

SIMULACIÓN VIRTUAL INTERACTIVA DE PROCESOS DE CORTE Y PUNZONADO

A. Moreno¹, G. Velez¹, I. Barandiaran¹, A. Ruíz de Infante², R. Chopitea²

¹ Vicomtech-IK4, Mikeletegi Pasealekua 57, 20009, Donostia-San Sebastián, Gipuzkoa

² Lantek Investigación y Desarrollo, Ferdinand Zeppelin 2, 01510 Miñano, Álava

RESUMEN

La simulación del mecanizado es una fase del proceso productivo que permite mejorar de forma sustancial el rendimiento, al facilitar la detección de cualquier anomalía no prevista en la fase de diseño. Reduce la cantidad de pruebas del programa en la máquina real y de esta forma, permite ahorrar en tiempo de máquina, materiales, energía, reduciendo a la vez las posibles averías o accidentes laborales. Este trabajo presenta los resultados parciales de un módulo software que, de manera gráfica e interactiva, ayuda a obtener una representación visual del comportamiento tanto de la máquina como de los efectos del punzonado sobre la chapa. Dicha representación gráfica utiliza algoritmos geométricos y visuales ya existentes pero adaptados para su utilización en entornos de simulación 3D. Además, se han diseñado e implementado nuevos algoritmos que dan solución a situaciones específicas del punzonado: *i)* existencia de punzonados que deforman la chapa, *ii)* extracción de piezas a través de trampilla y *iii)* extracción de piezas a través de brazo robótico.

1. INTRODUCCIÓN

En la industria manufacturera, los avances tecnológicos han incrementado la productividad y el rendimiento del negocio. Por lo general, estos avances han venido de la mano de los avances meramente mecánicos, obteniendo una maquinaria más eficiente, más rápida y más versátil, lo cual mejoró los resultados económicos.

La gestión y la optimización de los recursos son dos de los retos más interesantes, ya que pequeñas mejoras en el tiempo o en los recursos consumidos puede llevar a grandes ahorros en procesos manufactureros que producen miles de piezas. En el sector de la manufactura de chapa es aún más importante, puesto que la propia naturaleza del proceso de corte genera material de desecho, que debe ser devuelto a la industria de fundición para su reciclaje.

Así pues, las herramientas de simulación ayudan a testear y validar los programas en la fase de diseño tantas veces como sea necesario antes de ir a la máquina real. Después de una serie de tests con el programa optimizado, el programa ya puede ser utilizado en producción. Por el contrario, un programa no optimizado o no correctamente validado puede contener errores que pueda comprometer la integridad de la máquina o producir alguna rotura en las herramientas.

Este trabajo presenta una herramienta de simulación de programas CNC para corte de chapa mediante corte lineal y punzonado. En la siguiente sección se presenta la motivación y

encaje del simulador respecto de las iniciativas mundiales, europeas y locales, bajo los conceptos generales de “Industria 4.0”. En las siguientes secciones se presentan las capacidades del simulador, los algoritmos que simulan los procesos de corte y los métodos de cálculo de piezas sueltas. Finalmente, se presentan las conclusiones y el trabajo futuro.

2. ENCAJE DEL TRABAJO EN EL ENTORNO INTERNACIONAL

La Manufactura Avanzada se define como el tipo de fabricación que implica una rápida transferencia de nuevos conocimientos a los procesos y productos industriales [13]. Está ampliamente aceptado que las TIC son tecnologías clave que aceleran y mejoran la productividad en la industria manufacturera. El despliegue de las TIC en los sistemas de producción al final de la década de los 60 era en realidad una revolución industrial. Las fábricas competitivas de hoy en día no se pueden concebir sin la pirámide actual de automatización industrial (que incluye PLC, MES, ERP y otras tecnologías clave) en la producción, o sin la gestión del ciclo de vida del producto, con el apoyo de herramientas avanzadas de CAD / CAM / CAE, por mencionar sólo unas pocas casos.

La evolución reciente de las TIC está abriendo posibilidades revolucionarias para la fabricación y la producción, siendo la más importante la aplicación de las últimas tecnologías relacionadas con Internet en la industria. Debido a varias razones técnicas, de mercado y culturales, la industria es, paradójicamente, uno de los últimos nichos en ser conquistado por los avances generalizados y ubicuos asociados con el Internet de las Cosas y los servicios asociados.

En EE.UU., el llamado "*Industrial Internet - the Third Wave*" es un término acuñado por *General Electric* en su visionario artículo [1] y que ya es ampliamente aceptado en muchas entidades académicas estadounidenses como *NSF I/URCC Center for Intelligence Maintenance Systems (IMS)* y en otro tipo de actores industriales en EE.UU. Tiene un fuerte enfoque en el mayor grado de inteligencia junto con la potencia computacional, el análisis, la sensorización de bajo costo y los nuevos niveles de conectividad permitida por Internet [6]. Tres elementos caracterizan esta visión: (i) las máquinas inteligentes, (ii) la analítica avanzada, y (iii) personas en el trabajo. El ciclo de los datos de Internet Industrial señala varios aspectos muy relevantes para los que la simulación de procesos podría ser una tecnología habilitadora clave.

Por otro lado, la iniciativa estratégica *Industrie 4.0* [5] ha creado un punto de referencia internacional (no solo en Alemania) para establecer la visión, oportunidades tecnológicas y los retos científicos relacionados con la entrada de la nueva generación de tecnologías TIC, incluida el Internet de las Cosas y sus Servicios y los sistemas de ciber-físicos (o CPS) [8,14], en los sistemas de producción industriales. El término CPS (*Cyber-Physical-System*) se refiere a la convergencia del mundo físico y el mundo digital. Cuando se aplica a la producción, CPS se especializa en CPPS o sistemas de producción ciber-físicas.

Dada la relevancia de esta nueva tendencia o revolución industrial están apareciendo diferentes programas, tanto a nivel local como a nivel internacional, para tratar de favorecer y acelerar el desarrollo de las tecnologías relacionadas hasta convertir el paradigma de *Industrie 4.0* en una realidad tangible con impacto en las empresas.

Una de las estrategias más importantes por la Comisión Europea en materia de tecnología e Innovación es la denominada RIS3, o especialización inteligente. Esta estrategia es una evolución de las anteriores estrategias Regionales de Innovación (RIS).

El Gobierno Vasco dentro del desarrollo de su Plan de Ciencia y Tecnología 2020 ha adoptado la estrategia RIS3 como uno de los pilares fundamentales de dicho plan. Tal y como define la estrategia RIS3 el Gobierno Vasco definió como uno de los sectores con mayor capacidad productiva y con mayores capacidades de desarrollo, el sector industrial. El sector industrial es el que más peso exportador soporta dentro Euskadi y supone aproximadamente el 25% del PIB.

Dentro de la estrategia RIS3 para Euskadi se han definido tres prioridades de especialización inteligente fundamentales: Fabricación avanzada, Energía y Biociencias.

Las herramientas de simulación encajan tanto con la visión de “*Industria 4.0*” como con la idea subyacente a la estrategia RIS3, ya que representan un duplicado digital de un elemento real y físico como son las máquinas de mecanizado. La simulación del proceso de corte trata de reflejar el comportamiento real del proceso de corte, pero de una manera virtual. Del realismo de la simulación dependerá cómo de cercanos estarán el mundo real y el mundo digital o virtual.

En este trabajo se trabajará con representaciones virtuales teóricas, en el sentido de que se simularán programas CNC según salen de la fase de diseño. Las simulaciones previas de estos programas evitarán en gran medida infrutilizar los recursos reales y evitar posibles problemas y defectos en las máquinas. Una detección de posibles errores en tiempo de simulación, permitirá refinar el programa CNC una y otra vez hasta que se estime suficientemente robusto y eficiente. En este momento, se podrán realizar los tests necesarios con la maquinaria real, cuyo número total será bastante menor comparado con una situación donde no hubiera simulación de los programas CNC.

3. SIMULADOR DE CORTE Y PUNZONADO DE CHAPA

Este trabajo presenta un simulador interactivo del corte y punzonado de chapas metálicas. El simulador está orientado y focalizado en la obtención de resultados geoméricamente correctos y de alta calidad gráfica de los siguientes procesos de mecanizado:

- Corte lineal de chapa utilizando láser, plasma, oxicorte o agua
- Punzonado utilizando diferentes tipos de punzones con perfil arbitrario

Las técnicas de simulación permiten replicar virtualmente el comportamiento de la máquina involucrada en los procesos. El simulador comienza a partir del código NC (normalmente Código G o algún dialecto del mismo), traducido a un lenguaje común e intermedio representado en XML, que lista de manera ordenada la secuencia de movimientos que debe hacer la máquina para obtener el resultado esperado.

La simulación del proceso de corte difiere del punzonado en varios aspectos. El más importante de cara a la definición geométrica de la chapa es que el corte lineal se produce de manera continua y el punzonado, de manera discreta o instantánea.

En los siguientes apartados se describe la arquitectura del sistema y las funcionalidades más importantes del simulador.

3.1 Arquitectura del Sistema

El simulador está estructurado en una arquitectura de múltiples niveles, donde cada nivel encapsula diferentes técnicas y métodos. El nivel más bajo se encarga de los cálculos geométricos (*Geometric Kernel*) mediante el uso de operaciones booleanas que modifican la geometría de la chapa.

La capa de nivel bajo tiene acceso a la funcionalidad del sistema gráfico, que es el responsable de renderizar en pantalla los resultados de la simulación.

La capa de nivel medio de la arquitectura ofrece la lógica del simulador y del dominio del mismo, es decir, que modela los objetos principales del simulador: *Part, Herramienta, Máquina, Ejes*, etc. Cada elemento del sistema encapsula su propia funcionalidad en una clase distinta, dentro de la capa de nivel medio. Las animaciones de los elementos móviles también se representan en esta capa. En concreto, se calculan los volúmenes de barrido de las herramientas sobre la chapa, que serán usados para realizar las operaciones de corte entre la chapa y las herramientas.

La capa de alto nivel abstrae la funcionalidad orientada al usuario, que incluye el interfaz de usuario y la gestión de ficheros de entrada (XML, modelos 3D...):

- Interfaz de usuario multilenguaje
- Diferentes modelos de navegación 3D en el mundo virtual, utilizando el ratón como elemento principal.
- Gestión y visualización de las instrucciones NC reales, mostrando cual es la instrucción en curso.
- Gestión y visualización de los objetos 3D de la escena y sus componentes.

Tanto el interfaz de usuario como el tratamiento y lectura de ficheros XML se ha implementado utilizando Qt.

3.2 Interfaz de Usuario

La interfaz de usuario (ver Figura 1) muestra una interfaz sencilla y limpia para acceder a la funcionalidad del simulador. Se ha desarrollado utilizando Qt, que ofrece la tecnología necesaria para crear interfaces y controles de usuario de calidad profesional. La aplicación está estructurada mediante una serie de paneles móviles que se pueden acoplar a distintas

partes de la ventana o dejarlas flotando libremente. El control 3D donde se visualiza la simulación se mantiene en siempre en posición central.

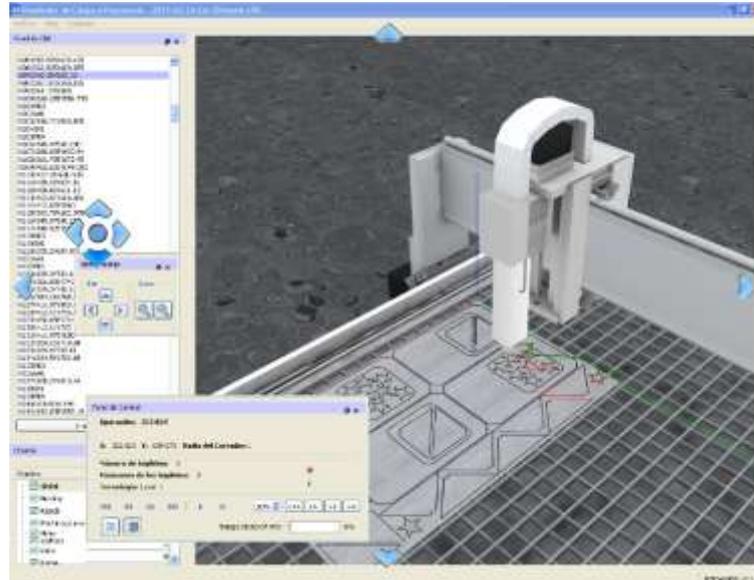


Figura 1. Interfaz de usuario del simulador CNC.

Hay dos maneras de controlar la simulación. Una es ejecutar una simulación continua, donde un factor elegido por el usuario se utiliza para controlar la velocidad de la simulación. La otra manera es una simulación rápida hasta una instrucción determinada. La simulación se hace en segundo plano y se muestra al usuario el resultado de la misma cuando está disponible. La combinación de ambos modos permite a los usuarios ir en modo rápido hasta una instrucción determinada, y simular de manera continua a partir de ahí. El panel de control ofrece a los usuarios una serie de botones que permite acceder a estos modos de manera intuitiva.

El interfaz de usuario ofrece opciones de visualización para los distintos objetos virtuales de la escena 3D. La visibilidad y la transparencia de cualquier objeto se pueden modificar de una manera sencilla e intuitiva a través de controles mostrados en el Panel de Objetos. Los modos de cámara se pueden seleccionar en otro panel y consisten en modo 3D y 2D. El modo 3D ofrece una navegación libre de la escena. El modo 2D ofrece una vista cenital de la chapa, y limita los movimientos del usuario al plano XY.

Gracias al soporte multilinguaje de Qt, la interfaz de usuario se puede traducir a múltiples lenguajes sin necesidad de reconstruir la aplicación, incluyendo así, un mecanismo sencillo y práctico para añadir nuevos idiomas.

3.3 Representación de la máquina de corte lineal

Cada proceso de corte de chapa tiene su propia representación 3D. Aunque cada proceso se comporte de manera similar, las diferencias visuales son importantes para los

usuarios. En la Figura 2 se muestran los distintos sopletes usados para diferentes tecnologías de corte.

Las máquinas se componen de una mesa estática, un puente móvil sobre la mesa (eje X) y un soporte que se mueve a lo largo del puente (eje Y). En este soporte, se encuentra el cabezal de corte.

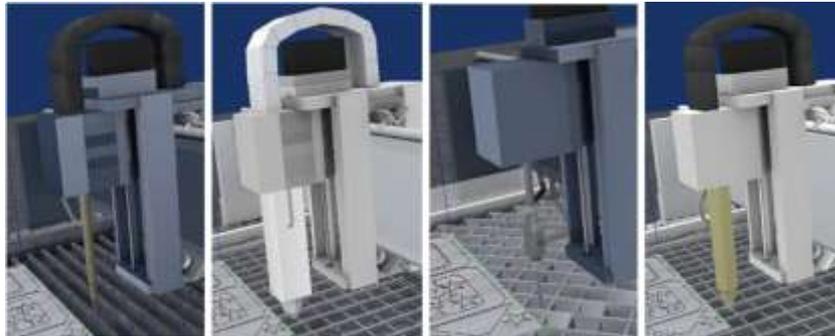


Figura 2. Cada tecnología tiene su propio modelo 3D para representar el objeto que corta la chapa.

La mesa difiere bastante según el tipo de mecanizado que se realice (ver Figura 3). Por ejemplo, usando tecnología de corte con agua, la mesa tiene un marco de plástico transparente que evita que salpique el agua al exterior de la máquina, y el fondo de la mesa es sólido. En otras máquinas, el fondo de la mesa es abierto, y permite que los trozos pequeños de metal puedan caer al suelo.

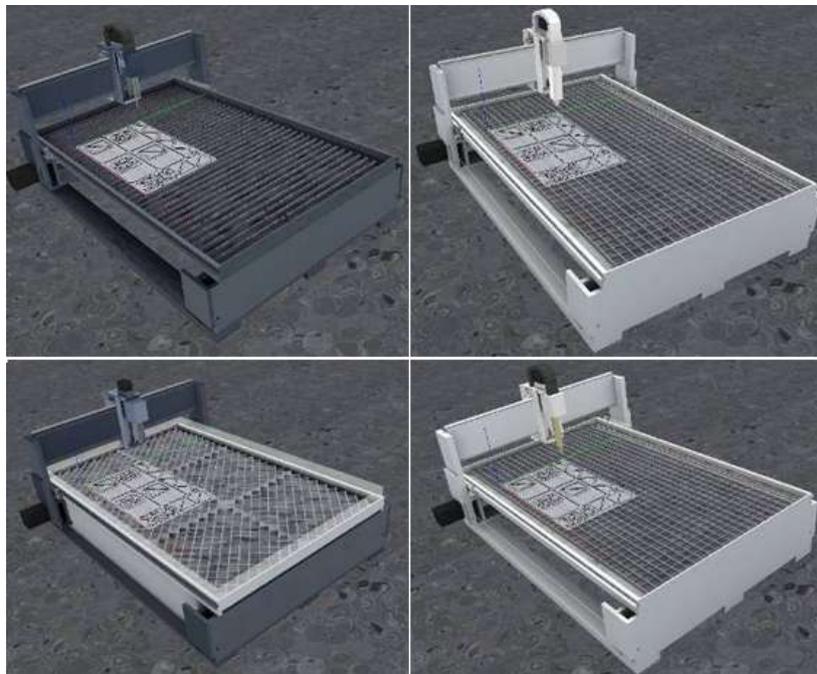


Figura 3. Modelos 3D de las máquinas de corte lineal. Hay una versión distinta según la tecnología utilizada (plasma, laser, oxi-corte y agua).

3.4 Efectos Visuales: Chispas

Los procesos de corte de chapa suelen producir chispas que dificultan la visualización. También los procesos de punzonado generan chispas, aunque suelen ser de una manera más discreta e instantánea.

El simulador permite añadir al sistema una gran variedad de efectos de chispas (ver Figura 4). Se utiliza un sistema de partículas como elemento gráfico, y modificando sus parámetros (color, velocidad, ángulo de dispersión, dirección...), se pueden simular chipas generadas por distintas tecnologías (laser, oxiacorte, agua...).

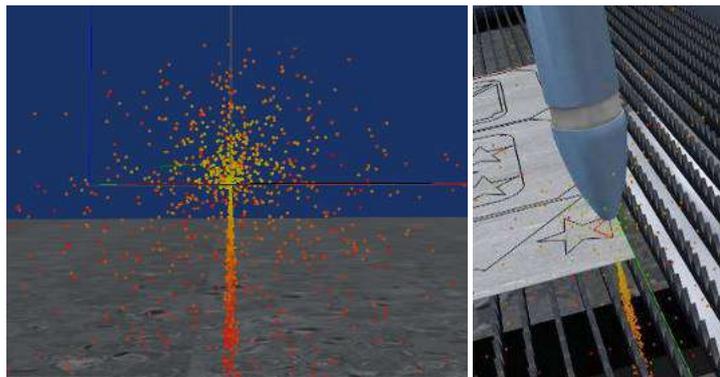


Figura 4. Efectos visuales que representan las chispas que ocurren durante los procesos de corte.

3.5 Representación de la máquina de punzonado

La representación jerárquica de la máquina de punzonado es diferente a la de las máquinas de corte lineal. En la máquina de punzonado, la chapa es un elemento móvil, mientras que en el corte lineal, la chapa era un elemento estático. Así pues, en punzonado la chapa se mueve en los ejes X e Y y los punzones se mueven en Z cuando se realizan los punzonados sobre la chapa.

El simulador de la máquina punzonadora representa los punzones como un conjunto de geometrías 3D construidas a partir de los contornos definidos en un fichero XML. La organización de los punzones en la torreta también se define en ficheros XML.

La visualización de los procesos de punzonado requiere activar y configurar la transparencia de los elementos que ocultan el mecanizado (ver Figura 5).

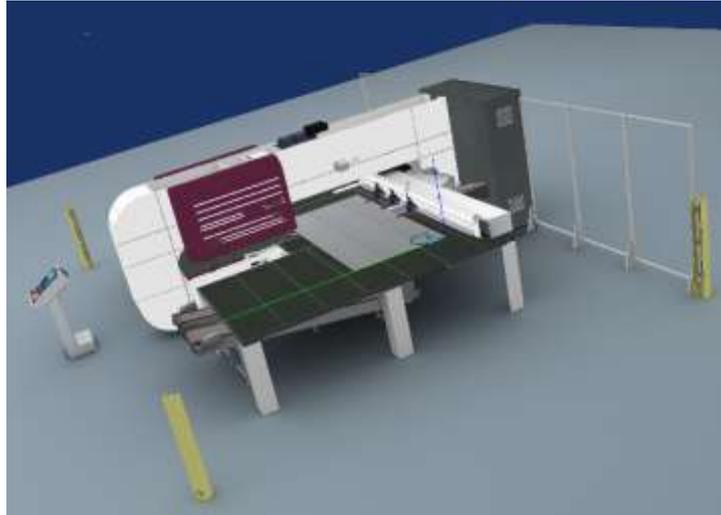


Figura 5. Modelo 3D de la máquina de punzonado.

La máquina de punzonado incluye no solo los punzones y la chapa, sino también la estructura y posicionamiento relativo de los punzones en ella. Una máquina punzonadora por lo general tiene alojados los punzones dentro de una torreta circular. En el simulador se ha optado por integrar un modelo de torreta circular con las siguientes características:

- Tiene estaciones individuales
- Tiene estaciones especiales (MultiTool), que contienen varias estaciones individuales dentro de ellas (subestaciones)

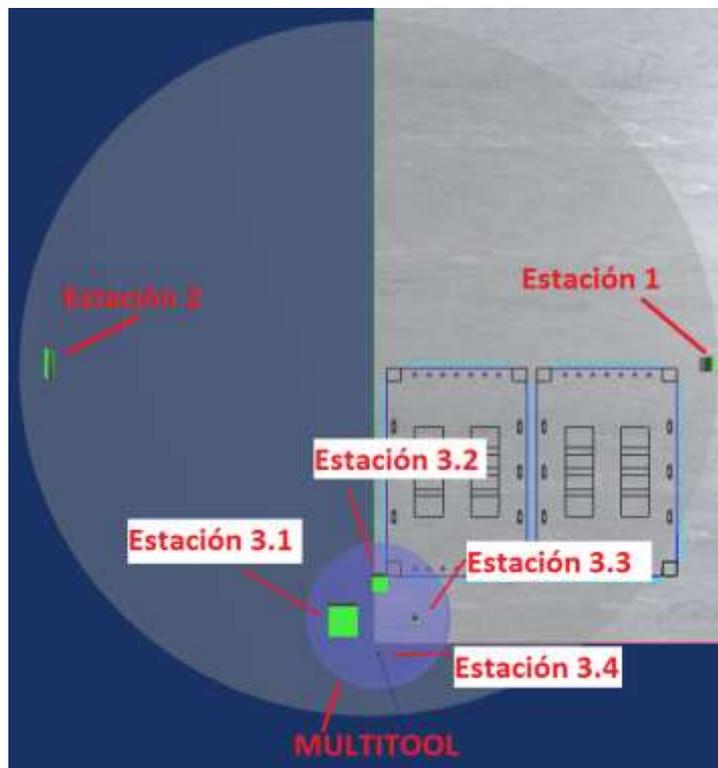


Figura 6. Ejemplo de configuración de Torreta Circular.

En la Figura 6 se puede ver cómo se conceptualiza una torreta circular. Se pueden distinguir estaciones individuales (estaciones 1 y 2) y una estación MultiTool con varias estaciones (estaciones 3.1, 3.2, 3.3 y 3.4).

Los punzonados se realizan en una posición fija de la torreta circular, por lo que cada cambio de punzón requiere realizar una serie de animaciones para colocar el punzón activo en la posición de golpeo. En concreto, primero se rota la torreta circular y, si hace falta, se rota la estación MultiTool que lo aloja. Una vez colocado el punzón activo en la posición de golpeo ya se pueden realizar los punzonados sobre la chapa.

3.6 Deformaciones de la chapa

Los procesos de punzonado utilizan una gran variedad de punzones. Los más típicos provocan un vaciado de material en la chapa, de mismo tamaño que el contorno que define el perfil del punzón. Sin embargo, existen otros punzones que provocan deformaciones en la chapa o una mezcla de corte y deformación.

El simulador ofrece la posibilidad de configurar las deformaciones en la chapa que provoca un punzón definido en el sistema y situado en la torreta circular. Para ello se hace uso de ficheros XML que definen los efectos que producen en la chapa.

Como se ha comentado con anterioridad, en el modo más básico de punzonado se obtiene un agujero con la misma forma que el punzón. El punzón puede tener prácticamente cualquier forma: redonda, triangular, rectangular, etc. Pero además, existen otros modos de punzonado más complejos que también pueden ejecutarse mediante máquinas punzonadoras CNC de torreta: Embutición, Abombamiento, Roldana, Cizallado, Marcado, Trazado, Bordoneado, Offset-up y offset-down, Roscado y Rejillas.

Los dos tipos de punzonado que deforman la chapa que se han elegido para su inclusión en el simulador son los siguientes:

- **Embutición:** consiste en dar a una chapa formas huecas más o menos profundas. Para ello, se coloca la chapa sobre la matriz y se presiona hacia la cavidad con ayuda de un punzón que tiene la forma en la cual quedará formada la lámina.
- **Rejillas:** Una aplicación típica de punzonado es la de crear rejillas o, más genéricamente, estructuras complejas sobre la chapa.

Para resolver estos tipos de punzonado se propone la siguiente estrategia: Una vez eliminado el trozo afectado de la chapa por el punzón, se rellena el hueco generado por geometría nueva que encaje en el hueco generado.

Esta técnica se utiliza para implementar las rejillas y, de manera general, para embuticiones de alta complejidad. La técnica se basa en modelar externamente (en 3D por lo general) el resultado de una operación de punzonado. En tiempo de ejecución, se utiliza esta plantilla para posicionarla en el hueco generado por el punzón.

Para la simulación de la rejilla, se ha modelado una rejilla básica de dimensiones conocidas ($10 \times 1 \times 1$ cm) y orientación conocidas (lado largo en X). Dado un punzón rectangular de $20 \times 10 \times 10$, se calcula la transformación necesaria que hay que aplicar a la plantilla para rellenar el hueco: $2 \times 10 \times 10$ en este caso. Usando esta técnica, se ha generado una alternativa para la embutición de punzones circulares. En este caso, en vez de generar la geometría de una semiesfera, se está utilizando una plantilla previamente modelada (ver Figura 7).

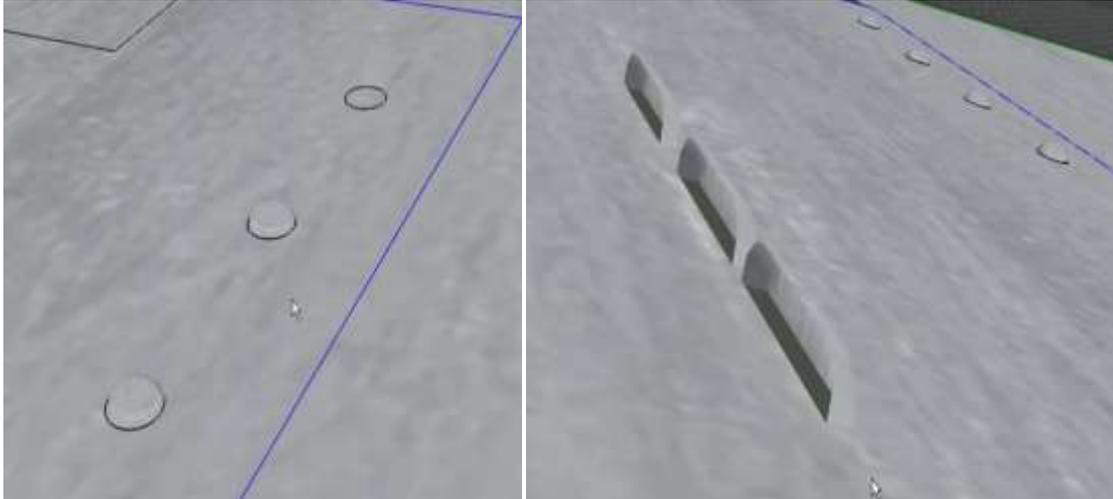


Figura 7. Punzonado avanzado mediante plantillas.

4. Operaciones de corte

En las secciones anteriores se han descrito la funcionalidad básica del simulador desde el punto de vista funcional. En esta sección se describirán cómo se representa la chapa de metal y cómo se realizan las operaciones de corte a partir de operaciones booleanas entre contornos 2D. Así mismo, se analizan los problemas de eficiencia que esta aproximación puede generar y se presenta la solución que se ha utilizado para obtener un rendimiento óptimo.

4.1 Representación virtual de la chapa

La representación de la chapa se puede ver como un rectángulo 2D. De hecho, cualquier representación instantánea de la chapa se puede representar como un conjunto de contornos coplanares y no auto-intersectantes. Así pues, las operaciones de mecanizado (corte o punzonado) utilizarán esta presentación de contornos 2D para realizar las operaciones booleanas entre la chapa y las herramientas.

Utilizando esta definición de la chapa de metal, la operación booleana 3D de alto nivel, se simplifica a una operación booleana 2D entre dos polígonos complejos, uno que representa la chapa y otro que representa la herramienta [15].

Normalmente, la chapa de metal comienza siendo un rectángulo definido por 4 vértices. Según avanza el proceso de corte, las herramientas generan volúmenes de barrido que intersectan la chapa. Aunque los sopletes tienen una punta muy fina (0.5 mm o menos), el área de barrido es significativa y hay que tenerlo en cuenta.

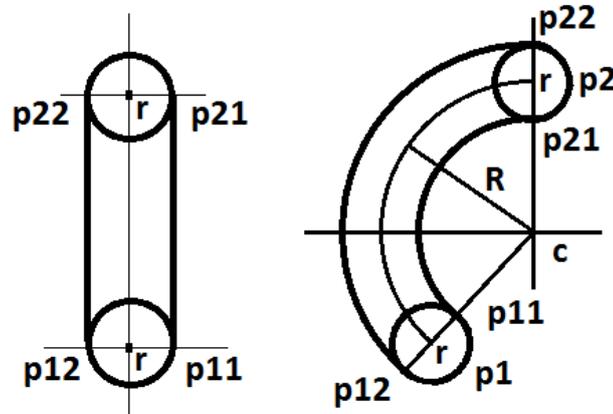


Figura 8. Una herramienta en movimiento produce volúmenes de barrido. Trabajando con contornos 2D, el barrido puede ser representado geoméricamente a partir del punto de inicio y final del movimiento (lineal o en arco) y los datos de la herramienta (radio).

El material eliminado de la chapa es el que corresponde con el volumen de barrido de la herramienta a lo largo de la trayectoria definida por su movimiento (ya sea lineal o en arco). Teniendo en cuenta que se utilizan contornos 2D y que la herramienta se puede aproximar como un cilindro vertical de un radio r , el contorno 2D que modela el movimiento de la herramienta se puede definir geométrica de una manera sencilla (ver Figura 8).

El punzonado de chapa es algo diferente, pero desde el punto geométrico el proceso es el mismo (operaciones booleanas de contornos 2D). La definición de los punzones se dan como el contorno que generan sobre la chapa, por lo que la forma 3D del punzón no corresponde con un cilindro de radio r como en el proceso de corte lineal. El volumen de barrido se calcula cuando el punzón baja sobre la chapa. Como los punzonados son operaciones discretas e instantáneas, la chapa de metal y el punzón no se mueven relativamente en el plano XY. Esta situación simplifica el cálculo del volumen de barrido. Al igual que en el corte lineal, la operación booleana entre el estado actual de la chapa y el contorno que define el punzón se realiza con la librería Clipper [7].

4.2 Renderizado

La visualización de la chapa de metal recae en el subsistema gráfico. *OpenSceneGraph* [11] es un motor gráfico de alto rendimiento que se utiliza como parte del sistema de simulación, animación y visualización. Para poder visualizar los contornos 2D que representan la chapa de metal en la pantalla, es necesario construir una geometría basada en triángulos a través de métodos de teleselación de polígonos 2D. Estos métodos tienen en cuenta que los polígonos 2D pueden ser complejos, es decir, tener contornos anidados que

actúan de agujeros o tener islas dentro de su definición. Este proceso se automatiza internamente a través del API GLUtesselator de OpenGL.



Figura 9. Renderizado de los contornos 2D anidados, utilizando OpenSceneGraph (GLUtesselator).

A pesar de la simplicidad aparente del método de teselación, los resultados obtenidos muestran una calidad geométrica y visual muy alta (ver Figura 9).

La chapa de metal puede ser muy fina, pero tiene espesor. Para visualizar la chapa en 3D, se genera un sólido duplicando la geometría 2D de la chapa a una distancia determinada por el espesor de la misma. Posteriormente se añaden las paredes verticales que unen ambas geometrías y que permiten obtener un sólido cerrado. Estas paredes verticales se generan a través de geometrías “*triangle_strip*”, cuya definición es muy apropiada para definir este tipo de geometría.

4.3 Optimizaciones

Una implementación directa de los métodos expuestos hasta ahora no da un rendimiento muy alto en la simulación. Según avanza la simulación, la chapa de metal aumenta su complejidad, tanto en número de contornos, como en número de vértices. Debido a esto, las operaciones booleanas son menos eficientes y el rendimiento global cae significativamente. Además, visualizar polígonos muy complejos (con muchos contornos anidados) no es una tarea sencilla para el subsistema gráfico debido a que el proceso de teselación no es trivial. La siguiente sección presenta las técnicas de subdivisión espacial que se han introducido para mejorar el rendimiento de la simulación.

4.4 Subdivisión espacial y resultados

La eficiencia de los algoritmos de “*polygon clipping*”, es decir, los algoritmos que calculan las operaciones booleanas entre polígonos 2D, depende directamente del número de contornos y vértices involucrados en las operaciones [10]:

$$O (n \times \log(n) + k + z \times \log(n))$$

donde n es el número de aristas (puntos), z es el número de contornos y k es el número de intersecciones de aristas. Según avanza la simulación, la chapa se va haciendo más y más compleja, incrementándose el número de puntos y contornos. Por este motivo, se utiliza un sistema de particionado, para limitar este crecimiento de puntos y contornos y así intentar contener la pérdida de rendimiento global de la simulación.

Tabla 1. Rendimiento de la aplicación de la subdivisión especial variando el número de regiones. Se presenta el número de operaciones booleanas de bajo nivel y el tiempo de simulación.

Model	Movements	Subdivision	OpBools	Time (s.)
Simple	834	1×1	774	10
		4×4	1008	5
		16×16	2700	6
		32×32	6556	10
Complex	12731	1×1	11037	2100
		4×4	12481	195
		16×16	19533	87
		32×32	36910	130

La partición especial descompone la chapa de metal en regiones más pequeñas. De esta manera, el efecto del crecimiento del coste computacional de las operaciones booleanas se limita al interior de una región de la chapa. Por el contrario, en movimientos que se extiendan a lo largo de varias regiones, el número de operaciones booleanas será mayor. La mejora de rendimiento tiene que ver con la reducción media de la complejidad de la chapa y del hecho de que las operaciones de corte son locales, es decir, suelen estar definidas en una región y ocasionalmente, en dos o cuatro.

En nuestros experimentos (ver Tabla 1), variando el número de subdivisiones de la chapa con el mismo ejemplo de mecanizado se obtiene un rendimiento máximo con una subdivisión de 16×16 . La Figura 10 muestra la complejidad de la chapa de metal después de la simulación completa. Aproximadamente, cada círculo contiene unos 100 vértices, 50 para el contorno interno y otros 50 para el contorno externo. La chapa completa contiene cerca del millón de vértices y gracias a la subdivisión espacial (16×16 en este caso), la visualización se realiza en tiempo real.

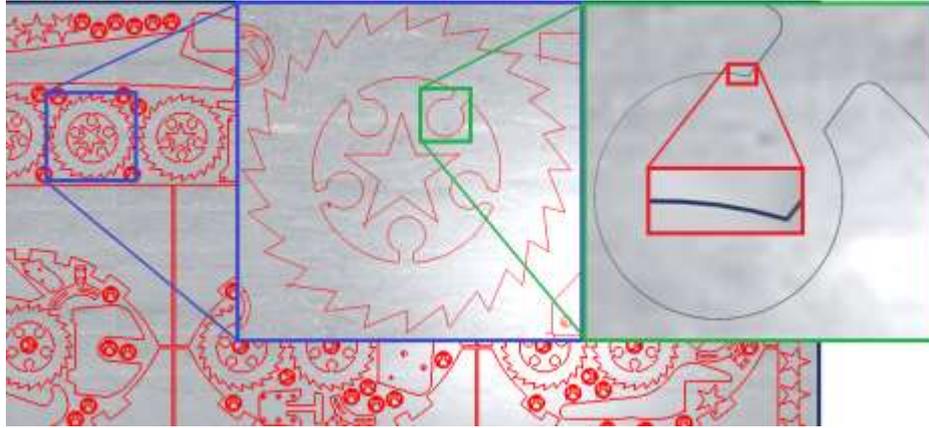


Figura 10. Resultado final de la simulación de un programa CNC de más de 12000 operaciones de corte.

5. Mecanismos de evacuación de piezas sueltas

En la sección anterior se ha presentado la definición de la chapa de metal y se ha introducido la subdivisión espacial como un mecanismo que permite mantener un rendimiento óptimo a lo largo de la simulación. La subdivisión espacial tiene un efecto colateral que es, a su vez, una de las razones del incremento de rendimiento: Los contornos se parten en regiones y por lo tanto se pierde la lógica y la representación jerárquica de las piezas que se generan en el proceso de corte.

Esta sección presenta los mecanismos de extracción de piezas sueltas en la chapa, teniendo en cuenta que la representación de la chapa (utilizando subdivisión espacial) ya no posee esa información. Se presenta un algoritmo que permite recuperar esta información sin gran coste computacional. Los mecanismos físicos de la máquina que se han añadido al simulador son la evacuación por trampilla y por brazo robótico.

5.1 Detección geométrica de piezas sueltas

La detección de piezas sueltas a través de la geometría de la chapa no es trivial si se tiene en cuenta la subdivisión espacial de la chapa. Conociendo de antemano las piezas que puede haber en la chapa, se ha ideado un método basado en imagen. Primero, se obtiene una imagen cenital de la chapa. Se hace uso de las capacidades de renderizado avanzado de la librería gráfica *OpenSceneGraph*. En concreto, se hace un RTT (*Render To Texture*) offline, es decir, sin salida gráfica en pantalla. En esta imagen, solo se dibuja la chapa, con un fondo negro para poder distinguir claramente qué es chapa y qué no.

El segundo paso es binarizar la imagen, es decir, convertir los colores de la imagen en imagen en blanco y/o negro. En este caso, el fondo será negro y la chapa será blanca. En ese paso, el ojo humano ya puede discernir qué es pieza y qué no: hay que buscar aquellas regiones blancas de la chapa que estén rodeadas de negro (el color del fondo).

El tercer paso es hacer uso de conocimiento a priori de las posibles piezas del programa. Por cada posible pieza se elige un punto o píxel interior que sea blanco (es decir, que sea chapa). En ese punto se ejecuta un algoritmo básico de los sistemas de tratamiento de imagen: *flood filling*, que consiste en rellenar una región con un color dado. En este caso, el algoritmo pinta de rojo aquellos píxeles blancos que encuentre hasta llegar a un valor distinto (el negro del fondo).

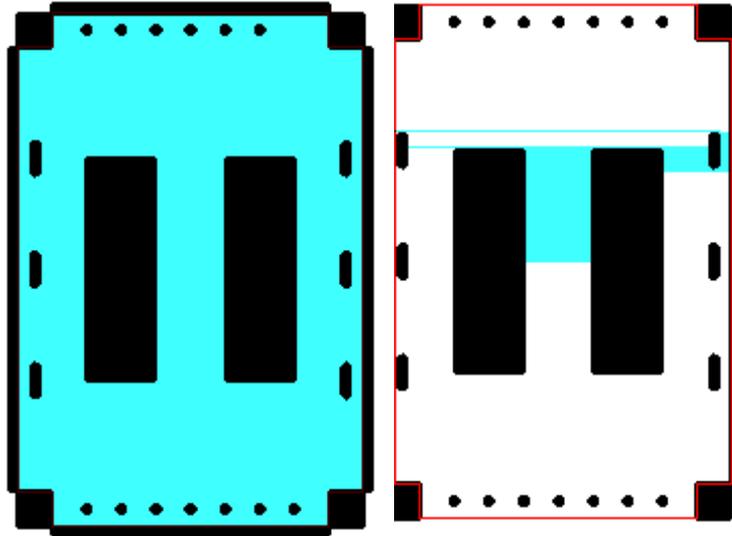


Figura 11. La detección de piezas ha sido satisfactoria en la imagen de la izquierda. En la derecha, la pieza ha sido descartada por no haber estado completamente cortada.

Este proceso (*flood filling*) acabará cuando se llegue a una frontera negra y/o cuando se llegue al borde de la imagen. Si se llega al borde de la imagen, se puede decir con seguridad que el contorno de pieza testeado no define una pieza real en la chapa y puede ser descartado. Si el proceso acaba antes, se puede decir que una pieza ha sido detectada.

En la Figura 11 se pueden ver dos ejemplos distintos, donde en uno se consigue detectar una pieza y en otro no. En el caso que no se detecta la pieza, el algoritmo *flood filling* acaba prematuramente al salirse la región azul de la línea roja que indica la potencial pieza a detectar.

5.2 Extracción mediante brazo robótico

A la máquina punzonadora se le puede adjuntar un brazo robótico para extraer las piezas detectadas de la chapa. Se ha diseñado un modelo 3D versátil, que permite ser configurado de manera personalizada.

Primeramente, el posicionamiento del brazo robótico no es fijo, ya que se puede mover siguiendo el eje Y. Es decir, todo el conjunto del brazo robótico se puede desplazar.

Los elementos individuales del brazo robótico son:

- Soporte del brazo robótico. Es una viga con soporte en el suelo por un lado y que se apoya en el puente de la máquina punzonadora.
- Carro. Es el elemento principal del brazo robótico. Se puede mover en el eje X y contiene todos los demás elementos del brazo robótico.
- Elevación en Z. Es el elemento del carro que permite subir y bajar. Se puede mover en el eje Z.

A partir de aquí, el modelo del brazo robótico puede tomar 2 configuraciones básicas. Con tablero o con bandas.

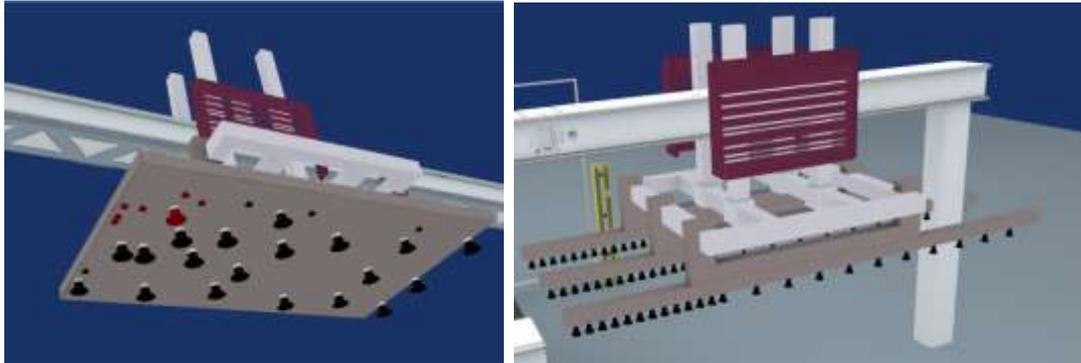


Figura 12. Configuraciones del brazo robótico: Con tablero y con bandas.

En la Figura 12 se puede ver la configuración con tablero. En este caso, el elemento tablero se compone de estos elementos:

- Tablero estático.
- Conjunto de ventosas distribuidas debajo del tablero según una configuración personalizada

En la Figura 12 también se puede ver la configuración con 3 bandas, aunque el número de bandas es configurable. En este caso, tenemos los siguientes elementos:

- Tablero de soporte. Elementos estáticos donde irán alojadas las bandas.
- Soporte de la banda. Cada elemento de estos puede moverse en los ejes Y.
- Banda. Cada banda va debajo del soporte de banda y se mueve en el eje X.
- Cada banda contiene un conjunto de ventosas según una configuración personalizada.

Para finalizar, el brazo robótico puede estar colocado a un lado o a otro de la máquina punzonadora.

5.3 Extracción mediante trampilla

El mecanismo de la trampilla es una modificación del modelo 3D de la mesa por la que caen las piezas sueltas que hubiere encima en el momento de su activación (ver Figura 13).

En el simulador la trampilla se activa a través de una instrucción especial. Tras su activación, se calculan las piezas sueltas que hay en la chapa utilizando los métodos

presentados anteriormente. Sin embargo, hay que limitar el detector de piezas sueltas a región donde está definida la trampilla y no de manera global.

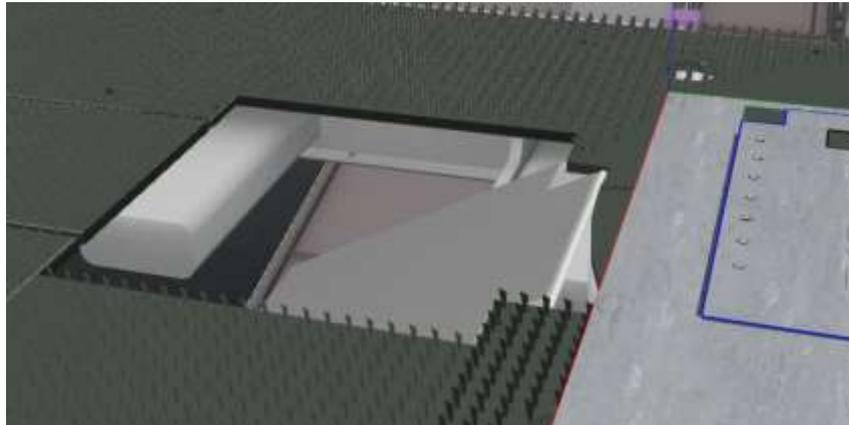


Figura 13. Trampilla de la máquina punzonadora.

6. Conclusiones

El simulador de corte y punzonado de chapa presentado en este trabajo se utiliza de manera integral a los programas de diseño y planificación de programas CNC. En cada instante del diseño se puede ejecutar el simulador para verificar que el resultado esperado y lo que el diseñador tiene en mente van convergiendo hacia una solución eficiente.

Las características principales del simulador permiten configurar muchos de los elementos fundamentales de las máquinas de corte y punzonado. Por un lado, se puede elegir la tecnología de corte (oxicorte, plasma, laser o agua) que incide en los modelos 3D visuales que se utilizan. Asimismo, las opciones de la máquina de punzonado permiten añadir y eliminar módulos según sean necesarios o no. Dentro de estos módulos encontramos los elementos de extracción de piezas sueltas: la trampilla y el brazo robótico, que a su vez puede ser configurado para dar cabida a múltiples representaciones visuales y funcionales.

Los procesos de corte son operaciones de alto coste computacional y se han diseñado algoritmos y metodologías que permiten optimizar la eficiencia global del simulador. Para ello, se ha introducido una subdivisión espacial a la representación de la chapa. Finalmente, se han cuidado los aspectos visuales, ya sean relacionados con el interfaz de usuario, con los modelos 3D o con los efectos visuales añadidos a los procesos de corte (chispas).

7. Trabajo Futuro

El simulador presentado en este trabajo tiene varias líneas en las que focalizarse para mejorar desde el punto de vista tecnológico.

Por un lado el rendimiento del simulador necesita un estudio más detallado sobre la influencia de la subdivisión espacial. Actualmente se utiliza una subdivisión espacial dada y fija para todos los programas CNC y durante toda la simulación. Este estudio podrá determinar si es más rentable utilizar distintos grados de subdivisión para cada programa CNC o si es interesante evaluar modificar la subdivisión espacial durante la simulación, de manera dinámica.

Por otro lado, se puede proveer al simulador de más funcionalidad, requerida por los usuarios para optimizar los resultados obtenidos. En este sentido destacamos la inclusión de un control de colisiones tanto lógico como geométrico. El control de colisiones geométrico calculará las colisiones entre distintas partes de la máquina 3D, ofreciendo avisos a los usuarios sobre esta circunstancia. El control de colisiones lógico será el encargado de controlar las dimensiones del espacio de trabajo, máximos y mínimos en los ejes de la máquina, etc.

Finalmente, podemos pensar en incluir una simulación más realista desde el punto de vista físico. La inclusión de aceleraciones y desviaciones respecto de la trayectoria teórica podrá revelar posibles incompatibilidades entre el programa CNC diseñado en el PC y el resultado obtenido según los parámetros físicos de una máquina determinada.

Agradecimientos

Este trabajo ha sido financiado por el Departamento de Industria del Gobierno Vasco a través del programa Gaitek.

REFERENCIAS

[1] Annunziata, M., Evans, P.C. "Industrial Internet: Pushing the Boundaries of Minds and Machine." General Electric, 2012.

[2] Blanchet, M. Rinn, T., von Thaden, G. de Thieulloy, G. "Industry 4.0: The New Industrial Revolution. How Europe will succeed." Roland Berger Strategy Consultants GmbH. March 2014.

[3] Drath, R.; Horch, A., "Industrie 4.0: Hit or Hype?" [Industry Forum]. Industrial Electronics Magazine, IEEE, vol.8, no.2, pp.56-58, June 2014.

[4] European Commission "Advanced computing" Available in <http://tinyurl.com/okpfpp4>, last visited 15/12/2014

[5] Final report of the Industrie 4.0 Working Group. "ACATECH: Recommendations for implementing the strategic initiative INDUSTRIE 4.0.", July 2014.

[6] Jay Lee, Edzel Lapira, Behrad Bagheri, and Hung-an Kao “Recent Advances and Trends in Predictive Manufacturing Systems in Big Data Environment.” *Manufacturing Letters* 1(1):38-41 (October 2013)

[7] Johnson A., 2012. Clipper - an open source freeware polygon clipping library. URL <http://www.angusj.com/delphi/clipper.php>.

[8] Kim, K.D, Kumar, P.R. “Cyber-Physical Systems: A Perspective at the Centennial.” *Proc. of the IEEE*, vol. 100, pp. 1287- 1308, May 2012.

[9] Lee, Edward A. "Cyber physical systems: Design challenges." *Object Oriented Real-Time Distributed Computing (ISORC)*, 2008 11th IEEE International Symposium on. IEEE, 2008.

[10] Leonov M.V., 1998. “Implementation of boolean operations on sets of polygons in the plane.”

[11] OpenSceneGraph, 2012. Open source 3D Graphics API over OpenGL. URL <http://www.openscenegraph.org/>.

[12] Preparata F.P. and Shamos M.I., 1985. “Geometry: An Introduction.” Springer-Verlag. ISBN 0-3879-6131-3.

[13] Ross, G. "Advanced Manufacturing and its implications for Business", 2013. Available online: <http://tiny.cc/5f5njx>

[14] Sha, L., Gopalakrishnan S., “Cyber-physical systems: A new frontier.” in *Machine Learning in Cyber Trust*. New York: Springer, pp. 3-13, 2009.

[15] Vatti B.R., 1992. “A Generic Solution to Polygon Clipping.” *Communications of the ACM*, 35(7), 56-63.