

# Interferencias entre las piezas de un ensamble mecánico contenido en un fichero STEP

**Edwin Estévez Parra**

Correo electrónico:edwin@cadcam.uho.edu.cu  
Universidad de Holguín, Cuba

**Artículo Original**

**Ricardo Lorenzo Ávila Rondón**

Correo electrónico:ricardo@cadcam.uho.edu.cu  
Universidad de Holguín, Cuba

## Resumen

La detección de las interferencias entre las piezas de un producto contribuye a determinar la factibilidad de las secuencias de ensamble según la geometría de las piezas involucradas. Sin embargo, en el ensamble pueden existir piezas con superficies de forma libre, lo que hace más complejo determinar automáticamente la relación de interferencia entre ellas. El objetivo de esta investigación es presentar un método para la obtención de las interferencias entre las piezas con superficies prismáticas, cilíndricas y de forma libre de un ensamble. El método es automatizado con una herramienta informática desarrollada sobre la biblioteca libre y de código abierto PythonOCC basada en OpenCascade. Se describen las técnicas para la extracción de la información geométrica del ensamble contenida en un fichero CAD 3D con formato estándar STEP, la detección de las colisiones entre las piezas en las direcciones de desensamble y la generación de las matrices de interferencia. Para demostrar la eficacia del método se analizan dos ejemplos de ensambles de productos mecánicos y se obtienen como resultado las matrices de interferencia correspondientes a las seis direcciones principales de desensamble.

Palabras claves: interferencia, ensamble, fichero CAD 3D, STEP

Recibido: 15 de marzo del 2014      Aprobado: 9 de octubre del 2014

## INTRODUCCIÓN

El conjunto de las secuencias en las que se puede ensamblar un producto constituye la base para la planificación del proceso del ensamble, en el cual se selecciona la secuencia mejor si se consideran los costos y tiempos de las operaciones, entre otros criterios. La obtención de las interferencias entre las piezas de un producto contribuye a determinar la factibilidad de las secuencias de ensamble según la geometría de las piezas involucradas. La factibilidad geométrica es la precondition para deducir las relaciones de precedencia del ensamble/desensamble a partir de las cuales generar el conjunto de las secuencias.

Algunos autores han representado la información de la relación entre las piezas de un ensamble en forma matricial.

Dini y Santochi [1] describieron un procedimiento para generar las secuencias posibles de ensamble, basado en un modelo matemático del producto, obtenido a través de la definición de tres matrices: matriz de interferencia, matriz de contacto y matriz de conexión. En este estudio, la matriz de interferencia representa las interferencias entre las piezas de un producto en los ejes de coordenadas principales  $+x$ ,  $+y$ ,  $+z$ . Huang y Huang [2] presentaron un método para almacenar los datos de las matrices de interferencia en la matriz de precedencia del desensamble, donde las posibles direcciones de desensamble están representadas por un número binario. Sinanoglu y Borklu [3] describieron un método para determinar los estados de la factibilidad geométrica mediante las matrices de intersección, las cuales son

utilizadas para representar la interferencia entre las piezas durante la operación de ensamble. Estas matrices se definen a lo largo de los seis ejes principales del sistema de coordenadas cartesianas y contienen elementos booleanos que representan la información de interferencia. Sin embargo, la representación matricial de Dini y Santochi [1] ha sido la de mayor utilización en investigaciones posteriores.

Otros estudios tratan algoritmos para determinar la interferencia entre las piezas del ensamble. Baldwin [4] propuso varios métodos y herramientas informáticas para la generación de las secuencias de ensamble mecánico. Estos métodos solo consideraban las interferencias de las piezas del ensamble causadas por condiciones de contacto, con las cuales la herramienta informática generaba posibles subensambles y realizaba consultas de la forma si/no a un usuario, para registrar su factibilidad. Sin embargo, el autor recomienda enriquecer sus métodos para automatizar completamente la generación de las secuencias de ensamble, de tal manera que las consultas planteadas al usuario fueran respondidas automáticamente con el uso de la geometría del ensamble. Como resultado, propone comprobar la interferencia geométrica ocasionada por la interferencia global de la pieza por medio de tres técnicas diferentes: prueba de interferencia volumétrica, prueba de interferencia de emisión de rayos y prueba de interferencia de proyección de superficies.

Sambhoos, Koc y Nagi [5] utilizaron un algoritmo de emisión de rayos para identificar las relaciones de contacto entre piezas clasificadas en dos tipos: indirecta e interferencia. La información geométrica era obtenida de un fichero con formato STEP (estándar para el intercambio de los datos del modelo del producto). No obstante, los métodos desarrollados solo consideraban las piezas en contacto con forma regular como cilíndrica y prismática. Pan, Smith y Smith [6] propusieron un método computacional para extraer la información geométrica de un fichero STEP, analizar las relaciones de interferencia entre las piezas del ensamble, y generar seis matrices donde se representaban las relaciones de interferencia entre las piezas en las direcciones de ensamble de los ejes principales de coordenada (+x, -x, +y, -y, +z, -z). Sin embargo, este método solo manipula piezas prismáticas y cilíndricas, y no considera piezas con superficies de forma libre. Para resolver dicha limitante, estos autores recomiendan utilizar aproximaciones triangulares. En este estudio, el método abordado manipula piezas al considerar cualquier superficie que la caracterice con el uso de una aproximación triangular según su descripción geométrica.

El método empleado para detectar las colisiones se denomina *Detección de interferencia múltiple*, y ha sido el más utilizado en el contexto de la detección de las colisiones en tiempo real entre objetos, ya que permite reducir el problema de detección de colisiones a múltiples llamadas a pruebas de interferencia estática [7]. La interferencia entre los objetos se comprueba en un lapso de tiempo determinado. Jiménez, Thomas, y Torras [7] apuntan que si este lapso de tiempo es muy grande podría no detectarse una colisión, y si es muy pequeño, podría ser costoso computacionalmente, pues se generarían demasiadas comprobaciones de

interferencia. Idealmente, el próximo lapso de tiempo debe ser el menor tiempo en el que puede ocurrir una colisión. Existen varias estrategias [7] que estiman este tiempo de diferentes maneras, lo que disminuiría la cantidad de comprobaciones de interferencia, pero estas se encuentran fuera del alcance de este estudio. El lapso de tiempo utilizado en esta investigación es pequeño y se ha comprobado que con este parámetro se detectan todas las colisiones entre las piezas del ensamble. No obstante, podría darse el caso que se realicen pruebas de interferencia en un momento en el que las piezas aún no hayan colisionado, por lo que se sugiere que se utilice una de estas estrategias en próximos estudios.

Pan, Smith y Smith [6] utilizan un método diferente para detectar las colisiones entre dos piezas. En su estudio, la existencia de una interferencia es comprobada si las proyecciones de dos piezas en un plano perpendicular a una dirección principal de ensamble se solapan. Además, para analizar el problema de planificación del ensamble, afirman que no es necesario utilizar el método de *Detección de interferencia múltiple*, pues fue diseñado para detectar las colisiones en tiempo real de objetos en movimiento en un entorno de simulación física. Sin embargo, en la vida real, el proceso de ensamble está sujeto a restricciones técnicas originadas por la estabilidad.

La estabilidad es un importante aspecto de la modularidad de los componentes de un ensamble y también desempeña un papel significativo en la factibilidad del ensamble paralelo. La estabilidad se refiere a la coherencia de un subensamble al estar encontrado bajo la acción de fuerzas, las cuales podrían ser fuerzas externas, como la gravedad y las vibraciones, y las fuerzas internas como las que son causadas por la deformación y el magnetismo, entre otras [8].

Las restricciones geométricas son frecuentemente estudiadas al asumirse que las fuerzas no están presentes, como es el caso de Pan, Smith y Smith [6] y la presente investigación. Sin embargo, el método presentado podría ampliarse para permitir el análisis de las relaciones de interferencia en ambientes donde las piezas o subensambles estén sometidas a la acción de varias fuerzas para determinar su estabilidad en el ensamble.

Las matrices de interferencia representan la restricción del movimiento de un componente por otros componentes en una dirección específica.

La matriz de interferencia correspondiente a la figura 1 al comprobarse las relaciones de interferencia en la dirección de desensamble +z es la siguiente:

$$F_{+z} = \begin{array}{c|cc} & A & B \\ \hline A & 0 & 1 \\ B & 0 & 0 \end{array}$$

En esta representación  $F_{+z}$  significa que el movimiento de la pieza A está restringida en la dirección +z por la pieza B; el movimiento del componente B no está restringido por la pieza A en la dirección +z. Esta matriz contiene la

información completa, porque la matriz de interferencia que se refiere a la dirección opuesta es la transpuesta de la original [8].

El objetivo de esta investigación es presentar un método para la obtención de las interferencias entre las piezas con superficies prismáticas, cilíndricas y de forma libre de un ensamble.

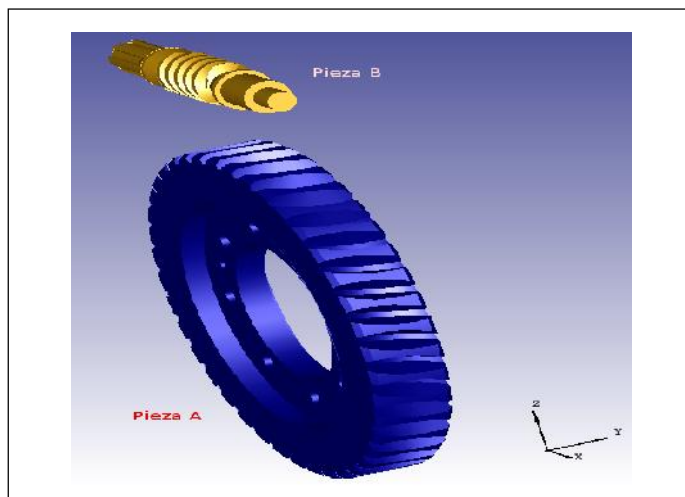


Fig. 1. Piezas A y B compuestas por superficies complejas

## MATERIALES Y MÉTODOS

Para el desarrollo de este trabajo se contó con la información de las piezas pertenecientes a un ensamble, la cual previamente ha sido salvada en un fichero de formato ISO-10303, también conocido como formato STEP. Este formato ya es común en la mayoría de los sistemas CAD comerciales existentes hoy en día.

La información mencionada se procesó en una computadora Pentium IV con las siguientes características: procesador dual core con velocidad de 2.0 GHz, memoria RAM de 2GB, espacio de disco duro necesario para su ejecución de 500 MB. El sistema operativo en que se ejecutó la herramienta informática desarrollada fue Ubuntu Linux 12.04 LTS a 64 bits, aunque también se realizaron pruebas en el sistema operativo Windows XP a 32 bits.

El módulo desarrollado se programó en el lenguaje Python, versión 2.7.3, y se utilizaron las bibliotecas de OpenCascade 6.5 por medio de PythonOCC, versión 0.6, así como las propias del PythonOCC. PythonOCC\* es una plataforma de desarrollo CAD/CAE/PLM 3D para el lenguaje de programación Python. Proporciona funciones tales como: operaciones topológicas y geométricas avanzadas, intercambio de datos en varios formatos, mallado 2D y 3D, simulación de cuerpos rígidos, y modelado paramétrico. Está basado fundamentalmente en el proyecto OCE\*\* (OpenCascade para la Comunidad). La topología del OCE está diseñada con referencia a la norma STEP ISO-10303-42 (Part 42 [9]), lo que significa que los tipos geométricos primitivos de OCE son una implementación de las entidades algebraicas y geométricas básicas compatible con STEP. En esta

\*<https://github.com/tpaviot/pythonocc>.

\*\*<https://github.com/tpaviot/oce>.

investigación, las funciones de intercambio de los datos CAD que implementa PythonOCC para extraer la información geométrica de los ficheros STEP Part 21 son utilizadas en una herramienta informática desarrollada con el lenguaje Python.

A continuación se presentan los métodos para la extracción de la información geométrica de las piezas relativas a un ensamble. Luego se describe el proceder para obtener las interferencias entre dichas piezas en las direcciones de los tres ejes principales de desensamble en la forma de matrices de interferencia que servirán como datos de entrada a una herramienta de planificación de desensamble.

### Extracción de la información geométrica de las piezas contenidas en un fichero STEP

Para habilitar el intercambio de los datos de un producto a través de su ciclo de vida, la ISO ha desarrollado el STEP. La especificación de los datos de STEP está representada por los Protocolos de Aplicación (APs). Entre los APs para el intercambio de los ficheros CAD se encuentran el AP203 y AP214, los cuales están centrados principalmente en la representación de la información geométrica en formato BRep (Representación de Fronteras) de las piezas y los ensambles mecánicos. Además, STEP Part 21 [10] especifica un formato de intercambio con el uso de una codificación en texto plano de los datos del producto. Actualmente, varias herramientas CAD generan ficheros con este formato para garantizar que el intercambio de sus datos se ajuste a este estándar. Por estas razones, el método propuesto utiliza la información geométrica para detectar la interferencia entre las piezas, la cual es extraída de un fichero con este formato.

Al cargarse un fichero STEP con las funciones de PythonOCC hay que tener en cuenta que el ensamble es representado como una entidad topológica de tipo *Compound* y cada pieza es de tipo *Solid*. Estos tipos pertenecen a un tipo genérico denominado *Shape*. Para una mejor comprensión y uso de las funcionalidades para importar ficheros con formato STEP, y manejar la información geométrica y topológica ver ejemplos distribuidos con el PythonOCC. Una vez que se haya cargado en la aplicación el fichero STEP, los datos extraídos se mapean automáticamente en estructuras de datos del OCE representadas con formato BRep. Entonces, cada pieza del ensamble representada por una forma de tipo *Solid* es almacenada en una lista de piezas P por la herramienta informática implementada.

### Generación de la aproximación triangular de las superficies de las piezas

Una vez que se tenga la lista de piezas (P), se generan las aproximaciones triangulares de sus superficies. La técnica utilizada en este momento es el mallado de los sólidos (que representan las piezas) con el constructor de mallas para estructuras BRep del OCE. Debe tenerse en cuenta que una mayor precisión de la malla permite una aproximación más exacta a la superficie original de la pieza, pero influye en el aumento del costo computacional. Posteriormente se transforman estos datos a una estructura denominada *TrimeshData*. Para llevar a cabo esta tarea, se utiliza una función similar a `_compute_trimesh()` que se encuentra

implementada en la clase *DynamicShape* del módulo DYN (simulación de cuerpos rígidos) del PythonOCC.

Hay que destacar que la malla resultante de cada pieza está representada por una estructura que se utiliza en el próximo paso y que es resultado de aplicar transformaciones geométricas al considerar el centro de masa de cada pieza. Al terminarse el proceso de mallado de todas las piezas, las mallas resultantes se almacenan en la lista de mallas *M*.

#### Detección de las interferencias entre las piezas

En los sistemas de ensamble automatizados, la mayoría de los componentes mecánicos son ensamblados a lo largo de los ejes principales [6]. Por lo que se hace necesario determinar si una pieza tiene una trayectoria libre de colisión al ser ensamblada o desensamblada en una dirección determinada. El método que se presenta analiza la interferencia entre las piezas a lo largo de los tres ejes principales de desensamble (+x, +y, +z). A partir de la lista *M*, que contiene el mallado superficial de la geometría de cada pieza, se comprueban las relaciones de interferencia entre cada par de mallas. Si una colisión existe entre las mallas  $M_i$  y  $M_j$ , donde  $M_i, M_j$  pertenecen a *M*, con  $M_i \neq M_j$  (es decir, no se analiza la interferencia de una malla consigo misma), mientras  $M_i$  se mantiene estática y  $M_j$  se mueve en una de las direcciones principales de desensamble, entonces se registra la información de interferencia entre las dos piezas involucradas. Al finalizar, la información de interferencia entre todas las piezas es almacenada por cada dirección de

desensamble en un conjunto de matrices de interferencia, para mantener la compatibilidad con algunas herramientas que utilizan esta información para la planificación de las secuencias de desensamble.

El módulo DYN del PythonOCC fue utilizado para la detección de las interferencias entre las piezas. Contiene una clase llamada *DynamicSimulationContext* que permite detectar la ocurrencia de colisiones entre las mallas triangulares. En este caso, se fija un objeto malla y se le aplica un movimiento (velocidad) a otro objeto malla en una dirección de desensamble. Se repite este proceso por cada par de mallas de las piezas en las tres direcciones principales de desensamble. De esta manera se detecta la información de interferencia entre las dos piezas analizadas y es almacenada en las matrices de interferencia.

#### Creación de las matrices de interferencia

Para el caso de la matriz  $F_{+z}$ , se analiza la interferencia entre las piezas A y B, al mantener fija la pieza B y mover la pieza A en la dirección de desensamble +z. Al detectarse una colisión, entonces se almacena el valor 1 en la segunda columna de la primera fila, sino se pone el 0 como valor. Los elementos de la diagonal principal de la matriz siempre van a ser igual a 0, porque no se analiza la relación de interferencia de una pieza consigo misma. Los pasos del método propuesto son resumidos en formato de pseudocódigo en la figura 2.

```

ALGORITMO: Relaciones de interferencia entre piezas del ensamble
ENTRADA: Modelo de ensamble CAD 3D con formato STEP
SALIDA: Matrices de interferencia  $F_{+x}, F_{-x}, F_{+y}, F_{-y}, F_{+z}, F_{-z}$ 

COMENZAR:
  /** Extracción de la información geométrica de las piezas */
  Extraer piezas del fichero STEP
  Almacenar piezas en la lista P, número de piezas n
  /** Generar la aproximación triangular de cada pieza */
  FOR k ← 1 to n {
    M[k] = obtenerAproxTriangular( P[k] )
  }
  /** Obtención de las interferencias entre las piezas */
  FOR i ← 1 to n {
    FOR j ← 1 to n {
      IF i <> j {
        /**  $F_{-x} = F_{+x}^T, F_{-y} = F_{+y}^T, F_{-z} = F_{+z}^T$  */
        IF  $M_i$  colisiona con  $M_j$  al moverse en la dirección de desensamble +x {
           $F_{+x}[i][j] = 1$ 
        } ELSE {  $F_{+x}[i][j] = 0$  }
        IF  $M_i$  colisiona con  $M_j$  al moverse en la dirección de desensamble +y {
           $F_{+y}[i][j] = 1$ 
        } ELSE {  $F_{+y}[i][j] = 0$  }
        IF  $M_i$  colisiona con  $M_j$  al moverse en la dirección de desensamble +z {
           $F_{+z}[i][j] = 1$ 
        } ELSE {  $F_{+z}[i][j] = 0$  }
      }
    }
  }
FINALIZAR

```

Fig. 2. Algoritmo para obtener las relaciones de interferencias entre piezas de un ensamble

## RESULTADOS

El método presentado para la creación de las seis matrices de interferencia a partir de la información geométrica extraída de un fichero con formato STEP fue comprobado para dos ejemplos de ensamble que contienen piezas con superficies complejas.

Una herramienta informática fue desarrollada al utilizar la plataforma PythonOCC con la que se implementó el método propuesto. La herramienta informática requiere como dato de entrada un fichero STEP que pudo haber sido generado por una herramienta CAD comercial, y el resultado son las matrices de interferencia correspondientes a las seis direcciones de desensamblaje principales. La figura 3 muestra un ensamble de un disco de freno. La pieza 1 corresponde al rotor del disco de freno, que contacta con la pieza 2 que es el centro del disco de freno. Las piezas 3, 4, 5 son pernos que fijan el centro del disco de freno al rotor del disco de freno. Este ensamble fue simplificado para una mejor comprensión del análisis del caso. Primeramente se determinan las relaciones de interferencia entre las piezas mientras cada una se desplaza en las tres direcciones de desensamblaje y luego se obtienen las siguientes matrices de interferencia:

$$\begin{array}{cc}
 \begin{array}{c} +x \\
 \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\
 0 & 0 & 1 & 1 & 1 \\
 1 & 1 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 0 \\
 1 & 1 & 1 & 0 & 0 \end{bmatrix}
 \end{array} &
 \begin{array}{c} -x \\
 \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\
 0 & 0 & 1 & 1 & 1 \\
 1 & 1 & 0 & 0 & 1 \\
 1 & 1 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 0 \end{bmatrix}
 \end{array}
 \end{array}$$

$$\begin{array}{cc}
 \begin{array}{c} +y \\
 \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\
 0 & 0 & 1 & 1 & 1 \\
 1 & 1 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 0 \end{bmatrix}
 \end{array} &
 \begin{array}{c} -y \\
 \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\
 0 & 0 & 1 & 1 & 1 \\
 1 & 1 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 0 \end{bmatrix}
 \end{array}
 \end{array}$$

$$\begin{array}{cc}
 \begin{array}{c} +z \\
 \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\
 0 & 0 & 1 & 1 & 1 \\
 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 \end{bmatrix}
 \end{array} &
 \begin{array}{c} -z \\
 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 0 \\
 1 & 1 & 0 & 0 & 0 \end{bmatrix}
 \end{array}
 \end{array}$$

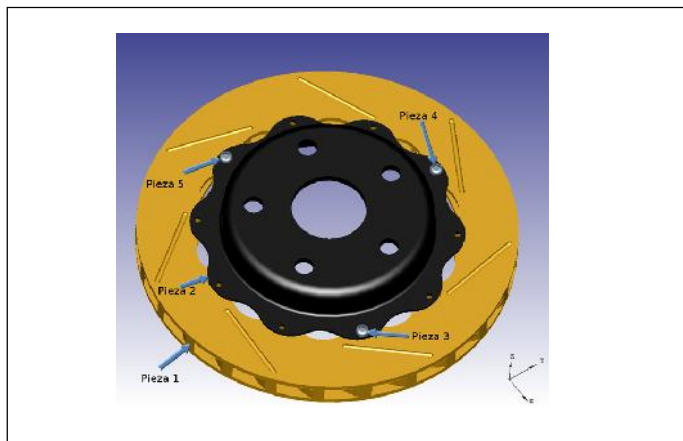


Fig. 3. Ensamble simplificado de un disco de freno

Las matrices que representan a las direcciones -x, -y, -z fueron generadas teniendo en cuenta que se corresponden con las transpuestas de las matrices +x, +y, +z. La información de la matriz de interferencia para la dirección de +z sugiere que la pieza 1 al desensamblarse en esa dirección está obstruida por las piezas 2, 3, 4 y 5, el movimiento de la pieza 2 está restringido por las piezas 3, 4 y 5, y ninguna pieza interfiere la extracción de las piezas 3, 4 y 5.

La figura 4 b) muestra un segundo ejemplo de un ensamble de un reductor, el cual también fue simplificado. Este ensamble está compuesto por siete piezas. La pieza 1 es el tornillo de izaje o cáncamo, la pieza 2 el cuerpo del reductor y la pieza 3 representa la tapa superior. La pieza 4 representa la rueda dentada conducida, la pieza 5 al árbol de salida del movimiento, la pieza 6 corresponde al piñón o rueda dentada conductora y la pieza 7 es el árbol de entrada del movimiento al piñón. En este modelo 3D de ensamble puede notarse la complejidad superficial de la geometría que caracteriza a las piezas del reductor, y al analizarse con la herramienta informática que implementa el método propuesto se obtienen las matrices de interferencia de desensamblaje siguientes:

$$\begin{array}{cc}
 \begin{array}{c} +x \\
 \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 1 & 1 \\
 1 & 0 & 0 & 1 & 1 & 1 \\
 0 & 1 & 1 & 0 & 1 & 1 \\
 0 & 1 & 1 & 1 & 0 & 1 \\
 0 & 1 & 1 & 0 & 0 & 1 \\
 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}
 \end{array} &
 \begin{array}{c} -x \\
 \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 1 & 1 \\
 1 & 0 & 0 & 1 & 1 & 1 \\
 0 & 1 & 1 & 0 & 1 & 1 \\
 0 & 1 & 1 & 1 & 0 & 1 \\
 0 & 1 & 1 & 1 & 0 & 1 \\
 0 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}
 \end{array}
 \end{array}$$

$$\begin{array}{cc}
 \begin{array}{c} +y \\
 \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 1 \\
 1 & 0 & 0 & 1 & 0 & 1 \\
 0 & 1 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 1 & 1 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}
 \end{array} &
 \begin{array}{c} -y \\
 \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 1 \\
 1 & 0 & 0 & 1 & 0 & 1 \\
 0 & 1 & 1 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 1 & 0 & 0 & 1 \\
 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}
 \end{array}
 \end{array}$$

$$\begin{array}{cc}
 \begin{array}{c} +z \\
 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 1 & 1 & 1 & 1 \\
 1 & 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 1 & 0 & 1 & 1 \\
 1 & 0 & 1 & 1 & 0 & 0 \\
 0 & 0 & 1 & 1 & 0 & 0 \\
 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}
 \end{array} &
 \begin{array}{c} -z \\
 \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 1 & 1 & 1 \\
 0 & 1 & 0 & 0 & 1 & 1 \\
 0 & 1 & 1 & 0 & 0 & 0 \\
 0 & 1 & 0 & 1 & 0 & 0 \\
 0 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}
 \end{array}
 \end{array}$$

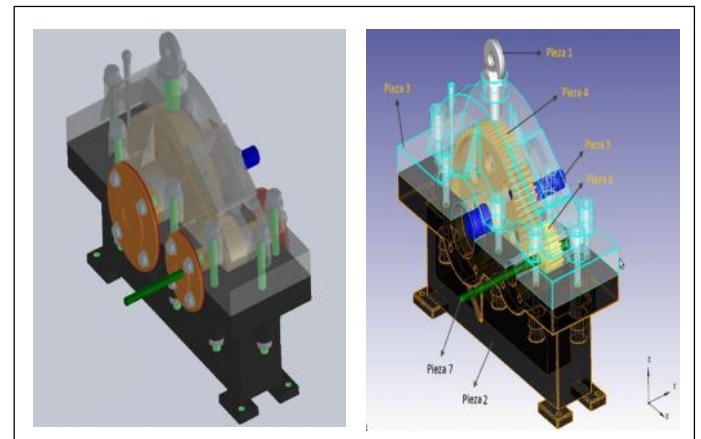


Fig. 4. Ensamble de un reductor: a) Ensamble del reductor con todos sus componentes; b) Ensamble del reductor simplificado

## DISCUSIÓN

Para detectar las interferencias entre todas las piezas de un ensamble, primeramente se realiza una aproximación triangular de la superficie de cada pieza mediante el mallado, y posteriormente se realizan pruebas de interferencias mientras la malla de una pieza se mantiene estática y la malla de otra se mueve en una dirección de desensamble hasta que ocurra o no una colisión. Las relaciones de interferencias entre las piezas se almacenan en matrices con una estructura ampliamente utilizada en la literatura, y sirven como base para la generación automatizada del conjunto de las secuencias de desensamble al considerarse la geometría.

Los resultados de esta investigación consideran las relaciones de interferencia entre piezas de superficies prismáticas, cilíndricas y además con superficies de forma libre, lo que implica que se puede extender el análisis de interferencia a otros ensambles de productos con geometrías más complejas. Sin embargo, solo se toman en cuenta las direcciones de desensamble de los ejes principales. La información contenida en las matrices de interferencia es importante para la generación automática de las reglas de precedencia de desensamble, y al considerar estas reglas se puede realizar la selección de las secuencias de desensamble factibles de un producto para lograr su posterior planificación. Sin embargo, a partir de la información de estas matrices solo se puede discriminar si una secuencia de desensamble es factible si se toma en cuenta la geometría de las piezas involucradas.

## CONCLUSIONES

En este artículo se presentó un método para la obtención de las relaciones de interferencia entre las piezas con superficies prismáticas, cilíndricas y de forma libre de un ensamble a partir de la geometría contenida en un fichero con formato STEP. Para automatizar el método descrito, fue programada una herramienta informática para leer ficheros STEP, y generar las matrices de interferencia correspondientes a las direcciones de desensamble de los seis ejes principales. Los resultados obtenidos muestran la eficacia del método al analizarse ensambles de productos compuestos por piezas con superficies complejas. Aunque puede darse el caso que un ensamble contenga piezas ubicadas en ejes de coordenadas diferentes a los ejes principales. En tal caso, el método no es aplicable, por lo que se recomienda considerar otras direcciones de desensamble en próximos estudios. La herramienta informática fue desarrollada sobre la plataforma libre y de código abierto PythonOCC, lo que implica que es una alternativa a los que se pueden crear al utilizarse herramientas privativas de código cerrado.

## REFERENCIAS

1. **DINI, Gino; SANTOCHI, Marco.** "Automated sequencing and subassembly detection in assembly planning". *CIRP Annals-Manufacturing Technology*, 1992, vol. 41, núm. 1, pp. 1-4.
2. **HUANG, Yuan Mao; HUANG, Chun-Ting.** "Disassembly matrix for disassembly processes of products". *International Journal of Production Research*, 2002, vol. 40, núm. 2, pp. 255-273. Disponible en Web: <http://ntur.lib.ntu.edu.tw/bitstream/246246/86461/1/18.pdf> [consultado febrero 2013].
3. **SINANOGU, Cem; BORKLU, H. Riza.** "An approach to determine geometric feasibility to assembly states by intersection matrices in assembly sequence planning". *Journal of Intelligent Manufacturing*, 2004, vol. 15, núm. 4, pp. 543-559.
4. **BALDWIN, Daniel F.** "Algorithmic methods and software tools for the generation of mechanical assembly sequences". Tesis de Maestría, Massachusetts Institute of Technology, 1990. Disponible en Web: <http://dspace.mit.edu/bitstream/handle/1721.1/14006/23130372.pdf?sequence=1> [consultado enero 2013].
5. **SAMBHOOS, Kedar; KOC, Bahattin; NAGI, Rakesh.** "Extracting assembly mating graphs for assembly variant design". *Journal of Computing and Information Science in Engineering*, 2009, vol. 9, núm. 3.
6. **PAN, Chunxia; S-F. SMITH, Shana; C. SMITH., Gregory.** "Determining Interference Between Parts in CAD STEP Files for Automatic Assembly Planning." *Journal of Computing and Information Science in Engineering*, 2005, vol. 5, núm. 1, pp. 56-62. Disponible en Web: [http://diyhl.us/~bryan/papers2/Determining interference between parts in CAD STEP files for automatic assembly planning.pdf](http://diyhl.us/~bryan/papers2/Determining%20interference%20between%20parts%20in%20CAD%20STEP%20files%20for%20automatic%20assembly%20planning.pdf) [consultado enero 2013].
7. **JIMÉNEZ, Pablo; THOMAS, Federico; TORRAS, Carme.** "3D collision detection: a survey". *Computers & Graphics*, 2001, vol. 25, núm. 2, pp. 269-285. Disponible en Web: <http://citeseerx.ist.psu.edu/viewdoc/download?jsessionid=B52B6FFAAEE79963172F043E693667456?doi=10.1.1.27.6188&rep=rep1&type=pdf> [consultado enero 2013].
8. **LAMBERT, Alfred J.D.; GUPTA, Sudendra M.** *Disassembly Modeling for Assembly, Maintenance, Reuse and Recycling*. 2005, CRC Press, Boca Raton, Florida. ISBN: 1-57444-334-8.
9. International Organization for Standardization. "Industrial automation systems and integration product data representation and exchange, Part 42: Integrated Generic Resources: Geometric and Topological Representation". ISO: 10303-42: 1994.
10. International Organization for Standardization. "Industrial automation systems and integration Product data representation and exchange, Part 21: Implementation methods: Clear text encoding of the exchange structure". ISO: 10303-21: 1994.

## **AUTORES**

### **Edwin Estévez Parra**

Ingeniero Informático, Profesor Asistente, Investigador, Centro de Estudios CAD/CAM, Facultad de Ingeniería, Universidad de Holguín, Cuba

### **Ricardo Lorenzo Ávila Rondón**

Licenciado en Contabilidad, Profesor Titular, Doctor en Ciencias Técnicas, Investigador, Centro de Estudios CAD/CAM, Facultad de Ingeniería, Universidad de Holguín, Cuba

# Interference between Parts of a Mechanical Assembly Contained in a STEP File

## **Abstract**

Obtaining interferences among parts of a product contributes to determine the feasibility of the assembly sequences according to the geometry of the parts involved. However, in assembly may exist parts with freeform surfaces which make difficult to determine the interference relationship between them automatically. The objective of this research is to present a method for obtaining interference among assembly parts with prismatic, cylindric and form free surfaces. The method is automated with a software developed on PythonOCC open source framework, which is based on OpenCascade. The techniques for extracting assembly geometric information which is contained in a 3D CAD file with STEP standard format, the detection of collisions between the parts in disassembly directions, and the generation of interference matrices are described. Two examples of mechanical assemblies are analyzed to demonstrate the effectiveness of the method and the interferences matrices corresponding to the six principal axis disassembly directions are obtained as a result.

Key words: interference, assembly, 3D CAD file, STEP