

3

Generador de movimientos de desensamblado

3.1. Introducción

Un aspecto muy importante a la hora de realizar un proceso de desensamblado consiste en determinar cuál es la trayectoria a seguir por un componente para poder separarlo del resto del producto.

En este capítulo se tratan las distintas maneras de abordar este problema, para, en base a ellas, definir el método aportado en esta tesis para calcular los movimientos de desensamblado de un componente.

A la hora de realizar el desensamblado de un producto existen distintos planteamientos sobre cuál debe ser la secuencia de movimientos, el camino a seguir, por un componente para que se separe del resto del producto.

Una primera aproximación es la utilizada por los autores que sólo plantean la generación de la secuencia de planificación, por lo tanto la información sobre los movimientos a realizar por cada componente se encuentra almacenada en la base de datos [48] [106] [107] [162] [180] [202] [206] [213] [214].

Dentro de este tipo de métodos se encuentran principalmente aquellos que sólo intentan minimizar una función de coste sencilla [213] [214]; también se incluyen entre este tipo los que utilizan sistemas expertos [202]. Por consiguiente, debido a que estos métodos no planean una aproximación a la generación de movimientos de desensamblado, no se van a tratar en este apartado ya que dichos métodos se han tratado con mayor profundidad en el capítulo 2.

Otro planteamiento se presenta cuando se calcula automáticamente la dirección de separación de un componente en base a información sobre sus contactos con otros componentes, con lo que si un componente está unido a otro tiene una restricción de movimiento en una dirección determinada, con lo que según se desensamblan los componentes se obtienen nuevas direcciones libres en las que se puede desplazar un componente para separarlo del resto [133] [148] [150] [299] [300] [301] [308].

Otro conjunto de métodos son aquellos métodos que permiten generar la secuencia de desensamblado sin intervención del usuario, calculan las posibles direcciones de separación utilizando un modelo CAD del producto; es decir, utilizan un modelo geométrico del producto sobre el que se aplican una serie de operaciones que permiten obtener las direcciones de desensamblado [62] [160] [161] [243] [247] [248] [249] [250].

Llegados a este punto hay que realizar la definición del concepto de sistema n -desensamblado, que es:

«Un sistema de desensamblado se denomina n -desensamblado si requiere, para separar un componente del resto, n movimientos en diferentes direcciones. Si un componente requiere $n+1$ movimientos para separarse del resto entonces dicho componente se considera no desensamblable.»

En la Figura 3.1 se muestra un ejemplo en el que el componente c_2 es 2-desensamblable ya que requiere de dos traslaciones para poder separarse del resto del producto.

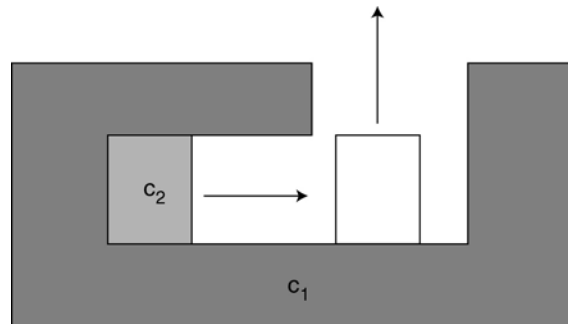


Figura 3.1: Representación de un componente 2-desensamblable.

Teniendo en cuenta esta definición la mayoría de los sistemas de desensamblado, son del tipo 1-desensamblado, es decir, se requiere un solo movimiento para desensamblar un componente [62] [89] [90] [91] [133] [240] [247] [248] [249] [250].

En esta tesis se aporta una solución al problema del n -desensamblado, utilizando para ello una representación geométrica del producto basada en un modelo CAD, ya que ello permite el cálculo automático de los movimientos de desensamblado.

Este capítulo se divide de tal manera que: en el siguiente apartado se explica el método de representación utilizado para llevar a cabo el cálculo de la trayectoria de desensamblado para un solo componente. A partir de esta, se presenta cómo calcular la dirección de desensamblado de un componente. Tras esto, se explica cómo calcular la trayectoria completa de desensamblado del componente y como detectar las colisiones con otros componentes. Por último se muestra un ejemplo del cálculo de la trayectoria de desensamblado

para un producto en concreto así como una evaluación en cuanto a costes computacionales del algoritmo planteado.

3.2. Representación geométrica del producto

La representación que se utiliza del producto debe permitir la generación automática de los movimientos de desensamblado, es por ello que se presenta a continuación un breve estado del arte en lo que respecta a representación geométrica de los productos. Tras él se plantea el método utilizado para convertir la información geométrica en un grafo, que se corresponde con el método utilizado en esta tesis.

3.2.1. Estado del arte

Las posibles representaciones de un producto mediante un programa CAD se pueden resumir en tres:

1. Utilización de un modelo basado en sólidos, lo que implica el manejo de modelos geométricos complejos y diferentes para cada componente del producto.
2. Un modelo poliédrico del producto, en el que cada componente se define como un poliedro. Este método simplifica el tratamiento de los distintos componentes ya que todos se simplifican en última instancia a un conjunto de polígonos [248] (Figura 3.2).

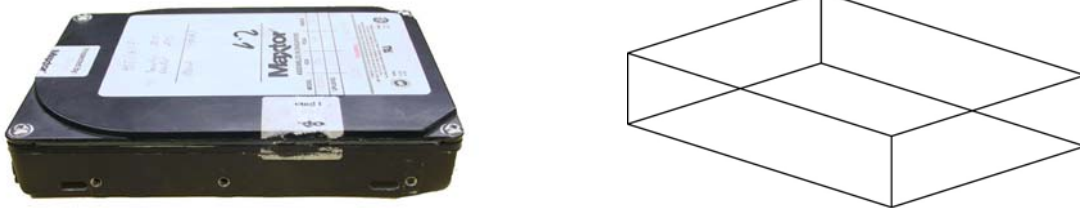


Figura 3.2: Disco duro y su representación simplificada mediante polígonos.

3. Un modelo basado en elementos finitos. Este modelo planea una solución bastante precisa en cuanto al diseño de los componentes, pero conlleva un cálculo complejo [42] (Figura 3.3).

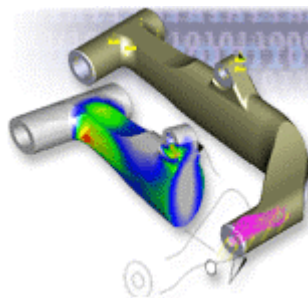


Figura 3.3: Representación de un componente utilizando elementos finitos (Cortesía de HEC).

Para que estas representaciones estén correctamente especificadas habría que incluir dos definiciones, la de polígono y la de poliedro [44]:

Un polígono se define como:

«Figura plana limitada por una línea poligonal cerrada. Los segmentos de la línea poligonal se llaman lados. Según el número de lados, los polígonos se

clasifican en: triángulos, cuadriláteros, pentágonos, hexágonos, heptágonos, octógonos, etc.»

Basándose en esta definición, se puede definir un poliedro como:

«Sólido limitado por polígonos llamados caras. La intersección de dos caras es una arista. Los puntos de intersección de 3 o más caras forman un vértice.»

Teniendo en cuenta estas características hay que indicar que la más utilizada consiste en la utilización del modelo poligonal del producto ya que permite una generalización mayor a la hora de definir las operaciones para el cálculo de la secuencia de movimientos.

Esto es debido a que todo programa CAD actual puede convertir cualquier modelo por complejo que sea a un conjunto de polígonos, normalmente triángulos, que definan los distintos componentes.

El problema que aparece a la hora de utilizar un modelo poliédrico de los productos consiste en la pérdida de resolución, ya que las superficies que no son planas: tubos, tornillos, etc., se tienen que transformar en superficies planas, realizando una aproximación de sus contornos. Esta aproximación se puede ajustar, en la mayoría de programas CAD, según el error que se pueda tolerar en el modelo, de manera que para obtener una mayor precisión simplemente se aumenta el número de polígonos utilizados para convertir el modelo CAD a un modelo poliédrico (Figura 3.4).

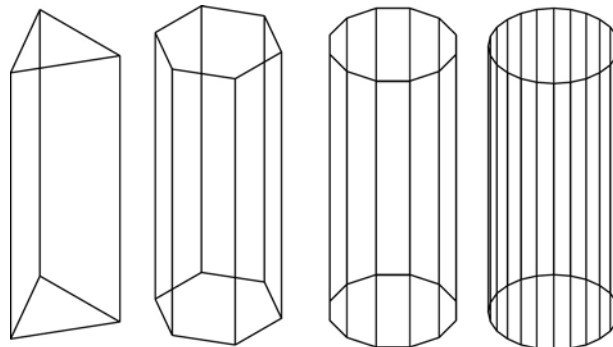


Figura 3.4: Modelo poliédrico de un cilindro utilizando 5, 8, 14 y 26 polígonos.

3.2.2. Representación utilizada

El planteamiento que se utiliza en esta tesis consiste en usar un modelo geométrico del producto. Este modelo se basa en un modelo CAD de cada uno de los componentes del producto a desensamblar, así como de las relaciones de posición existentes entre ellos. Utilizando dicho modelo se calcula la mejor dirección de desensamblado, y se realiza un desplazamiento lineal del componente, tras lo que se actualiza el modelo y se permite realizar al componente una secuencia de n movimientos lineales hasta que se consigue desensamblarlo, con lo que está planteando un sistema n -desensamblado, el cuál es una de las aportaciones originales de esta tesis.

Además se dispone de varios componentes que forman un producto, por lo tanto, no basta con tener el modelo individual de cada uno de ellos por separado, sino que hay que conocer cuál es la localización espacial de unos respecto a otros para tener un correcto modelado del producto.

Para representar, mediante un grafo, las caras existentes en un componente, se utiliza un grafo de adyacencia (Figura 3.5), donde se representa la relación de un componente con las caras que lo forman [161].

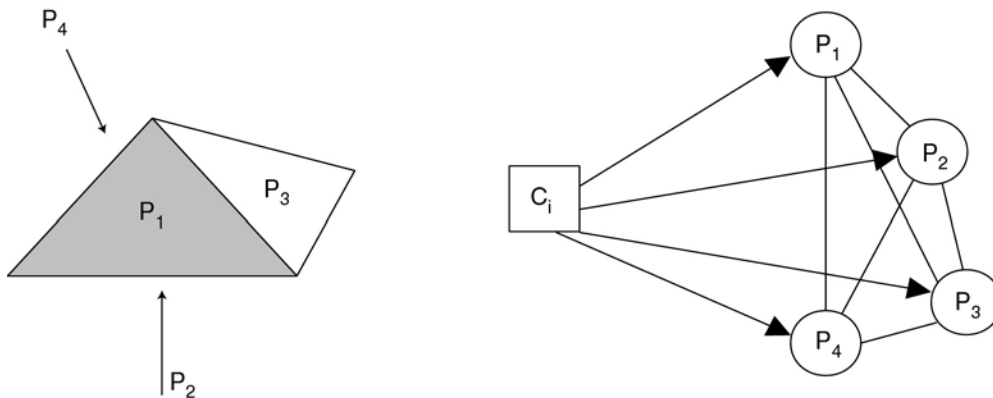


Figura 3.5: Componente poliédrico C_i de cuatro caras (P_1 , P_2 , P_3 , P_4) y el grafo de caras de adyacencia.

Una vez se ha definido el concepto de grafo de adyacencia se definen los grafos de adyacencia correspondientes a cada uno de los componentes del producto.

Con estos grafos hay que crear el grafo de caras de contacto, el cual representa las relaciones existentes entre las caras de los distintos componentes; es decir, si dos componentes están en contacto dicha información se verá reflejada en el grafo de caras de contacto.

Este grafo se genera partiendo de los grafos de adyacencia de todos los componentes del producto y añadiendo una arista entre cada par de nodos, los nodos que representan las caras, que poseen una relación de contacto entre ellos.

De esta manera en el nuevo grafo aparecen reflejadas las relaciones, ya no de un solo componente, sino entre todos los componentes del producto.

Para ilustrar estos conceptos se utiliza el producto compuesto por los tres componentes poliédricos de la Figura 3.6.

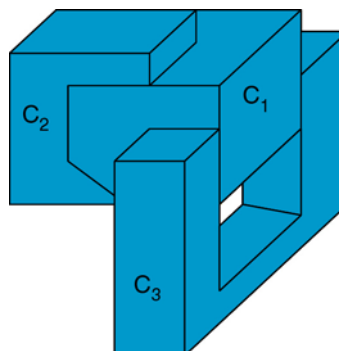


Figura 3.6: Producto compuesto por tres componentes.

En la Figura 3.7 se puede ver el grafo de adyacencia para el primer componente del producto de la Figura 3.6. En el que se ha simplificado el grafo, para una mejor visualización, eliminando las aristas que representan un contacto entre las caras del propio componente.

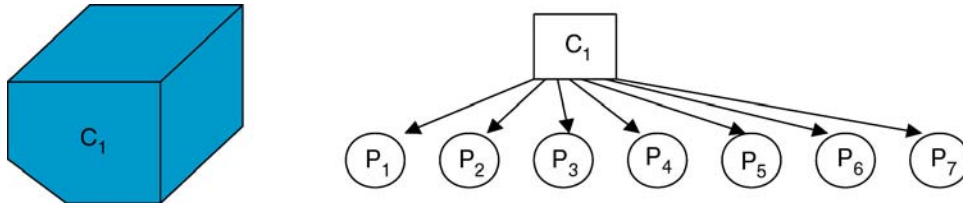


Figura 3.7: Grafo de adyacencia para el componente C_1 del producto de la Figura 3.6.

La Figura 3.8 presenta el grafo de caras de contacto que resulta de unir el componente de la Figura 3.7 con el del segundo componente del producto.

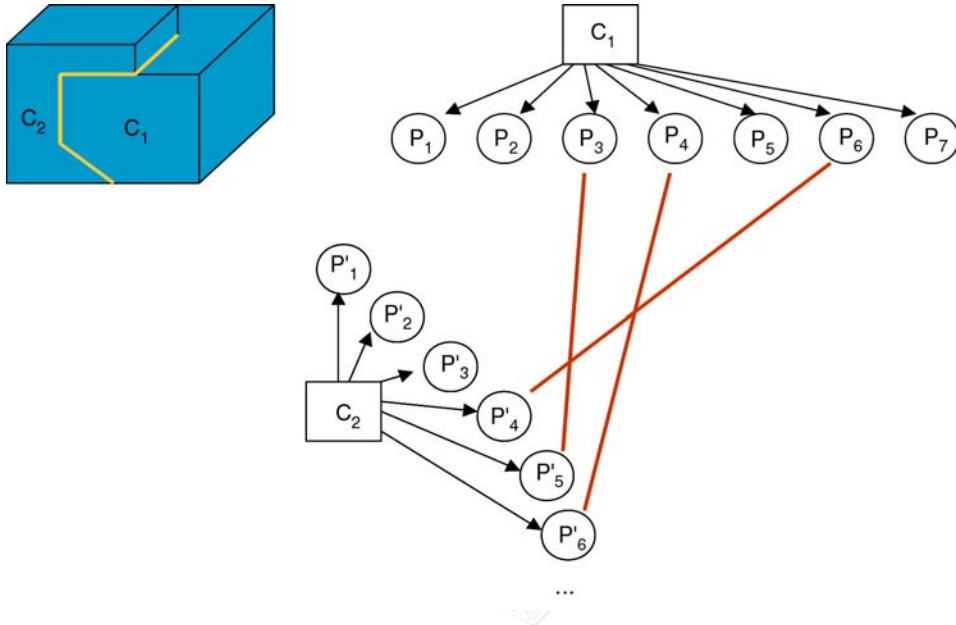


Figura 3.8: Grafo de caras de contacto para los componentes C_1 y C_2 del producto de la Figura 3.6.

En la Figura 3.9 se presenta el grafo de caras de contacto resultante de añadir a los dos componentes de la Figura 3.8 el tercer componente, con lo que se obtiene el grafo de caras de contacto completo del producto.

Cada arista que representa un contacto posee un descriptor que indica en cada momento qué tipo de contacto se está produciendo. En el esquema descrito (Figura 3.9) se observa como el tipo de contacto existente entre las caras de los componentes, es de tipo plano.

También se pueden producir otros tipos de contacto como de rosca entre un tornillo y un componente, de tal manera, que el tipo de contacto implica una u otra secuencia de movimientos de desensamblado que se pueden almacenar en una base de datos. Por ejemplo el contacto plano requiere un desplazamiento, un contacto de rosca requiere un desplazamiento y simultáneamente una rotación, etc.

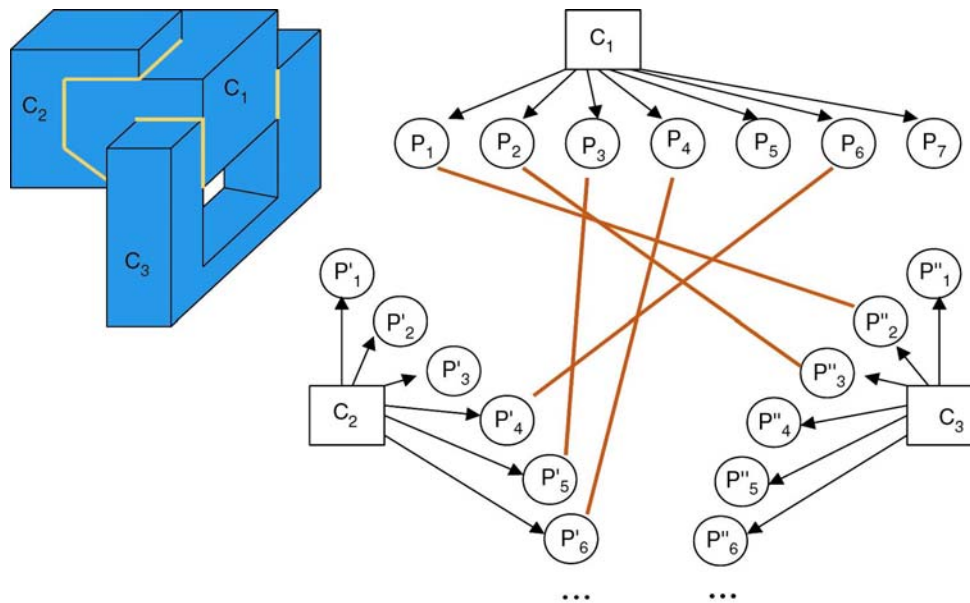


Figura 3.9: Grafo de caras de contacto para el producto de la Figura 3.6.

En el estudio que se va a realizar, las direcciones de desensamblado se calculan basándose en la existencia de contactos planos; si el contacto es de otro tipo, entonces la dirección de desensamblado vendrá indicada por el tipo de contacto, dado que no implicará desplazamientos lineales de los componentes.

3.3. Determinación de la dirección de separación de un componente

En este apartado se determina la dirección de separación de un componente respecto del resto del producto. En primer lugar, a partir de la información disponible en el grafo de caras de contacto, se establece un conjunto de posibles direcciones de separación, desde el que, finalmente, se determina una dirección concreta.

3.3.1. Estado del arte

A partir del grafo de caras de contacto entre los componentes del producto, hay que determinar los movimientos relativos de los componentes para su desensamblado. Estos desplazamientos corresponden, principalmente, con traslaciones que son modeladas a partir de relaciones entre componentes. Para abordar este problema existen dos métodos principales: el concepto de esfera unidad y las representaciones cinemáticas.

3.3.1.1. Esfera unidad

El concepto de esfera unidad, o esfera de Gauss es utilizado ampliamente y presentado en detalle en [244], [288] y [289]. Permite determinar los desplazamientos de traslación posibles de un componente en función de sus contactos. Los componentes son modelados utilizando una representación poliédrica, que permite definir una representación polinomial de los planos de contacto, que son necesarios para este método. La idea básica consiste en que

cada cara de contacto entre dos superficies divide la esfera en dos hemisferios.

Una característica importante a tener en cuenta es que este método no tiene en cuenta explícitamente los movimientos de rotación de los componentes.

Otro planteamiento [156] [225] [287] consiste en: a partir de una representación poliédrica de los componentes y del análisis de los contactos planos generar un grafo denominado como: *Non-Directional Blocking Graph*; que representa las direcciones en las que se puede desplazar cada componente.

También existen extensiones de esta idea que consideran los contactos cilíndricos y las dos direcciones de traslación asociadas a este tipo de contacto [244].

Una aproximación más detallada incluye, además, la posibilidad de tener en cuenta los ejes de rotación admisibles para los contactos planos [182]. Las rotaciones son determinadas a partir de la geometría del contacto, es decir, de las dimensiones de las superficies planas y son representadas con una esfera unidad.

3.3.1.2. Representaciones cinemáticas

Dentro de un sistema mecánico, la puesta en contacto de dos componentes implica condiciones o limitaciones de los valores posibles de sus parámetros de desplazamiento relativo. La unión así constituida viene caracterizada por la superficie o superficies de contacto entre los sólidos modelados de los componentes.

Los modelos de uniones suelen ser definidos por un modelo de representación de los grados de libertad correspondientes a los movimientos autorizados por la unión. La representación de los grados de libertad utilizando una notación cinemática permite distinguir explícitamente la orientación y la dirección de los movimientos relativos de traslación y de rotación de un contacto que son necesarios para romper este contacto.

En este caso se representan las uniones en forma parecida a una matriz de transformación homogénea que se denomina "pseudomatriz" o "freedom matrix". Esta matriz representa las transformaciones relativas necesarias para la eliminación de un determinado componente [155] [181]. Las traslaciones representadas en la pseudomatriz se pueden determinar automáticamente a partir del análisis de los modelos CAD mientras que las rotaciones son completadas manualmente si son necesarias para realizar el desensamblado [155] (Figura 3.10).

El concepto de esfera unidad es el más general y permite determinar las direcciones de traslación posibles para el desensamblado. Sin embargo, es difícil de extender para todos los tipos de contacto y de tener en cuenta los movimientos de rotación.

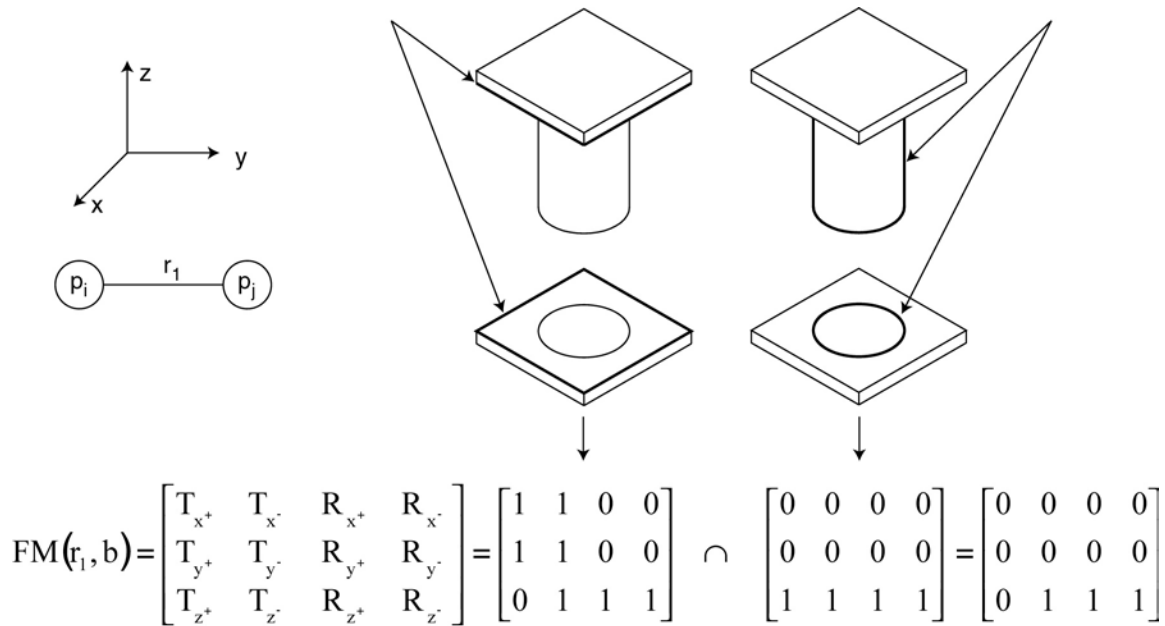


Figura 3.10: Ejemplo de aplicación de la técnica "freedom matrix".

3.3.2. Conjunto de direcciones de separación

La dirección de desensamblado de un componente consiste en calcular cuál es la dirección en la que se puede desplazar, para conseguir separarlo del resto. Este concepto presupone que, con un solo movimiento lineal, se puede separar el componente deseado; si se necesita más de un movimiento o movimientos no lineales (Figura 3.11), la dirección obtenida no sirve directamente para desensamblar el componente.

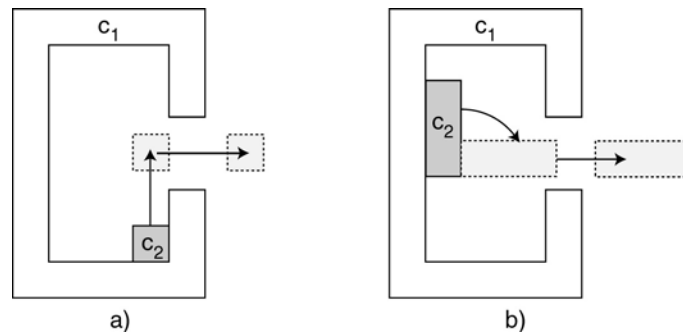


Figura 3.11: Combinación de movimientos. a) Se necesitan dos movimientos lineales para desensamblar c_2 . b) Se necesita un movimiento de rotación y uno lineal para desensamblar el componente c_2 .

También se presupone que todas las caras de los poliedros que pueden impedir un movimiento, se encuentran adyacentes unas a otras. Una cara que no tenga contacto con otra, no impide el movimiento (Figura 3.12).

Con estos dos conceptos, el sistema calcula una dirección de desensamblado para un determinado componente.

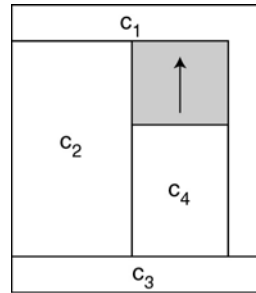


Figura 3.12: Los componentes c_1 y c_4 poseen una restricción que impide su movimiento en una determinada dirección (zona gris), pero no existe un contacto directo en esa dirección que indique la restricción.

Para poder llevar a cabo el desensamblado del componente c_1 de la Figura 3.6, al resto de componentes que forman el producto (c_2 y c_3), se les considera como si de un único componente se tratara, denominado componente grupo residual (Figura 3.13), conocido en la bibliografía como *residual component group (RCG)* [248].

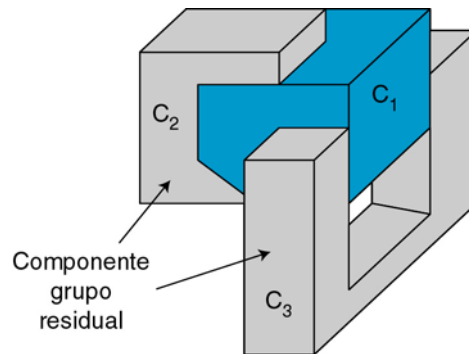


Figura 3.13: Representación del componente grupo residual para desensamblar el componente c_1 .

Para separar un componente del grupo residual, lo primero a realizar consiste en calcular cuáles son las caras de contacto entre ambos componentes, denominadas en la bibliografía como *interacting surfaces (IS)*. Las caras de contacto restringen las direcciones en las que pueden desensamblarse dos componentes.

Un aspecto importante a tener en cuenta es que las caras de contacto siempre aparecen por pares, es decir, si una cara del componente c_1 tiene contacto con una cara del componente c_2 , entonces, el componente c_2 tiene el contacto conjugado con el componente c_1 . La cara de contacto conjugada se expresa como IS^* . Cada par de caras adyacentes IS/IS^* recibe el nombre de grupo de superficies con interacción, denominada en la bibliografía como *set of interacting surfaces (SIS)*.

Las caras de contacto entre los distintos componentes del producto de la Figura 3.6 se muestran en la Figura 3.14.

Un grupo de superficies con interacción, expresa una restricción en el movimiento de un componente respecto a otro, de manera que define una serie de direcciones posibles de separación para el componente, denominado en la bibliografía como *set of possible directions of separation (SDS)*.

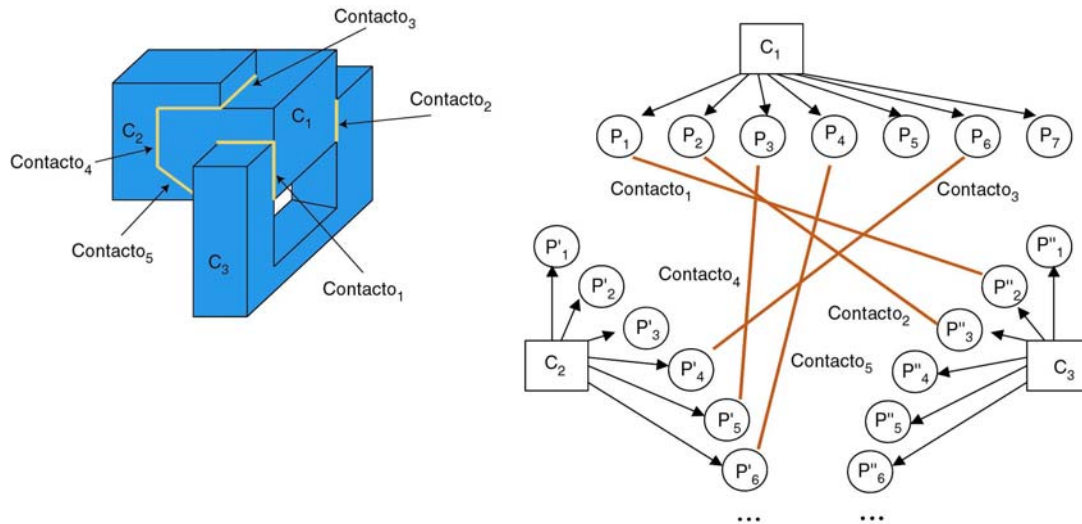


Figura 3.14: Representación de los contactos entre las caras del producto.

Las posibles direcciones de separación de un componente se pueden expresar en el espacio tridimensional como una esfera gaussiana [61], que representa todas las direcciones posibles de movimiento si el componente estuviese libre en el espacio 3D. Esta esfera representa todos los posibles vectores del espacio; cada vector es una dirección de desensamblado, para los que es posible realizar un movimiento del componente a desensamblar. De esta manera, y según las restricciones impuestas por el grafo de caras de contacto, se va cortando la esfera, hasta dejar las direcciones válidas para desensamblar un determinado componente (Figura 3.15).

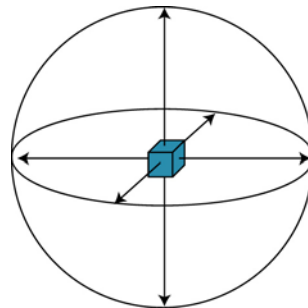


Figura 3.15: Esfera gaussiana para un componente libre en el espacio tridimensional.

Si se tiene en cuenta un contacto del componente por su cara inferior con una superficie plana, otro componente, una mesa de trabajo, etc.; entonces las direcciones posibles de desensamblado se reducen a una semiesfera (Figura 3.16).

Aplicando las definiciones anteriores se obtiene que el grupo de direcciones de separación de dos componentes, que tienen un contacto plano entre dos de sus caras, se reduce a una semiesfera.

De esta manera se pueden definir matemáticamente las posibles direcciones de desensamblado de un componente que tiene un contacto plano con otro de la siguiente manera:

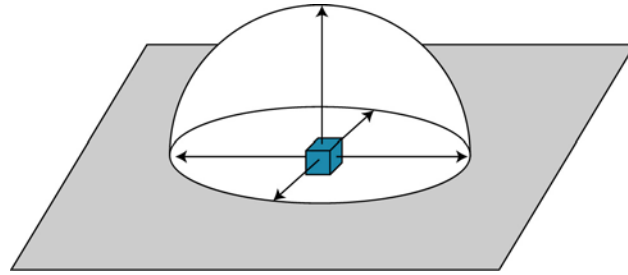


Figura 3.16: Semiesfera que define las posibles direcciones de desensamblado de un componente que posee un contacto plano con otro.

Considérese $\mathbf{n}=(n_1, n_2, n_3)$ como el vector normal a la superficie de contacto, cuyo sentido se corresponde a ir desde el punto de contacto en la superficie hacia el componente; y si se representa por $\mathbf{x}=(x_1, x_2, x_3)$ al vector que define una posible dirección de desensamblado, entonces se tiene que, para que una dirección sea válida, debe cumplir:

$$\mathbf{n} \times \mathbf{x} = n_1x_1 + n_2x_2 + n_3x_3 > 0 \quad (3.1)$$

Si en lugar de tener un único contacto plano existen múltiples contactos planos, para que un vector \mathbf{x} sea una dirección de desensamblado válida, debe de cumplir:

$$\sum_{j=1}^3 n_{ij}x_j > 0 \quad \text{para } i = 1, 2, \dots, n \quad (3.2)$$

Donde n_{ij} son las componentes de los vectores \mathbf{n}_i normales a cada uno de los planos de contacto.

De esta manera, para que un determinado componente tenga una solución válida, debe de tener al menos una solución para el vector \mathbf{x} en la ecuación 3.2.

El conjunto de posibles direcciones en las que un componente puede ser desensamblado, se denomina como conjunto de direcciones de separación, denominado en la bibliografía como *set of directions of removal (SDR)*. Este conjunto de direcciones se calcula realizando la intersección de todas las direcciones de separación del componente.

En la Figura 3.17 se muestra la unión de las direcciones de desensamblado obtenidas para cada contacto, según se va reduciendo la esfera por las direcciones posibles de desensamblado.

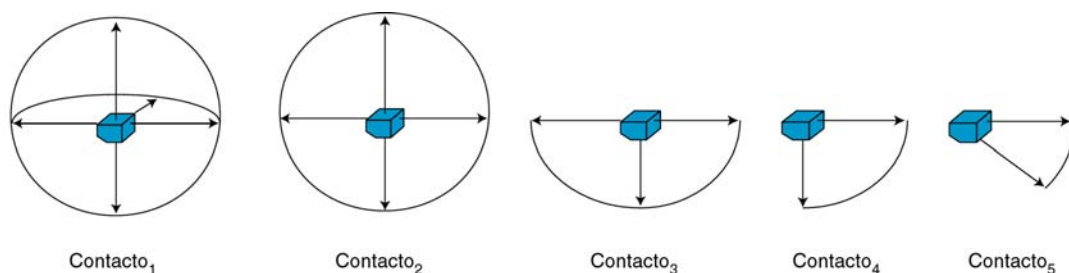


Figura 3.17: Unión de las direcciones de desensamblado del componente c_1 para obtener el conjunto de direcciones de separación.

Aplicando estos resultados al desensamblado del componente c_1 se obtiene que, para desensamblarlo, se puede desplazar hacia la derecha, hacia la parte inferior derecha o en cualquier dirección comprendida entre ambas; según el rango de direcciones posibles que han quedado de la esfera de la Figura 3.17

3.3.3. Determinación de una dirección de separación

De entre todas las posibles direcciones de separación, determinadas a partir del grafo de caras de contacto, se selecciona una de ellas, considerando las localizaciones del espacio en las que hay menor probabilidad de que el componente a desensamblar colisione con el resto del producto, con el que no mantiene contactos físicos.

Como se puede apreciar en la Figura 3.18 el algoritmo determina unas direcciones de desensamblado posibles; pero dado que el obstáculo no posee una cara de contacto con el componente a desensamblar, entonces existen direcciones de la solución que plantean una mayor probabilidad de colisión que las demás.

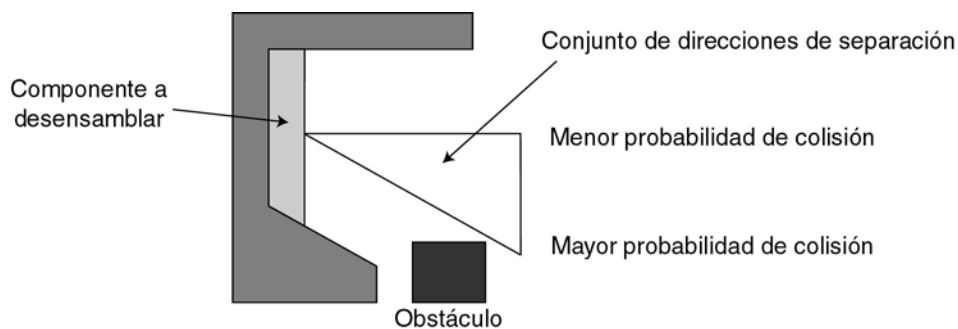


Figura 3.18: Probabilidad de colisión en un determinado conjunto de direcciones de desensamblado.

Al llegar a este punto hay que determinar cuál es la dirección en la que se desplaza el componente para realizar su desensamblado, evitando las colisiones con los objetos con los que no posee un contacto.

Para solucionar este problema, se plantea un algoritmo que permite el cálculo de dónde se pueden producir las colisiones. Para ello, el entorno de trabajo se discretiza en celdas cúbicas (Figura 3.19), de tal manera que, en cada celda se indica la probabilidad de encontrar un obstáculo; así es posible determinar cuál es la dirección en la que existe una menor probabilidad de encontrar otro componente.

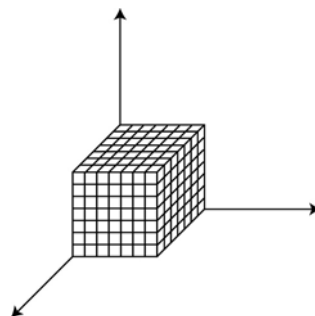


Figura 3.19: Discretización del entorno de trabajo en celdas cúbicas.

En cada celda se acumula información acerca de la existencia de posibles obstáculos en esa posición, conteniendo el valor 1 cuando haya certeza de que se encuentre un obstáculo en esa posición, y 0 cuando no la haya.

Para determinar la existencia de un obstáculo, se simula una técnica similar a la empleada por ultrasonidos o láser en robótica móvil. Es decir, a partir del modelado geométrico existente en ese momento, se simula la existencia de un haz esférico de rayos, que parte del centro del componente a desensamblar, indicando el valor de la detección para cada rayo en las coordenadas correspondientes del mundo discretizado.

Con la información almacenada, se crea un histograma que indica, para cada dirección del conjunto determinado a partir de las caras de contacto, la probabilidad de que exista un obstáculo. El histograma es generado a partir de la información obtenida dentro de una esfera de un cierto radio preestablecido, de tal manera que, a mayor radio, más información se tendrá en cuenta; pero por el contrario, mayor será el coste computacional para realizar los cálculos.

Para generar el haz de rayos, se ha seleccionado hacerlo desde el centro geométrico del componente. Dicho punto de referencia puede ser perfectamente cualquier punto del componente, ya que lo que se pretende con él es aumentar el conocimiento del entorno. Dicha información será más o menos precisa según la elección del punto de referencia, pero siempre válida. Una alternativa interesante consiste en generar una semiesfera de rayos desde el centro de cada una de las caras del componente, con lo que se obtiene una mayor información del entorno, pero se incurre en un mayor coste computacional, que se incrementa rápidamente con la utilización de componentes complejos.

Ya que si se posee un componente con n caras, y se lanza un haz de m rayos desde el centro del componente, en caso de lanzar el mismo haz desde cada cara, se tendrían $n \cdot m$ rayos, lo que incrementa claramente el coste computacional y está claro que se tendrá una mayor información del entorno, debido a que se han lanzado mayor cantidad de rayos.

A continuación se emplea una estrategia similar a la empleada en [216]. Se generan unas fuerzas de repulsión y atracción sobre el componente que se está separando. Las fuerzas de repulsión en una dirección resultan proporcionales a los valores de probabilidad almacenados en el histograma para esa dirección e inversamente proporcional a la distancia del obstáculo. Las fuerzas de atracción se generan desde las direcciones libres de obstáculos. Combinando ambas, se determina la dirección de menor repulsión, dentro de las comprendidas en el conjunto de direcciones de separación, obtenidas por el grafo de caras de contacto.

A continuación se detallan los pasos a seguir para calcular la dirección en la que existe una menor probabilidad de encontrar un obstáculo; es decir, menor repulsión:

1. Una vez lanzados los rayos desde el componente, hay que determinar las casillas del mundo discretizado donde existe un obstáculo, es decir, donde el rayo colisiona con otro componente. La posición del punto de choque se expresa en coordenadas, respecto al centro del componente que se está considerando. De tal manera que se asigna un valor de 1 a aquellas casillas en las que existe un

obstáculo. Un ejemplo de estos conceptos se ilustra en la Figura 3.20 en representación bidimensional.

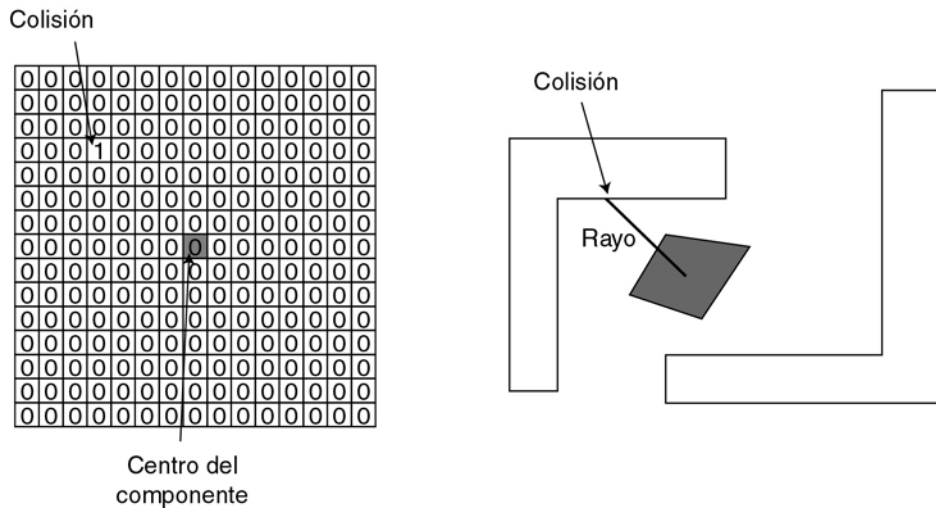


Figura 3.20: Detección de colisiones e incremento del mundo discretizado.

Tras enviar un haz de rayos se obtiene para el mundo discretizado una mayor información sobre el entorno del componente. Además hay que tener en cuenta que la zona a considerar para generar la trayectoria, se reduce a una distancia máxima de r' (Figura 3.21)

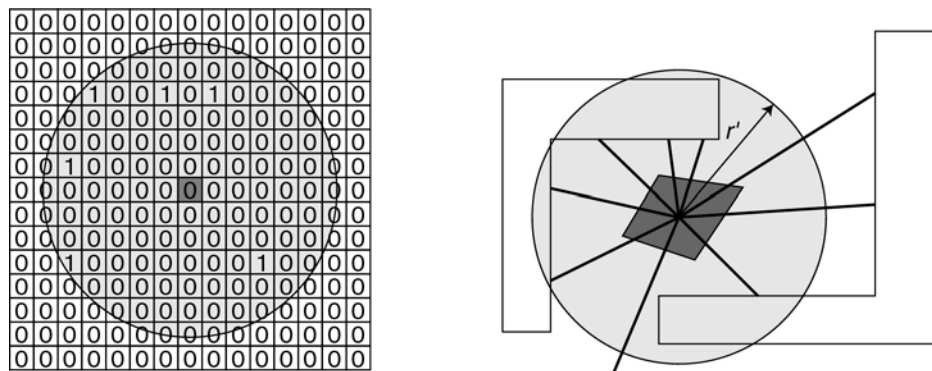


Figura 3.21: Mundo discretizado tras un haz de rayos.

2. El siguiente paso del algoritmo consiste en la generación del histograma del mundo discretizado, que indica la distancia y posición de los objetos en el mundo discretizado. Para generar el histograma, las coordenadas cartesianas del mundo discretizado (x, y, z) se convierten a coordenadas esféricas (r, θ, ϕ) (Figura 3.22), considerando el origen de coordenadas en el punto que representa el centro del componente.

$$r = \sqrt{x^2 + y^2 + z^2} \tag{3.3}$$

$$\theta = \arctan \frac{y}{x} \tag{3.4}$$

$$\phi = \arctan \frac{\sqrt{x^2 + y^2}}{z} \quad (3.5)$$

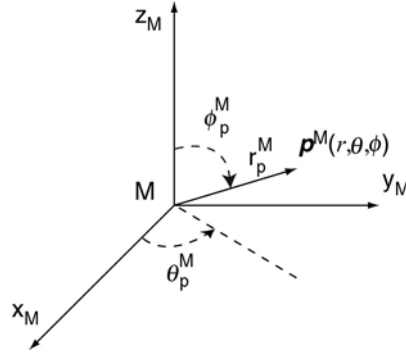


Figura 3.22: Transformación a coordenadas esféricas.

En base a esta información en coordenadas esféricas, se genera información sobre los obstáculos existentes en una determinada dirección, es decir para cada par (θ, ϕ) se calcula la fuerza de repulsión del objeto en la dirección correspondiente, que es inversamente proporcional a la distancia; a mayor distancia menor es la fuerza de repulsión. Por lo tanto la fuerza de repulsión viene definida como:

$$\mathbf{f}(\theta, \phi) = \begin{bmatrix} f_x(\theta, \phi) \\ f_y(\theta, \phi) \\ f_z(\theta, \phi) \end{bmatrix} = - \sum_{\forall p / p(r, \theta, \phi) \neq 0} \frac{\text{hist}(\theta, \phi)}{r} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (3.6)$$

Donde el $\text{hist}(\theta, \phi)$ representa el valor del histograma para dichos ángulos.

- Una vez se han calculado las fuerzas de repulsión, se añade un conjunto de fuerzas de atracción, que se corresponden con cada dirección en la que un rayo del haz no ha colisionado con otro componente. Por lo tanto estas fuerzas se suman a las anteriores de repulsión para establecer una dirección de desensamblado de entre el conjunto dado previamente por el grafo de caras de contacto.

Una vez seguidos estos pasos, se tiene generada la dirección de desensamblado a realizar, que se corresponde con la resultante de sumar todas las fuerzas obtenidas.

En la Figura 3.23 se muestra un ejemplo bidimensional de la determinación de la dirección de separación de un componente respecto al resto, en el que se representa el mundo discretizado y el histograma obtenido, a partir del cual es posible determinar la fuerza de repulsión y la dirección de separación.

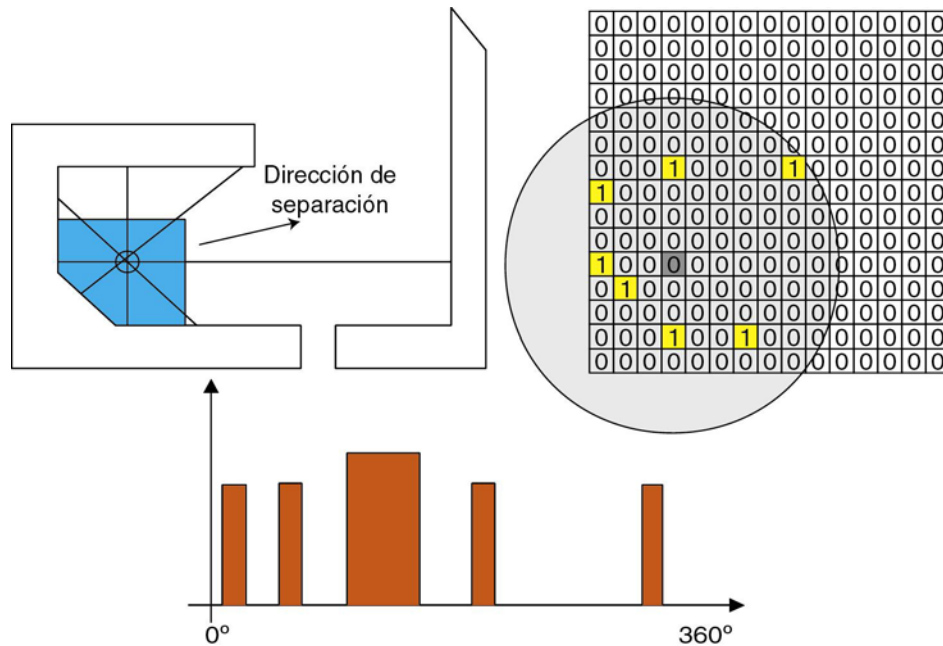


Figura 3.23: Determinación de la dirección de separación a partir de la obtención de la máxima fuerza de repulsión.

3.4. Generación de una trayectoria de desensamblado

Una vez determinada la dirección de separación de un componente del resto del montaje, ésta será válida si no existe ningún obstáculo en esa dirección. Por el contrario si existen obstáculos que impidan un desensamblado del componente en esa dirección, (Figura 3.23) se necesita determinar un conjunto de movimientos de separación simples, según la metodología explicada en el anterior apartado, que constituyan una trayectoria cartesiana para desensamblar un componente [211].

3.4.1. Estado del arte

En los estudios de accesibilidad de un componente en el contexto de desensamblado o de mantenimiento se emplean principalmente dos técnicas: búsqueda automática de trayectorias y la simulación interactiva con la ayuda de herramientas de robótica o de realidad virtual.

3.4.1.1. Búsqueda automática de trayectorias

Estas técnicas están basadas en el conocimiento del entorno de los componentes a manipular. Pueden ser de tres tipos:

- Globales: basadas en un conocimiento global del entorno.
- Locales: utilizan sólo una percepción local del entorno.
- Mixtas: usando principios locales y globales.

El entorno es denominado espacio de configuraciones. Este espacio, denominado C , es el conjunto de todas las configuraciones y contiene, por tanto:

- Espacio libre, denominado C -libre, representa el conjunto de las configuraciones sin colisión entre los componentes y los obstáculos.

- El espacio de los obstáculos, denominado C-obstáculos, representa el conjunto de configuraciones que provocan colisiones. C-obstáculos es el complementario de C-libre en C.

En la Figura 3.24 se puede ver una división de las búsquedas automáticas de trayectorias [218]:

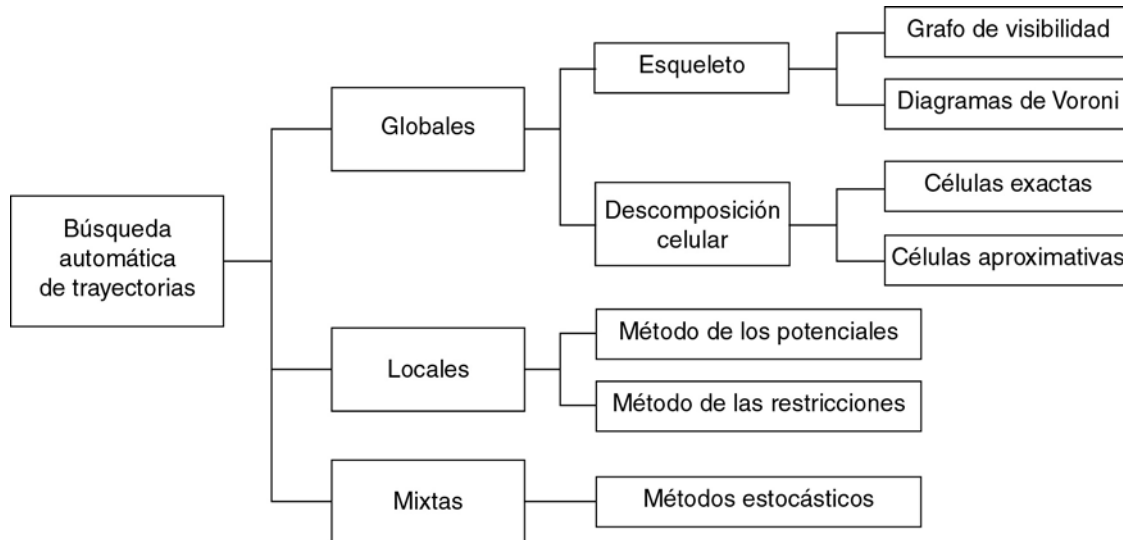


Figura 3.24: Técnicas de búsqueda automática de trayectorias (Fuente: [218]).

Globales

Las técnicas globales están basadas en la discretización del espacio de configuraciones. Una solución global construye una representación del espacio de configuraciones a partir del modelo completo del entorno. En general, este tipo de aproximación transforma el problema de búsqueda de trayectorias en un problema de búsqueda en un grafo. Se pueden clasificar en dos categorías:

- Esqueleto: intenta construir una representación del espacio de configuraciones con subespacios de dimensión más pequeña que el espacio de configuraciones. Las propiedades de conexión son guardadas en una nueva representación llamada esqueleto. En este planteamiento se utiliza una representación explícita de los espacios libres conexos. El conjunto de trayectorias en el espacio libre es almacenado en un grafo de manera que este grafo representa todas las regiones del espacio libre. De esta manera toda trayectoria en el espacio libre está representada por un conjunto de arcos y de nodos. Para determinar una trayectoria de una configuración inicial a otra final, basta con proyectar las configuraciones inicial y final sobre dos nodos del grafo. A continuación se utiliza un método de búsqueda para recorrer el grafo y encontrar el conjunto de nodos y arcos que describen la trayectoria. Se tienen, principalmente, dos planteamientos para el cálculo de los espacios libres:
 - Grafos de visibilidad: utilizados para entornos compuestos por obstáculos poligonales. El grafo es entonces construido con los nodos de los componentes y las configuraciones inicial y final [282].
 - Diagramas de Voronoi: construyendo el grafo libre a partir de puntos equidistantes entre los obstáculos [258].

- Descomposición celular: consiste en dividir el espacio de búsqueda en subespacios denominados células. La búsqueda de una trayectoria consiste entonces en la construcción de un camino entre dos subespacios. Cada célula libre de obstáculos representa un nodo. Dos nodos están conectados entre ellos si las células son adyacentes [173]. Las descomposiciones en células son exactas o aproximadas:
 - Para las exactas: hay que obtener una representación exacta de todas las células completamente libres de obstáculos.
 - En las aproximadas: el conocimiento de un subespacio del espacio libre puede ser representado, por ejemplo, en la forma de un quadtree. Para este tipo de representación, se consideran tres tipos de células: células libres, células C-obstáculos y células mixtas. Las células libres son subespacios del espacio libre, las células C-obstáculos son subespacios de C-obstáculos y las células mixtas contienen un subespacio del espacio libre y un subespacio de los C-obstáculos.

Las técnicas globales construyen una representación del espacio de configuraciones a partir de un modelo completo del entorno. Son eficaces para un pequeño número de grados de libertad pero, en general, son costosos en tiempo de cálculo. Uno de los inconvenientes de la utilización del grafo de visibilidad o bien de los diagramas de Voronoi es que se aplican solamente a dos dimensiones.

Locales

Este planteamiento utiliza una representación local del entorno para intentar generar una trayectoria hacia la configuración final. Esta basado en la optimización de una función orientada al guiado del manipulador hacia la configuración final. En general, se utiliza cuando la configuración inicial y final son vecinas. Las aproximaciones locales son también utilizadas en combinación con los métodos globales para alcanzar las configuraciones intermedias con el fin de mejorar la búsqueda de una trayectoria completa. Las aproximaciones locales principales son:

- Método de los potenciales: consideran el componente como una partícula bajo la influencia de fuerzas de atracción y repulsión. Estas fuerzas son generadas respectivamente por la configuración final y los obstáculos. Estas fuerzas pueden ser representadas por un campo de curvas equipotenciales [217]. La partícula se encuentra sometida a una energía potencial. Si se considera que la partícula está en un punto más alto que la posición final, la energía potencial obliga entonces a la partícula a descender por un camino libre de colisiones debido a la fuerza de repulsión de los obstáculos.
- Método de las restricciones: representa localmente el espacio de configuraciones por una lista de restricciones [52]. El problema de búsqueda de trayectorias es entonces transformado en un problema de optimización bajo restricciones o de minimización de una función objetivo. Esta función es definida a partir de la especificación de la configuración a alcanzar. La minimización bajo restricciones permite entonces determinar localmente la dirección de movimiento del

componente con el fin de alcanzar la configuración final sin colisión con los obstáculos.

Las técnicas locales son rápidas y eficaces. Sin embargo, es difícil definir una función potencial o restricciones que determine que un mínimo local representa la configuración final.

Mixtas

Las técnicas mixtas utilizan principios locales y globales. Tienen por objetivo: evitar los mínimos locales, permitir un número importante de grados de libertad, utilizar entornos complejos; así como, disminuir el tiempo de cálculo.

El método estocástico permite una descomposición de la búsqueda de la trayectoria [9]. Esta técnica propone la colocación de marcas intermedias definidas aleatoriamente. El módulo encargado de calcular una trayectoria sin colisiones para el transporte de objetos manipulados debe elegir un conjunto de posiciones intermedias que garanticen esquivar los obstáculos durante el desplazamiento. La complejidad de este problema está unida directamente al número de grados de libertad del componente.

Otro planteamiento propone muestrear aleatoriamente el espacio de configuraciones del robot [126]. En este caso, las configuraciones libres de colisiones, son alcanzadas con trayectorias simples. El resultado es un grafo denominado: *Probabilistic roadmap for Path Planning*. Las configuraciones inicial y final del robot son alcanzadas por las configuraciones libres de obstáculos en este mapa.

Otra solución consiste en la utilización de facultades humanas. En efecto, el hombre tiene la facultad de visualizar y de analizar rápidamente un entorno siguiendo la representación que se le propone. Entonces, el hombre gracias a su visión global ve que camino debe seleccionar. Un estudio presentado en [112] está basado en esta aproximación y utiliza muestras intermedias elegidas por el hombre a lo largo de un recorrido. La búsqueda de trayectorias es entonces realizada por sistemas autónomos que intentan alcanzar las configuraciones de paso sucesivas. La utilización de la visión global del hombre conduce a una división coherente del problema inicial de búsqueda de trayectorias.

Las técnicas mixtas utilizan principios locales y globales. Los métodos propuestos anteriormente son básicamente secuenciales. La búsqueda global está simplificada por la inserción de muestras intermedias. De esta manera, estos métodos confirman que las intervenciones secuenciales del hombre o una aproximación estocástica mejoran los tiempos de búsqueda.

3.4.1.2. Simulación interactiva

Actualmente, existen herramientas de robótica que permiten la simulación de robots. Estas herramientas permiten el desplazamiento interactivo de un objeto, es decir, un desplazamiento de manera secuencial siguiendo los seis grados de libertad del objeto manipulado. Este desplazamiento, generalmente largo y delicado, se realiza por el operador gracias a su facultad de visión global. De esta manera, una funcionalidad de detección de colisiones permite al operador definir una trayectoria que evite colisiones.

La manipulación interactiva es igualmente accesible en realidad virtual [269]. Con este propósito se utilizan periféricos de comando especiales. El operador manipula entonces el componente en un entorno virtual. De todas maneras, a pesar de la funcionalidad de detección de colisiones que permite al operador definir una trayectoria libre de colisiones, es necesario señalar la ausencia de evitación automática de colisiones con el entorno, ausencia de periféricos que hagan "retroceder el esfuerzo".

3.4.2. Solución utilizada

Para calcular cual es la trayectoria de desensamblado que debe realizar un determinado componente para separarse del resto, sigue el siguiente algoritmo, el cual se corresponde con una aproximación local que utiliza el método de los potenciales para su cálculo:

1. Determinar la dirección de desensamblado según las caras de contacto, tal y como se explicó en el apartado 3.3.
2. Determinar el punto de colisión más cercano al componente según la dirección elegida. Para ello se hace uso de un detector de colisiones que se explica en el siguiente apartado.
3. Una vez encontrado el punto de colisión más cercano y determinada la distancia a la que se encuentra dicho punto, se realiza el desplazamiento del componente según dicha dirección. De esta manera desaparecen algunas de las caras de contacto del componente actual y aparecen otras, información que es necesario actualizar en el grafo que representa el producto.
4. Una vez se ha desplazado el componente, se vuelve al paso 1 del algoritmo teniendo en cuenta la nueva configuración del producto y el entorno de trabajo discretizado con la información actual.

De esta manera se obtiene la secuencia de desplazamientos que conforman la trayectoria de desensamblado.

En la Figura 3.25 se muestra la secuencia que se sigue para obtener la trayectoria de desensamblado de un componente del producto de la Figura 3.23. La primera dirección de desensamblado mostrada en la Figura 3.25a, se corresponde con la obtenida por el algoritmo inicialmente, que permite desplazar el componente una cierta distancia (Figura 3.25b y Figura 3.25c) según la dirección de desensamblado obtenida. En ese momento el modelo se actualiza con las nuevas posibles colisiones y determina una nueva dirección de desplazamiento (Figura 3.25d); en la Figura 3.25e se muestra como se desplaza el componente hasta llegar a la posición representada en la Figura 3.25f, donde se calcula de nuevo la dirección de desensamblado, que en este caso, ya permite realizar el desensamblado del componente, con lo que se tendría calculada la trayectoria de desensamblado para el componente.

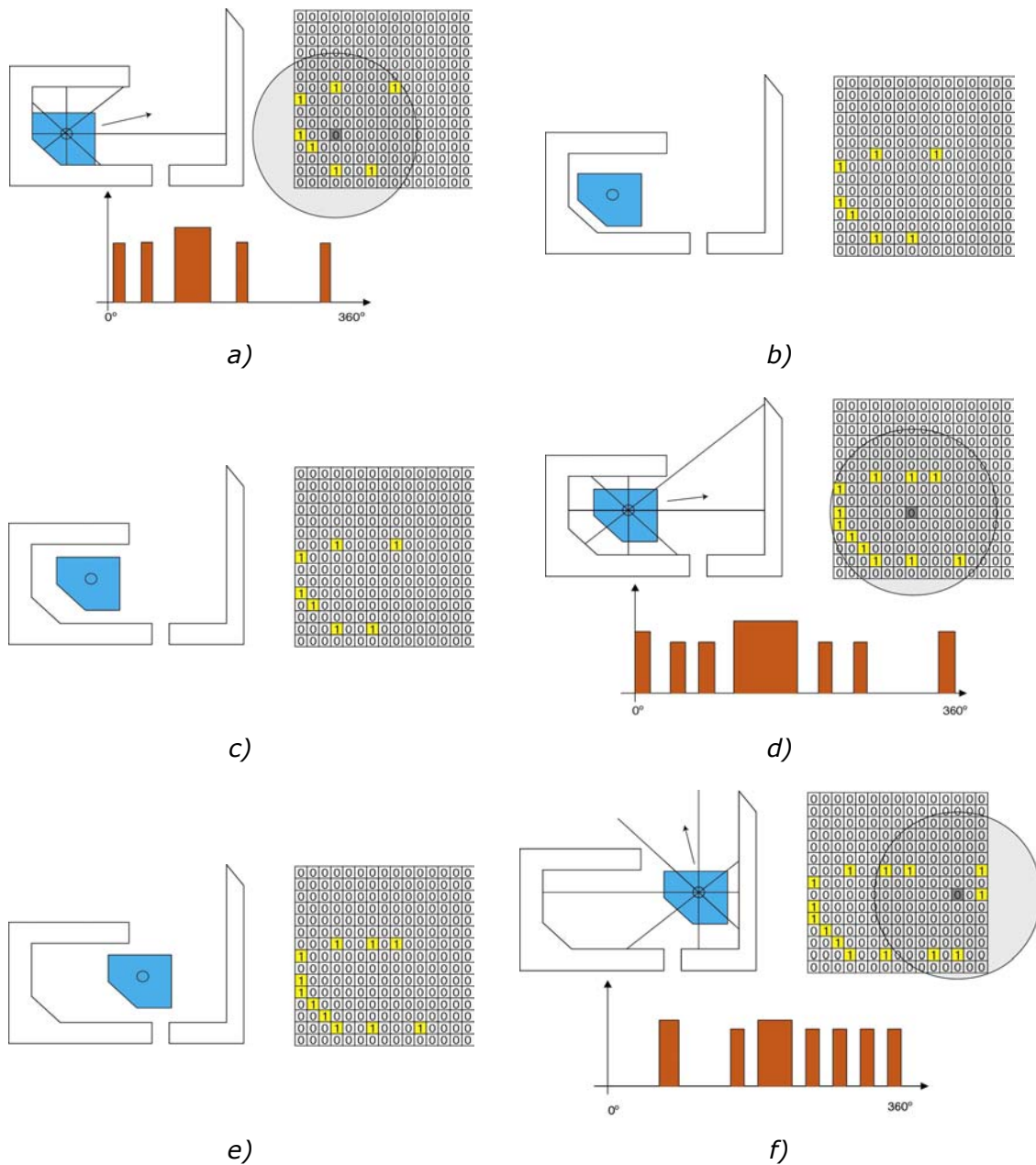


Figura 3.25: Secuencia del algoritmo para determinar la trayectoria de desensamblado de un componente en un producto formado por tres componentes .

Del algoritmo explicado para calcular la trayectoria de desensamblado, cabe destacar un dato importante sobre el que no se ha especificado un valor; este dato es la distancia que se desplaza el componente. Una primera aproximación podría pensar en desplazarlo lo máximo posible hasta que colisionara, en cuyo caso se podrían plantear bucles como el de la Figura 3.26.

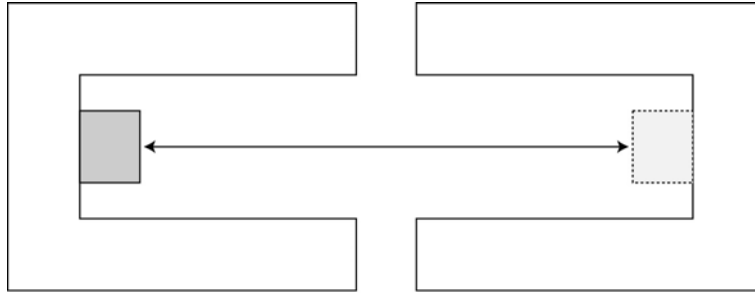


Figura 3.26: Bucle al desplazar totalmente un componente.

Una manera de evitar este problema, consiste en prohibir que un componente se desplace en la misma dirección, pero en sentido opuesto, al que se ha desplazado en el movimiento inmediatamente anterior.

Otro método sería desplazarlo sólo una celda del entorno discretizado. Este método plantea un costo computacional elevado, dado que se pueden generar una gran cantidad de desplazamientos en la misma dirección; pero obtiene la trayectoria de desensamblado que implica un menor desplazamiento total del componente.

3.5. Detección de colisiones

Otro aspecto de interés en la determinación de movimientos de desensamblado, es la detección de colisiones y la representación eficiente de la posición y orientación de los componentes. Una parte esencial, a la hora de determinar movimientos de desensamblado, es un sistema de razonamiento geométrico que pueda detectar colisiones entre los componentes en el espacio de trabajo, así como planificar caminos libres de colisiones a lo largo de este espacio de trabajo. Las restricciones en el movimiento del componente son causadas por los obstáculos correspondientes a superficies en el espacio de configuración del componente.

Bibliografía relacionada con la computación gráfica y la realidad virtual a menudo hace referencia a métodos para la determinación de colisiones entre objetos en movimiento [97].

Un objeto sólido tridimensional tiene tres grados de libertad de traslación y otros tres de rotación. Su orientación es típicamente especificada usando ángulos de Euler, pero esto provoca que las restricciones en el espacio de configuración contenga funciones trascendentales [174].

Hay algoritmos de planificación 3-D que utilizan distintos conceptos para representar la orientación: una representación angular [46], una representación usando los coeficientes de una matriz de rotación de tamaño 3x3, dando las restricciones algebraicas [235]. Esta última aproximación implica un espacio de configuración de 12 dimensiones, por lo que se tienen que introducir 6 restricciones adicionales para asegurar que todas las configuraciones corresponden a los coeficientes válidos de una matriz de rotación.

La orientación de un componente puede ser representada usando un cuaternio con cuatro componentes. Esto aumenta el espacio de configuración a 7 dimensiones, con una restricción adicional para asegurar que el cuaternio

tiene magnitud unidad. Debido a que la rotación de un vector por un cuaternio implica sólo multiplicación y suma, las restricciones serán algebraicas. De hecho es posible eliminar las variables extra usando las propiedades de escalado de los cuaternios.

Existen métodos para detectar si un componente poliédrico en movimiento colisiona con obstáculos también poliédricos [23]. En esta solución, los componentes se mueven con un movimiento de traslación uniforme así como con una rotación también uniforme a lo largo del espacio de configuración, que corresponde aproximadamente con una rotación uniforme en el espacio real. Expresando cada coordenada del objeto como una función polinomial de varios parámetros que varían a lo largo del camino, es posible detectar colisiones a lo largo de los caminos más generales. Así mismo si la rotación se representa mediante cuaternios las restricciones son puramente algebraicas en un espacio de mayor dimensión. Por simple manipulación, las restricciones pueden ser proyectadas en un espacio de seis dimensiones sin incrementar la complejidad. Usando esta formulación, se deriva un test eficiente y exacto para la intersección de un componente que se está trasladando y rotando entre medio de obstáculos.

3.5.1. Planteamiento utilizado

Como se ha comentado, se necesita calcular las posibles colisiones de un determinado componente con el resto del producto; para ser más exactos, se necesita conocer el punto de colisión de cada cara del componente que se está desensamblando (componente móvil), con cada una de las caras pertenecientes al resto de componentes del producto (componente estático).

Para ello se considera que el componente a desensamblar está en movimiento y el resto está estático, lo cual se corresponde con la notación presentado por Lozano-Pérez [174] para determinar las interacciones entre un componente poliédrico y obstáculos poliédricos, donde al componente en movimiento se le denomina A y al componente estático, el obstáculo, como B.

Teniendo en cuenta esta notación, existen tres tipos diferentes de colisiones que dan lugar a ciertas restricciones [23]:

- Tipo-A: Se producen cuando un vértice de A colisiona con una cara de B (Figura 3.27).

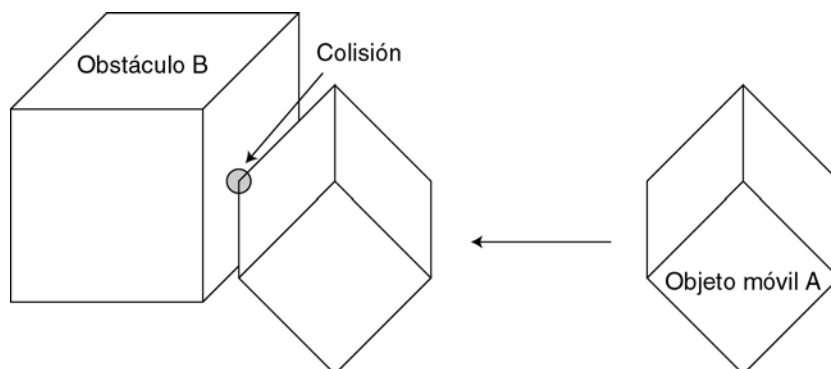


Figura 3.27: Colisión de tipo-A entre dos objetos.

- Tipo-B: Se producen cuando una cara de A colisiona con un vértice B (Figura 3.28).

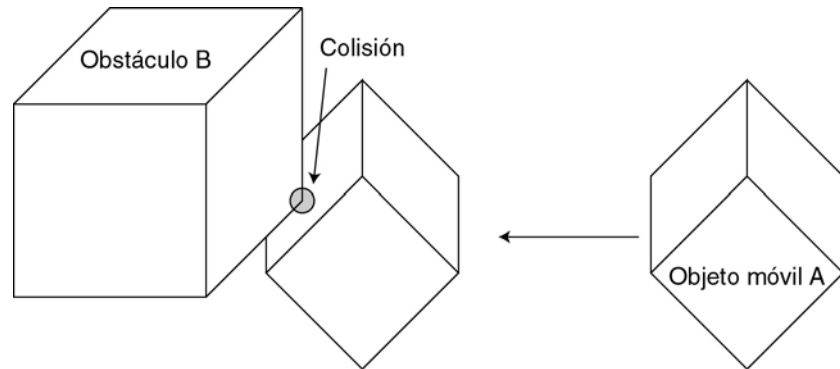


Figura 3.28: Colisión de tipo-B entre dos objetos.

- Tipo-C: Se producen cuando una arista de A colisiona con un arista B (Figura 3.29).

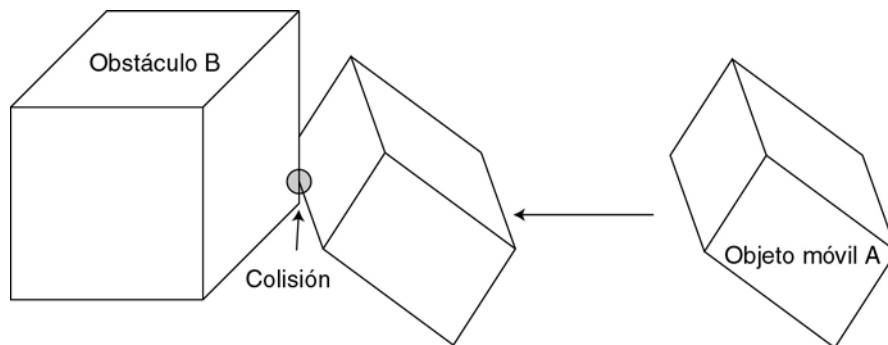


Figura 3.29: Colisión de tipo-C entre dos objetos.

De manera general, la localización de un objeto viene dada por un par (\mathbf{x}, \mathbf{Q}) , donde \mathbf{x} es un vector de posición, que indica el desplazamiento que ha sufrido el componente respecto del origen de coordenadas, y \mathbf{Q} es un cuaternino, que representa la rotación del componente respecto al origen. De tal manera que, la localización del componente se obtiene realizando su rotación respecto del origen, según el cuaternino \mathbf{Q} y posteriormente trasladándolo según \mathbf{x} .

Para cada uno de los contactos indicados existen una serie de restricciones entre ambos componentes, de tal manera, que se puede definir una región del espacio, en la que existe una interferencia entre el objeto en movimiento y los obstáculos.

Si el componente A se encuentra en el origen, entonces un punto \mathbf{y} en una cara de A debe satisfacer:

$$\mathbf{y} \cdot \mathbf{n}_A - d_A = 0 \quad (3.7)$$

Donde:

- \mathbf{n}_A es el vector unitario normal a la superficie de la cara dirigido hacia el exterior del componente.
- d_A es la distancia desde el origen a la cara en la dirección normal.

Además la rotación del componente por un cuaternino \mathbf{Q} viene dada por:

$$\mathbf{y} \cdot \mathbf{Rot}(\mathbf{n}_A, \mathbf{Q}) - d_A = 0 \quad (3.8)$$

Donde:

- $\mathbf{Rot}(\mathbf{n}_A, \mathbf{Q})$ es la rotación del vector \mathbf{n}_A por el cuaternino \mathbf{Q} .

Si se traslada el componente A según el vector \mathbf{x} , es el mismo efecto que una traslación de \mathbf{y} por $-\mathbf{x}$, ya que un punto perteneciente a la cara de A rotada y desplazada debe satisfacer:

$$(\mathbf{y} - \mathbf{x}) \cdot \mathbf{Rot}(\mathbf{n}_A, \mathbf{Q}) - d_A = 0 \quad (3.9)$$

De esta manera para calcular las colisiones de tipo-A se parte de la ecuación de una cara de B:

$$\mathbf{y} \cdot \mathbf{n}_B - d_B = 0 \quad (3.10)$$

Donde:

- \mathbf{n}_B es el vector unitario normal a la superficie de la cara dirigido hacia el exterior del componente.
- d_B es la distancia desde el origen a la cara en la dirección normal.

En la ecuación 3.10 se sustituye \mathbf{y} por un vértice de A denominado \mathbf{p}_A , que se encuentra rotado \mathbf{Q} y desplazado \mathbf{x} , de tal manera que la restricción queda:

$$(\mathbf{Rot}(\mathbf{p}_A, \mathbf{Q}) + \mathbf{x}) \cdot \mathbf{n}_B - d_B = 0 \quad (3.11)$$

Por otra parte para las colisiones de tipo-B, donde un vértice del obstáculo \mathbf{p}_B colisiona con una cara de A, la restricción corresponde a sustituir en la ecuación 3.9 el valor de \mathbf{y} por \mathbf{p}_B es:

$$(\mathbf{p}_B - \mathbf{x}) \cdot \mathbf{Rot}(\mathbf{n}_A, \mathbf{Q}) - d_A = 0 \quad (3.12)$$

Para calcular las colisiones de tipo-C se aplica el hecho de que dos rectas no colineales, que se intersectan en el espacio tridimensional, forman un plano. De tal manera que, definiendo las aristas como rectas, se puede calcular el plano perpendicular en el que se intersectan. En base a lo que se puede decir:

$$(\mathbf{y} - \mathbf{p}_B) \cdot (\mathbf{e}_A \times \mathbf{e}_B) = 0 \quad (3.13)$$

Donde:

- \mathbf{e}_A es el vector que define la dirección de la arista a considerar del componente A.
- \mathbf{e}_B es el vector que define la dirección de la arista a considerar del obstáculo B.
- \mathbf{p}_B es un punto del obstáculo.

Condición que cumple el plano que pasa por la arista de B y es paralelo a la arista de A. Para comprobar la intersección entre ambas aristas se sustituye \mathbf{y} por las coordenadas de un punto en la arista A, el punto \mathbf{p}_A . Dado que el objeto A ha sido rotado según \mathbf{Q} y trasladado según \mathbf{x} se tiene:

$$\mathbf{y} = \mathbf{Rot}(\mathbf{p}_A, \mathbf{Q}) + \mathbf{x} \quad (3.14)$$

Y además se puede ver que la dirección de la arista de A viene dada por:

$$\mathbf{Rot}(\mathbf{e}_A, \mathbf{Q}) \quad (3.15)$$

Con lo cual, sustituyendo en la ecuación 3.13 las ecuaciones 3.14 y 3.15 se obtiene que la restricción de tipo-C es la siguiente:

$$(\mathbf{Rot}(\mathbf{p}_A, \mathbf{Q}) + \mathbf{x} - \mathbf{p}_B) \cdot (\mathbf{Rot}(\mathbf{e}_A, \mathbf{Q}) \times \mathbf{e}_B) = 0 \quad (3.16)$$

Estas expresiones permiten determinar si un punto está obtenido realmente por una colisión y además saber que tipo de colisión lo ha provocado.

3.5.1.1. Restricciones debidas al movimiento del componente

Cuando un componente se encuentra en movimiento en una determinada dirección, se generan una serie de restricciones que dependen de la propia geometría del objeto, así como del movimiento realizado.

En este apartado, así como en el resto de la tesis, se va a considerar que los componentes no pueden rotar.

Estas restricciones se aplican sólo a una parte del espacio, la zona que se puede ver afectada por el desplazamiento del componente; para ser más exactos, se realiza un estudio de las colisiones de cada una de las caras del componente en la dirección del movimiento, de manera que, el espacio de búsqueda, se reduce a los lugares donde dicha cara puede colisionar con otra (Figura 3.30).

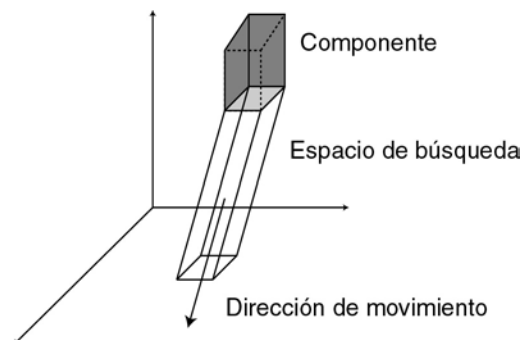


Figura 3.30: Espacio de búsqueda de colisiones, teniendo en cuenta las restricciones, por la dirección de desensamblado para la cara inferior de un componente.

De tal manera que cada arista de la cara que se está analizando, genera un plano paralelo a la dirección del movimiento, cuyo vector director apunta hacia el exterior del volumen encerrado por el movimiento de la cara.

Para calcular cada uno de los planos que definen las restricciones, se parte de los dos vértices que forman una arista (\mathbf{v}_1 y \mathbf{v}_2) de la cara, de la dirección de desensamblado que se está considerando \mathbf{d} y de otro vértice de la cara \mathbf{v}_3 ; este último vértice servirá para determinar el sentido del vector director del plano. Teniendo en cuenta esta información, se determina la ecuación del plano que pasa por dos puntos y contiene la dirección dada por un vector como:

$$\det \begin{vmatrix} 1 & 1 & 0 & 1 \\ x_{v_1} & x_{v_2} & x_d & x \\ y_{v_1} & y_{v_2} & y_d & y \\ z_{v_1} & z_{v_2} & z_d & z \end{vmatrix} = 0 \quad (3.17)$$

En donde expresado el plano como:

$$Ax + By + Cz + D = 0 \quad (3.18)$$

Se obtendría el siguiente valor para los parámetros A, B, C y D, teniendo en cuenta que se pretende que el vector director del plano sea unitario:

$$A' = y_{v_2}z_d - z_{v_2}y_d + z_{v_1}y_d - y_{v_1}z_d \quad (3.19)$$

$$B' = -x_{v_2}z_d + x_{v_2}x_d - z_{v_1}x_d + x_{v_1}z_d \quad (3.20)$$

$$C' = x_{v_2}y_d - y_{v_2}x_d + y_{v_1}x_d - x_{v_1}y_d \quad (3.21)$$

$$\text{Mod} = \sqrt{A'^2 + B'^2 + C'^2} \quad (3.22)$$

$$A = \frac{A'}{\text{Mod}}; \quad B = \frac{B'}{\text{Mod}}; \quad C = \frac{C'}{\text{Mod}} \quad (3.23)$$

$$D = \frac{-x_{v_1}(y_{v_2}z_d - z_{v_2}y_d) + x_{v_2}(y_{v_1}z_d - z_{v_1}y_d) - x_d(y_{v_1}z_{v_2} - z_{v_1}y_{v_2})}{\text{Mod}} \quad (3.24)$$

Teniendo en cuenta que todo punto (x, y, z) que se encuentre en la zona de búsqueda delimitada por el plano debe cumplir:

$$Ax + By + Cz + D \leq 0 \quad (3.25)$$

El tercer punto del polígono \mathbf{v}_3 debe cumplir la condición 3.25 con lo que:

$$Ax_{v_3} + By_{v_3} + Cz_{v_3} + D \leq 0 \quad (3.26)$$

Luego si no se cumple, el resultado obtenido en las ecuaciones 3.23 y 3.24 se debe multiplicar por -1 para obtener el plano que posea el sentido, de su vector director, adecuado para el resto de cálculos.

3.5.1.2. Detección de colisiones mediante un problema de programación lineal

Una vez definidos los distintos tipos de restricciones que aparecen a la hora de realizar el movimiento de una cara según una determinada dirección, se van a calcular los puntos de colisión con las caras de otros componentes del producto, mediante un problema de programación lineal [21].

Este método resulta muy útil para representar las restricciones, debido a que todas ellas se pueden expresar de manera sencilla mediante ecuaciones lineales, que van limitando el espacio de búsqueda de la solución.

Programación lineal

El método de programación lineal busca maximizar la función z para n variables independientes $\mathbf{x}=(x_1, x_2, \dots, x_n)$.

$$z = a_{01}x_1 + a_{02}x_2 + \dots + a_{0n}x_n \quad (3.27)$$

Donde se tiene como restricciones principales:

$$x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0 \quad (3.28)$$

Y además la función 3.27 se encuentra sometida a $M = m_1 + m_2 + m_3$ restricciones extras. De las que m_1 son de la forma:

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \leq b_i \quad (b_i \geq 0) \quad i = 1, \dots, m_1 \quad (3.29)$$

Y m_2 restricciones son de la forma:

$$a_{j1}x_1 + a_{j2}x_2 + \dots + a_{jn}x_n \geq b_j \quad (b_j \geq 0) \quad j = m_1 + 1, \dots, m_1 + m_2 \quad (3.30)$$

Y por último m_3 restricciones son de la forma:

$$a_{k1}x_1 + a_{k2}x_2 + \dots + a_{kn}x_n = b_k \geq 0 \quad k = m_1 + m_2 + 1, \dots, m_1 + m_2 + m_3 \quad (3.31)$$

Donde los coeficientes a_{ij} pueden ser positivos, negativos o cero.

Existen dos clases de métodos para la solución de problemas de programación lineal: el método denominado en la bibliografía como *simplex*, y los métodos interiores. El método *simplex* [2] considera el problema como un espacio n -dimensional donde cada vector \mathbf{x} es un punto en el espacio, y cada restricción define un hiperplano que divide el espacio en dos. De esta manera las restricciones definen una región cerrada del espacio que contiene las soluciones del problema [255]. El algoritmo comienza con un vértice en una superficie n -dimensional y se mueve en la dirección que consigue una mejora de la solución hasta encontrar la óptima.

Los métodos interiores comienzan en un punto dentro del conjunto de soluciones válidas y van aproximando el punto hacia el óptimo [2].

Como se puede ver para el caso que nos ocupa, interesa el método *simplex* en lugar de un método interior, ya que no se conoce a priori ningún punto del conjunto de soluciones.

Existen dos casos en los que no se puede encontrar una solución:

- Cuando no hay solución posible, las restricciones son incompatibles.
- No existe un máximo. Este caso se da cuando una o más variables pueden tomar un valor infinito y seguir cumpliendo las restricciones, de tal manera que la función objetivo no estaría acotada.

Generación del problema de programación lineal

Para describir el problema de programación lineal e identificar claramente cada una de las restricciones existentes se va a realizar el diseño esquemático de las restricciones utilizando un par de componentes, tal y como se muestran en la Figura 3.31.

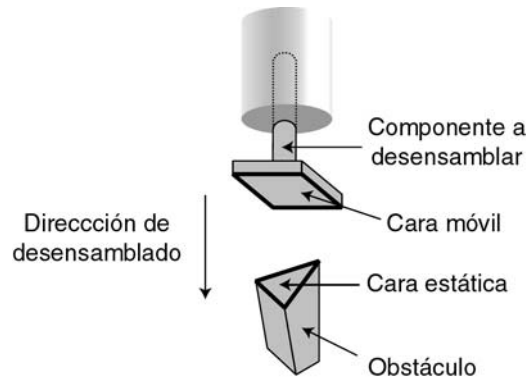


Figura 3.31: Componentes sobre los que se aplica el problema de programación lineal.

Si se considera la cara del componente móvil implicada en la detección de la colisión como:

$$f(\mathbf{x}) = Ax + By + Cz + D \quad (3.32)$$

Se tendría un problema ya que se tiene un término independiente, D , que en la ecuación objetivo no existe (Ecuación 3.27). Para solventar este problema se ha optado por aumentar la dimensión del espacio de tal manera que se introduce una nueva variable v cuyo valor es 1 con lo que la ecuación objetivo queda:

$$f(\mathbf{x}) = Ax + By + Cz + Dv \quad (3.33)$$

Y por tanto hay que introducir una primera restricción, para que el sistema no varíe con la introducción de la nueva variable, que es:

$$v = 1 \quad (3.34)$$

Teniendo esto en cuenta, la función a maximizar es $-f(\mathbf{x})$ que vale:

$$\text{Maximizar} \equiv -Ax - By - Cz - Dv \quad (3.35)$$

Donde además de la restricción comentada, se deben aplicar otras restricciones que dependen de los componentes implicados y la dirección del movimiento. En estas restricciones se ha hecho directamente la extrapolación de ellas, para que incluyan la variable v :

- Restricciones por los planos paralelos a la dirección de movimiento que pasan por cada par de vértices de las aristas de la cara del componente móvil (Figura 3.32). Si se considera que la ecuación del plano obtenido es de la forma:

$$f_m(\mathbf{x}) = A_m x + B_m y + C_m z + D_m v \quad (3.36)$$

Entonces la restricción tiene la forma:

$$A_m x + B_m y + C_m z + D_m v \leq 0 \quad (3.37)$$

Esta restricción limita el espacio de búsqueda de soluciones basándose en la geometría de la cara móvil y en la dirección en la que se está realizando el movimiento.

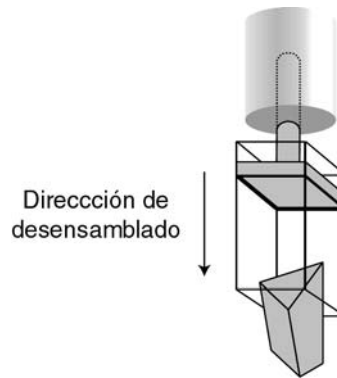


Figura 3.32: Restricciones por los planos paralelos a la dirección de movimiento de la cara del componente móvil.

- Restricciones por los planos paralelos a la dirección de movimiento que pasan por cada par de vértices de las aristas de la cara del componente estático que se está considerando (Figura 3.33).

Si se considera que la ecuación del plano obtenido es de la forma:

$$f_e(\mathbf{x}) = A_e x + B_e y + C_e z + D_e v \quad (3.38)$$

La restricción es de la forma:

$$A_e x + B_e y + C_e z + D_e v \leq 0 \quad (3.39)$$

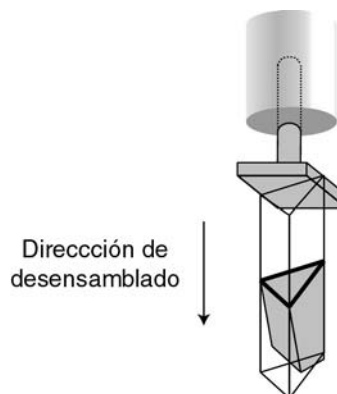


Figura 3.33: Restricciones por los planos paralelos a la dirección de movimiento de la cara del componente estático.

Esta restricción sirve para limitar el volumen del espacio por el que tiene que desplazarse la cara del componente móvil para colisionar con la cara estática.

- Restricción por la cara que se está considerando del componente móvil. (Figura 3.34). Si se considera la ecuación del plano de la cara como:

$$f_{cm}(\mathbf{x}) = A_{cm} x + B_{cm} y + C_{cm} z + D_{cm} v \quad (3.40)$$

La restricción es de la forma:

$$A_{cm} x + B_{cm} y + C_{cm} z + D_{cm} v \geq 0 \quad (3.41)$$

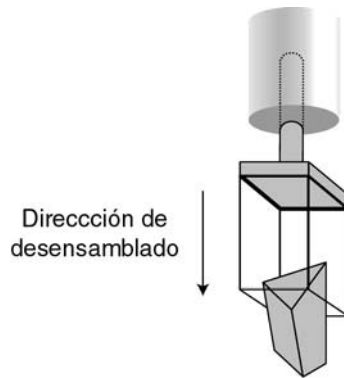


Figura 3.34: Restricción por la cara del componente móvil.

Esta restricción representa el espacio que se encuentra delante de la cara móvil.

- Restricción por la cara que se está considerando del componente estático. (Figura 3.35). Si se considera que la ecuación del plano de la cara como:

$$f_{ce}(x) = A_{ce}x + B_{ce}y + C_{ce}z + D_{ce}v \quad (3.42)$$

Con lo que la restricción tiene la forma:

$$A_{ce}x + B_{ce}y + C_{ce}z + D_{ce}v = 0 \quad (3.43)$$

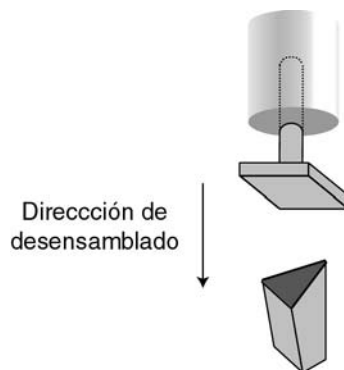


Figura 3.35: Restricción por la cara del componente estático.

Esta restricción implica que la colisión tiene que producirse en el plano que genera la cara estática, ya que se colisiona con él.

De esta manera se tiene definida la zona del espacio donde se puede producir la colisión entre ambas caras (Figura 3.36).

Con lo que se plantea el problema de programación lineal correspondiente, el cual obtiene el punto de colisión entre la cara móvil y la cara del componente estático (Figura 3.36).

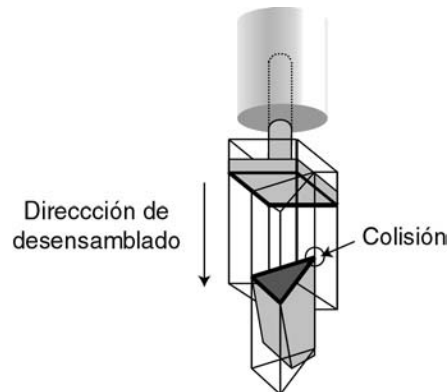


Figura 3.36: Zona del espacio donde se puede producir colisión y la colisión.

3.6. Cálculo de la trayectoria de desensamblado de una fuente de alimentación

En este apartado se ilustran los conceptos vistos durante el presente capítulo, aplicándolos al cálculo de la trayectoria de desensamblado de la fuente de alimentación de un ordenador, considerando solamente la existencia de dos componentes: la fuente de alimentación y la carcasa interior del mismo.

En la Figura 3.37 se muestran ambos componentes así como la relación de posición existente entre ellos.

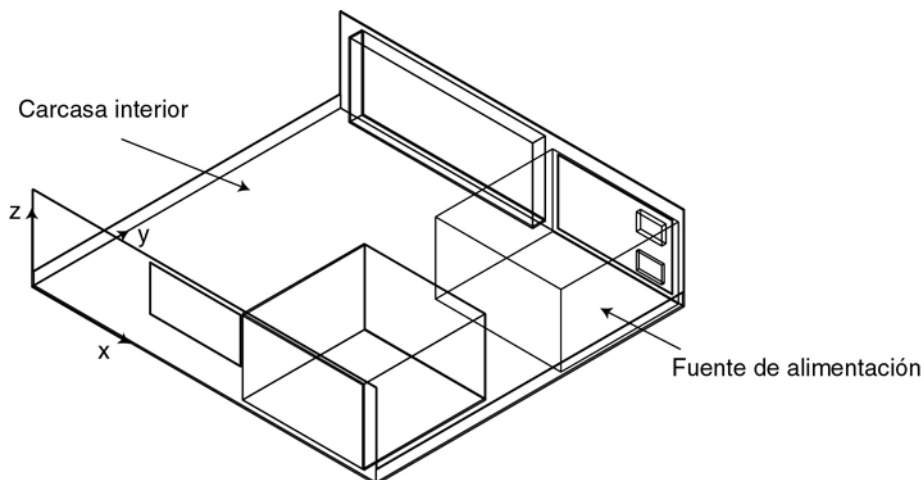


Figura 3.37: Modelo geométrico de la carcasa interior y la fuente de alimentación.

En base a la información geométrica de cada uno de los componentes, se genera el grafo de adyacencia correspondiente.

Para la fuente de alimentación, se tiene un grafo de adyacencia formado por 16 caras rectangulares, denominadas como $p_1, p_2, p_3, \dots, p_{16}$ (Figura 3.38).

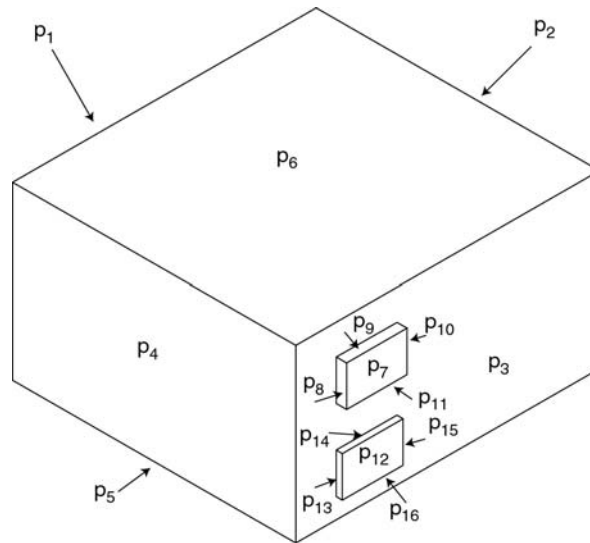


Figura 3.38: Identificación de las caras en la fuente de alimentación.

Como puede observarse cada una de las caras está formada por cuatro puntos, cuyas coordenadas así como el plano que forman, se detallan a continuación. Para ello se supone que el origen de coordenadas se encuentra en la esquina inferior izquierda del ordenador, tal y como muestra la Figura 3.37.

Cara	Punto 1	Punto 2	Punto 3	Punto 4	Plano
P ₁	(254, 225, 85)	(404, 225, 85)	(404, 225, 0)	(254, 225, 0)	-y + 225 = 0
P ₂	(254, 225, 85)	(254, 225, 0)	(254, 366, 0)	(254, 366, 85)	-x + 254 = 0
P ₃	(254, 366, 0)	(254, 366, 85)	(404, 366, 85)	(404, 366, 0)	y - 366 = 0
P ₄	(404, 366, 85)	(404, 366, 0)	(404, 225, 0)	(404, 225, 85)	x - 404 = 0
P ₅	(404, 225, 0)	(254, 225, 0)	(254, 366, 0)	(404, 366, 0)	-z = 0
P ₆	(254, 225, 85)	(404, 225, 85)	(404, 366, 85)	(254, 366, 85)	z - 85 = 0
P ₇	(354, 370, 50)	(384, 370, 50)	(384, 370, 70)	(354, 370, 70)	-y - 370 = 0
P ₈	(384, 366, 50)	(384, 370, 50)	(384, 370, 70)	(384, 366, 70)	x - 384 = 0
P ₉	(384, 370, 70)	(384, 366, 70)	(354, 366, 70)	(354, 370, 70)	z - 70 = 0
P ₁₀	(354, 366, 70)	(354, 370, 70)	(354, 370, 50)	(354, 366, 50)	-x + 354 = 0
P ₁₁	(354, 370, 50)	(354, 365, 50)	(384, 366, 50)	(384, 370, 50)	-z + 50 = 0
P ₁₂	(354, 368, 10)	(384, 368, 10)	(384, 368, 30)	(354, 368, 30)	-y - 368 = 0
P ₁₃	(384, 368, 10)	(384, 368, 30)	(384, 366, 30)	(384, 366, 10)	x - 384 = 0
P ₁₄	(384, 368, 30)	(384, 366, 30)	(354, 366, 30)	(354, 368, 30)	z - 30 = 0
P ₁₅	(354, 366, 30)	(354, 368, 30)	(354, 368, 10)	(354, 368, 10)	-x + 354 = 0
P ₁₆	(354, 368, 10)	(354, 366, 10)	(384, 366, 10)	(384, 368, 10)	-z + 10 = 0

Por otro lado, para la carcasa interior del ordenador, se genera un grafo de adyacencias que indique las caras que posee, así como la información relevante a las mismas. Para este componente se han especificado sólo 17 caras, que son las que se consideran más relevantes para el resto de la resolución, ya que la definición de la totalidad de la carcasa poseería muchas más y su desglose es innecesario para los cálculos subsiguientes. En la Figura 3.39 se muestran las caras más relevantes identificadas sobre el modelo geométrico del componente.

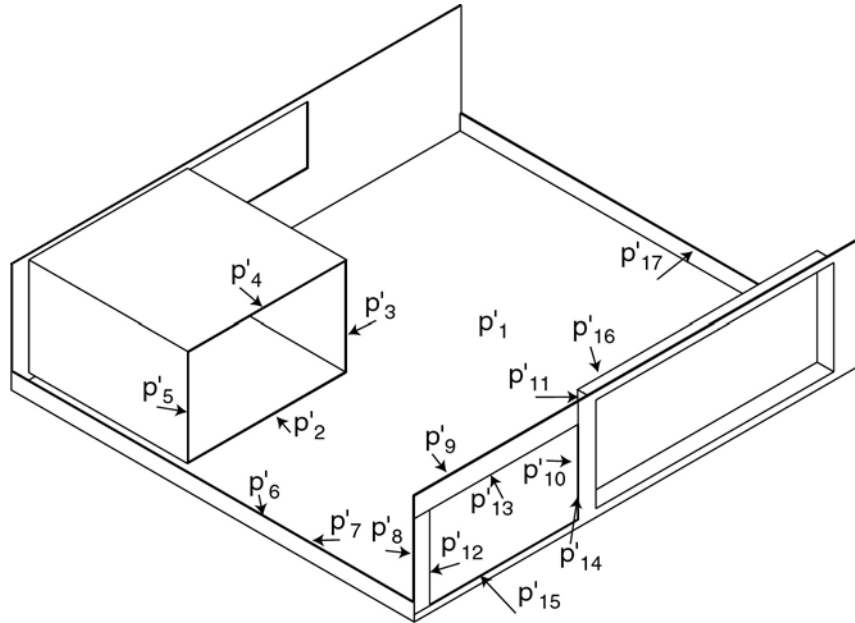


Figura 3.39: Identificación de las caras en la carcasa interior.

En la siguiente tabla se muestra la información relativa a las coordenadas de cada una de las caras de la carcasa interior, así como la información sobre los planos que forman.

Cara	Punto 1	Punto 2	Punto 3	Punto 4	Plano
P' ₁	(0, 0, 0)	(0, 366, 0)	(408, 366, 0)	(408, 0, 0)	$z = 0$
P' ₂	(394, 145, 6)	(250, 145, 6)	(250, 145, 7)	(394, 145, 7)	$y - 145 = 0$
P' ₃	(249, 145, 7)	(250, 145, 7)	(250, 145, 94)	(249, 145, 94)	$y - 145 = 0$
P' ₄	(250, 145, 94)	(250, 145, 95)	(394, 145, 95)	(394, 145, 94)	$y - 145 = 0$
P' ₅	(394, 145, 94)	(393, 145, 94)	(393, 145, 7)	(394, 145, 7)	$y - 145 = 0$
P' ₆	(409, 366, 14)	(408, 366, 14)	(408, 1, 14)	(409, 1, 14)	$z - 14 = 0$
P' ₇	(408, 366, 14)	(408, 1, 14)	(408, 1, 0)	(408, 366, 0)	$-x + 408 = 0$
P' ₈	(409, 366, 81)	(395, 366, 81)	(395, 366, 5)	(395, 366, 81)	$-y + 366 = 0$
P' ₉	(409, 366, 99)	(409, 366, 81)	(244, 366, 81)	(244, 366, 99)	$-y + 366 = 0$
P' ₁₀	(259, 366, 81)	(259, 366, 5)	(244, 366, 5)	(244, 366, 81)	$-y + 366 = 0$
P' ₁₁	(244, 351, 93)	(244, 351, 5)	(244, 366, 5)	(244, 366, 93)	$x - 244 = 0$
P' ₁₂	(395, 366, 81)	(395, 366, 5)	(395, 367, 5)	(395, 367, 81)	$-x + 395 = 0$
P' ₁₃	(395, 366, 81)	(395, 367, 81)	(259, 367, 81)	(259, 366, 81)	$-z + 81 = 0$
P' ₁₄	(259, 367, 81)	(259, 366, 81)	(259, 366, 5)	(259, 367, 5)	$x - 259 = 0$
P' ₁₅	(259, 366, 5)	(259, 367, 5)	(395, 367, 5)	(395, 366, 5)	$z - 5 = 0$
P' ₁₆	(244, 351, 93)	(244, 351, 5)	(25, 351, 5)	(25, 351, 93)	$-y + 351 = 0$
P' ₁₇	(0, 0, 0)	(0, 0, 14)	(0, 366, 14)	(0, 366, 0)	$x = 0$

Utilizando esta información, se calcula el grafo de caras de contacto, que genera las relaciones mostradas en la Figura 3.40. Que expresa los contactos existentes entre la cara correspondiente a la base de la fuente de alimentación (p_5) con la correspondiente de la carcasa interior (p'_1); y además las relaciones entre la parte posterior de la carcasa (p'_8 , p'_9 , p'_{10}) con la parte posterior de la fuente de alimentación (p_3).

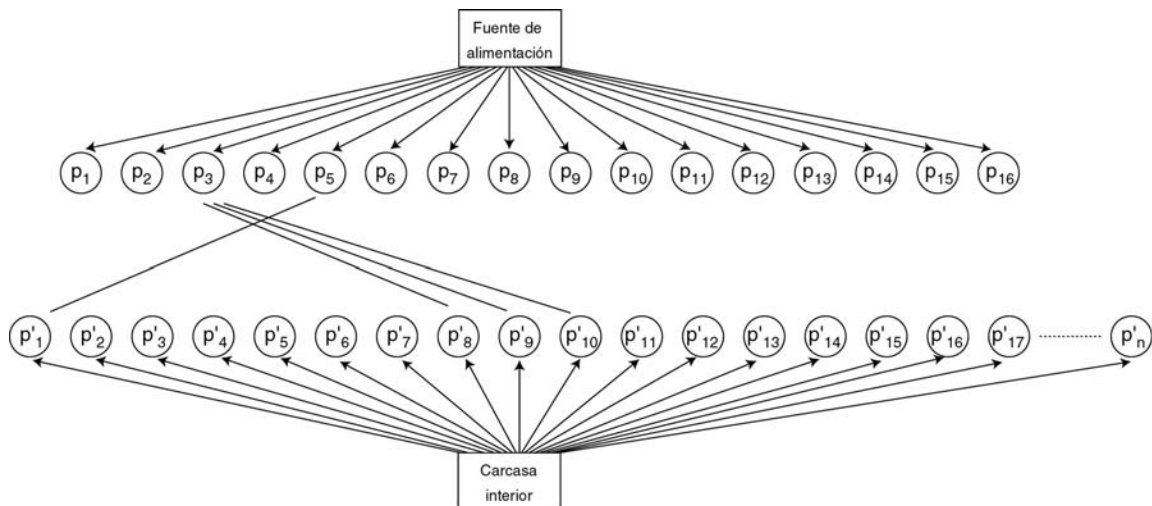


Figura 3.40: Grafo de relaciones.

Utilizando esta información se genera el conjunto de posibles direcciones de desensamblado para el componente "fuente de alimentación", mediante la intersección de las superficies esféricas correspondientes a cada cara de contacto. Cabe destacar al respecto que las caras p_8 , p_9 y p_{10} de la carcasa interior pertenecen al mismo plano, luego tendrán la misma esfera de direcciones de desensamblado.

En la Figura 3.41 se muestra cada una de las esferas que representan las posibles direcciones de desensamblado del componente así como el resultado de su combinación.

De entre el conjunto de direcciones de desensamblado, ahora se tiene que calcular cuál es la que posee una menor probabilidad de colisión con el entorno; para ello se procede a la generación de un haz de rayos desde el centro del objeto, con lo que se obtiene información acerca de la existencia de otros componentes en las cercanías y a qué distancia se encuentra cada uno de ellos. Con esta información se construye un histograma que indica las direcciones con mayor probabilidad de colisión.

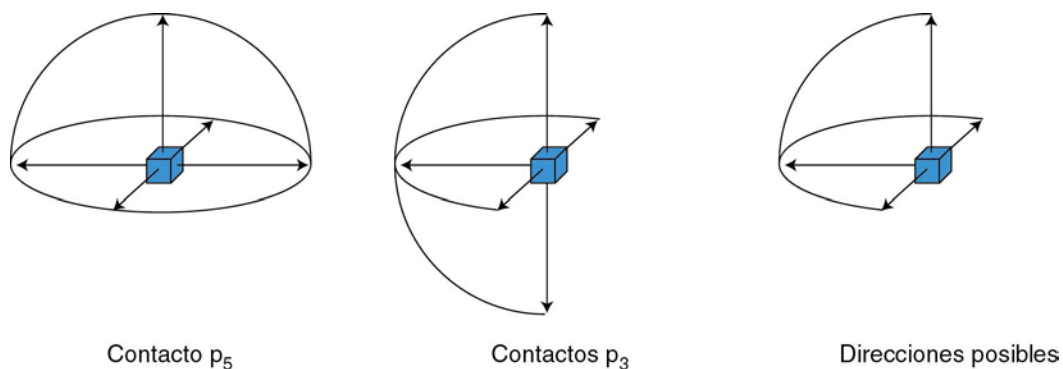


Figura 3.41: Direcciones de desensamblado para la fuente de alimentación.

En este caso se ha considerado un haz de rayos desde el centro de la fuente de alimentación con un ángulo de 45° entre cada rayo. De esta manera sólo existen fuerzas de atracción, en el cuarto de esfera considerado, porque no colisiona ningún rayo con otro componente. Por lo tanto la dirección de desensamblado viene definida por la suma de dichas direcciones:

$$d = (0, -0.7071, 0.7071) \quad (3.44)$$

Una vez se ha determinado la dirección de desplazamiento, hay que aplicar el método de programación lineal para calcular la distancia a la que se encuentra un obstáculo según dicha dirección, por lo tanto, se aplica el problema de programación lineal a todas aquellas caras de la fuente de alimentación cuyo vector director, del plano correspondiente, forme un ángulo comprendido entre -90° y 90° , que serían las caras p_1 , p_6 , p_9 y p_{14} . Por otra parte, los puntos de colisión con las caras del componente estático a considerar son aquellos que formen con la dirección de desensamblado un ángulo entre 90° y -90° . Por lo tanto se utilizan las caras p'_2 , p'_3 , p'_4 , p'_5 , p'_{13} .

De esta manera hay que plantear un problema de programación lineal para cada posible cara del componente móvil con cada una de las posibles caras del componente estático. El problema de programación lineal, para la cara p_1 y la cara p'_2 , queda definido por las siguientes ecuaciones:

Función objetivo:

$$\text{Maximizar } \equiv y - 225v \quad (3.45)$$

Las restricciones provenientes de los planos paralelos a la dirección del movimiento, tanto de la cara del componente móvil como de la cara del componente estático son:

$$-0.707107y - 0.707107z + 159.0990v \leq 0 \quad (3.46)$$

$$x - 394v \leq 0 \quad (3.47)$$

$$0.707107y + 0.707107z - 107.4802v \leq 0 \quad (3.48)$$

$$-x + 254v \leq 0 \quad (3.49)$$

Hay que considerar que, en este caso, deberían ser 8 las ecuaciones correspondientes a los planos paralelos a la dirección del movimiento, 4 correspondientes a las caras del componente móvil y los otros 4 a las caras del componente estático. Estas ecuaciones se han reducido a 4 porque las otras 4 definen planos paralelos menos restrictivos, con lo que se puede simplificar el problema, utilizando, en caso de planos paralelos y con el mismo vector director, aquellos que son más restrictivos.

Por otro lado la restricción por la cara móvil es:

$$-y + 225v \geq 0 \quad (3.50)$$

Por último queda la restricción debida a la cara estática:

$$y - 145v = 0 \quad (3.51)$$

Y la correspondiente a v , que se muestra en la ecuación 3.34.

El algoritmo de programación lineal, teniendo en cuenta estas ecuaciones informa al sistema que no hay punto de colisión en la dirección del movimiento entre las caras implicadas, por lo tanto, dicha cara del componente estático no impide el desplazamiento en la dirección seleccionada. Esta característica se

puede observar fijándose en la Figura 3.37, donde se aprecia que las cara p_1 y p'_2 no pueden llegar a afectarse, teniendo en cuenta la actual dirección de desensamblado. Esto mismo sucede con la cara p_1 y la p'_{13} .

A continuación hay que seguir resolviendo el problema de programación lineal para el resto de caras del componente estático respecto de la cara actual del componente móvil.

Si consideramos p_1 respecto a la cara estática p'_4 el problema de programación lineal planteado es el siguiente:

Función objetivo:

$$\text{Maximizar } \equiv y - 225v \quad (3.52)$$

Las restricciones correspondientes de los planos paralelos a la dirección del movimiento, tanto de la cara del componente móvil como de la cara del componente estático son:

$$-0.707107y - 0.707107z + 168.9985v \leq 0 \quad (3.53)$$

$$x - 394v \leq 0 \quad (3.54)$$

$$0.707107y + 0.707107z - 169.7056v \leq 0 \quad (3.55)$$

$$-x + 254v \leq 0 \quad (3.56)$$

La restricción por la cara móvil es:

$$-y + 225v \geq 0 \quad (3.57)$$

Por último queda la restricción debida a la cara estática:

$$y - 145v = 0 \quad (3.58)$$

En donde se obtiene como punto de corte más cercano entre ambas caras el punto:

$$(x, y, z) = (254, 145, 94) \quad (3.59)$$

Lo que implica una distancia máxima permitida en la dirección del movimiento de 113mm.

Por otra parte, si se calcula el problema de programación lineal de la cara móvil p_1 con la cara estática p'_3 se obtiene que no existe solución, es decir dichas caras no se cortan en el desplazamiento a lo largo de la dirección seleccionada, tal y como pasaba con la cara p'_2 y p'_{13} .

Sin embargo para la cara estática p'_5 sí que existe solución al problema de programación lineal, que viene definido por las siguiente restricciones:

Restricciones por los planos paralelos a la dirección del movimiento:

$$-0.707107y - 0.707107z + 159.0990v \leq 0 \quad (3.60)$$

$$x - 394v \leq 0 \quad (3.61)$$

$$0.707107y + 0.707107z - 168.9985v \leq 0 \quad (3.62)$$

$$-x + 393v \leq 0 \quad (3.63)$$

La restricción correspondiente a la cara móvil es la misma que en los casos anteriores, ya que se trata de la misma cara móvil. Por la misma razón la función a maximizar también es la misma.

La última restricción a considerar es la correspondiente a la cara estática:

$$y - 145v = 0 \quad (3.64)$$

En donde se obtiene, como punto de corte más cercano entre ambas caras, el punto:

$$(x, y, z) = (393, 145, 80) \quad (3.65)$$

que se encuentra a una distancia de 113mm según la dirección del movimiento. Esta distancia era de esperar si se observa la estructura del producto, ya que es el mismo valor obtenido para p'_4 . Esto es debido a que ambas caras definen el mismo plano, pero se diferencian en las aristas que lo delimitan.

Considerando como cara móvil p_6 , se define un nuevo problema de programación lineal con cada una de las caras correspondientes de la carcasa interior. En este caso se puede apreciar directamente en la Figura 3.37 que no existe colisión de dicha cara con ninguna de las del componente estático.

Por lo tanto hay que seguir calculando las colisiones correspondientes a p_9 y p_{14} . Estas caras tampoco colisionan con ninguna otra en la dirección seleccionada; de esta manera ya se tiene resuelto el problema, que implica un desplazamiento máximo de 113mm.

El componente se desplaza una unidad en dicha dirección; en este caso se ha tomado el valor de una unidad como 50mm. Con lo que se desplaza en la dirección del movimiento dicha cantidad, quedando, el producto, tal y como muestra la Figura 3.42.

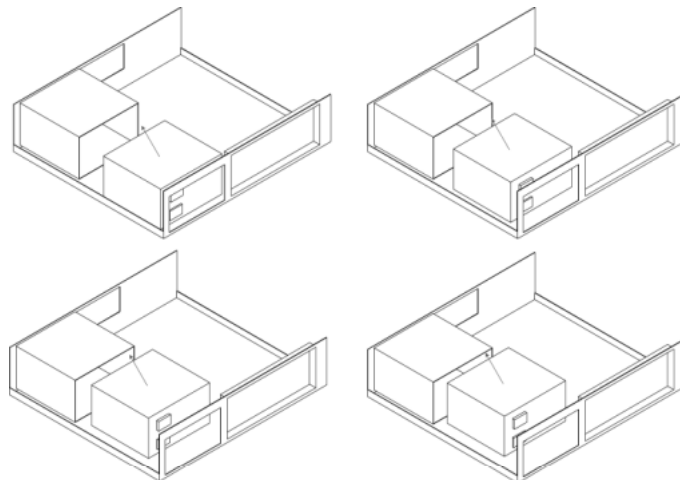


Figura 3.42: Secuencia de desplazamiento de la fuente de alimentación según la dirección $(0, -0.7071, 0.7071)$.

Tras realizar el desplazamiento del objeto, hay que actualizar el grafo de relaciones del producto. En este caso no existiría ninguna relación entre las caras de los dos componentes, ya que no existe ningún contacto. Por lo tanto, no se eliminaría ninguna dirección para el cálculo de las trayectoria de desensamblado.

Además hay que actualizar los valores correspondientes a los polígonos que forman el componente "fuente de alimentación" para que contemplen el desplazamiento, con lo que las caras quedarían definidas de la siguiente manera:

Cara	Punto 1	Punto 2	Punto 3	Punto 4	Plano
P ₁	(254, 190, 120)	(404, 190, 120)	(404, 190, 35)	(254, 190, 35)	-y + 190 = 0
P ₂	(254, 190, 120)	(254, 190, 35)	(254, 331, 35)	(254, 331, 120)	-x + 254 = 0
P ₃	(254, 331, 35)	(254, 331, 120)	(404, 331, 120)	(404, 331, 35)	y - 331 = 0
P ₄	(404, 331, 120)	(404, 331, 35)	(404, 190, 35)	(404, 190, 120)	x - 404 = 0
P ₅	(404, 190, 35)	(254, 190, 35)	(254, 331, 35)	(404, 331, 35)	-z + 35 = 0
P ₆	(254, 190, 120)	(404, 190, 120)	(404, 331, 120)	(254, 331, 120)	z - 120 = 0
P ₇	(354, 335, 85)	(384, 335, 85)	(384, 335, 105)	(354, 335, 105)	-y - 335 = 0
P ₈	(384, 331, 85)	(384, 335, 85)	(384, 335, 105)	(384, 331, 105)	x - 384 = 0
P ₉	(384, 335, 105)	(384, 331, 105)	(354, 331, 105)	(354, 335, 105)	z - 105 = 0
P ₁₀	(354, 331, 105)	(354, 335, 105)	(354, 335, 85)	(354, 331, 85)	-x + 354 = 0
P ₁₁	(354, 335, 85)	(354, 360, 85)	(384, 331, 85)	(384, 335, 85)	-z + 85 = 0
P ₁₂	(354, 333, 45)	(384, 333, 45)	(384, 333, 65)	(354, 333, 65)	-y - 333 = 0
P ₁₃	(384, 333, 45)	(384, 333, 65)	(384, 331, 60)	(384, 331, 45)	x - 384 = 0
P ₁₄	(384, 333, 65)	(384, 331, 65)	(354, 331, 65)	(354, 333, 65)	z - 65 = 0
P ₁₅	(354, 331, 65)	(354, 333, 65)	(354, 333, 45)	(354, 333, 45)	-x + 354 = 0
P ₁₆	(354, 333, 45)	(354, 331, 45)	(384, 331, 45)	(384, 333, 45)	-z + 45 = 0

Tras generar un nuevo haz de rayos, se obtiene un histograma más detallado del entorno, y se selecciona la dirección de desensamblado como:

$$d = (0.7071, 0, 0.7071) \quad (3.66)$$

Con este resultado hay que volver a aplicar el problema de programación lineal para cada una de las caras del componente móvil, la "fuente de alimentación", con las caras del componente estático, la "carcasa interior".

Una vez realizado este cálculo, se observa que no existe ninguna colisión entre ambos componentes, con lo que, ya se tiene definida la trayectoria de desensamblado del componente, como un desplazamiento según la dirección definida en la ecuación 3.44 de 50mm, seguido de un desplazamiento según la dirección 3.66, que le permite desplazarse hasta el infinito, es decir, hasta el punto en el que se considera que el componente ya no interfiere con el resto, y de esa posición, trasladarlo hasta el almacén correspondiente.

3.7. Evaluación del algoritmo

En este apartado se presenta un estudio sobre el coste del algoritmo planteado a lo largo del capítulo.

Este método implica un coste en función del número de polígonos que compongan cada componente. Si se supone que el componente móvil posee n polígonos y el resto del producto esta compuesto por m polígonos entonces el coste del algoritmo viene determinado, una vez que se selecciona la dirección de desensamblado, por el coste de calcular a que distancia se puede desplazar el componente en la dirección señalada sin que colisione con otro componente. Para ello hay que plantear para cada cara del componente móvil un problema

de programación lineal con cada una de las caras del resto del producto. Lo que implicaría $n \cdot m$ problemas de programación lineal, que pueden ser reducidos a una cuarta parte si se consideran sólo las caras móviles cuyo vector director esté en el plano de la dirección de movimiento con lo que se tendría $n/2$ caras; y de igual manera si se consideran sólo las caras estáticas cuyo vector director se encuentre en la dirección opuesta a la dirección del movimiento tendremos que considerar $m/2$ caras.

En la Figura 3.43 se aprecia el coste de calcular las colisiones con otros componentes según la dirección de desensamblado para un número caras variable.

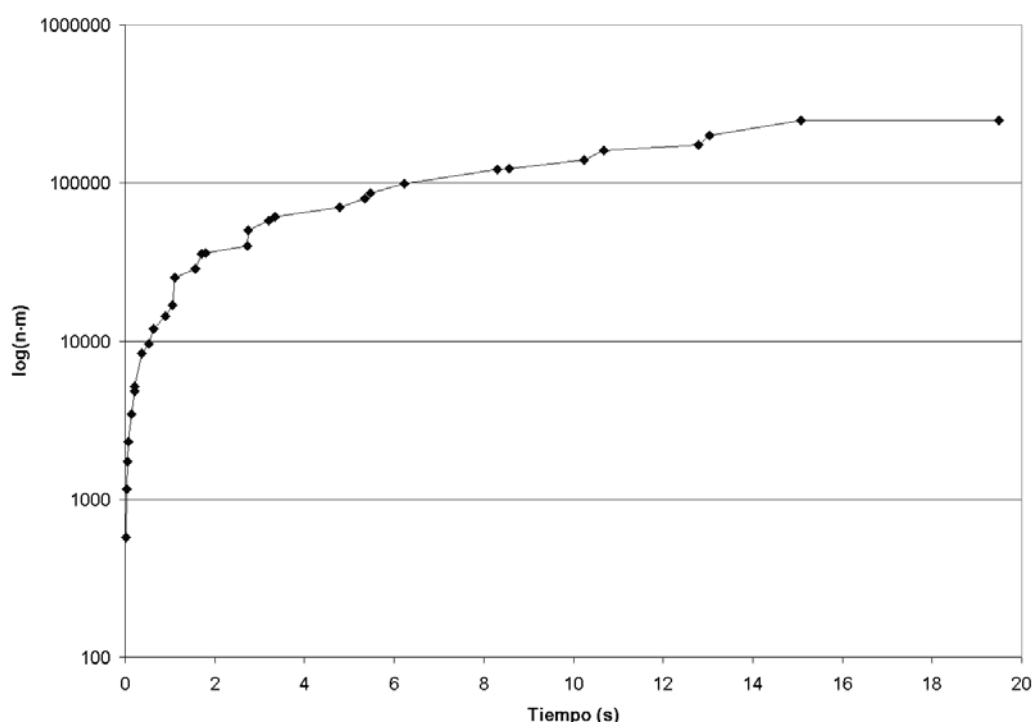


Figura 3.43: Coste del algoritmo de cálculo de colisiones según las caras existentes en el producto.

Este valor viene definido en el eje de ordenadas por el producto $n \cdot m$, correspondiente a la cantidad total de veces que hay que ejecutar el algoritmo de programación lineal para realizar la detección de las posibles colisiones. Se ha expresado dicha magnitud en valor logarítmico para una mayor claridad en la representación gráfica.

En el eje de abscisas se ha representado el tiempo empleado en la realización de los problemas de programación lineal en segundos.

Este tiempo se ha calculado utilizando un ordenador Pentium III a 450Mhz con 128Mbs de memoria RAM para realizar la ejecución de los problemas de programación lineal.

Como puede apreciarse, en la Figura 3.43, al aumentar el número de caras a analizar aumenta rápidamente el coste computacional necesario para llevarlo a cabo.