

Tipo de artículo: Artículo original
Temática: Soluciones Infrmáticas
Recibido: 2/12/18 | Aceptado: 11/02/18 | Publicado: 23/02/18

Módulo para automatizar el diseño de engranajes de tornillo Sin Fin

Module to automate the design of worm gear

Leandro Texidor Basterrechea ¹, Augusto César Rodríguez²

¹ Universidad de las Ciencias Informáticas, leandro.texidor@nauta.cu

² Universidad de las Ciencias Informáticas, augustor@uci.cu

* Autor para correspondencia: usuario@dominio.com

Resumen

En el presente trabajo se exponen los resultados obtenidos en el proceso de desarrollode un módulo para automatizar el modelado de engranajes de tornillo sin fin, para una aplicación de diseño asistido por computadoras (más conocido por el termino en inglés “Computer Aided Desing” o “CAD” por sus siglas) basada en tecnologías de código abierto; los sistemas comerciales tienen implementados aceleradores de diseño con los que se pueden generar este tipo de componente. Debido a las restricciones del bloqueo impuesto por el gobierno de los Estados Unidos contra Cuba y los altos precios de los sistemas comerciales con estas prestaciones, se dificulta el acceso a los mismos. La solución que se presenta ha sido desarrollada con las tecnologías de código abierto Open Cascade y el marco de trabajo Qt, por lo que está alineada con la necesidad del país de aumentar la soberanía tecnológica.

Palabras clave: tornillo sin fin; industria; ingeniería.

Abstract

In the present work the results obtained in the process of development of a module to automate the modeling of worm gears for a computer-aided design application, based on open source technologies are exposed; commercial systems have implemented design accelerators with which this type of component can be generated. Because of the restrictions of the blockade imposed by the United States government against Cuba and the high prices of commercial systems with these benefits, access to them is difficult. The solution presented has been developed with Open Cascade open source technologies and the Qt framework, so it isaligned with the country's need to increase technological sovereignty.

Keywords: worm gear; industry; engineering.

Introducción

Una de las principales diferencias de los sistemas propietarios con respecto a los de software libre en el mundo del diseño mecánico es que presentan aceleradores de diseño de distintos tipos de engranajes entre los que se encuentran los de tornillo sin fin lo que permite que los ingenieros se ahorren tiempo en el proceso de diseño de estos tipos de mecanismos.

En el presente trabajo se muestran los resultados alcanzados en el desarrollo de un acelerador de diseño de engranajes de tornillo sin fin y la integración a la aplicación de código abierto "Ingeniero".

En la actualidad existen una gran cantidad de herramientas de diseño asistido por computadoras entre las que se encuentra el Catia, Inventor, Solid Works, Solid Edge entre otros, todos ellos de carácter comercial. Basados en tecnologías de código abiertos se encuentran el FreeCad, LibreCad, Salome Meca.

El acceso a los sistemas informáticos comerciales por parte de Cuba, se dificulta por las restricciones del bloqueo impuesto por el gobierno de los Estados Unidos y por los precios prohibitivos de los mismos (5), pero la existencia de implementaciones en código abierto se ofrece como una alternativa para el desarrollo tecnológico, en la rama del diseño asistido por computadoras y de las funcionalidades para la construcción de imágenes de aspecto realista a partir de prototipos virtuales.

La búsqueda de información sobre los sistemas para la construcción de engranajes de tornillo sin fin, el estudio de las metodologías de cálculo y la forma en que se usan por diferentes aplicaciones, permitió establecer las ideas rectoras para el desarrollo del presente trabajo; en los apartados siguientes se exponen los materiales y métodos utilizados, el resultado obtenido y algunas consideraciones sobre el mismo.

Materiales y métodos o Metodología computacional

Se utilizaron los siguientes recursos tecnológicos: como lenguaje de programación C++11 ya que la mayor parte del desarrollo en la esfera de las tecnologías CAD está conformado en ese lenguaje, de esta forma se garantiza plena compatibilidad con el núcleo gráfico; el framework Qt 5.5.1 del que se emplearon las librerías para el diseño de las interfaces gráficas; el entorno de desarrollo QtCreator 4.5.0 y la tecnología OpenCascade en su versión OCE 0.17 (Open CASCADE Community Edition).

Para el desarrollo se empleó una computadora personal con 4GB de memoria RAM, un procesador Intel Core i5 4200U, con velocidad de 2GHz y sistema operativo LinuxMint 18.1 Cinnamon basado 64-bit en la versión Ubuntu 16.04 LTS.

La base metodológica requerida durante el proceso de desarrollo implicó a la metodología “Agile Unified Process” (4) en la versión propuesta por la Universidad de las Ciencias Informáticas, específicamente en el escenario 4 (1); también se hizo uso, en lo fundamental, de los métodos teóricos y empíricos de investigación relacionados a continuación:

Análisis y síntesis: Empleados durante el estudio de los fundamentos teóricos necesarios para enfrentar el proceso de desarrollo; en este sentido se realizó el estudio las metodologías de cálculos; se indagó sobre las implementaciones comerciales existentes empleadas en los sistemas CAD existentes; también durante el estudio de las opciones empleadas por las tecnologías Open Cascade como librerías para el apoyo del modelado en 3 dimensiones. El proceso de síntesis permitió conformar un grupo de requisitos funcionales para el desarrollo del módulo.

Observación: Se empleó durante el estudio de las funcionalidades de algunos sistemas propietarios como Inventor, Solid Edge y SolidWorks, referenciados en diversas fuentes. Algunos datos de los sistemas propietarios se tomaron de videos descargados de internet y con la información disponible se definió otro grupo de requisitos funcionales del módulo.

Experimentación: Fundamentalmente en dos aspectos, el primero para verificar la efectividad de las funcionalidades implementadas en el módulo y el segundo para comprobar la calidad de la implementación mediante pruebas unitarias y de integración.

Se escogió como estilo arquitectónico el de Llamada y Retorno, pues permite que un diseñador de software obtenga una estructura de programa que resulta relativamente fácil de modificar e incorporar nuevas funcionalidades. Dentro de este estilo arquitectónico se decidió el empleo del modelo en capas para la confección del componente, pues soporta bien los cambios y el desarrollo incremental. También, cuando las interfaces de las capas cambian o se añaden nuevas funcionalidades a una de ellas, solamente se ven afectadas las adyacentes. (2)

Resultados y discusión

El módulo desarrollado funciona acoplado a una aplicación en desarrollo denominada “Ingeniero”, sigue los criterios del estilo arquitectónico denominado llamada y retorno, específicamente el modelo por capas. Contiene una interfaz desarrollada en QT (ver Imagen 1), permite introducir los parámetros, relación de transmisión, longitud del tornillo,

ángulo de la hélice, módulo normal, número de dientes de la rueda y números de hilos del tornillo, ancho del diente, ángulo de presión y el paso normal.

El código de la aplicación está distribuido en las siguientes clases:

WormGear: Esta es la clase principal que de recoger todos los datos de la clase WormWheel, Screw y ejecutar la función de crear el modelo.

WormWheel: En esta clase es que se realizan las funciones para la construcción del piñón.

Screw: En esta clase es que se realizan las funciones para la construcción del tornillo.

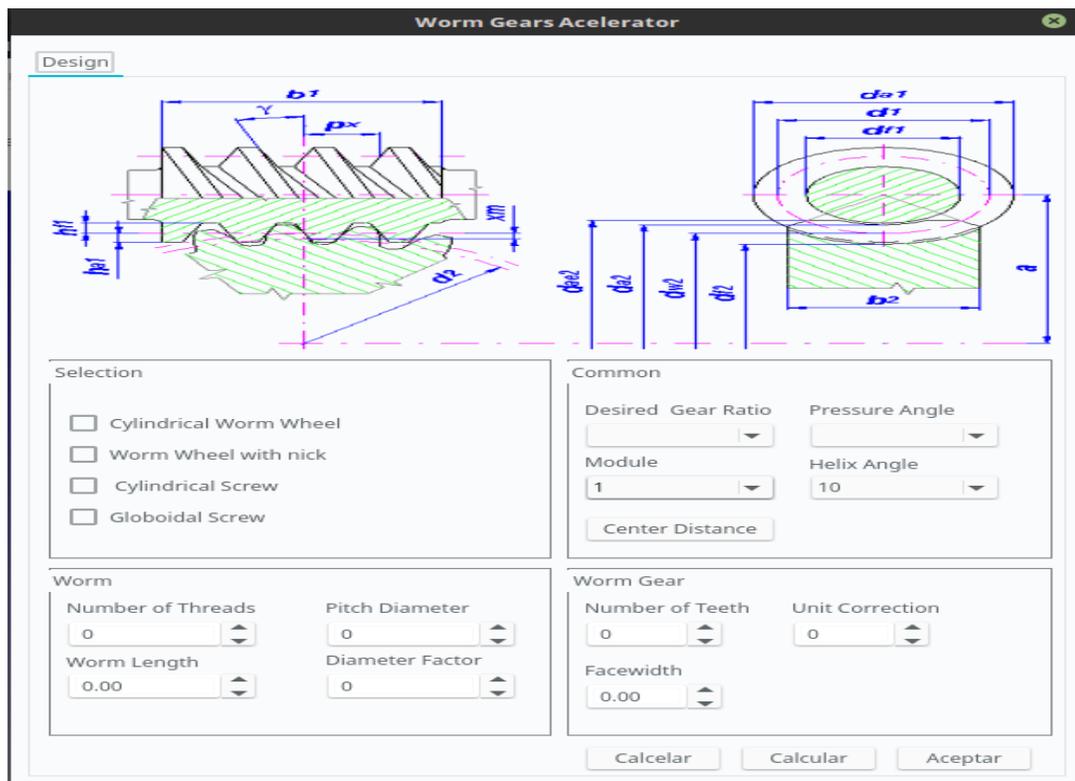


Figura 1: Interfaz principal.

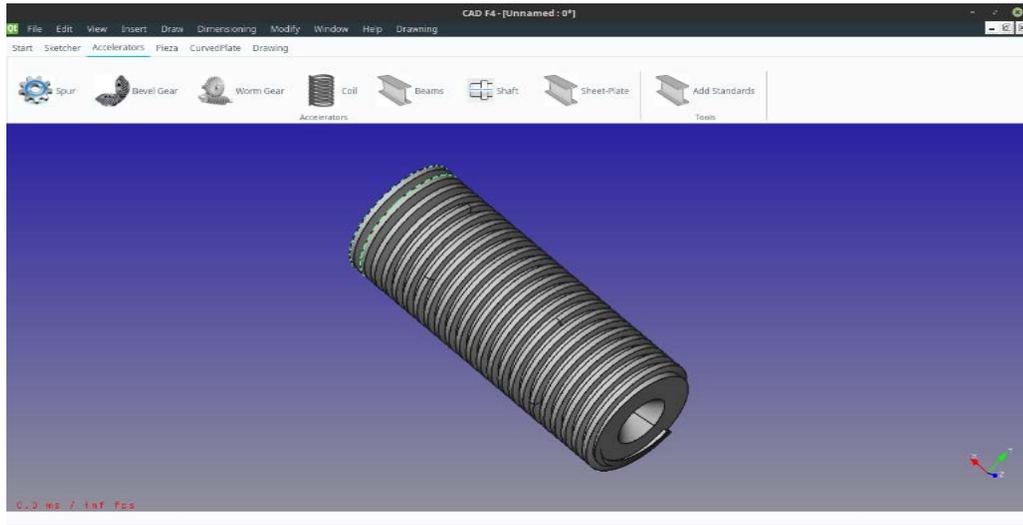


Figura 2: Tornillo.

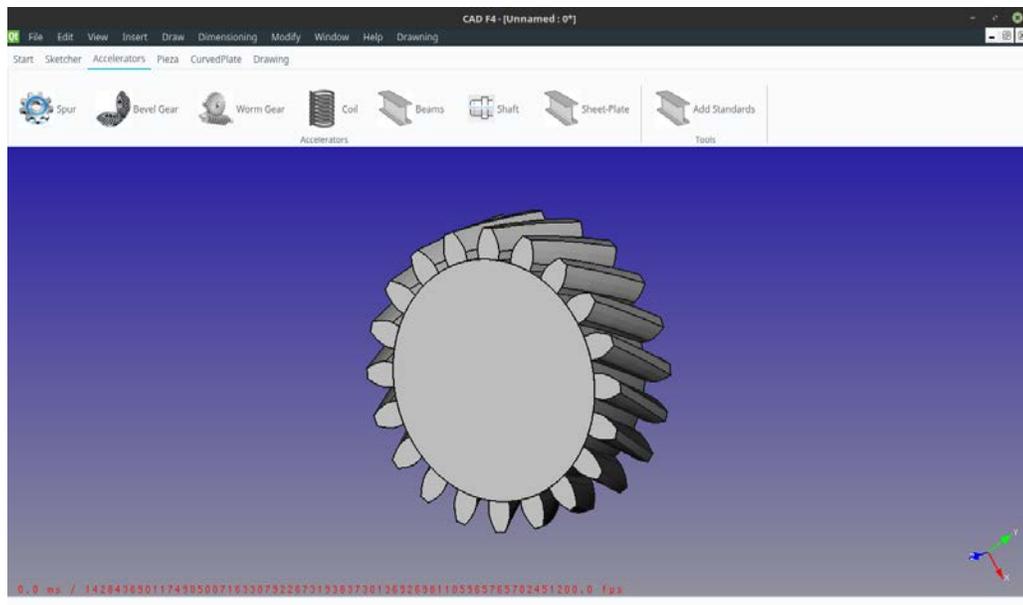


Figura 3: Piñón.

Conclusiones

Con el módulo obtenido se evidencia la factibilidad de desarrollar componentes con funcionalidades útiles para los sistemas de diseño asistido por computadoras, tomando como base las tecnologías de código abierto, que nos permiten estudiar implementaciones existentes de algoritmos.

Se obtuvo un acelerador de diseño que se integre a la aplicación “Ingeniero”, para construir modelos en tres dimensiones de engranajes de tornillo sin fin.

Referencias

- [1] TAMARA RODRÍGUEZ SÁNCHEZ. Metodología de desarrollo para la actividad productiva de la UCI. 2015.
- [2] Roger S Pressman. Ingeniería de software: Un enfoque práctico. McGraw-Hill, Nueva York, EUA, 6ta edición, 2007.
- [3] OCC: Overview. [en línea], 2017. [Consulta: 25 enero 2017]. Disponible en: <https://dev.opencascade.org/doc/overview/html/index.html>.
- [4] JAMES RUMBAUGH, IVAR JACOBSON and GRADY BOOCH. El Proceso Unificado de Desarrollo de Software. 2000.
- [5] MONTANO, J.L.M. de O., 2015. La migración hacia software libre en Cuba: complejo conjunto de factores sociales y tecnológicos en el camino de la soberanía nacional. Revista Universidad y Sociedad, vol. 7, pp. 119-125.