73rd International Astronautical Congress (IAC), Paris, France, 18-22 September 2022.

IAC-22,D1,6,x70297

Path generation and control of humanoid robots during extravehicular activities

J. L. Ramón[a]*, R. Calvo, A. Trujillo, J. Pomares, L. Felicetti[b]

[a] University of Alicante, San Vicente del Raspeig, Alicante, 03690, Spain, jl.ramon@ua.es, ramon1999cg@gmail.com, adrii.trujillo@gmail.com, jpomares@ua.es
[b] Cranfield University, Cranfield, MK43 0AL, United Kingdom, Leonard.Felicetti@cranfield.ac.uk
* Corresponding Author

Abstract

This paper proposes and investigates strategies that can be used to plan the motion and control of humanoid robots in some elementary tasks that characterize extravehicular activities. The humanoid robot taken into account is a torso with two arms and two grippers at their extremities. This study addresses the problem of robot motion on the complex system of handrails and handles that characterize the International Space Station. Such a complex task has been divided into two elementary sub-tasks: motion planning and tracking the planned trajectories. First, an optimization procedure is presented to plan and coordinate the robot's arms motions and graspers to achieve the desired location using handrails. Then, a low-level controller is used to guarantee that the robots' actuators can follow these previously generated trajectories. Simulation results assess the applicability of the proposed strategy in different typical operations that potentially can be performed in an extravehicular activity scenario.
Keywords: space robot, humanoid robot, robot control, path planning, optimization.

Acronyms/Abbreviations

International Space Station (ISS), Robot Operating System (ROS).

1. Introduction

Future and under-development missions will require extensive use of on-orbit assembling and manufacturing to build new commercial, research and exploration infrastructures in space. On-orbit assembling involves complex tasks with strict requirements in terms of the operations' accuracy, dexterity, reliability, and safety. The utilization of astronauts performing extravehicular activities might still be considered a viable option, but it is challenged by numerous technical and technological limitations. The utilization of autonomous robots is indeed a preferable option in the future, especially for tasks where operations are repetitive, structured, and standardized. In some cases, robots will be expected to operate with tools and in an environment strongly characterized by human presence. Workshop tools, screwdrivers, brackets and pliers are made to be easily used by astronauts during their extravehicular activities. Stations such as the ISS are made by modules which have numerous handles and handrails that allow for safe movements of astronauts around them. For this reason, humanoid robots are a preferable option over other kinds of robotic systems in such kinds of scenarios. This paper addresses the problem of planning and control of a humanoid robot which is supposed to crawl itself towards a desired location by grasping handrails outside the modules of the ISS.

The paper outlines the definition of the optimal path planning problem and the tracking controller that can be used for such kind of robotic systems by considering that the system would be weightless and operating in an almost floating and frictionless condition. The presented research focuses on the analysis of trajectories, torques and forces to be provided for performing on-orbit operations. Simulations, carried on in ROS/Gazebo environment, show that the overall control architecture (trajectory optimizer + controller) is sufficiently robust and allows for complex and articulated motion of the humanoid robot in an eventual extravehicular activity outside the ISS in free-floating conditions.

Some of the previous works on path planning use an optimization-based trajectory planning method for free-flying spacecraft [1] and for free-flying space manipulators [2][3] that embed the multibody dynamics in free floating conditions within their formulation. Other approaches use a two-step approach: first, to solve the kinematic path planning problem and then to compensate for the robot unbalancement by correcting the pose of the robot by considering the centroid and contact dynamics [4][5]. In this paper, a non-linear programming solver is used to generate the sequence of grasping trajectories for the two arms that allow the robot to reach a given desired location on the ISS.

Once the trajectories have been generated, a controller must be designed to perform the tracking of these trajectories. A survey of the main approaches for modeling and controlling free-floating manipulators is given in [6]. A systematical approach for forward and inverse kinematics of free-floating space-robots is presented in [7] based on screw theory without

employing inertial tensors. The presented path planning approach provides the desired trajectory to be tracked in the Cartesian space. Therefore, a task-space controller should be defined. The work described in [8] presents a task-space control of a free-floating space robot. The controller is developed by transforming the equations of motion into the task-space. A computed-torque inner-feedback loop is then developed utilizing the pseudo-inverse solution that minimizes the norm of the actuator torques. Another method is proposed to minimize the base disturbance during the visual servoing process in [9]. In this case, an offline optimal control method is applied to achieve the optimization, and a pose planning method is presented to achieve a second-order continuity of quaternion. A common control of the spacecraft base and manipulator under structural disturbances and system uncertainties is also presented in [10]. Additionally, approaches like the one described in [11] can be used to compensate for the target motion during the visual guidance. In this case, the target motion is predicted and included as the velocity feedforward.

In this paper, a Cartesian controller that takes into account the humanoid robot dynamics and the free-floating conditions are presented for the tracking of the previously generated trajectories.

The remaining part of the paper is divided into the following sections. Section 2 develops and demonstrates the stability of the proposed controller for the tracking of the trajectories generated by the path planning method. The motion planning problem and the nonlinear programming formulation used to solve the problem are described in Section 3. Simulation results, showing the applicability and robustness of the proposed approach in selected test case scenarios, are described and commented in Section 4. Concluding remarks and future areas of development are presented in Section 5.

2. Tracking trajectories

This section describes the controller used for the tracking of the trajectories generated by the path planning method that will be described in the next section. The equations of motion that characterize the system dynamics are described in Section 2.1. Then, the controller formulation is described in Section 2.2.

2.1 System dynamics and kinematics

Fig. 1 shows the ROS/Gazebo simulation of the humanoid robot and the ISS.

Fig. 1. Humanoid robot grasping a ISS handrail

The humanoid robot configuration can be defined by the vector $\boldsymbol{\epsilon} = [\boldsymbol{r}^T, \boldsymbol{\theta}^T, \boldsymbol{q}_1^T, \boldsymbol{q}_2^T]^T$, where $\boldsymbol{\theta}$ contais the yaw, pitch, roll Euler angles representing the orientation of the torso body frame B with respect the inertial coordinate frame. The robot dynamics provides a mathematical function that relates the robot force and torques and the system accelerations. The main body acceleration is defined as $\dot{\boldsymbol{v}}_b = [\ddot{\boldsymbol{r}}^T, \dot{\boldsymbol{\omega}}^T]^T \in \Re^6$ with respect to the inertia coordinate frame. This vector represents both the linear and angular acceleration of the robot torso. Additionally, the joint accelerations are $\ddot{\boldsymbol{q}}_1$ and $\ddot{\boldsymbol{q}}_2$ for both manipulators. The system dynamics for both manipulators can be combined in the following equation:

$$\boldsymbol{u} - \boldsymbol{u}_e = \begin{bmatrix} \boldsymbol{M}_{bb} & \boldsymbol{M}_{b1} & \boldsymbol{M}_{b2} \\ \boldsymbol{M}_{b1}^T & \boldsymbol{M}_{11} & 0 \\ \boldsymbol{M}_{b2}^T & 0 & \boldsymbol{M}_{22} \end{bmatrix} \ddot{\boldsymbol{\epsilon}} + \begin{bmatrix} \boldsymbol{c}_b \\ \boldsymbol{c}_1 \\ \boldsymbol{c}_2 \end{bmatrix} \tag{1}$$

where $\boldsymbol{u}$ is the input vector, $\boldsymbol{M}_{bb} \in \Re^{6\times6}$ is the inertia matrix of the robot torso, $\boldsymbol{M}_{b1} \in \Re^{6\times n}$ is the coupled inertia matrix of the robot torso and the left robot arm, $\boldsymbol{M}_{11} \in \Re^{n\times n}$ is the inertia matrix of the left arm, $\boldsymbol{M}_{b2} \in \Re^{6\times n}$ is the coupled inertia matrix of the robot torso and the right arm, $\boldsymbol{M}_{22} \in \Re^{n\times n}$ is the inertia matrix of the right arm; $\boldsymbol{c}_b$, $\boldsymbol{c}_1$, and $\boldsymbol{c}_2 \in \Re^6$ are a velocity/displacement-dependent, nonlinear terms for the body, left and right arm, respectively.

The dynamic equations for only one of the two robot arms (with $\ddot{\boldsymbol{q}}$ as the joint accelerations) can be represented as:

$$\boldsymbol{M}^*\ddot{\boldsymbol{q}} + \boldsymbol{C}^* = \boldsymbol{u}^* \tag{2}$$

where $\boldsymbol{M}^* \in \Re^{n\times n}$ is the generalized inertia matrix, $\boldsymbol{C}^* \in \Re^n$ is the generalized Coriolis and centrifugal vector for the manipulator arm, defined explicitly as:

$$\boldsymbol{M}^* = \boldsymbol{M}_{ii} - \boldsymbol{M}_{bi}^T \boldsymbol{M}_{bb}^{-1} \boldsymbol{M}_{bi} \tag{3}$$

$$\boldsymbol{C}^* = \boldsymbol{c}_i - \boldsymbol{M}_{bi}^T \boldsymbol{M}_{bb}^{-1} \boldsymbol{c}_b \tag{4}$$

In this last equation, the subindex i represents the manipulator i considered in the dynamic equation (i = 1, 2). The linear and angular momenta of the system $\left[\boldsymbol{\ell}^{\mathrm{T}}, \boldsymbol{\Psi}^{\mathrm{T}}\right]^{\mathrm{T}} \in \Re^6$ are defined as:

$$\begin{bmatrix} \boldsymbol{\ell} \\ \boldsymbol{\Psi} \end{bmatrix} = \boldsymbol{M}_{bb}\boldsymbol{v}_b^I + \boldsymbol{M}_{bi}\dot{\boldsymbol{q}} \tag{5}$$

where $\dot{\boldsymbol{q}} \in \Re^n$ represents the joint speeds of the arm. The relationship between these joint speeds and the corresponding end-effector's absolute linear and angular velocities can be obtained by computing the robot manipulator Jacobian, $\boldsymbol{J}_m \in \Re^{6\times n}$, and the Jacobian of the robot torso, $\boldsymbol{J}_b \in \Re^{6\times 6}$, using the following equation:

$$\dot{\boldsymbol{p}} = \boldsymbol{J}_m\dot{\boldsymbol{q}} + \boldsymbol{J}_b\boldsymbol{v}_b \tag{6}$$

where $\dot{\boldsymbol{p}} \in \Re^6$ is the linear and angular velocity of the robot manipulator end-effector in the inertial frame. Using Equation (5) and (6) it is possible to derive the value of the Generalized Jacobian Matrix, $\boldsymbol{J}_g$. This matrix can be used to derive the relationship between the joint velocities and the end-effector velocities taking into account the free-floating dynamics. The following relationship can be obtained:

$$\dot{\boldsymbol{p}} = \boldsymbol{J}_g\dot{\boldsymbol{q}} + \boldsymbol{J}_b\boldsymbol{M}_{bb}^{-1}\begin{bmatrix} \boldsymbol{\ell} \\ \boldsymbol{\Psi} \end{bmatrix} \tag{7}$$

where the so-called Generalized Jacobian Matrix can be obtained by using the following expression:

$$\boldsymbol{J}_g = \boldsymbol{J}_m - \boldsymbol{J}_b\boldsymbol{M}_{bb}^{-1}\boldsymbol{M}_{bm} \tag{8}$$

In this paper we want to derive an acceleration-based controller. Therefore, it is necessary to define the equations that define the relations between the joint accelerations and the end-effector accelerations. This can be obtained from Equation (7):

$$\ddot{\boldsymbol{p}} = \boldsymbol{J}_g\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_g\dot{\boldsymbol{q}} + \dot{\boldsymbol{v}}_{gm} \tag{9}$$

where $\boldsymbol{v}_{gm} = \boldsymbol{J}_b\boldsymbol{M}_{bb}^{-1}\begin{bmatrix} \boldsymbol{\ell} \\ \boldsymbol{\Psi} \end{bmatrix}$.

2.2 Path tracking controller

The proposed resolved acceleration controller requires the second derivative relationship presented in Equation (9). This equation defines how to obtain the end-effector acceleration from the joint acceleration. Additionally, a reference acceleration must be defined by two diagonal positive definite matrices (proportional and derivative matrices).

The system acceleration reference can be obtained by using both proportional and derivative matrices using the following expression:

$$\ddot{\boldsymbol{p}}_r = \ddot{\boldsymbol{p}}_d + \boldsymbol{K}_d(\dot{\boldsymbol{p}}_d - \dot{\boldsymbol{p}}) + \boldsymbol{K}_p(\boldsymbol{p}_d - \boldsymbol{p}) \tag{10}$$

where $\boldsymbol{p}_d$, $\dot{\boldsymbol{p}}_d$, and $\ddot{\boldsymbol{p}}_d$ are the desired position, velocities and acceleration to be tracked. These desired values will be obtained from the trajectories coming from the path planning method.

The outputs of the controller are the torques to be applied to the joints of the manipulator of the humanoid robot during the tracking of the trajectories. In order to obtain these torques, we derive first the control action expressed in joints accelerations. The following expression can be obtained for the joint accelerations from Equation (9):

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}_g^{+}\left(\ddot{\boldsymbol{p}}_r - \dot{\boldsymbol{J}}_g\dot{\boldsymbol{q}} - \dot{\boldsymbol{v}}_{gm}\right) \tag{11}$$

where the end-effector accelerations, $\ddot{\boldsymbol{p}}$, are replaced by the reference accelerations, $\ddot{\boldsymbol{p}}_r$, generated by Equation (10).

Finally, the control action expressed in joint torques can be obtained by taking into account the dynamics equation indicated in (2). Therefore, the final control law of the resolved acceleration controller can be defined by the following equation:

$$\boldsymbol{\tau} = \boldsymbol{M}^*\boldsymbol{J}_g^{+}\left(\ddot{\boldsymbol{p}}_r - \dot{\boldsymbol{J}}_g\dot{\boldsymbol{q}} - \dot{\boldsymbol{v}}_{gm}\right) + \boldsymbol{C}^* \tag{12}$$

When applying the control law defined by Equation (12), the following closed-loop behavior is obtained for the robot manipulator:

$$\boldsymbol{M}^*\ddot{\boldsymbol{q}} + \boldsymbol{C}^* = \boldsymbol{M}^*\boldsymbol{J}_g^{+}\left(\ddot{\boldsymbol{p}}_r - \dot{\boldsymbol{J}}_g\dot{\boldsymbol{q}} - \dot{\boldsymbol{v}}_{gm}\right) + \boldsymbol{C}^* \tag{13}$$

which leads to:

$$\boldsymbol{J}_g\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{p}}_r - \dot{\boldsymbol{J}}_g\dot{\boldsymbol{q}} - \dot{\boldsymbol{v}}_{gm} \tag{14}$$

From Equation (14) and by taking into account the value of $\ddot{\boldsymbol{p}}$ defined in Equation (9), we can obtain the following identity:

$$\ddot{\boldsymbol{p}} = \ddot{\boldsymbol{p}}_r \tag{15}$$

and, therefore by considering the definition of the reference acceleration given by Equation (10), we obtain the following relation:

$$0 = \ddot{\boldsymbol{p}}_d - \ddot{\boldsymbol{p}} + \boldsymbol{K}_d(\dot{\boldsymbol{p}}_d - \dot{\boldsymbol{p}}) + \boldsymbol{K}_p(\boldsymbol{p}_d - \boldsymbol{p}) \quad (16)$$

Demonstrating that this controller achieves asymptotic tracking in operational space.

3. Path planning

This section describes the path planning method that generates the trajectories. The proposed path planning method is based on the algorithm presented in [1]. However, several modifications are made to consider the inputs given by the camera located at the head of the humanoid robot.

The trajectory optimization generates trajectories for the linear position of the center of mass of the humanoid robot, $\boldsymbol{r}(t)$, its attitude, $\boldsymbol{\theta}(t)$ and the position and forces exerted by each arm, i, $\boldsymbol{p}_i(t)$ and $\boldsymbol{f}_i(t)$, respectively. The humanoid robot is equipped with a 3D camera located on its the head, as shown in Figure 2. This camera provides information about the depth of each point. This information is stored in the height map, $h(x, y)$ that provides the map of the altitudes (z coordinate) for each point in the work space defined by the coordinates x and y.

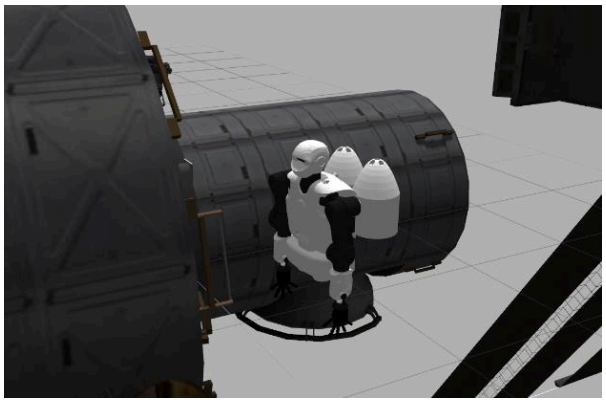

Fig. 2. Humanoid robot and camera at the head

Additional information must be provided to the path planning method to generate the trajectory. This includes the desired position for the robot's torso, the number of grasping positions, *N*, and the duration of the maneuvre, *T*.

In addition, optimization constraints are required to guarantee the feasibility of the generated trajectories given the robot properties and the workspace constraints. All the constraints are defined in the Cartesian workspace to reduce the number of variables estimated by the optimization algorithm. Specifically, both constraints on the kinematics and dynamics are necessary to define the problem correctly.

The kinematic constraint should guarantee that the generated trajectories are within the robot workspace. This is achieved mathematically through the following equation:

$$|\boldsymbol{R}(\boldsymbol{\theta})[\boldsymbol{p}_i(t) - \boldsymbol{r}(t)] - \overline{\boldsymbol{p}}_i| < \boldsymbol{\varepsilon} \quad (17)$$

where $\boldsymbol{R}(\boldsymbol{\theta})$ is the rotation matrix representing the orientation of the robot torso with respect the inertial frame, and $\overline{\boldsymbol{p}}_i$ is the nominal position of the *i-th* arm end-effector relative to the body frame. Thus, the arm's workspace is approximated by a cube of edge length 2 $\boldsymbol{\varepsilon}$ centred, for each hand, at position $\overline{\boldsymbol{p}}_i$

An additional constraint should be defined to guarantee that the generated trajectories fulfil the robot dynamics. Specifically, the following equation can be used to model the linear accellerations induced to the robot torso by the forces generated by the end effectors of the manipulators:

$$m\ddot{\boldsymbol{r}} = \boldsymbol{f}_1(t) + \boldsymbol{f}_2(t) \quad (18)$$

On the other hand, $\boldsymbol{\omega}(t)$ represents the angular velocity that can be computed from the Euler angles $\boldsymbol{\theta}(t)$ (the corresponding time derivatives are $\dot{\boldsymbol{\theta}}(t)$), and $\boldsymbol{M}_I$ is a constant rotational moment of inertia calculated from the nominal robot configuration.

A similar relationship to the one presented in (18) but for the angular acceleration can be obtained by considering the following equation:

$$\begin{aligned} &\boldsymbol{M}_I\dot{\boldsymbol{\omega}}(t) + \boldsymbol{\omega}(t) \times \boldsymbol{M}_I\boldsymbol{\omega}(t) = \\ &\boldsymbol{f}_1(t) \times \big(\boldsymbol{r}(t) - \boldsymbol{p}_1(t)\big) + \\ &\boldsymbol{f}_2(t) \times \big(\boldsymbol{r}(t) - \boldsymbol{p}_2(t)\big) \end{aligned} \quad (19)$$

Different constraints should be defined when the robot is in contact with the bodies belonging to the workspace (i.e., handles and handrails) and when the robot is not in contact. For example, when the humanoid robot is grasping a handrail, the robot end effector does not slide and maintains a fixed position during the grasping phase; therefore, the robot velocity in a grasping position must be zero.

Additionally, the grasping positions should be coincident with positions stored in the height map. As it is previously indicated, the height map provides information about the depth for each position of the workspace. Therefore, the z component of the grasping position should coincide with the corresponding value in the height map.

These last two constraints, defined when the robot is in the grasping position, can be summarized in the following equation:

$$\dot{\boldsymbol{p}}_i(t) = 0, \qquad p_i^z = h_w\left(p_i^x, p_i^y\right) \tag{20}$$

where $\boldsymbol{p}_i = \left(p_i^x, p_i^y, p_i^z\right)$ is the grasping position.

On the other hand, another typology of constraints is defined when the robot is not in contact with the handrails. In such a case, we need to guarantee that contact forces are null, as there is no interaction with handrails:

$$\boldsymbol{f}_i(t) = 0 \tag{21}$$

The functions mapping the behaviour of the components of the end-effector position $\boldsymbol{p}_i(t)$ and contact forces $\boldsymbol{f}_i(t)$ are assumed to be cubic polynomials. More specifically, three cubic polynomials are used to arbitrary generate the shape of the trajectories, $\boldsymbol{p}_i(t)$, and the contact forces, $\boldsymbol{f}_i(t)$.

4. Results

This section describes the simulation results obtained in the application of the proposed path planning and control algorithm to the humanoid robot presented in Fig. 1 and Fig. 2.

The dimensions and inertia properties of the torso of the humanoid robot and of the two arms, with seven degrees of freedom each, are provided in Table 1 . In the next subsections, simulations results are shown during the path planning of the humanoid robot and the tracking of the generated trajectories.

4.1 Path planning

To test the path planning algorithm, a simulation is perfomed, where the humanoid robot is supposed to move 0.6 *m* along the *x* direction. As it is indicated in Section 3, the optimization generates trajectories for the linear position of the centre of mass, $\boldsymbol{r}(t)$, its attitude, $\boldsymbol{\theta}(t)$ and the position and forces exerted by each arm, *i*, $\boldsymbol{p}_i(t)$ and $\boldsymbol{f}_i(t)$, respectively. To do this, the user should indicate the time required to perform the trajectory (T = 5,5 secs) and the number of grasping position ($N = 4$) that need to be used during the maneuvre.

Fig. 3 and Fig. 4 illustrate the main results obtained during this experiment. Specifically, Fig. 3 shows the contact forces at the end effectors and the linear accelerations at the robot's body due to the contact forces. In this simulation, only four contact occasions are generated, and peaks of the contacts forces are produced when the end effectors touch and grasp the handrails. These contact forces are responsible for the linear accelerations that move the main body of the humanoid robot. Fig. 4 represents the robot's body trajectory and the end-effectors positions during the movement. As it can be seen in Fig. 4. the robot body moves forward of 0.6 m along x direction while the displacement in the others direction is zero at the end of the trajectory. The desired location for the robot's body is achieved in the desired duration $T = 5.5$ s.

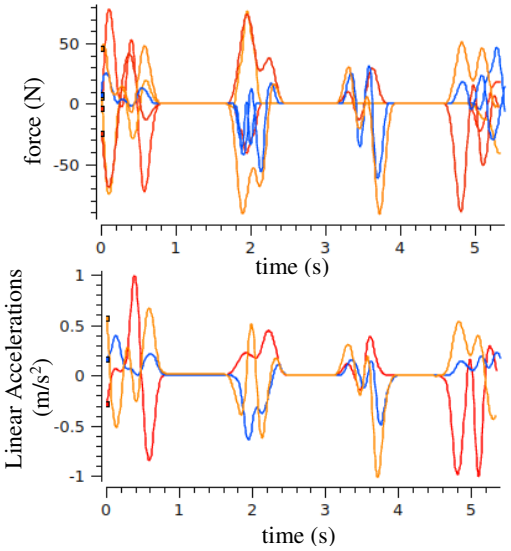

Fig. 3. Forces and linear accelerations of the robot body during the path planning

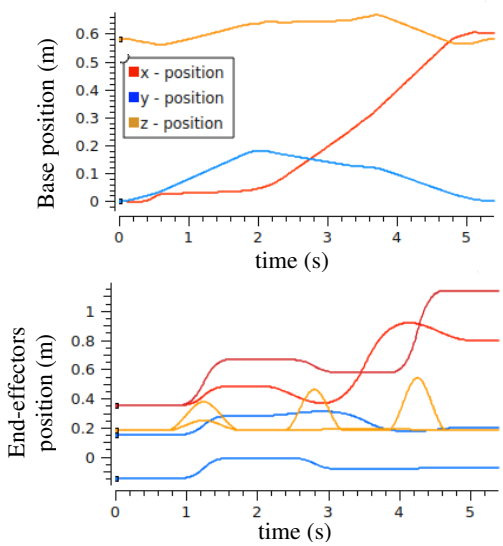

Fig. 4. Position of the robot torso and end-effectors during the path planning.

Table 1. Kinematic and dynamic parameters of the humanoid robot

	Mass (kg)	Height (m)	Inertia (kg·m2)					
			Ixx	Iyy	Izz	Ixy	Ixz	Iyz
Body Parameters	93	0.843	18.6	15.4	4.1	-0.008	-0.027	0.058

	Mass (kg)	Length (m)	Inertia (kg·m2)					
			Ixx	Iyy	Izz	Ixy	Ixz	Iyz
Link 1	2.741	0.28	0.0124	0.0042	0.0136	3.6e-05	7.1e-05	-0.0002
Link 2	2.425	0.144	0.013	0.0138	0.0049	1.2e-05	-0.0032	-0.0001
Link 3	2.209	0	0.007	0.0069	0.0039	-0.0001	0.0007	0.0004
Link 4	0.877	0.274	0.0025	0.0027	0.0012	0.0001	-0.0003	0.0004
Link 5	1.878	0.265	0.0035	0.0044	0.0023	1.3e-05	1.03e-05	-9.7e-05
Link 6	0.409	0	0.0001	0.00014	0.00015	-8.9e-08	-4.4e-08	4.2e-07
Link 7	0.308	0	0.0003	0.0002	0.00017	-1.6e-06	1.7e-06	-1.2e-05

4.2 Trajectory tracking

Once the trajectories are generated by using the proposed path planning algorithm, the controller described in Section 2 is used to perform the tracking of the previous generated trajectory. This section describes the main results of the controller during the grasping of a handrail. Fig. 5 represents the trajectory considered in this experiment. This trajectory consists of a grasping trajectory. Fig. 6 shows the main results obtained during the tracking of the grasping trajectory. Fig. 6.a and Fig 6.b represents the tracking error (position and attitude respectively). These two last figure show that the tracking error remains low during the tracking phase and, indeed, the desired location is correctly achieved by tracking the desired trajectory. Fig 6.c represents the control action generated by the proposed controller in Equation (12) showing that the torques remain limited and below 1 Nm during the entire maneuvre.

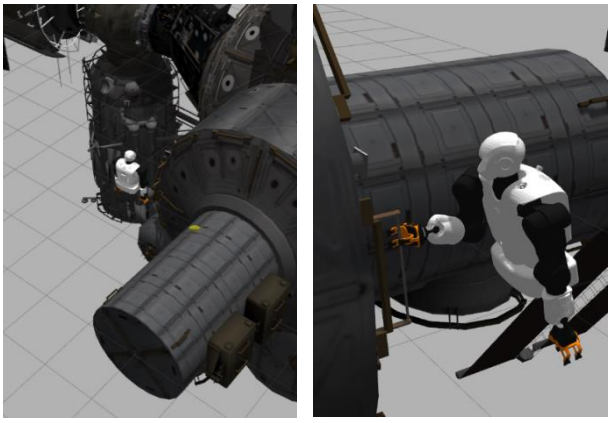

Fig. 5. 3D representation of the desired trajectory

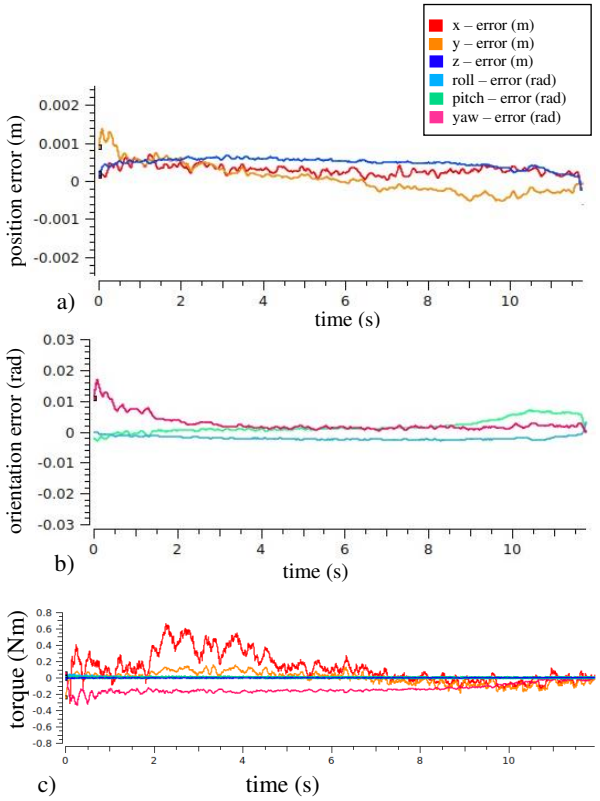

Fig. 6. Trajectory tracking. a) Position error. b) Orientation error. c) Control action (torque)

6. Conclusions

The path planning and control of a humanoid robot in free-floating conditions was presented in this paper. An optimization formulation is presented for defining the path planning problem, while a task-space controller that integrates the robot dynamics is used for the tracking of the generated trajectories. Simulation results carried on in ROS/Gazebo environment show that the overall

control architecture (trajectory optimizer + controller) is sufficiently robust and allows for complex and articulated motion of the humanoid robot in an eventual extravehicular activity outside the ISS in free-floating conditions.

References

[1] M. Mote, M. Egerstedt, E. Feron, A. Bylard, M. Pavone, Collision-Inclusive Trajectory Optimization for Free-Flying Spacecraft, Journal of Guidance, Control, and Dynamics, 43(7) (2020), 1247–1258.

[2] G. Misra, X. Ba, Optimal Path Planning for Free-Flying Space Manipulators via Sequential Convex Programming, Journal of Guidance, Control, and Dynamics, 40(11) (2017) 3019–3026.

[3] G. Misra, X. Ba, Task-Constrained Trajectory Planning of Free-Floating Space-Robotic Systems Using Convex Optimization. Journal of Guidance, Control, and Dynamics. 40(11) (2017) 2857–2870.

[4] A. Herzog, S. Schaal, and L. Righetti, Structured contact force optimisation for kino-dynamic motion generation, in IEEE International Conference on Intelligent Robots and Systems, 2016, 2703–2710.

[5] A. Herzog, N. Rotella, S. Schaal, and L. Righetti, Trajectory generation for multicontact momentum control, in IEEE-RAS International Conference on Humanoid Robots, IEEE, 2015, 874–880.

[6] A. Seddaoui, C. Saaj, M. Nair. Modeling a Controlled-Floating Space Robot for In-Space Services: A Beginner's Tutorial. Frontiers in Robotics and AI, 8. (2021).

[7] Y. Wang, X. Liang, K. Gong, Y. Liao. Kinematical Research of Free-Floating Space-Robot System at Position Level Based on Screw Theory. International Journal of Aerospace Engineering. (2019).

[8] D. R. Isenberg, A Task-Space Control Law for Free-Floating Space Robots. 25th International Conference on Systems Engineering (ICSEng), 2017, pp. 33-38.

[9] X. Zhao, Z. Xie, H. Yang, J. Liu, J. Minimum Base Disturbance Control of Free-Floating Space Robot during Visual Servoing Pre-capturing Process. Robotica, 38(4), (2020) 652-668.

[10] S. Kraïem, M. Rognant, J.-M. Biannic, Y. Brière. Robust control of rotation-floating space robots with flexible appendages for on-orbit servicing. IFAC-PapersOnLine, 54(20), (2021), 134-140.

[11] R. Wang, C. Liang, D. Pan, X. Zhang, P. Xin, X. Du. Research on a Visual Servo Method of a Manipulator Based on Velocity Feedforward. Space: Science and Technology. (2021).

[12] J L. Ramón, R. Calvo, A Trujillo, J Pomares, L Felicetti. "Trajectory Optimization and Control of a Free-Floating Two-Arm Humanoid Robot", Journal of Guidance, Control, and Dynamics (2022), 1-15.