeles, CA, USA, 10-12 Nov. 2004; pp. 547-565.
- [Mirtich-1998] Mirtich, B., "V-Clip: Fast and robust polyhedral collision detection". *ACM Transactions on Graphics*, 17(3) pp. 177-208, 1998.
- [Mitobe-2006] Mitobe, K.; Kaiga, T.; Yukawa, T.; Miura, T.; Tamamoto, H.; Rodgers, A.; Yoshimura, N., "Development of a motion capture system for a hand using a magnetic three dimensional position sensor", In *ACM International Conference and Exhibition on Computer Graphics and Interactive Techniques (SIGGRAPH)*, Boston, MA, USA, 30 Jul. - 3 Aug. 2006; pp. 102.
- [Moller-1997] Moller, T., "A Fast Triangle-Triangle Intersection Test". *Journal of Graphics, GPU and Game Tools*, 2(2) pp. 25-30, 1997.
- [Montana-1988] Montana, D. J., "The Kinematics of Contact and Grasp". *The International Journal of Robotics Research*, 7(3) pp. 17-32, 1988.
- [Montecillo-Puente-2010] Montecillo-Puente, F.-J.; Sreenivasa, M. N.; Laumond, J.-P., "On real-time whole-body human to humanoid motion transfer", In 7th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2010), Funchal, Madeira, Portugal, 15-18 June 2010; pp. 22-31.
- [MotionAnalysis-2011] MotionAnalysis, "Raptor RealTime MoCap System". [Online: 2011, April]. Available: <http://www.motionanalysis.com/>.
- [Musso-2001] Musso, C.; Oudjane, N.; Gland, F. L., "Improving Regularized Particle Filters". In *Sequential Monte Carlo Methods in Practice (Statistics for Engineering and Information Science)*, 1 ed.; A. Doucet, N. de Freitas and N. Gordon, Eds.; Springer: New York, 2001; pp. 247-271.

- [Okamura-2000] Okamura, A. M.; Smaby, N.; Cutkosky, M. R., "An Overview of Dexterous Manipulation", In IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 24-28 Apr. 2000 2000; pp. 255-262.
- [Ong-2001] Ong, C. J.; Gilbert, E. G., "Fast versions of the Gilbert-Johnson-Keerthi distance algorithm: Additional results and comparisons". IEEE Transactions on Robotics and Automation, 17(4) pp. 531-539, Aug 2001.
- [Onishi-2006] Onishi, M.; Odashima, T.; Luo, Z.; Hosoe, S., "An immersion-type 3D dynamic simulation environment for developing human interactive robot systems". Systems and Computers in Japan, 37(10) pp. 47-57, Sept. 2006.
- [PhaseSpace-2011] PhaseSpace, "PhaseSpace Impulse System". [Online: 2011, April]. Available: <http://www.phasespace.com/productsMain.html>.
- [Polhemus-2011] Polhemus, "Liberty Motion Tracking System". [Online: 2011, April]. Available: http://www.polhemus.com/?page=Motion_Liberty.
- [Pomares-2009] Pomares, J.; Gil, P.; Corrales, J. A.; García, G. J.; Puente, S. T.; Torres, F., "Using Visual and Force Information in Robot-Robot Cooperation to Build Metallic Structures". Journal of Automation, Mobile Robotics and Intelligent Systems, 3(3) pp. 96-101, 2009.
- [Pomares-2010] Pomares, J.; Candelas, F. A.; Torres, F.; Corrales, J. A.; García, G. J., "Safe Human-Robot Cooperation Based on an Adaptive Time-Independent Image Path Tracker". International Journal of Innovative Computing, Information and Control, 6(9) pp. 3819-3842, Sept. 2010.
- [Priyantha-2005] Priyantha, N. B. "The Cricket Indoor Location System". Thesis, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, 2005.
- [Rahman-2011] Rahman, M. M.; Mitobe, K.; Suzuki, M.; Takano, C.; Yoshimura, N., "Analysis of dexterous finger movement for piano education using motion capture system". International Journal of Science and Technology Education Research, 2(2) pp. 22-31, Feb. 2011.
- [Redon-2000] Redon, S.; Kheddar, A.; Coquillart, S., "An algebraic solution to the problem of collision detection for rigid polyhedral objects", In IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, April 2000; pp. 3733-3738.
- [Redon-2005] Redon, S.; Lin, M. C.; Manocha, D.; Kim, Y. J., "Fast continuous collision detection for articulated models". Journal of Computing and Information Science in Engineering, 5(2) pp. 126-137, Jun. 2005.
- [Ribo-2002] Ribo, M.; Lang, P.; Ganster, H.; Brandner, M.; Stock, C.; Pinz, A., "Hybrid Tracking for Outdoor Augmented Reality Applications". IEEE Computer Graphics and Applications, 22(6) pp. 54-63, 2002.
- [Ribo-2004] Ribo, M.; Brandner, M.; Pinz, A., "A Flexible Software Architecture for Hybrid Tracking". Journal of Robotic Systems, 21(2) pp. 53-62, Jan. 2004 2004.
- [Roetenberg-2006] Roetenberg, D. "Inertial and Magnetic Sensing of Human Motion". Thesis, University of Twente, Enschede, The Netherlands, 2006.

- [Roetenberg-2007a] Roetenberg, D.; Baten, C. T. M.; Veltink, P. H., "Estimating Body Segment Orientation by Applying Inertial and Magnetic Sensing Near Ferromagnetic Materials". IEEE Transactions on Neural Systems and Rehabilitation Engineering, 15(3) pp. 469-471, Sept. 2007a.
- [Roetenberg-2007b] Roetenberg, D.; Slycke, P. J.; Veltink, P. H., "Ambulatory Position and Orientation Tracking Fusing Magnetic and Inertial Sensing". IEEE Transactions on Biomedical Engineering, 54(5) pp. 883-890, 2007b.
- [Rus-1999] Rus, D., "In-Hand Dexterous Manipulation of Piecewise-Smooth 3-D Objects". The International Journal of Robotics Research, 18(4) pp. 355-381, 1999.
- [Sarcos-2011] Sarcos, "Sarcos Sensuit". [Online: 2011, April]. Available: http://www.sarcos.com/telepic_sensuit.html.
- [Saut-2006] Saut, J.-P.; Sahbani, A.; Perdereau, V., "A Global Approach for Dexterous Manipulation Planning Using Paths in n-fingers Grasp Subspace", In IEEE International Conference on Automation, Robotics and Vision (ICARCV'06), Singapore, Singapore, 5-8 Dec. 2006; pp. 1-6.
- [Saut-2007] Saut, J.-P.; Sahbani, A.; El-Khoury, S.; Perdereau, V., "Dexterous Manipulation Planning Using Probabilistic Roadmaps in Continuous Grasp Subspaces", In IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 2007; pp. 2907-2912.
- [Saut-2011] Saut, J.-P.; Sahbani, A.; Perdereau, V., "Generic Motion Planner for Robot Multi-fingered Manipulation". Advanced Robotics, 25(1-2) pp. 23-46, 2011.
- [Schneider-2003] Schneider, P.; Eberly, D. H., "Geometric Tools for Computer Graphics". Elsevier: San Francisco, CA, USA, 2003.
- [Schon-2007] Schon, T. B.; Tornqvist, D.; Gustafsson, F., "Fast Particle Filters for Multi-Rate Sensors", In European Signal Processing Conference (EUSIPCO 2007), Poznan, Poland, 2007.
- [Schwarzer-2005] Schwarzer, F.; Saba, M.; Latombe, J. C., "Adaptive dynamic collision checking for single and multiple articulated robots in complex environments". IEEE Transactions on Robotics, 21(3) pp. 338-353, Jun 2005.
- [Schweikard-1991] Schweikard, A., "Polynomial-time collision detection for manipulator paths specified by joint motions". IEEE Transactions on Robotics and Automation, 7(6) pp. 865-870, Dec 1991.
- [Shadow-2011] Shadow, "Shadow Hand". [Online: 2011, May]. Available: <http://www.shadowrobot.com>.
- [Shoemake-1985] Shoemake, K., "Animating Rotation with Quaternion Curves". ACM SIGGRAPH Computer Graphics, 19(3) pp. 245-254, July 1985.
- [Shotton-2011] Shotton, J.; Fitzgibbon, A.; Cook, M.; Sharp, T.; Finocchio, M.; Moore, R.; Kipman, A.; Blake, A., "Real-Time Human Pose Recognition in Parts from a Single Depth Image", In IEEE Computer Vision and Pattern Recognition Conference (CVPR 2011), Colorado Springs, CO, USA, 21-23 June 2011.

- [Simon-2001] Simon, D., "Kalman Filtering". Embedded Systems Programming, 14(6) pp. 72-79, June 2001.
- [Smith-2011] Smith, R., "Open Dynamics Engine (ODE)". [Online: 2011, May]. Available: <http://opende.sourceforge.net/>.
- [Steinbach-2006] Steinbach, K.; Kuffner, J.; Asfour, T.; Dillmann, R., "Efficient collision and self-collision detection for humanoids based on sphere trees hierarchies", In 6th IEEE-RAS International Conference on Humanoid Robots (HUMANOIDS'06), Genoa, Italy, December 2006; pp. 560-566.
- [Tao-2008] Tao, Y.; Hu, H., "A Novel Sensing and Data Fusion System for 3-D Arm Motion Tracking in Telerehabilitation". IEEE Transactions on Instrumentation and Measurement, 57(5) pp. 1029-1040, May 2008.
- [Thrun-2005] Thrun, S.; Burgard, W.; Fox, D., "Probabilistic Robotics". MIT Press: Cambridge, MA, USA, 2005.
- [Torres-Solis-2010] Torres-Solis, J.; Falk, T. H.; Chau, T., "A Review of Indoor Localization Technologies: towards Navigational Assistance for Topographical Disorientation". In Ambient Intelligence, F. J. V. Molina, Ed. InTech: Vukovar, Croatia, 2010; pp. 51-84.
- [Townsend-2000] Townsend, W., "The BarrettHand grasper - programmably flexible part handling and assembly". Industrial Robot: An International Journal, 27(3) pp. 181-188, 2000.
- [Trinkle-1991] Trinkle, J. C.; Hunter, J. J., "A Framework for Planning Dexterous Manipulation", In IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, 09-11 Apr. 1991; pp. 1245-1251.
- [Ubisense-2011] Ubisense, "Ubisense Real-Time Location System". [Online: 2011, April]. Available: www.ubisense.net.
- [Vass-1999] Vass, G.; Payandeh, S.; Lantos, B., "Object Manipulation with Dexterous Robotic Hand Using Simulated Annealing". Periodica Polytechnica Ser. El. Eng., 43(3) pp. 199-206, 1999.
- [Vass-2005] Vass, G. "Object Manipulation Planning for Dexterous Robot Systems". Thesis, Budapest University of Technology and Economics, Budapest, Hungary, 2005.
- [Vicon-2011] Vicon, "Vicon MX system". [Online: 2011, April]. Available: <http://www.vicon.com/products/viconmx.html>.
- [Wan-2010] Wan, S.; Foxlin, E., "Improved Pedestrian Navigation Based on Drift-Reduced MEMS IMU Chip", In Institute of Navigation - International Technical Meeting 2010 (ITM 2010), San Diego, CA, USA, 25-27 Jan. 2010; pp. 390-399.
- [Want-1992] Want, R.; Hopper, A.; Falcao, V.; Gibbons, J., "The active badge location system". ACM Transactions on Information Systems (TOIS), 10(1) pp. 91-102, Jan. 1992.

- [Watt-1992] Watt, A.; Watt, M., "Advanced Animation and Rendering Techniques". Addison-Wesley: New York, New York, USA, 1992.
- [Welch-2002] Welch, G.; Foxlin, E., "Motion Tracking: No Silver Bullet, but a respectable arsenal". IEEE Computer Graphics and Applications, 22(6) pp. 24-38, 2002.
- [Wie-2004] Wie, Y.; Goodwine, B., "Stratified Motion Planning on Nonsmooth Domains with Robotic Applications". IEEE Transactions on Robotics and Automation, 20(1) pp. 128-132, 2004.
- [Xiao-2001] Xiao, J.; Ji, X., "Automatic Generation of High-Level Contact State Space". The International Journal of Robotics Research, 20(7) pp. 584-606, July 2001.
- [Xsens-2011a] Xsens, "Xsens MTx IMU". [Online: 2011a, April]. Available: <http://www.xsens.com/en/general/mtx>.
- [Xsens-2011b] Xsens, "Xsens MVN Motion Capture Suit". [Online: 2011b, April]. Available: <http://www.xsens.com/en/general/mvn>.
- [Xu-2006] Xu, J.; Lou, Y.; Li, Z., "Hybrid Automaton: A Better Model of Finger Gaits", In IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 2006; pp. 4628 - 4633
- [Xu-2007a] Xu, J.; Koo, T. J.; Li, Z., "Finger Gaits Planning for Multifingered Manipulation", In IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 2007a; pp. 2932 - 2937
- [Xu-2007b] Xu, J.; Liu, D. K.; Fang, G., "An efficient method for collision detection and distance queries in a robotic bridge maintenance system". In Robotic Welding, Intelligence and Automation, T.-J. Tarn, S.-B. Chen and C. Zhou, Eds.; Springer Berlin Heidelberg: New York, 2007b; Vol. 362, pp. 71-82.
- [Xu-2008] Xu, J.; Li, Z., "A Kinematic Model of Finger Gaits by Multifingered Hand as Hybrid Automaton". IEEE Transactions on Automation Science and Engineering, 5(3) pp. 467-479, Jul. 2008.
- [Xu-2010] Xu, J.; Koo, T.-K. J.; Li, Z., "Sampling-based Finger Gaits Planning for Multifingered Robotic Hand". Autonomous Robots, 28(4) pp. 385-402, 2010.
- [Yamada-1997] Yamada, Y.; Hirasawa, Y.; Huang, S.; Umetani, Y.; Suita, K., "Human-Robot Contact in the Safeguarding Space". IEEE/ASME Transactions on Mechatronics, 2(4) pp. 230-236, Dec. 1997.
- [Yashima-2002] Yashima, M.; Yamaguchi, H., "Dynamic Motion Planning Whole Arm Grasp Systems Based on Switching Contact Modes", In IEEE International Conference on Robotics and Automation, Washington, DC, USA, 2002; pp. 2492-2499.
- [Yashima-2003] Yashima, M.; Shiina, Y.; Yamaguchi, H., "Randomized Manipulation Planning for a Multi-Fingered Hand by Switching Contact Modes", In IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 2003; pp. 2689-2694.

- [Yashima-2004] Yashima, M., "Manipulation Planning for Object Re-orientation Based on Randomized Techniques", In IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, 26 Apr. - 1 May 2004 2004; pp. 1245-1251.
- [Yoshikawa-2010] Yoshikawa, T., "Multifingered robot hands: Control for grasping and manipulation". Annual Reviews in Control, 34(2) pp. 199-208, Dec. 2010.
- [You-1999] You, S.; Neumann, U.; Azuma, R., "Orientation Tracking for Outdoor Augmented Reality Registration". IEEE Computer Graphics and Applications, 19(6) pp., 1999.
- [You-2001] You, S.; Neumann, U., "Fusion of Vision and Gyro Tracking for Robust Augmented Reality Registration", In IEEE Virtual Reality 2001 Conference, Yokohama, Japan, 13/03/2001-17/03/2001 2001; pp. 71-78.
- [Zhang-1996] Zhang, H.; Tanie, K.; Maekawa, H., "Dextrous Manipulation Planning by Grasp Transformation", In IEEE International Conference on Robotics and Automation, Minneapolis, MN, USA, 22-28 Apr. 1996 1996; pp. 3055-3060.
- [Zhang-2009] Zhang, Z.; Huang, Z.; Wu, J., "Hierarchical Information Fusion for Human Upper Limb Motion Capture", In 12th International Conference on Information Fusion (FUSION'09), Seattle, WA, USA, 6-9 July 2009; pp. 1704-1711.

Reunido el Tribunal que suscribe en el día de la fecha acordó otorgar, por
a la Tesis Doctoral de Don/Dña. Juan Antonio Corrales Ramón la calificación de
..... .

Alicante, de de

El Secretario,

El Presidente,

Universitat d'Alacant
Universidad de Alicante

**UNIVERSIDAD DE ALICANTE
CEDIP**

La presente Tesis de D. _____ ha sido
registrada con el nº _____ del registro de entrada correspondiente.

Alicante ____ de _____ de _____

El Encargado del Registro,

La defensa de la tesis doctoral realizada por D/D^a Juan Antoinio Corrales Ramón se ha realizado en las siguientes lenguas: y, lo que unido al cumplimiento del resto de requisitos establecidos en la Normativa propia de la UA le otorga la mención de “Doctor Europeo”.

Alicante, de de

EL SECRETARIO

EL PRESIDENTE



Universitat d'Alacant
Universidad de Alicante

