



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Rutina de postprocesado de G-code para la impresión de
grandes piezas de termoplástico a 45 grados

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: García Gascón, Luis

Tutor/a: Solanes Galbis, Juan Ernesto

CURSO ACADÉMICO: 2023/2024

Agradecimientos

En primer lugar, le agradezco a mi tutor, Ernesto Solanes, su apoyo, disponibilidad y ayuda para la realización y redacción de este TFG.

A mis compañeros del IDF y, en especial, a mi amigo Javi Bas, por su ayuda en el uso de la impresora y su constante disposición a colaborar para solucionar cualquier problema.

Y, por supuesto, agradezco a mis padres, a mi hermano y a mi tía Bego, que me han acompañado y apoyado incondicionalmente siempre.

Resum

Este TFG s'enfoca principalment en el desenvolupament d'un programa dissenyat per a modificar la inclinació de les figures generades per mig de software de còdic obert *Ultimaker Cura*, amb el propòsit de facilitar la impressió d'objectes de gran altura. El problema principal es relaciona amb la necessitat de realitzar impressions que superen el límit d'altura a la qual arriba la impressora i, per tant, és necessari recórrer a la divisió de la figura en parts per a després unir-les una vegada finalitzada les impressions, o bé, a l'ús de suports per a imprimir la peça en un pla inclinat.

En este projecte es proposa una rutina de càlcul per a la planificació de trajectòries de deposició a 45 graus en màquines de Fabricació Additiva de Gran Format. L'ús de filtres a 45 graus permet la fabricació de peces de gran grandària sense la necessitat d'usar suports. Esta estratègia de deposició s'està implantant en indústries dels materials compòsits per a fabricar aerogeneradors, ales d'avió, etc. La solució proposada es basa en la capacitat del programa *Ultimaker Cura* per a inclinar les figures, permetent d'esta manera, generar el G-code amb la inclinació necessària per a poder imprimir en un angle que evite la fragmentació i acoblament manual, optimitzant, d'esta manera, la impressió 3D de peces de gran grandària. S'usa el codi ISO convencional (G-code) generat per programes de codi obert i, mitjançant el programa desenvolupat en este projecte, s'efectua una transformació de coordenades per a posicionar la peça en l'espai, amb la inclinació que definisca l'usuari.

L'aplicació s'ha programat en Python i té una interface gràfica per a la visualització en 3D de les geometries. A més, compta amb una interface gràfica d'usuari (GUI) que permet seleccionar els paràmetres bàsics de l'operació, com ara angle de gir, posició en la taula de treball, inclusió d'ordes pre/post del G-code, fitxers d'entrada/eixida, etc. A més, el temps de càlcul s'ha minimitzat, ja que la rutina ha de gestionar un gran nombre de línies de codi, en alguns casos pot arribar a centenars de milers.

Mitjançant el desenvolupament d'esta eina es busca optimitzar el procés d'impressió 3D eliminant l'etapa de divisió d'impressions.

Paraules clau: G-code, Cura, Software, Fabricació Additiva

Resumen

Este TFG se enfoca principalmente en el desarrollo de un programa diseñado para modificar la inclinación de las figuras generadas mediante software de código abierto *Ultimaker Cura*, con el propósito de facilitar la impresión de objetos de gran altura. El problema principal se relaciona con la necesidad de realizar impresiones que superan el límite de altura a la que llega la impresora 3D y, por lo tanto, es necesario recurrir a la división de la figura en partes, para luego unir las una vez finalizadas las impresiones, o bien, al uso de soportes para imprimir la pieza en un plano inclinado.

En este proyecto se propone una rutina de cálculo para la planificación de trayectorias de deposición en máquinas de Fabricación Aditiva de Gran Formato, con boquillas a 45 grados. El uso de boquillas a 45 grados permite la fabricación de piezas de gran tamaño sin la necesidad de usar soportes. Esta estrategia de deposición se está implantando en industrias de los materiales compuestos para fabricar aerogeneradores, alas de avión, etc. La solución propuesta se basa en la capacidad del programa *Ultimaker Cura* para inclinar las figuras, permitiendo de este modo, generar el G-code con la inclinación necesaria para poder imprimir en un ángulo que evite la fragmentación y ensamblaje manual, optimizando, de este modo, la impresión 3D de piezas de gran tamaño. Se usa el código ISO convencional (G-code) generado por programas de código abierto y, mediante el programa desarrollado en este proyecto, se efectúa una transformación de coordenadas para posicionar la pieza en el espacio, con la inclinación que defina el usuario.

La aplicación se ha programado en Python y tiene una interface gráfica para la visualización en 3D de las geometrías. Además, cuenta con una Interfaz Gráfica de Usuario (GUI) que permite seleccionar los parámetros básicos de la operación, tales como ángulo de giro, posición en la mesa de trabajo, inclusión de órdenes pre/post del G-code, ficheros de entrada/salida, etc. Además, el tiempo de cálculo se ha minimizado, ya que la rutina debe gestionar un gran número de líneas de código que, en algunos casos, puede llegar a cientos de miles.

Mediante el desarrollo de esta herramienta se busca optimizar el proceso de impresión 3D eliminando la etapa de división de impresiones.

Palabras clave: G-code, Cura, Software, Fabricación Aditiva

Abstract

This TFG is mainly focused on the development of a program designed to modify the inclination of the figures generated using open source software *Ultimaker Cura*, with the purpose of facilitating the printing of tall objects. The main problem is related to the need to make prints that exceed the height limit of the 3D printer can reach and, therefore, it is necessary to divide the figure into parts and then join them together once the prints are finished, or to the use of supports to print the part on an inclined plane.

In this project we propose a calculation routine for the planning of deposition paths in large format Additive Manufacturing machines, with nozzles at 45 degrees. The use of 45-degree nozzles allows the production of large parts without the use of supports. This deposition strategy is being implemented in the composite material industries to manufacture wind turbines, aircraft wings, etc. The proposed solution is based on the ability of the *Ultimaker Cura* program to tilt the figures, thus allowing to generate the G-code with the necessary inclination to be able to print at an angle that avoids fragmentation and manual assembly, thus optimizing the 3D printing of large parts. The conventional ISO code (G-code) generated by open source programs is used and, by means of the program developed in this project, a coordinate transformation is performed to position the part in space, with the inclination defined by the user.

The application has been programmed in Python and has a graphical interface for 3D visualization of the geometries. In addition, it has a Graphical User Interface (GUI) that allows to select the basic parameters of the operation, such as angle of rotation, position on the workbench, inclusion of pre/post G-code commands, input/output files, etc. In addition, the computation time has been minimized, as the routine must handle a large number of lines of code, in some cases up to hundreds of thousands.

The development of this tool seeks to optimize the 3D printing process by eliminating the print splitting stage.

Key words: G-code, Cura, Software, Additive Manufacturing

Índice general

Índice general	IX
Índice de figuras	XI
Índice de tablas	XII
<hr/>	
1 Introducción	1
1.1 Motivación	2
1.2 Objetivos	3
1.3 Estructura de la memoria	3
2 Impresión 3D	5
2.1 Estado del arte de la Impresión 3D	7
2.2 Fabricación Aditiva por Extrusión de Materiales (MEX)	10
2.3 Impresión de Gran Formato	13
2.4 Antecedentes en procesos FDM	15
3 Tecnologías usadas	19
3.1 Software CAD/CAM/CAE	19
3.2 Slicers: Ultimaker Cura y Simplify3D	20
3.3 Software de programación: Matlab, Python	21
3.3.1 Matlab	21
3.3.2 Python	22
3.4 Lenguaje de programación G-code	23
3.5 Impresora 3D	26
4 Propuesta de rutina	35
4.1 Requisitos	36
4.2 Diseño de la interfaz gráfica de usuario	37
4.3 Implementación	39
4.3.1 Implementación de la interfaz gráfica de usuario: GUI_rotate.py	39
4.3.2 Implementación del código principal: rotate.py	42
5 Ejemplos Prácticos	47
5.1 Pala de aerogenerador	48
5.2 Tapas FSUPV	49
5.3 Atenuador impactos FSUPV	50
6 Conclusiones	53
6.1 Conclusiones del trabajo	53
6.2 Relación del trabajo con los Estudios cursados	54
7 Líneas futuras	57
Bibliografía	59
<hr/>	
APÉNDICES	
A Presupuesto	63
B Objetivos de Desarrollo Sostenible	65

Índice de figuras

1.1	Flujo de acciones para generar el fichero G-code.	1
2.1	Los 9 pilares tecnológicos de la Industria 4.0 [11]	5
2.2	Fabricación Aditiva frente a Fabricación Sustractiva.	6
2.3	Principales Procesos de Fabricación Aditiva.	9
2.4	Evolución de las tecnologías de AM.	10
2.5	Procesos de extrusión de materiales (MEX) [21].	11
2.6	Esquema equipo de extrusión de material.	12
2.7	Equipo Large Scale Additive Manufacturin (LSAM).	14
2.8	Ángulo límite impresión voladizos [35]	16
2.9	Principales causas de error en impresión 3D convencional [17]	17
2.10	Técnica ILP propuesta en [36] para impresión FDM.	18
3.1	Software de impresión <i>Cura</i>	21
3.2	Software de impresión <i>Simplify3D</i>	21
3.3	Formato instrucción G-code.	24
3.4	Ejemplo trozo de código G-code.	25
3.5	Super Discovery 3D printer.	27
3.6	Movimientos del cabezal de impresion.	27
3.7	Cabezal de impresion.	28
3.8	Vista de las diferentes zonas de la cama de impresión.	28
3.9	Estación de trabajo con vista detalle de los controladores y botones disponibles.	29
3.10	Cuadro eléctrico y microcontroladores.	30
3.11	Panel de control para la programación de la unidad de almacenamiento y acondicionamiento de material.	31
3.12	Macros para encender el calentamiento de la cama de impresión.	31
3.13	Software de control de la impresora.	32
3.14	Macro para crear un mapa de altura.	32
4.1	Esquema trabajo rutina Phyton.	35
4.2	Impresión de una pala de aerogenerador.	36
4.3	Diseño de la interfaz de usuario.	37
4.4	Ventana secundaria para modificar el inicio y el final del G-code.	38
4.5	Creación de la ventana principal.	39
4.6	Diseño de la sección para elegir el ángulo de rotación.	40
4.7	Selección de fichero G-code.	40
4.8	Actualiza la pieza descrita en el G-code_original.	41
4.9	Transforma el G-code_original y muestra las gráficas de las piezas original y rotada.	41
4.10	Flujo del código rotate.py	42
4.11	Lectura del fichero G-code.	43
4.12	Extrae las coordenadas de los puntos.	43
4.13	Matriz de transformación de coordenadas.	44

4.14	Cálculo de nuevas coordenadas.	44
4.15	Busca instrucciones anteriores y posteriores.	45
4.16	Escribe todas las instrucciones en el fichero G-code rotado.	45
5.1	Ejemplo componentes FSUPV G-code.	47
5.2	Ejemplo diseño álabe aerogenerador.	48
5.3	Ejemplo álabe aerogenerador construido.	48
5.4	Interfaz gráfica para la generación del nuevo G-code de la pala de aerogenerador.	49
5.5	Ejemplo tapa cover bottle FSUPV.	50
5.6	Interfaz gráfica para la generación del nuevo G-code de la tapa cover bottle FSUPV.	51
5.7	Interfaz gráfica para la generación del nuevo G-code de la tapa damper FSUPV.	51
5.8	Interfaz gráfica para la generación del nuevo G-code del atenuador de impactos.	52
5.9	Ejemplo impresión atenuador impactos FSUPV.	52
6.1	Empresas colaboradoras G-code.	54

Índice de tablas

2.1	Procesos de Fabricación Aditiva.	8
A.1	Costes de personal.	63
A.2	Costes de material.	63

CAPÍTULO 1

Introducción

La Fabricación Aditiva (*Additive Manufacturing, AM*) o impresión 3D comprende un conjunto de tecnologías de fabricación que se basan en el diseño y la construcción de objetos 3D agregando, capa a capa, algún tipo de material, desde plásticos a metales, a partir de un modelo virtual. Por ello, el primer paso consiste en modelar digitalmente el objeto que se pretende fabricar mediante algún software asistido por ordenador (CAD). El archivo 3D generado, normalmente en formato STL (fichero estereolitografía), se aproxima mediante triángulos y se divide en varias capas utilizando un software denominado *slicer* (en este caso, *Ultimaker Cura* [29] o *Simplify3D* [26]). Así, se genera un fichero G-code [28] que contiene la información de cada capa que se va a imprimir y en el que se pueden seleccionar los parámetros de impresión. Los archivos G-code se envían a la impresora 3D ya que contienen las instrucciones con la descripción de todas las acciones necesarias para imprimir la pieza final. La Figura 1.1 muestra el flujo de acciones para generar un archivo imprimible.

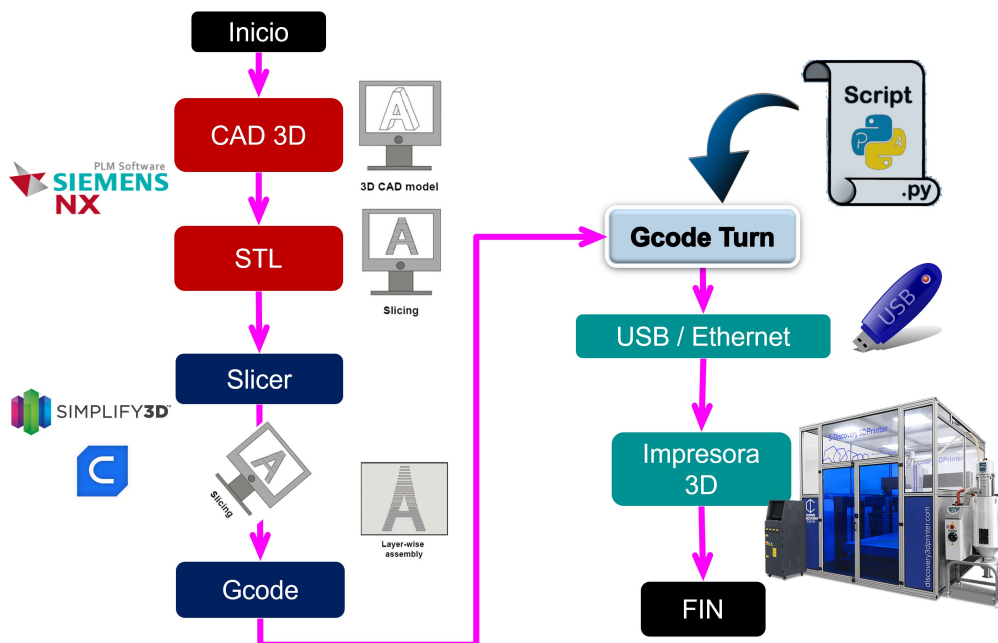


Figura 1.1: Flujo de acciones para generar el fichero G-code.

Durante el postprocesado de la pieza se seleccionan los datos de la impresora, los parámetros de fabricación de la pieza y se pueden programar nuevas rutinas para personalizar la impresión. Es en esta parte donde se ha visto la necesidad de incorporar una rutina específica que automatice la impresión de la pieza a 45 grados, dado que los pro-

gramas de laminado (*slicing*) y, en particular, el programa *Ultimaker Cura*, que es el más extendido en impresión 3D, no incluyen esta funcionalidad.

En este trabajo se propone la implementación de una técnica de impresión a 45 grados que requiere, por un lado, una impresora 3D con una boquilla que se pueda girar 45 grados alrededor de un eje horizontal, mientras que el movimiento de los ejes móviles (X, Y y Z) seguirá siendo idéntico al de una impresora convencional. No obstante, esta estrategia de impresión difiere de una estrategia convencional y requiere algoritmos y funcionalidades que los programas comerciales más usados (slicer *Ultimaker Cura* o *Simplify3D*) aún no han implementado. Por ello, por otro lado, será necesario desarrollar un software que sea capaz de cortar las piezas implementando la funcionalidad descrita. Esto conlleva una modificación del fichero G-code, que es el que define el movimiento de la impresora para imprimir la pieza final. Dado que se trata de una tarea relativamente compleja, en este trabajo se propone una rutina en Python [22] que, a partir del G-code generado por el slicer para la pieza original rotada un ángulo de 45 grados, devuelva un G-code modificado rotando las coordenadas de las trayectorias, de modo que contenga las órdenes necesarias para imprimir la pieza final con la inclinación de la boquilla a 45 grados. El software desarrollado en este trabajo está representado en el flujo de acciones de la Figura 1.1 como un script y tiene por objetivo transformar el G-code generado por el slicer para implementar esta funcionalidad.

Aunque en este documento se describe la versión de la rutina en Python, dado el carácter científico-técnico del problema que se plantea, se ha comenzado diseñando e implementando la rutina en Matlab. Esta primera versión ha servido para chequear y corregir posibles errores, así como para perfilar el diseño final de la aplicación. No obstante, debido a que Python es un lenguaje de uso más general, gratuito, con una gran colección de librerías y, por tanto, más flexible, la versión final se ha implementado en Python.

1.1 Motivación

Los procesos de Fabricación Aditiva abarcan la totalidad de materiales y campos de la Ingeniería, por lo tanto, hay una gran variedad de procesos de conformado. En el presente proyecto, se ha trabajado en la generación de trayectorias de deposición en los procesos de extrusión directa de termoplásticos, conocidos como MEX. A diferencia de otros procesos, el MEX utiliza como sustrato para la deposición de capas únicamente el material depositado en la capa anterior. Por lo tanto, si en la capa anterior hay zonas sin material se pierde el soporte y no es posible imprimir correctamente. Los procesos MEX tienen claros inconvenientes a la hora de fabricar voladizos o estructuras que superan la denominada regla de los 45 grados, utilizada por los diseñadores para determinar el máximo ángulo de impresión.

Un voladizo o *overhang* es cualquier parte de la pieza que se extiende más allá de la base creada por la capa anterior y que no posee, por tanto, un soporte. Los voladizos tienen limitaciones en su tamaño y la configuración de la máquina es fundamental para su correcta impresión. En cualquier caso, un correcto diseño deberá limitar en la medida de lo posible este tipo de estructuras. En general, se considera que una pieza puede imprimirse correctamente siempre que el ángulo máximo con respecto a la capa inferior esté entre 45 y 60 grados [35]. A partir de estos grados, es necesario generar estructuras de soporte para poder imprimir correctamente. El uso de estructuras de soporte tiene importantes desventajas, sobre todo, en componentes de gran escala. Así, por ejemplo:

- La impresión requiere más material, lo que supone un coste adicional.
- Incremento del tiempo de impresión.

- Postproceso de la pieza, es decir, retirar las estructuras de soporte.
- Riesgo de dañar la pieza en las operaciones de postprocesado.

La impresión con boquillas orientadas a 45° respecto a la vertical de la máquina de impresión permite la fabricación de piezas con voladizos hasta 90° grados, lo que supone poder hacer piezas totalmente huecas que serían imposibles de fabricar de manera convencional. Asimismo, la impresión a 45° grados permite fabricar los componentes con un menor número de reposicionamientos de la boquilla.

Para superar las limitaciones anteriormente mencionadas en cuanto a la fabricación de voladizos y soportes, numerosos fabricantes han desarrollado métodos de impresión alternativos e introducido innovaciones técnicas. Las impresoras con cabezal a 45° han aparecido recientemente en el mercado, y sus referentes industriales en el mercado son pocos.

Para la elaboración del presente trabajo se ha usado un equipo de deposición fundida de gran formato de granza. La impresora es la Discovery3D (CNC Bárcenas, Valdepeñas, España) y fue diseñada específicamente para la Universitat Politècnica de València para poder hacer impresiones a 45° , con una cama lo suficientemente grande como para poder imprimir moldes de gran formato para las industrias de la automoción, naval y eólica. En el momento de la compra de la máquina, no se disponía de software capaz de imprimir a 45° y su desarrollo quedará finalizado con el presente trabajo fin de grado. Se parte de la programación de algunas rutinas preliminares de software de postprocesado en Matlab, que han servido de apoyo a la realización de este TFG. La rutina final desarrollada está programada en Python y se tiene prevista su distribución entre las empresas colaboradoras con la UPV. Actualmente, la están testeando empresas de España, Finlandia y Canadá.

1.2 Objetivos

El objetivo principal de este proyecto es el diseño e implementación de software que permita imprimir piezas, e incluso moldes, en Fabricación Aditiva de Gran Formato (*Large Format Additive Manufacturing, LFAM*) mediante extrusión de material (*Material EXtrusion, MEX*) de pellets de termoplástico, con una inclinación de 45 grados, sin usar estructuras de soporte. La rutina de cálculo se programará en Python y se basará en una transformación de la geometría que se realizará sobre un G-code convencional. En una primera aproximación se realizará una rotación del plano de impresión para dejarlo a 45° , pero se dejará preparada la rutina para poder aplicar cualquier transformación geométrica de las coordenadas de las líneas de deposición.

1.3 Estructura de la memoria

La memoria está estructurada en diferentes capítulos, cuyo contenido se expone brevemente a continuación.

- **Capítulo 1: Introducción**, donde se describe la motivación, objetivos y estructura del proyecto.
- **Capítulo 2: Impresión 3D**. En este capítulo se presenta de manera breve el contexto y las bases de la impresión 3D, especialmente para el caso de la fabricación aditiva para piezas de gran formato. Se expone brevemente su evolución, ligada a la de las

tecnologías de la información, así como los antecedentes de la problemática objeto de estudio de este TFG y las soluciones propuestas en la bibliografía para este problema, aplicadas en otros procesos de impresión 3D.

- **Capítulo 3: Tecnologías usadas.** En este apartado se comenta el software empleado (CAD, CAM, slicers y de programación: Matlab y Python), así como las características de la impresora 3D con la que se ha realizado la parte experimental y que ha permitido comprobar los beneficios del código desarrollado. Finalmente, para entender la rutina es necesario estar familiarizado con las instrucciones del lenguaje que se usa en impresión 3D, de ahí que se explique la estructura y algunos de los comandos más importantes de cualquier archivo G-code.
- **Capítulo 4: Propuesta de rutina.** En este capítulo se explica con detalle el código Python descrito para implementar la funcionalidad objeto de este TFG. Se exponen las partes más relevantes para entender el programa.
- **Capítulo 5: Ejemplos Prácticos.** En este capítulo se muestran varias piezas reales, de gran formato, fabricadas con la impresora 3D, usando el G-code generado por el código descrito.
- **Capítulo 6. Conclusiones.** Aquí se exponen las conclusiones del trabajo. Además, se incluye un apartado en el que se comenta la relación con diferentes asignaturas cursadas durante la carrera.
- **Capítulo 7. Trabajos futuros.** En esta parte se proponen posibles mejoras de la rutina que podrían incluirse en un futuro, enriqueciendo el código final.

CAPÍTULO 2

Impresión 3D

La Fabricación Aditiva (AM), a menudo denominada impresión 3D, es uno de los pilares de la Cuarta Revolución Industrial, también denominada Industria 4.0, concepto introducido en 2011 para referirse a una nueva forma de fabricar productos, y que ha supuesto una revolución que involucra a todos los factores implicados en el desarrollo industrial, desde los recursos, las empresas o su gestión, hasta la sociedad y el mundo en el que vivimos.

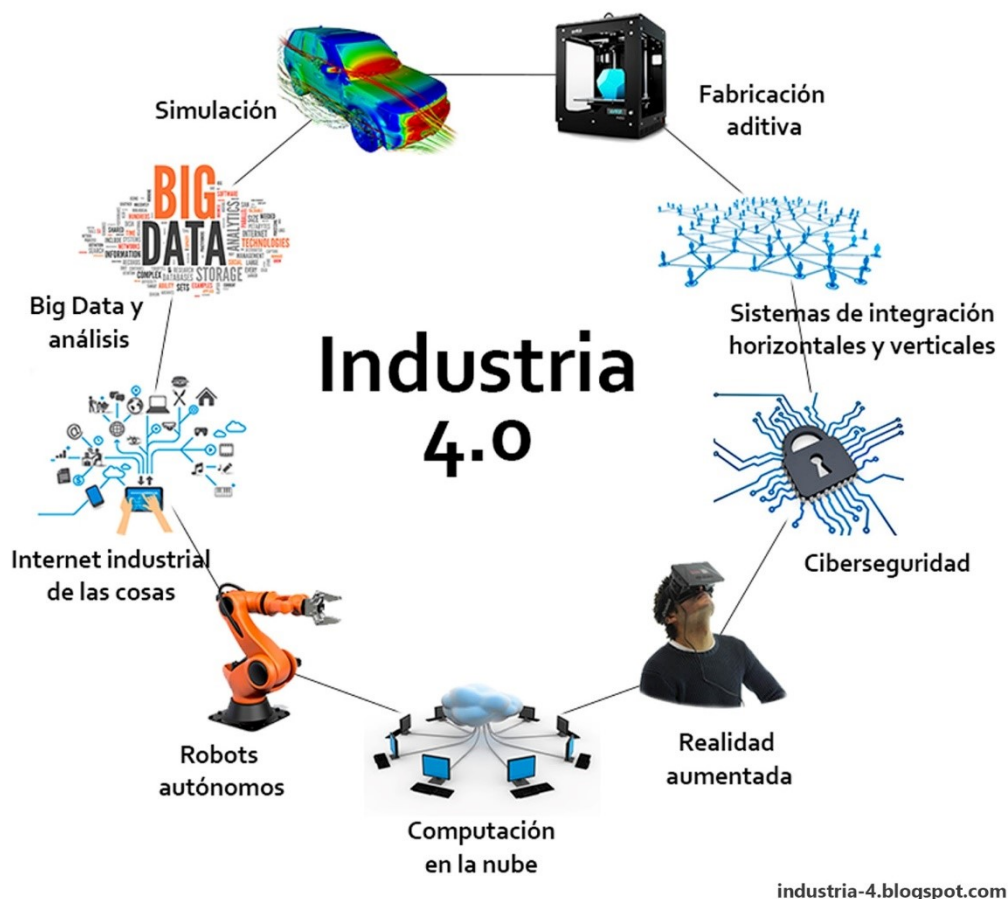


Figura 2.1: Los 9 pilares tecnológicos de la Industria 4.0 [11]

La industria 4.0 se puede interpretar como una estrategia para mejorar los sistemas productivos y mantener la competitividad de las empresas. Se basa en la interconexión de activos de todo tipo (productos, recursos, ...) así como en la introducción de las úl-

timas tecnologías de la información y las comunicaciones en la Industria [7, 23], consiguiendo que las fábricas sean inteligentes, interconectadas, fácilmente monitorizables, auto-reparables y con poca intervención humana.

Como se observa en la Figura 2.1, junto a la impresión 3D, la Industria 4.0 se sustenta en otras técnicas digitales, todas ellas basadas en los avances en las Tecnologías de la Información, entre las que destacan el Análisis de Datos (Big Data), la Computación en la Nube (Cloud Computing), la Inteligencia Artificial, la Simulación Virtual, los Robots Autónomos (drones, entre ellos), la Realidad Aumentada, la Ciberseguridad, la Modularización o el Internet de las Cosas.

La impresión 3D o tecnología AM ha sido definida por la International Standards Organization/American Society for Testing and Materials Standards [2] como *el proceso de unión de materiales para fabricar piezas a partir de datos de modelos 3D, normalmente capa sobre capa*. Esto contrasta con la Fabricación Sustractiva (SM) convencional, en la que el material se mecaniza a partir de una pieza de trabajo hasta obtener la forma deseada.

Las técnicas de fabricación digital se dividen en dos grandes grupos: técnicas SM y técnicas AM. Las primeras, como una fresadora o un torno, usan un sistema informático para controlar mediante Control Numérico Computarizado (CNC) los movimientos de una máquina en un sistema de coordenadas XYZ. Así, la máquina sigue las instrucciones detalladas en un programa G-code para llegar, mediante cortes del material, a fabricar la pieza que se desea, eliminando el material sobrante hasta conseguir la forma de la pieza. Los procesos del mecanizado por arranque de viruta son un claro ejemplo de este tipo de tecnologías, que suponen un gran desperdicio de energía y material. Sin embargo, su gran polivalencia ha hecho que sean los más utilizados en la mayoría de las industrias. Dichas técnicas han ido evolucionando desde las primeras máquinas automatizadas por CNC y, junto con el desarrollo de la Informática, concretamente, con el avance del software de Diseño Asistido por Ordenador (CAD) y la Fabricación Asistida por Ordenador (CAM), hasta llevar a las técnicas de AM, en las que se fabrica una pieza mediante la superposición de capas del material, siguiendo el diseño geométrico 3D de la pieza a fabricar, previamente generado en un fichero STL, donde cada capa representa una sección transversal de la pieza que se quiere fabricar. La Figura 2.2 muestra cómo la Fabricación Aditiva es ventajosa frente a la Sustractiva en cuanto a reducción de coste cuando el número de unidades a fabricar es pequeño, o también al aumentar la complejidad de la fabricación.

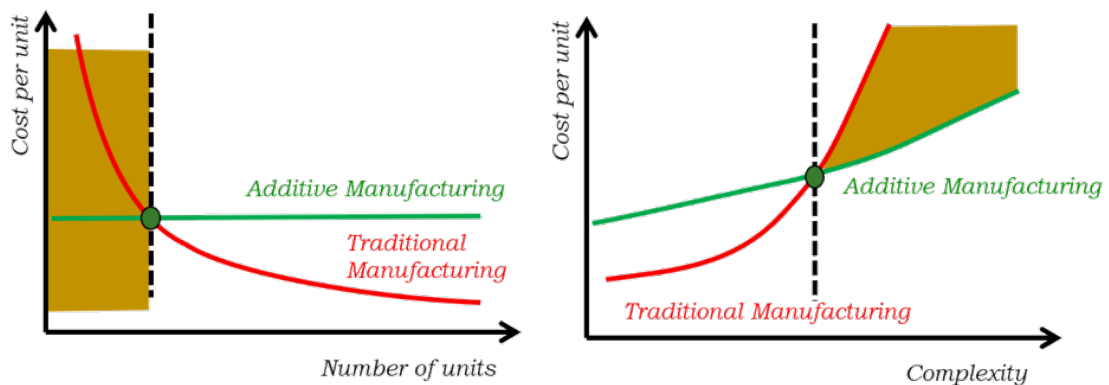


Figura 2.2: Fabricación Aditiva frente a Fabricación Sustractiva.

Las tecnologías aditivas se basan en aportar el material en lugar de eliminarlo, lo que hace que los residuos sean muy inferiores y, por lo tanto, mucho más ecológico. Estos

procesos se basan en transformar un modelo digital del componente a fabricar en sucesivas capas que se van depositando, siguiendo unas trayectorias definidas numéricamente mediante un software laminador (*Slicer*). El lenguaje utilizado para comunicar todas las órdenes de deposición del material sobre la pieza, denominado G-code, se basa en el código ISO definido para el control de las máquinas de CNC. Por ello, la AM supone un ahorro de material, ya que se basa en ir añadiendo capa a capa lo necesario para obtener la pieza, mientras la SM realiza el mecanizado CNC eliminando material de un bloque hasta obtener la forma deseada, dejando más residuos. No obstante, en algunos casos, puede ser más interesante la fabricación sustractiva, en particular, cuando el número de unidades es alto, debido a que la impresión 3D todavía es un proceso lento, o también cuando se fabrican piezas de gran tamaño. Por otro lado, el coste de las impresoras 3D es elevado y el postprocesado de las piezas puede ser más elaborado. En general, los factores que determinarán la elección de un sistema u otro serán el coste, el número de unidades, el tamaño y la complejidad de las piezas que se quieran fabricar [13].

2.1 Estado del arte de la Impresión 3D

Uno de los primeros procesos de AM surge en la década de los 80, gracias a la introducción de la creación rápida de prototipos (*Rapid prototyping, RP*). Se basa en crear capa por capa un objeto 3D utilizando diseño asistido por ordenador, dando lugar así al primer proceso comercial en AM, *la estereolitografía, SLA*, que surgió en 1986 y fue patentada por Charles Hull (fundador de 3D Systems) [16].

Originalmente, los procesos de AM eran conocidos como Prototipos Rápidos (RP), en los que las piezas finales no eran funcionales y sólo servían para verificar dimensiones, ensamblajes o como demostradores. El beneficio industrial asociado a estos primeros trabajos era muy reducido, pero se puso de manifiesto el gran potencial de esta tecnología. En la década de los 90 se empezaron a desarrollar los primeros procesos de piezas funcionales, tanto en materiales poliméricos como metálicos (Nylon, ABS o aleaciones metálicas como acero inoxidable, aleaciones base níquel o base titanio). Se empieza a hablar de *Rapid Manufacturing*. Actualmente, se denomina impresión 3D a los procesos de bajo coste, generalmente, del entorno doméstico o semiprofesional. Fabricación Aditiva es el último término aplicado y se utiliza para describir toda la tecnología.

El desarrollo de la RP supuso un gran avance en el desarrollo de productos y su evolución fue de la mano de otras tecnologías, cuya combinación hizo posible la impresión de objetos tridimensionales [5, 10, 18]. Nos referimos a técnicas de Computer-Aided Design (CAD), Computer-Aided Manufacturing (CAM) y Computer Numerical Control (CNC). Entre las ventajas de la RP destacan la reducción de tiempo y costes, la posibilidad de crear piezas, no solo modelos, y una mayor participación humana en el proceso.

La AM ha experimentado un gran crecimiento a partir de la aparición del RP, gracias al uso de nuevos materiales como consecuencia del avance en ciencia e ingeniería de los materiales y la liberalización de patentes, sobre todo a partir de 2014 (por ejemplo, la patente del proceso FDM de la empresa norteamericana Stratasys). Por otra parte, la aparición del movimiento RepRap (*Replicating Rapid Prototyper*), proyecto de hardware libre, iniciado por el profesor Adrian Browyer en 2005 [15], con la creación de la impresora 3D autorreplicable, junto con la difusión de la impresión 3D de Tecnología Abierta (*Open Source*), que permite el uso libre de código a cambio de compartirlo, ha hecho que la fabricación aditiva experimente un gran avance durante los últimos años.

La creación rápida de prototipos tiene, sin embargo, limitaciones como que las dimensiones de las piezas podrían ser mayores que las de las impresoras de AM, o que no se

pueda imprimir con todo tipo de materiales de uso común en la fabricación de materiales [10].

Hay un gran número de procesos de conformado AM, dependiendo básicamente de los materiales empleados, cómo se crean las capas y cómo se unen entre ellas. En la Tabla 2.1 se pueden ver las principales tecnologías de fabricación por AM. La tecnología AM puede clasificarse en procesos basados en sólidos, líquidos y polvos en función del estado físico de la materia prima antes de iniciar el proceso de AM, es decir, sólidos (filamentos, alambres u otros), pastas (suspensión de material granular), gránulos (piezas pequeñas), etc.

Material	Herramientas	Procesos	Nombre
Líquido	Radiación Láser	Solidificación	Fotopolimerización con láser, Estereolitografía (SLA)
			Fotopolimerización por proyección por máscara (DLP)
			Impresión láser UV de sólidos
	Lámpara de luz UV		Fotopolimerización
Sólido	Cabezal de extrusión	Fundido/Solidificado	Modelado por deposición fundida (FDM)
	Cabezal de inyección		Material Jetting - Polyjet
	Laminación	Laminación	Fabricación por laminación (SDL)
Polvo	Aglutinante	Unión	Impresión 3D (3DP)
	Láser	Sinterizado	Sinterizado por láser (SLS-DMLS)

Tipo	Tecnologías	Materiales
Foto-Polimerizado	Estereolitografía (SLA)	Fotopolímeros
	Fotopolimerización por proyección por máscara (DLP)	Resinas líquidas
Extrusión	Modelado por deposición fundida (FDM)	Termoplásticos (PLA, ABS, etc.), aleaciones eutécticas de metales, productos comestibles
Impresión	Inyección de material (Material Jetting)	Ceras, fotopolímeros o metales
	Inyección de aglutinante (Binder Jetting)	Cualquier material en polvo (Usualmente metales y cerámicos, aunque también maderas)
Fusión en lecho de polvo	Sinterizado por láser (SLS-DMLS)	Materiales en forma de polvo, principalmente metales y aleaciones, materiales compuestos, cerámicos, termoplásticos y elastómeros termoplásticos.
	Fundido Selectivo por Láser (SLM)	
	Fusión por haz de electrones (EBM)	
Laminación	Laminated Object Manufacturing (LOM)	Láminas de termoplásticos, papel/cartón, fibra de carbono y otros materiales compuestos o metales (chapas).
	Selective Deposition Lamination (SDL)	
Deposición directa de energía	Laser Cladding	Materiales en polvo o compactados, principalmente metales y aleaciones.
	Fusión de iones (IFF)	

Tabla 2.1: Procesos de Fabricación Aditiva.

En la Figura 2.3 se muestra una visión general de los principales procesos de Fabricación Aditiva, divididos según el tipo de material utilizado (polímeros o metales). Los procesos más conocidos son la estereolitografía (SLA), modelado por deposición de hilo (FDM), la sinterización selectiva por láser (SLS, SLM) y la fundición por haz de electrones (EBM). En [16] se proporciona una buena descripción general de la AM.

En la Figura 2.4 se representa la evolución histórica de las tecnologías de Fabricación Aditiva, indicando los principales hitos. Se pone de manifiesto que el crecimiento es ex-

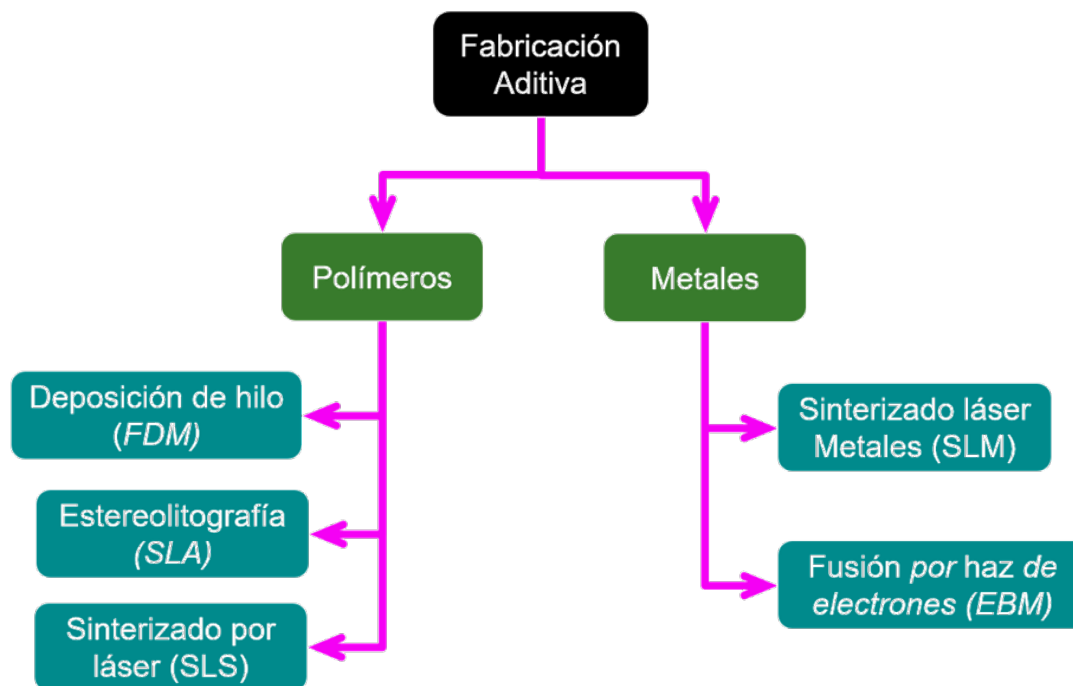


Figura 2.3: Principales Procesos de Fabricación Aditiva.

ponencial, tanto en patentes industriales como en publicaciones de investigación. Cabe destacar que este crecimiento se está produciendo de manera transversal a la totalidad de las actividades industriales. Se considera que el primer referente de estas tecnologías es la patente publicada en 1983 por el ingeniero Charles Hull, que supuso el nacimiento de la empresa 3D Systems. Es uno de los inventores de la impresora 3D SLA, la primera tecnología comercial de prototipado rápido, y del formato de archivo STL, ampliamente utilizado.

Como se ha señalado, la *Estereolitografía (SLA)* es el proceso de impresión 3D más antiguo y se basa en la fotopolimerización, es decir, crea la pieza 3D solidificando, capa por capa, resinas fotosensibles mediante un láser ultravioleta. La *fusión por haz de electrones (EBM)* usa un rayo de electrones en lugar de un láser, como fuente de energía, para fundir el polvo de metal capa por capa. La técnica de *sinterizado selectivo por láser (SLS)* usa un láser CO₂ para sinterizar la capa de polvo a través de unos puntos concretos, provocando la solidificación de las partículas, mientras que la *fusión selectiva por láser (SLM)* usa un láser para fundir polvos metálicos.

Actualmente, *Fused Deposition Modelling (FDM)* es considerado el proceso de impresión 3D más común y extendido. Se trata del primer proceso de AM basado en la extrusión, que fue inventado en 1989 por S. Scott Crump [20], cofundador de Stratasys Ltd. (Rehovot, Israel y Eden Prairie, EE. UU.). El mecanismo FDM utiliza la extrusión de filamento licuado a través de la boquilla de un cabezal de impresión para depositar sucesivamente capas y fabricar así las piezas deseadas. Esta invención fue patentada y comercializada por Stratasys dos años más tarde, en 1991 [23]. Los materiales típicos utilizados en FDM son polímeros de alto contenido, como el acrilonitrilo butadieno estireno (ABS) [23], polilactidos (PLA) y otros bioplásticos [5].

El proceso FDM es una de las tecnologías de AM más comunes puesto que es uno de los métodos de impresión 3D más sencillos y no requiere componentes caros como un láser. Se puede conseguir tanto con impresoras FDM de sobremesa como con sistemas de impresión profesionales y, aun así, sigue siendo capaz de fabricar geometrías y

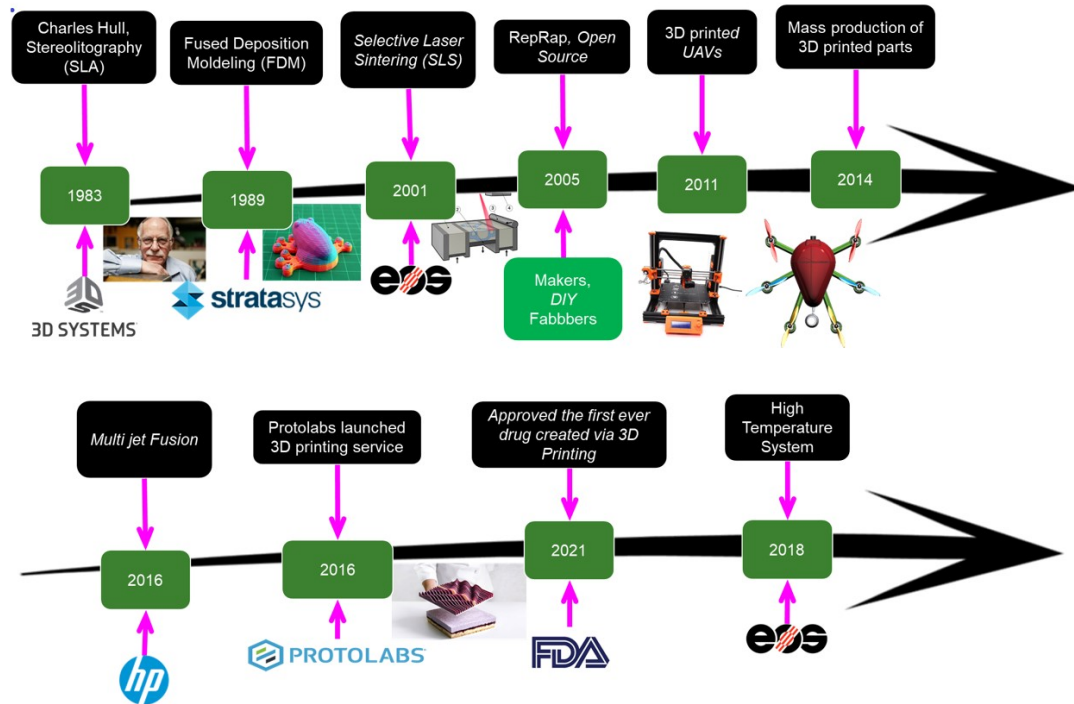


Figura 2.4: Evolución de las tecnologías de AM.

cavidades complejas con una precisión dimensional razonable. No obstante, todavía se pueden encontrar algunas restricciones e inconvenientes frente a las piezas creadas por métodos convencionales, como técnicas de inyección y compresión, principalmente en cuanto a propiedades mecánicas inferiores de las piezas creadas por FDM, cuya calidad en ese aspecto es menor.

Por otro lado, los procesos FDM son adecuados para imprimir piezas pequeñas y medianas, con paredes finas, geometrías complejas y orificios pequeños que requieren boquillas de menor diámetro para conseguir una altura de capa menor y un ancho de línea más pequeño. Su productividad es de hasta 80 gr/hora. Sin embargo, existen industrias, tales como las de energías renovables, aeroespacial, construcción naval y defensa, entre otras, que requieren imprimir piezas de gran formato (LFAM). Por ello, existen numerosas iniciativas para el desarrollo de impresoras de gran formato que sean capaces de imprimir piezas finales e incluso moldes para composites. En particular, este proyecto se enmarca en los procesos de Fabricación Aditiva de Gran Formato por Extrusión de Materiales.

2.2 Fabricación Aditiva por Extrusión de Materiales (MEX)

En los procesos MEX, la materia prima compuesta de termoplástico con o sin carga de fibra se calienta hasta que el plástico se ablanda y puede extruirse a través de una boquilla de impresión. A continuación, el material impreso se deposita en el lecho de impresión, que se calienta para aumentar la adherencia entre las piezas impresas y el lecho, de modo que la pieza 3D se crea capa por capa siguiendo el modelo CAD. Dependiendo del sistema de alimentación de la impresora, el proceso MEX se puede clasificar en tres tipos: (a) de tornillo, (b) de émbolo y (c) de filamento, tal y como se detalla en la Figura 2.5.

En la Figura 2.6 se muestran los principales componentes mecánicos de un equipo de impresión por extrusión de termoplásticos. Cuenta con un motor paso a paso para con-

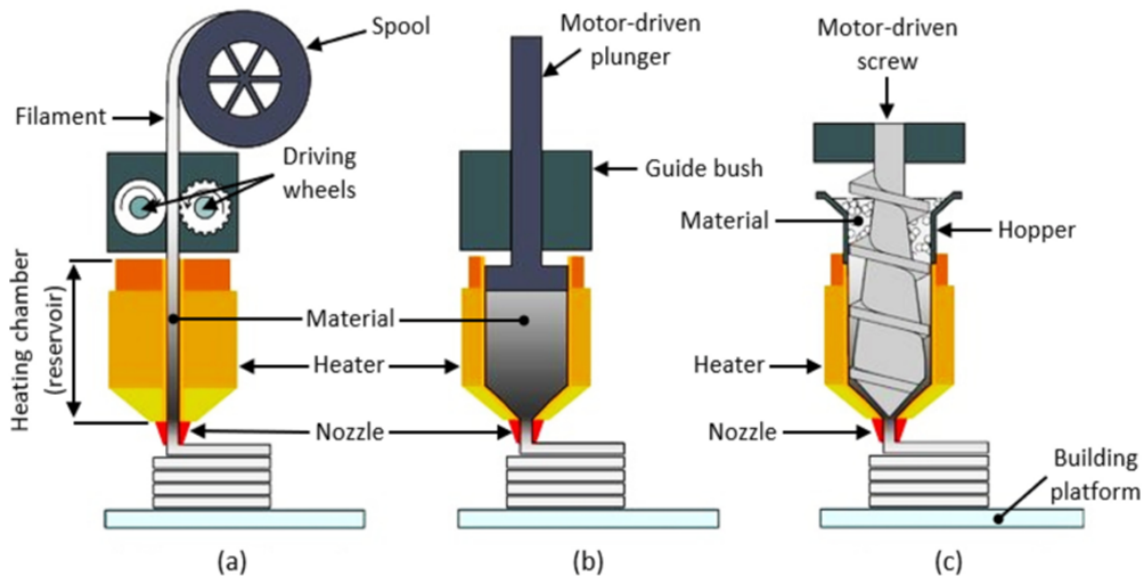


Figura 2.5: Procesos de extrusión de materiales (MEX) [21].

trolar el movimiento en los tres grados de libertad de la geometría de la pieza (XYZ) y un cuarto grado de libertad para controlar el caudal de deposición del material. Además, incorpora diferentes sensores de temperatura, posición, etc. La temperatura de impresión se controla mediante una resistencia ubicada en la boquilla de extrusión, en el caso de la extrusión por hilo, y de varias en el husillo de extrusión, para el caso de las impresoras de grana. Finalmente, posee un control de temperatura en la superficie de deposición para asegurar una correcta adhesión de la primera capa de la pieza y permite que no se desprenda durante la impresión. Las impresoras cuentan con una placa de control donde se leen las instrucciones de impresión (G-code) para generar los pasos para el desplazamiento de los motores y regular los movimientos con la información suministrada por los sensores.

La AM basada en la extrusión puede ser de dos tipos: fabricación basada en filamentos y fabricación basada en gránulos. En el primer caso, el material es suministrado en forma de filamento de una manera continua. La impresora se controla por un ordenador y tiene al menos dos extrusores: uno para el material base y otro para el material de soporte. En esta máquina las piezas se fabrican depositando material en capas sobre una plataforma para darle una forma prescrita controlada por el ordenador. El material depositado se va enfriando y uniéndose al material vecino. Una vez finalizada la deposición de una capa, la plataforma se desplaza hacia abajo una distancia determinada, la altura del filamento, y se continúa depositando la siguiente capa sobre la última depositada [21].

Las impresoras extrusoras de hilo utilizan extrusores Bowden o extrusores de accionamiento directo. Ambos extrusores funcionan de manera similar, solo se diferencian en la ubicación. Normalmente, los extrusores Bowden se sitúan en el bastidor de la impresora, lo que permite que el filamento llegue al extremo caliente a través de un tubo de politetrafluoroetileno (PTFE). La disposición del tubo Bowden ofrece la ventaja de un movimiento más limpio, ya que el peso sobre el carro es significativamente menor; por lo tanto, los movimientos son más rápidos y silenciosos, y la calidad de impresión es alta. Entre los inconvenientes de esta configuración figura el hecho de que el extrusor debe empujar el filamento a través de un tubo largo y, por ello, necesita un motor con gran potencia. Además, el tiempo de respuesta es bajo e implica complicaciones de materiales, ya que algunos materiales pueden desgastar el tubo Bowden.

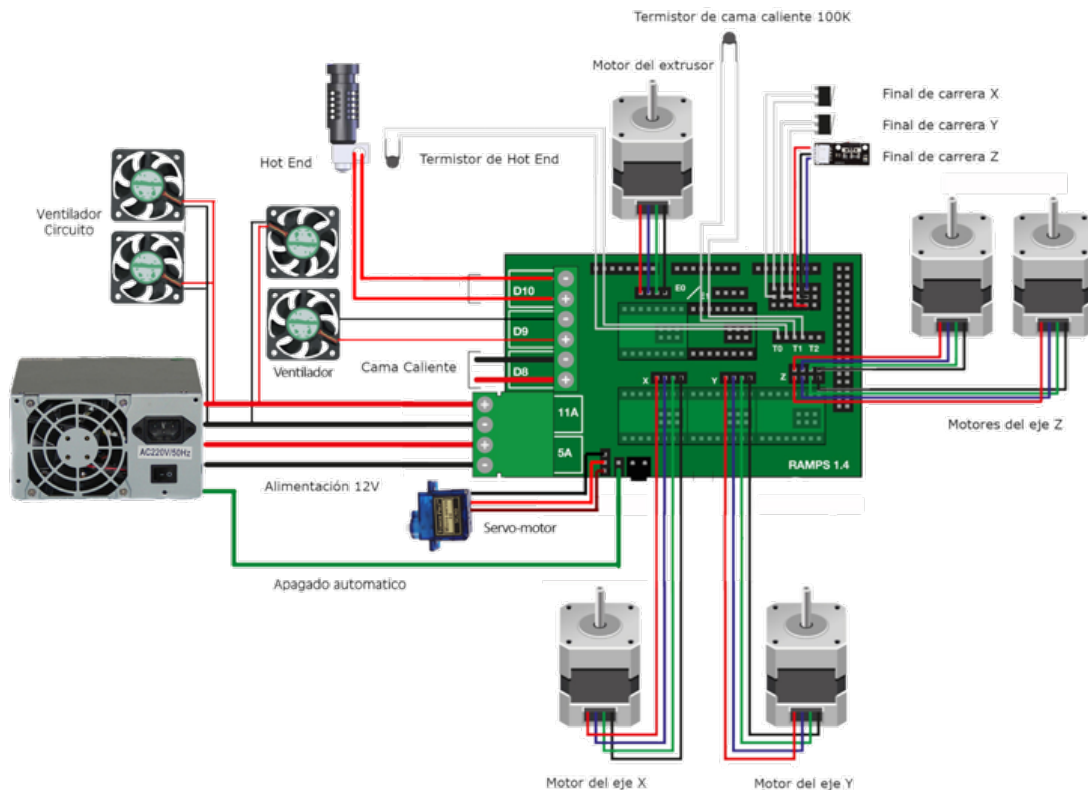


Figura 2.6: Esquema equipo de extrusión de material.

Por otra parte, en una disposición de accionamiento directo el extrusor se coloca sobre el cabezal de impresión, empujando el filamento directamente hacia el extremo caliente. Entre las ventajas de esta configuración destacan que la extrusión es más fiable, una mejor retracción, se requiere un motor de menor potencia y se pueden usar una amplia gama de materiales de filamento. En cuanto a los inconvenientes, destaca la presencia de vibraciones y que requiere un mayor mantenimiento debido al peso añadido sobre el cabezal de impresión. Este peso adicional provoca la reducción de la velocidad, aumento del bamboleo y, posiblemente, pérdida de precisión en los movimientos X e Y.

En la impresión 3D por extrusión basada en gránulos, como se muestra en la Figura 2.5, los gránulos de material se introducen en un barril mediante una tolva que, a menudo, contiene numerosas zonas de aporte de calor. En este caso, los gránulos son impulsados mediante un tornillo motorizado dentro del barril para descargarlos a través de una boquilla de impresión, para lo cual el material fundido se somete a altas presiones. Se distinguen tres zonas en el tornillo: de alimentación, de compresión y de fusión. El sistema de extrusión basado en gránulos es de menor coste que el basado en filamentos, dado que no requiere material en forma de filamento, lo que supone un ahorro en el coste del material y, además, el tiempo de impresión requerido también es inferior. Actualmente, están alcanzando gran protagonismo las impresoras 3D de extrusión basadas en pellets, especialmente para materiales con alto contenido en fibra de carbono, que se utilizan con frecuencia para minimizar el alabeo y aumentar la resistencia.

Los extrusores de pellets de menor coste ofrecen muchas posibilidades para la impresión 3D y, por lo general, pueden acelerar una impresión en un tiempo considerable. Por otro lado, un extrusor de filamento puede generar impresiones con tolerancias más aceptables y formas más complicadas. Con los avances en la química de materiales y la síntesis de polímeros, se crean y validan constantemente nuevos materiales poliméricos para su procesabilidad y adecuación a la AM. Hoy en día, la impresión de polímeros a

alta temperatura está ganando popularidad debido a sus aplicaciones en distintos campos, como productos de consumo, biomedicina, aeroespacial, automoción y aplicaciones de alto estrés. Estos materiales son muy útiles ya que son aplicables en diferentes sectores ambientales donde la resistencia al calor juega un papel crítico, en lugar de estos materiales otros pueden fundirse o ablandarse en estas condiciones ambientales.

2.3 Impresión de Gran Formato

La Fabricación Aditiva de Gran Formato (LFAM) mediante Extrusión de Material (MEX) de pellets de termoplástico ha surgido como una tecnología prometedora para producir piezas de gran tamaño en diversas industrias, como la automovilística, aeroespacial, construcción, energía, etc. A diferencia de las tecnologías convencionales de AM que se centran en piezas pequeñas y medianas, la LFAM permite la fabricación de piezas con dimensiones que alcanzan varios metros [30]. Esta tecnología utiliza principios de deposición por capas de material, empleando técnicas como el Modelado por Deposición Fundida (FDM), la Fabricación de Granulado Fundido (Fused Granulate Fabrication, FGF) o la Deposición de Polvo.

Sin embargo, a diferencia de las impresoras 3D tradicionales, la LFAM utiliza sistemas de extrusión y boquillas más grandes para depositar material en volúmenes significativamente mayores. Esta capacidad permite la fabricación de objetos a escala real, incluidos moldes, piezas arquitectónicas, estructuras de refuerzo, prototipos de automoción o componentes aeronáuticos. Entre las ventajas del LFAM destaca su rapidez de producción. Al utilizar boquillas más grandes, el proceso de deposición se acelera considerablemente, lo que permite la fabricación de piezas de gran tamaño. Además, ofrece la flexibilidad de utilizar una amplia gama de materiales, incluidos termoplásticos, termoestables, cerámicos y metales. Esto permite seleccionar el material más adecuado para aplicaciones específicas.

Los procesos de fabricación Aditiva de Gran Tamaño de Granza (*Large Fuse Granzze Deposition, LFGD*) pertenecen a la familia de los procesos de Fabricación Aditiva por MEX. Este término se aplica a los equipos capaces de depositar material en granza en volúmenes superiores a un metro cúbico. La terminología de las tecnologías de AM se describe en la norma ASTM F2792-12a (Standard Terminology for Additive Manufacturing Technologies) [2].

Los pellets son una alternativa muy ventajosa a la tradicional extrusión de polímeros por hilo debido a su rentabilidad en comparación con los filamentos. Además, la eliminación de la etapa de extrusión de filamentos en el proceso de fabricación aumenta aún más la eficiencia de estos sistemas. Se han utilizado varios nombres para designar los sistemas de extrusión basados en granza de termoplástico en AM. En cuanto a los sistemas de extrusión se han propuesto varias alternativas; por ejemplo, Volpato et al. [31] propusieron un sistema de extrusión accionado por pistón que utiliza polipropileno, adecuado para procesos de extrusión continua [33]. El concepto de Deposición de Granza Fundida (FGD) fue introducido por Wang et al. [32], a mayor escala. Este enfoque innovador combina sistemas de fabricación capa a capa de la impresión 3D convencional con sistemas de extrusión de husillo adaptados de las máquinas convencionales de extrusión a los procesos aditivos. Esta adaptación permite aumentar el tamaño de las boquillas y la altura de las capas, lo que facilita la producción de piezas de gran tamaño con alta precisión y resolución [3].

En los procesos LFGD, el equipo de impresión calienta solo el lecho de impresión y la boquilla, lo que provoca una transferencia de calor continua durante todo el proceso de fabricación. Esta transferencia de calor puede afectar a la calidad de la pieza final. El pro-

ceso deposita de manera continua material fundido caliente, estableciendo un gradiente térmico dentro de la pieza. El historial térmico de cada capa puede variar significativamente en función de la geometría de la pieza impresa y de la trayectoria definida de la boquilla. Esta variabilidad puede dar lugar a una contracción no homogénea y al desarrollo de tensiones residuales que, en última instancia, provocan la deformación de la pieza. Estos efectos, que contribuyen a la formación de gradientes térmicos significativos, son particularmente pronunciados cuando se trata de piezas impresas gruesas y tiempos de capa más largos [8]. Por este motivo, en este proyecto se propone una estrategia para modificar las trayectorias de los equipos de deposición de manera que se optimice la impresión de la pieza en impresoras de gran formato.

Las impresoras de gran formato industriales europeas más importantes son: la Delta WASP 3MT y las impresoras de la compañía sueca BLB Industries. La primera, de origen italiano, posee una superficie de impresión cilíndrica de más de 3 metros cúbicos, mientras que las suecas tienen una superficie de impresión de 5,6 metros cúbicos. A menor escala, empresas francesas, alemanas y estadounidenses han incorporado estos sistemas de extrusión en sus máquinas. Ejemplo de ello son Pollen AM Inc (París, Francia) y AIM3D (Rostock, Alemania). En Estados Unidos, varias empresas siguen un planteamiento similar. La impresora Big Area Additive Manufacturing (BAAM), desarrollada en 2015 por Cincinnati Incorporated en colaboración con Oak Ridge National Laboratories [24], utiliza un sistema de extrusión y alimentación que permite un flujo continuo de pellets y son un referente mundial en esta tecnología. Ofrece un volumen máximo de 25 metros cúbicos y una velocidad de deposición de material de 50 kg/h [4].

Del mismo modo, la Fabricación Aditiva de Gran Tamaño de Thermwood (Large Scale Additive Manufacturing, LSAM) combina un sistema de extrusión alimentado por pellets con una segunda herramienta, un cabezal de fresado de cinco ejes y una cámara térmica dentro de los mismos sistemas de movimiento cartesiano [1]. Las siglas LSAM son una marca comercial de la empresa Thermwood. En la Figura 2.7 se muestra una máquina LSAM de la empresa Thermwood (Cincinnati, USA).



Figura 2.7: Equipo Large Scale Additive Manufacturing (LSAM).

Otro equipo destacado es el desarrollado por Titan Robotics, una empresa estadounidense que ofrece una impresora personalizable de Fabricación de Filamento Fundido (Fuse Filament Fabrication, FFF) con un volumen de más de un metro cúbico, basada en un sistema autónomo multicabezal llamado Chronus. Este sistema se desarrolló en colaboración con Autodesk en un proyecto denominado Escher, en el que Autodesk programó su propio software de laminado (slicer). El proyecto más ambicioso actualmente en curso es el de la empresa estadounidense Ingersoll, en colaboración con Oak Ridge National Laboratories, lo han denominado Wide and High Additive Manufacturing, WHAM [12]. El objetivo de este proyecto es desarrollar y demostrar las tecnologías necesarias para la AM de gran anchura y altura, que abre nuevas áreas de fabricación en USA para utillajes de gran tamaño en industrias del transporte y la energía, reduciendo significativamente los costes y plazos de entrega.

Como se ha puesto de manifiesto, son numerosas las iniciativas para el desarrollo de impresoras de gran formato que son capaces de imprimir piezas finales e incluso moldes para composites. Las industrias más cercanas para beneficiarse de la LFAM son las de energías renovables, aeroespacial, construcción naval y defensa.

A diferencia del proceso FDM, la impresión FGF es adecuada para piezas de tamaño medio a grande, con paredes gruesas, geometrías de baja complejidad que no presentan voladizos (no requieren material de soporte). La productividad - hasta 10 kg/h - es un factor clave cuando se trata de fabricar piezas grandes, especialmente si se pueden procesar posteriormente, por ejemplo, mediante mecanizado CNC.

2.4 Antecedentes en procesos FDM

La mayoría de las aplicaciones de AM están basadas en la fabricación de piezas capa por capa en una sola dirección de construcción fija, lo que provoca restricciones en la resistencia de la pieza y la calidad de la superficie, así como la necesidad de usar estructuras de soporte. La metodología propuesta en este proyecto para imprimir piezas inclinadas supone una innovación en los procesos LFAM. Sin embargo, recientemente, se ha abordado esta problemática en FDM que, como ya se ha comentado, es uno de los procesos más comunes y extendidos en impresión 3D.

Uno de los problemas de la impresión FDM convencional es la imposibilidad de imprimir voladizos con más de 45 grados sin estructuras de soporte [34]. Las impresoras FDM más modernas se limitan a movimientos de tres ejes con una dirección de construcción fija en Z. En consecuencia, el software de corte divide la pieza (normalmente formato STL) en capas individuales con un grosor de capa constante, que está limitado por el diámetro de la boquilla. Para cada corte horizontal se calcula una trayectoria basada en el diámetro de la boquilla en 2D, centrándose en la geometría exterior. El resultado es un archivo G-code que será interpretado por la máquina. El principal inconveniente de este enfoque es la limitación a una sola dirección de construcción fija, lo que provoca una posible falta de resistencia, el efecto escalera y la necesidad de estructuras de soporte [35]. El uso de estructuras de apoyo adicionales conlleva un derroche de material, tiempo y energía.

En la Figura 2.8 se muestra la impresión de las capas de un voladizo de 45°, donde se observa cómo los filamentos extruidos no están totalmente apoyados, lo que puede provocar una reducción en la calidad de impresión de la pieza si se usa una boquilla de impresión tradicional a 90°; sin embargo, una de las soluciones más novedosas para evitar imprimir soportes es la propuesta en [35] que sugiere usar un cabezal de impresión con una inclinación de 45° para imprimir voladizos sin usar estructuras de soporte.

Existen numerosos programas comerciales que permiten generar código slicer como por ejemplo *Cura 4.11* (Ultimaker, Utrecht, The Netherlands) o *Simplify3D software* (Simplify3D, Cincinnati, OH, USA), sin embargo, son muy pocos los que permiten generar trayectorias en planos o direcciones diferentes a los planos paralelos a la mesa de trabajo. Prácticamente la totalidad de piezas son impresas en planos paralelos a la mesa de la impresora, de hecho el software de procesamiento se conoce como *laminador* o *slicer*. Las piezas se imprimen añadiendo capas paralelas con el mismo espesor, lo que restringe la calidad de las piezas impresas. No existe ningún condicionante tecnológico que impida la correcta deposición de capas en trayectorias totalmente 3D, sin embargo el proceso se puede complicar para ciertas geometrías y no es posible automatizar totalmente el proceso de generación del G-code.

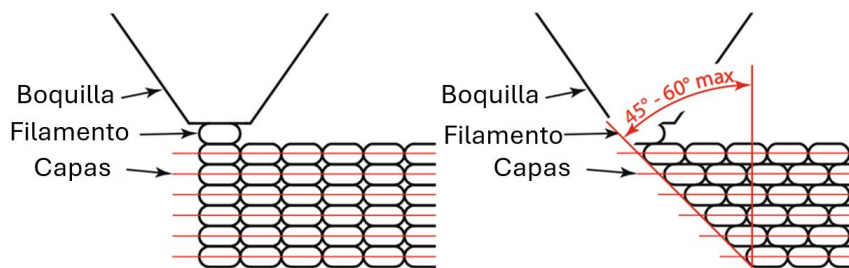


Figura 2.8: Ángulo límite impresión voladizos [35]

En los procesos MEX, las trayectorias de deposición se obtienen a partir del modelo 3D de la geometría de la pieza. En primer lugar, el software CAD exporta una versión STL simplificada del modelo 3D. En el segundo paso, se extraen los contornos de la pieza para cada capa, en lo que se denomina *slicing*, usando para ello un programa denominado *slicer*. Durante el corte (*slicer*), el modelo se trocea de forma escalonada mediante un conjunto de planos horizontales que intersectan la geometría. Cada intersección da un polígono que representa el contorno de la pieza para ese plano. Los diferentes cortes se apilan para representar la pieza.

El cálculo de las trayectorias a partir del modelo 3D incluye errores intrínsecos (fundamentalmente, de dos tipos) en diferentes fases del proceso. El primer error intrínseco se produce en el primer paso de conversión, cuyo objetivo es simplificar el modelo 3D aproximándolo mediante una malla triangular. El error se produce al aproximar regiones curvas mediante una malla de triángulos planos, por lo que el mallado no coincide completamente con la geometría del modelo 3D. El segundo error se produce durante la fase de corte (*slicing*). Normalmente, las impresoras 3D de tres ejes convencionales usan el rebanado horizontal plano como única alternativa para fabricar los modelos. El corte horizontal introduce una fuente de error en las regiones inclinadas de la pieza, que no pueden generarse perfectamente con capas planas. Debido a ello, la superficie de la pieza final tendrá forma de escalera, que se conoce comúnmente como *efecto escalera*. El efecto se ilustra en la Figura 2.9 para una pieza sencilla con una superficie superior ligeramente inclinada. La reducción del error del archivo STL implica aumentar el número de triángulos de la malla STL. Esto supone un compromiso entre precisión y eficacia, ya que la complejidad de la rutina de corte aumenta con el número de triángulos.

La deposición de material en los procesos MEX son esencialmente líneas de polímero. Estas líneas son más resistentes en la dirección de deposición porque las moléculas están literalmente unidas a lo largo de la extrusión. Sin embargo, cuando se apilan unas líneas sobre otras, como se hace con las capas impresas, la unión entre las capas es mucho menos fuerte. Esto significa que las impresiones 3D MEX son sistemáticamente más débiles en

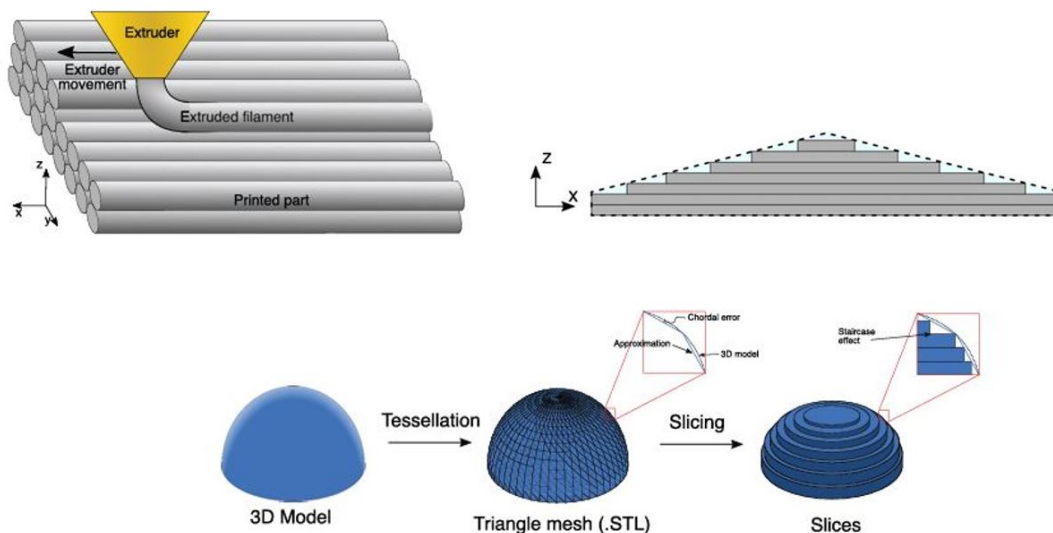


Figura 2.9: Principales causas de error en impresión 3D convencional [17]

las uniones. Para artículos decorativos, esto no importa, pero sí es importante para las piezas mecánicas. Algunos slicers de impresoras 3D hacen un esfuerzo considerable para orientar perfectamente sus piezas y garantizar que las líneas de las capas estén alineadas con las fuerzas esperadas. Sería muy positivo si se pudiera imprimir un modelo 3D de forma no plana para garantizar que la fuerza de extrusión está exactamente donde se necesita.

Sin embargo, el problema es tan complejo que es básicamente imposible crear un slicer no planario de uso general, si es que se le puede llamar así. La impresión 3D no plana es una línea en desarrollo para la comunidad científica con grandes retos mecánicos y computacionales. Por otro lado, hay muchas geometrías que podrían imprimirse de forma no plana porque su forma no generaría colisiones. El enfoque presentado en este trabajo es una solución intermedia entre la impresión no plana y la tradicional. En este caso, se propone hacer una impresión en un plano rotado 45 grados respecto de la mesa de trabajo.

En [34] se exponen las ventajas de la AM multiteje frente a la impresión convencional, donde la mayoría de los procesos se basan en imprimir capas horizontales y en una dirección de construcción vertical fija. Sin embargo, la impresión 3D multiteje requiere cambios tanto en la tecnología de la impresora 3D y, especialmente, en el cabezal de impresión, como en el diseño y fabricación en la geometría de la pieza a imprimir.

En [9] se presenta una revisión de alternativas al uso de soportes en procesos FDM, entre ellas, destaca la técnica ILP (*Inclined Layer Printing*) propuesta en [36] cuya idea es similar a la adoptada en este proyecto para imprimir piezas de gran tamaño. Esta técnica permite imprimir estructuras inclinadas, sin soporte, y puede integrarse fácilmente en el proceso actual de impresión 3D. El procedimiento se puede observar en la Figura 2.10 y consiste en rotar la pieza con una orientación optimizada, con un ángulo adecuado y adoptando después el corte por capas horizontales a la parte rotada. Este proceso se ejecuta mediante el motor de código abierto *Ultimaker Cura*. A continuación, toma el fichero G-code generado por el slicer y aplica la correspondiente matriz de transformación con el fin de restaurar la pieza a la orientación original. De esta manera, la trayectoria de la herramienta está inclinada y sincronizada con el ángulo de corte óptimo. No obstante,

en la referencia [36] se menciona la necesidad de elaborar una rutina que transforme el G-code, pero no se proporciona información sobre la misma.

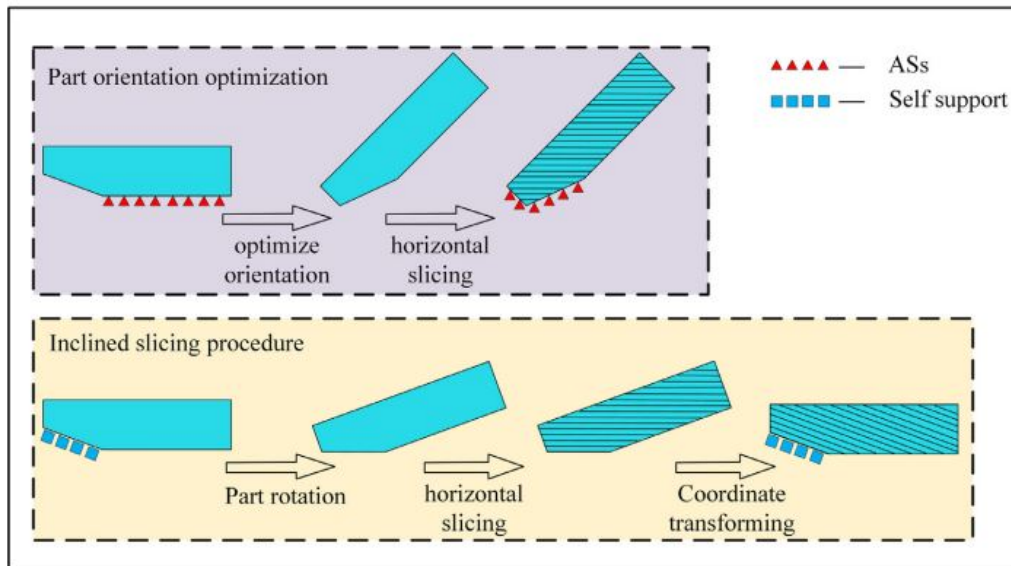


Figura 2.10: Técnica ILP propuesta en [36] para impresión FDM.

CAPÍTULO 3

Tecnologías usadas

Como se ha comentado en la introducción, la fabricación de una pieza en AM empieza con la generación del modelo 3D de la pieza. Para ello, tenemos dos opciones: diseñar la pieza mediante un software CAD o reconstruirla en formato digital mediante un escáner 3D. A continuación, se necesita laminar la pieza (slicing) con el fin de generar un fichero con las trayectorias que deberá seguir la máquina para construir cada una de las capas. Para elaborar este archivo necesitaremos un tipo de software CAM, denominado laminador (slicer), que genera ficheros escritos en lenguaje máquina (G-code) e interpretables por cualquier máquina de CNC. En este paso (postprocesado) se implementa una rutina para modificar el G-code de una pieza diseñada mediante las tecnologías previas (slicer) con una inclinación de 45 grados, resultando un G-code que contenga las instrucciones para imprimir la pieza original mediante una impresora con una boquilla girada 45 grados alrededor del eje horizontal. Esta rutina se ha generado con Python, tras una primera versión programada con Matlab. Finalmente, este fichero de tipo G-code, que contiene todas las instrucciones necesarias para imprimir capa a capa la pieza, será enviado a la impresora 3D, mediante algún puerto USB.

A continuación, se describe brevemente la elección de la tecnología empleada en base a sus características más relevantes.

3.1 Software CAD/CAM/CAE

El software CAD es una tecnología que utiliza sistemas informáticos para ayudar en la creación, modificación, análisis y optimización de un diseño. Se clasifica como software CAD cualquier programa que incluya gráficos por ordenador y contenga aplicaciones para facilitar las tareas de ingeniería en el diseño. El objetivo principal del software CAD es definir la geometría del diseño: una pieza mecánica, el ensamblaje de un producto, una estructura arquitectónica, un circuito electrónico, la disposición de un edificio, etc. Gracias al software CAD se puede ahorrar tiempo en el diseño de la pieza y también se reducen los errores a causa de la redefinición de la geometría desde el principio cada vez que se necesita.

La tecnología CAM incluye software y maquinaria para automatizar la planificación, gestión y control de los procesos de fabricación mediante una interfaz informática con los recursos de producción en diferentes industrias, entre las que se encuentra la impresión 3D. Una parte fundamental de un sistema CAM es el CNC. Se trata de un sistema que controla los movimientos y posición de una máquina a partir de una serie de comandos y operaciones programadas mediante software informático. Se usa para controlar diferentes tipos de máquinas, según el proceso empleado (corte, fresado, torneado, etc). La

programación de robots es otra de las funciones interesantes de los sistemas CAM. La planificación de procesos también es un objetivo de la automatización informática.

Finalmente, la Ingeniería Asistida por Ordenador o tecnología CAE (Computer Aided Engineering) es un proceso de ingeniería que incluye desde el diseño hasta la simulación del producto. Utiliza técnicas de análisis y simulación mediante software informático para analizar las funciones y propiedades de un producto creado mediante CAD. Esto permite a los diseñadores simular y validar las propiedades del producto así como la optimización de su diseño.

Existen herramientas CAE para distintos tipos de análisis. Por ejemplo, los programas de análisis cinemático pueden utilizarse para determinar las trayectorias de movimiento y las velocidades de conexión de los mecanismos. Los programas de análisis dinámico pueden utilizarse para determinar cargas y desplazamientos en conjuntos complejos, como los automóviles. Uno de los métodos de análisis más populares es el Método de los Elementos Finitos (MEF). Este enfoque puede utilizarse para determinar la tensión, la deformación, la transferencia de calor, la distribución del campo magnético, el flujo de fluidos y otros problemas de campo continuo que suelen ser demasiado difíciles de resolver con cualquier otro enfoque.

El software de diseño en 3D para la fabricación utilizado durante la realización de este proyecto ha sido *Siemens NX* [25]. NX software fue creado en General Motors y después se adaptó a otras áreas, cambiando su nombre a UGS y en la actualidad se le conoce como NX, que fue comprado por SIEMENS y se pasó a denominar SIEMENS NX PLM. SIEMENS NX es un software que proporciona soluciones completas en diseño, simulación y fabricación a las empresas. Ofrece un conjunto de herramientas para obtener y optimizar el diseño de productos de una forma más eficaz y rápida. Se trata de software CAD/CAM/CAE ampliamente usado en la ingeniería de productos y procesos, puesto que cubre todos los aspectos necesarios para la creación y fabricación de los mismos.

3.2 Slicers: Ultimaker Cura y Simplify3D

Un *slicer* para impresoras 3D es una parte fundamental del proceso de impresión. Un slicer es un software de corte 3D que transforma un modelo digital, normalmente un fichero con la extensión STL (aunque admite otras, tales como 3MF u OBJ), en un código G o G-code, que contiene las instrucciones necesarias para que la impresora imprima la pieza. Básicamente, el slicer divide la pieza en capas imprimibles y permite además ajustar muchos de los parámetros necesarios para optimizar la impresión. Existen muchos slicers de impresión, tanto gratuitos como comerciales. Durante la realización del proyecto se han utilizado dos: *Ultimaker Cura* y *Simplify3D*.

Desde hace casi una década, *Ultimaker Cura* [29] es el slicer de referencia para millones de personas, es fácil de usar y se trata de código abierto, por lo que se actualiza y mejora constantemente por la comunidad de usuarios. Además es compatible con la mayoría de impresoras 3D de escritorio, siendo el software de corte predeterminado por muchos fabricantes de impresoras 3D.

Cura también tiene un excelente mercado de complementos creados por la comunidad, así como integraciones en algunos conocidos programas CAD como Autodesk Inventor. En la Figura 3.1 se muestra un pantallazo del programa con la pieza demo de impresión por excelencia.

Simplify3D es también otro de los slicers más populares para impresoras 3D. Ofrece una solución de simulación 3D que ayuda a gestionar la impresión 3D, controlar las máquinas, hacer análisis de mallas y mucho más, en varios idiomas. Permite visualizar si-

mulaciones previas a la impresión mediante animaciones, creando secuencias, controlando la velocidad de impresión e identificando problemas de diseño, reduciendo el número total de impresiones fallidas, lo que supone un ahorro de tiempo y material. Asimismo, ofrece a los usuarios la posibilidad de optimizar los ajustes de cada capa en detalle.

La última versión añade mejoras que los hacen todavía más interesante (patrón de relleno 3D, reparación de mallado, etc), consiguiendo piezas de gran precisión, optimizando la calidad de la pieza final (bordes, soportes, etc). A diferencia del *Ultimaker Cura* el principal inconveniente de este software es que es de pago; no obstante, el grupo de investigación cuenta con licencia para su uso, precisamente por la alta calidad que se consigue. En la Figura 3.2 se muestra un pantallazo del programa con la figura Bunny preparada para imprimir.

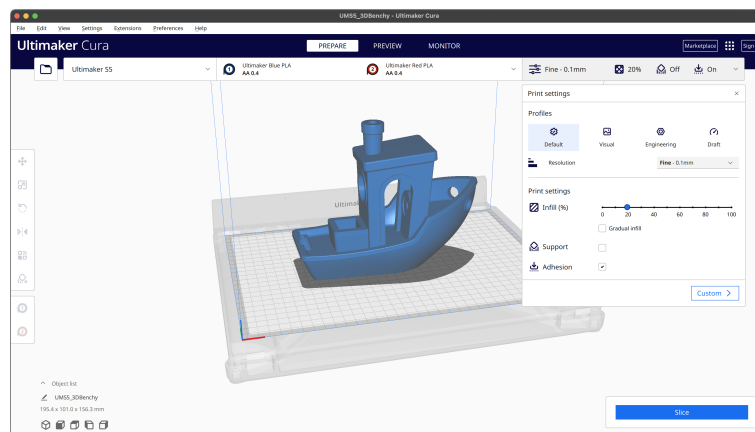


Figura 3.1: Software de impresión *Cura*.

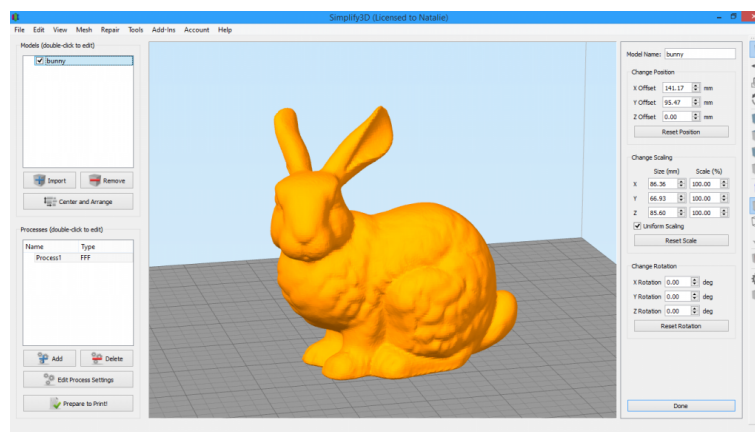


Figura 3.2: Software de impresión *Simplify3D*.

3.3 Software de programación: Matlab, Python

3.3.1. Matlab

MATLAB [14] es una aplicación de software y lenguaje de programación propiedad de MathWorks, basado en el cálculo matricial. Incluye una amplia biblioteca de herramientas (*toolboxes*) que permiten trabajar, de manera bastante rápida y eficiente, con diferentes algoritmos. Contiene gran número de funciones y comandos para realizar comple-

jos cálculos matemáticos, así como para generar nuevas funciones o programar scripts. Permite representar e interpretar gráficas de diferentes tipos, diseñar interfaces gráficas de usuario, crear apps web y de escritorio. Facilita tareas complejas de análisis de datos como la implementación de algoritmos, la interacción con otras aplicaciones y la manipulación de una matriz de datos.

Otras características destacables de MATLAB son:

1. Es un lenguaje de alto nivel que puede interactuar con otros lenguajes como Java y Perl.
2. Proporciona a los usuarios un entorno interactivo.
3. Proporciona un entorno para ejecutar tecnologías cognitivas exigentes y visión por ordenador.
4. Contiene numerosas herramientas para el procesamiento de imágenes y audio. Permite a los usuarios acceder a datos de fuentes como imágenes y archivos de audio. También admite datos de vídeo, binarios y telemétricos.
5. Cuenta con una caja de herramientas de aprendizaje automático y de aprendizaje profundo, que proporciona comandos sencillos de MATLAB utilizados para construir y enlazar las capas de una red neuronal profunda.
6. Dispone de una caja de herramientas de biología computacional que permite a los biólogos construir y analizar modelos biológicos.
7. Contiene numerosas herramientas para el procesamiento de señales.

Por todo ello, se trata de un software ampliamente usado para resolver problemas científicos y de ingeniería. Dado el carácter científico-técnico del problema que se propone resolver en este TFG hay que decir que una versión preliminar de la rutina descrita en este TFG se programó con Matlab.

3.3.2. Python

La elección del lenguaje de programación de Python [22] en este TFG obedece a diversos motivos. Actualmente, Python es uno de los lenguajes de programación más utilizado. Es gratuito, de código abierto, con un gran número de usuarios en todo el mundo que comparten recursos, tutoriales, documentación y librerías, lo que lo hace muy atractivo y contribuye a su crecimiento. Es muy versátil, relativamente fácil de usar, con una sintaxis similar a la de las personas, por lo que es sencillo encontrar errores, y se utiliza en todo tipo de aplicaciones.

Por otro lado, el propósito principal de este trabajo es definir una rutina para resolver un problema de carácter científico-técnico, con una fuerte base matemática, y Python es un lenguaje que, aún siendo de uso general, dispone de numerosas librerías que permiten usar un gran número de funciones en el ámbito científico y de ingeniería (al igual que Matlab), tanto para el desarrollo de algoritmos como para el diseño de la interfaz de usuario pero, además, dispone de otras bibliotecas con múltiples aplicaciones empresariales, siendo especialmente aplicado en el campo de la Ciencia de Datos y de la Inteligencia Artificial, áreas en las que es el lenguaje de programación más utilizado.

Aparte de todo ello, Python posee otras características especialmente interesantes [6]. Entre ellas, destacamos:

- Es un lenguaje interpretado, lo que permite ejecutar directamente el código sin necesidad de compilar el código fuente.
- Es un lenguaje orientado a objetos, organizado mediante clases y objetos, lo que facilita su reutilización.
- Usa tipado dinámico, de manera que no es necesario definir el tipo de las variables, que se asigna en función de su valor.
- Es multiplataforma, pudiendo ejecutarse en diferentes sistemas operativos (windows, linux, etc).
- Es compatible con diferentes interfaces gráficas de usuario.
- Es portable, lo que facilita su traslado.
- Es escalable, aumentando su complejidad según sea necesario.
- Es modulable. Admite módulos y paquetes, lo que fomenta la modularidad del programa y la reutilización del código.

Python dispone de un gran número de librerías que facilitan la tarea de programar. Entre ellas, citamos las usadas en este proyecto:

- **Numpy**: Biblioteca que contiene funciones para crear y almacenar vectores y matrices de grandes dimensiones e incluye numerosas funciones matemáticas para operar con ellas [19].
- **MatPolib**: Librería usada para generar los gráficos, a partir de los datos guardados en arrays (matrices).
- **Tkinter**: Librería de Python que facilita el desarrollo de la interfaz gráfica de usuario. Contiene widgets o funciones empleadas para el diseño de la ventana principal, cuadros de texto, creación de áreas de gráficos, personalización del color, etc.
- **PIL**: Biblioteca para procesar imágenes, con funciones que facilitan insertar, modificar, mostrar imágenes, etc.

3.4 Lenguaje de programación G-code

G-code es un código de procesamiento de propósito general ampliamente utilizado en la fabricación aditiva y sustractiva, donde el procesamiento de AM se define mediante un G-code estandarizado que es interpretado directamente por una impresora 3D.

Las impresoras 3D y, en general, las máquinas de CNC usan programas escritos en un lenguaje de programación denominado G-code, que contienen las instrucciones necesarias para indicar a la impresora los parámetros de la impresión (temperatura usada en el extrusor, número de capas de la pieza, etc) así como los pasos a seguir para construir la pieza diseñada previamente.

El G-code en este estudio ha sido generado por un software de corte desarrollado de forma independiente. En este proyecto se propone un método basado en el postprocesamiento del fichero de texto generado por un slicer convencional, utilizando expresiones regulares para interpretar y extraer la información de éste. El G-code almacena toda la información de puntos y capas durante el proceso de impresión, y los datos necesarios se pueden extraer mediante el reconocimiento de las características únicas. Para entender el

funcionamiento de la rutina propuesta, repasamos algunas de las instrucciones básicas que componen un G-code.

Una instrucción G-code básica se forma con una letra (código) seguida de un número (valor del parámetro). Así, un programa G-code usa una serie de comandos sencillos que comienzan con una letra **G** (Geométricos) para indicar movimientos de la máquina, ó **M** (Misceláneos) para otras acciones como, por ejemplo, apagar/encender, seguidos de números y combinados con otras funciones, representadas también por letras seguidas de números. Entre los comandos habituales, cabe destacar **Xx, Yy, Zz** para indicar que las coordenadas del punto se colocan en (x, y, z), **Ff** (siendo f la velocidad de movimiento del extrusor en mm/min), **Ee** (donde e representa la cantidad de filamento que pasa a través de la boquilla del extrusor, expresada en mm), **Ss** (en combinación con un comando **G**, s indica la temperatura que debe alcanzar el extrusor, expresada en grados centígrados; y a continuación de un comando **M**, s representa la velocidad del ventilador que enfría la pieza). La Figura 3.3 muestra el formato típico de una instrucción estándar:



Figura 3.3: Formato instrucción G-code.

Así, para entender la implementación de nuestra rutina, conviene tener en cuenta el significado de algunas de las instrucciones que, con más frecuencia, se usan en G-code. Destacamos las siguientes:

- **G0** y **G1**: Indican desplazamiento rápido y controlado, respectivamente.
- **G4**: Pausa.
- **G10** y **G11**: Retracción y desretracción, respectivamente.
- **G20** y **G21**: Para establecer unidades en pulgadas o mm, respectivamente.
- **G28**: Coloca la impresora en la posición inicial, es decir, los tres ejes en el origen.
- **G29**: Para autonivelar.
- **G90** y **G91**: Indican posicionamiento absoluto y relativo, respectivamente.
- **G92**: Establece la posición, indicando, a continuación, el extrusor.
- **M0**, **M1** y **M2**: Indican parada, pausa y fin de programa, respectivamente.
- **M104**: Para indicar, a continuación, la temperatura del extrusor.
- **M106**: Para ajustar, a continuación, la velocidad del ventilador.
- **M107**: Apaga el ventilador.
- **M140**, **M190**: Para ajustar, a continuación, la temperatura de la cámara.
- **M116**: Espera hasta que se estabilicen las temperaturas.
- **M112**: Indica parada de emergencia.

En este caso los ficheros G-code se han generado con el software *Ultimaker Cura*. Así, el G-code generado por el software *Cura* contiene, aparte del G-code necesario para imprimir una pieza concreta, el G-code inicial y final que, incluyen, respectivamente, las instrucciones previas y posteriores a la impresión. Se trata de parámetros que suelen usarse de manera más habitual para imprimir la mayoría de las piezas. Entre estos valores estarían las temperaturas de impresión o la colocación de los ejes en el origen, en el caso del G-code inicial; o el enfriamiento y apagado de la máquina, al terminar la impresión. Así, cabe destacar los siguientes comandos:

- **G28**: Coloca la impresora en la posición inicial, es decir, los tres ejes en el origen.
- **G92 E0**: Este comando prepara el extrusor para que inicie la impresión.
- **G1 Zz**: Coloca el eje Z en la posición z. Esta acción suele utilizarse para que el cabezal de impresión no raspe a través de la cama de impresión cuando se mueve a la posición inicial de impresión y al finalizar la misma.
- **G1 Ff E-e**: Retrae el extrusor. Este comando se usa para que la impresora retraiga o tire hacia atrás e mm de filamentos utilizando el extrusor a velocidad de f mm/min. Se suele utilizar al acabar la impresión.
- **G1 Xx Yy Ff Ee**: Coloca el cabezal de impresión en la posición (x,y) e indica a la impresora la velocidad de deposición f (mm/min), así como la cantidad de filamento que pasa a través de la boquilla del extrusor.
- **M106 Ss**: Indica la velocidad del ventilador que enfría la pieza. Suele usarse para enfriar el hotend o apagar el ventilador de enfriamiento después de acabar una impresión.
- **M104 S0**: Apaga el extrusor.
- **M84**: Apaga todos los motores de la impresora. Se recomienda usar estos dos últimos comandos al final de la secuencia de comandos de código G.

Si analizamos un trozo de código podemos observar, además que los comentarios van precedidos por;

```

;LAYER_COUNT:330
:LAYER:0
Velocidad ventilador → M107
G1 F1200 Z4.5
Velocidad sin impresión → G0 F18000 X474.333 Y302.047 Z4.5 ← Altura capa
;MESH:Cubo.stl
;TYPE:WALL-OUTER
G1 F1200 Z1.5
Velocidad impresión → G1 F3600 E13.05577
G1 F884.2 X475.099 Y302.126 E0.45598 ← Long. Extrusión
G1 F884.3 X650.239 Y302.125 E103.69276
G1 X825.379 Y302.124 E103.69276
Coordenadas desplazamiento ↑

```

Figura 3.4: Ejemplo trozo de código G-code.

Aunque es posible modificar estos scripts G-code cambiando algunos parámetros de ajuste en *Ultimaker Cura*, uno de los problemas que se han encontrado es que las trayectorias en los ficheros G-code generados con el software *Cura*, tradicionalmente, se basan en desplazamientos de los ejes en ángulo recto, no permitiendo cambiar directamente desde el slicer *Cura* el ángulo de inclinación del cabezal que, como ya se ha comentado, puede mejorar notablemente la impresión 3D de determinadas piezas. Dado que esta funcionalidad no está incluida en el slicer más popular usado en impresión 3D, nos hemos propuesto generar una rutina de postprocesamiento para modificar el G-code original generado por el slicer *Cura* para una pieza rotada, de manera que devuelva un G-code modificado que permita cambiar los parámetros necesarios para imprimir mediante desplazamientos horizontales, la pieza original con una inclinación *alpha* del cabezal de impresión.

3.5 Impresora 3D

El software diseñado está siendo validado con la impresora industrial 3D Super Discovery, fabricada por CNC Bárcenas (Valdepeñas, Ciudad Real, España), para su uso en el Instituto Diseño para la Fabricación de la Universitat Politècnica de València. La impresora tiene unas dimensiones de trabajo de 1300 mm x 2500 mm x 1000 mm. Se trata de una impresora industrial que funciona por extrusión directa de gránulos de plástico, que se van depositando capa a capa, hasta obtener la pieza diseñada, dispone de nivelado automático de cama y servomotores Panasonic. A diferencia de las impresoras convencionales FFF (*Fused Filament Fabrication*), basadas en la extrusión de filamentos, se usa para fabricar piezas de gran volumen, lo que supone una gran ventaja, permitiendo reducir el coste y mejorar la eficiencia en la fabricación de piezas grandes.

Las características principales de la máquina son:

- Suministros de granza ilimitado. Trabaja directamente con extrusión directa y está dotada de un alimentador automático que permite la impresión sin límite de material o peso de la pieza.
- Materiales: Permite la utilización de una amplia gama de materiales termoplásticos como ABS, ABS CF, PC CF, PPE CF, 3D850, ASA.
- Dimensiones de trabajo: 1.3x2.5x1 metros, pero estas medidas pueden ser modificadas a medida.
- Mesa: Se trata de una cámara cerrada con cama caliente que permite una temperatura de hasta 150 grado centígrados con autonivelado para compensar los desniveles durante la impresión.
- Extrusor: El cabezal de extrusión puede calentar los materiales de impresión hasta 400 grados.

En líneas generales se puede decir que la impresora consta de las siguientes áreas necesarias para su funcionamiento, como se aprecia en la Figura 3.5:

1. Estación de impresión.
2. Estación de trabajo.
3. Cuadro eléctrico y microcontroladores.
4. Compresor y calderín.

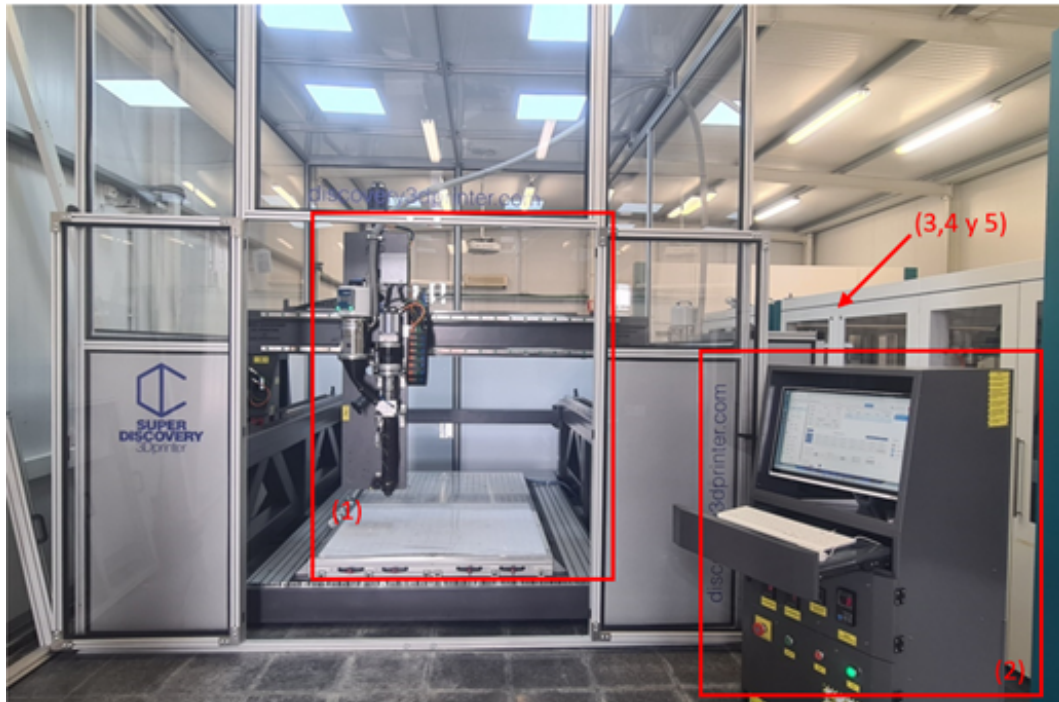


Figura 3.5: Super Discovery 3D printer.

5. Unidad de almacenamiento y acondicionamiento del material.

El área de impresión contiene la estructura principal de la impresora a la vez de la estructura de raíles y motores para realizar el movimiento del cabezal de impresión o fresado en los ejes XYZ, la orientación de dichos ejes se puede observar en la Figura 3.6.

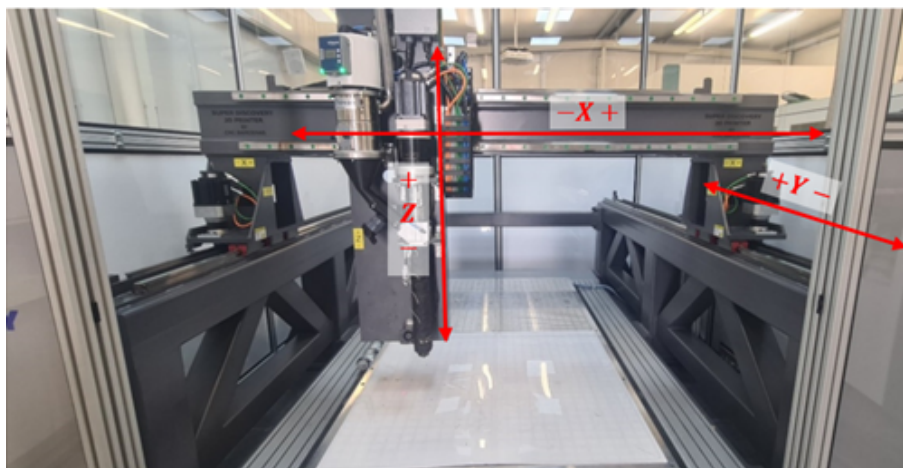


Figura 3.6: Movimientos del cabezal de impresión.

A su vez, el cabezal de impresión puede moverse hacia arriba y hacia abajo, en dirección Z, controlado a través de un husillo. Dicho cabezal está colgado sobre un raíl que se desplaza en dirección del eje Y, montado sobre una barra transversal. La barra transversal está montada en una estructura de soporte localizada a ambos lados de la mesa de impresión y permite el movimiento en la dirección del eje X. Todos estos movimientos son realizados por motores paso a paso eléctricos.

El cabezal de impresión incluye tanto la boquilla como el extrusor, como se muestra con más detalle en la Figura 3.7. El extrusor tiene cuatro bobinas calefactoras en su interior, que derriten el material entrante. Se puede aplicar una bobina calefactora adicional a la boquilla si es necesario. Actualmente utilizada en el uso de la boquilla de 45 grados.

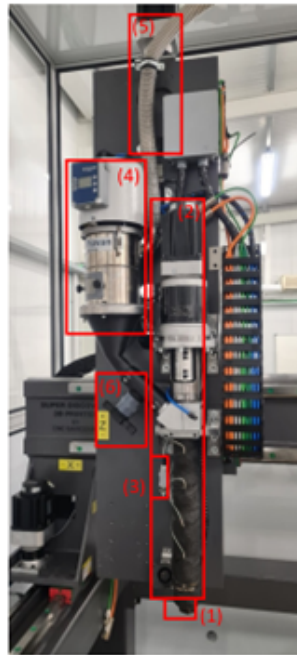


Figura 3.7: Cabezal de impresión.

Por otro lado, el cabezal de impresión incluye un contenedor de material en el que se almacena y precalienta el material antes de ingresar al extrusor. El material se sopla en este contenedor a través de un tubo de suministro conectado a la unidad de almacenamiento y acondicionamiento de materiales.

La cama de impresión (Figura 3.8) es una placa de aluminio de tamaño 1500x2500 mm y un perfil fresado en la superficie. Esta cama se puede calentar a diversas temperaturas, que son controladas desde el panel de control. El calentamiento de la cama permite que la adhesión de las primeras capas de la pieza sea mejor, reduce las tensiones y deformaciones que se dan durante el enfriamiento y facilita la retirada del objeto.

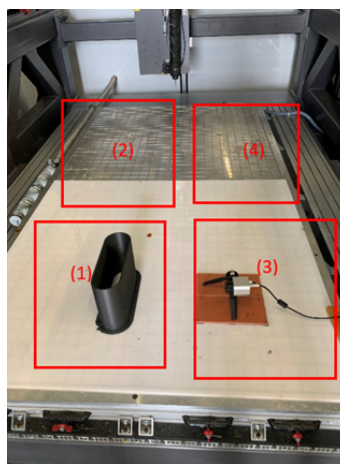


Figura 3.8: Vista de las diferentes zonas de la cama de impresión.

La impresión de piezas no se puede realizar directamente sobre la cama de aluminio, sino que es necesario colocar una placa de policarbonato sobre ella. Dicha placa se fija a la cama mediante un sistema de vacío que se aplica sobre las ranuras de la placa de aluminio. Existen cuatro zonas en la cama de impresión sobre las cuales se puede aplicar el vacío, dependiendo del tamaño de placa de policarbonato que se quiera utilizar o el tamaño de pieza a imprimir. La numeración de las zonas queda marcada en la Figura 3.8, siendo del 1 al 4, partiendo de la casilla ubicada delante-izquierda y acabando en la zona trasera-derecha.

El control del vacío se puede aplicar de dos maneras, mediante los denominados *Macros*, que permiten realizar ciertas acciones de manera sistemática con el uso del software del ordenador. Además del control por *Macros*, se disponen de llaves de paso que permiten abrir y cerrar los puertos de vacío de forma manual.

Por otro lado, como se puede observar en la Figura 3.9, la estación de trabajo consta de diversas partes:

- Ordenador.
- Paneles de control de temperatura.
- Botones de funciones básicas.



Figura 3.9: Estación de trabajo con vista detalle de los controladores y botones disponibles.

El ordenador de trabajo es simplemente un ordenador estándar en el que se han instalado todos los programas necesarios para las actividades de impresión y fresado. Entre ellos cabe destacar la versión actual de *Ultimaker Cura* (David Braam, Ultimaker, Utrecht, Países Bajos) y *PANATERM*. Por último, una de las zonas más importantes en el funcionamiento de la impresora es la caja de interruptores eléctricos y microcontroladores. Esta se encuentra situada en el lado derecho del área de trabajo, como se puede ver en la Figura 3.10.

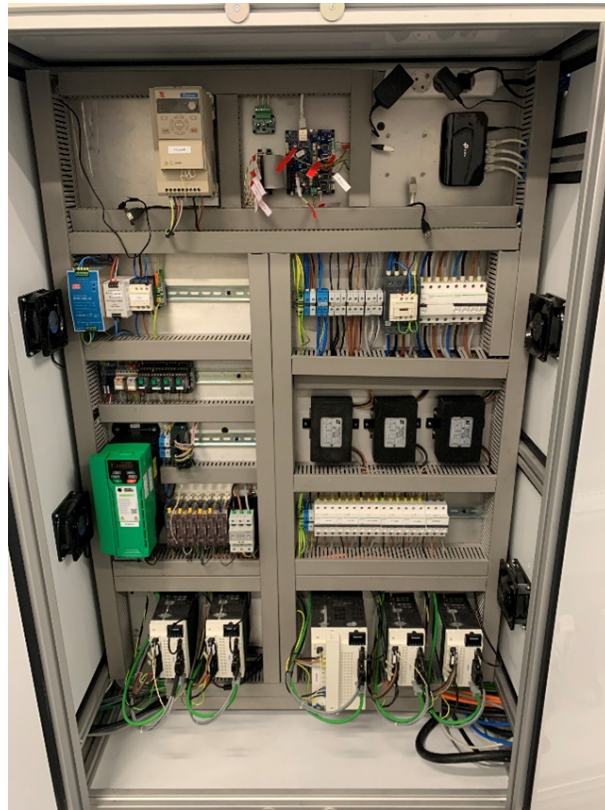


Figura 3.10: Cuadro eléctrico y microcontroladores.

El armario contiene diversas unidades de control internas, fusibles e interruptores. En la Figura 3.10, se pueden observar todos los componentes que tiene dicha caja. Entre ellos, destacan la Duet 2 Wifi y Rapberri Pi 4.

En la Duet podemos encontrar la tarjeta de memoria, que es común en impresoras no industriales de menor tamaño, y se utiliza para almacenar el G-code. Sin embargo, en esta situación todo el G-code se debe transferir a través del software de control, permitiendo su ejecución directa desde el mismo. No se requiere extraer la tarjeta para almacenar o ejecutar el G-code de manera independiente.

En el lateral superior derecho de la caja de interruptores eléctricos se encuentra un panel de control alojado en una caja gris, tal y como se aprecia en la Figura 3.11. Desde este punto es posible programar los tiempos de funcionamiento de la unidad de almacenamiento y acondicionamiento de materiales. Puede activarse y desactivarse manualmente, aunque también es posible programar períodos automáticos de inicio y detención según la hora del día, de forma independiente para cada día de la semana.

En el interior de la unidad de almacenamiento y acondicionamiento, el material, normalmente en forma de pellets, se acondiciona para la impresión variando su temperatura y contenido de humedad. Al iniciarse este proceso, el material experimenta un ciclo de calentamiento y deshumidificación con diversas fases, indicadas en el panel de control como *Regeneración o Calentamiento*. En el panel de control de la unidad de acondicionamiento, se pueden ajustar la temperatura y otros parámetros según el material utilizado. La temperatura predefinida para ABS es de 80°C. Para asegurarnos de un adecuado acondicionamiento del material se requiere un mínimo de cuatro horas. Por ello, es de vital importancia asegurarse de iniciar el acondicionamiento al menos el tiempo mínimo antes de imprimir. A su vez, es fundamental que la unidad de acondicionamiento permanezca encendida durante todo el proceso de impresión. Puede notarse una disminución de



Figura 3.11: Panel de control para la programación de la unidad de almacenamiento y acondicionamiento de material.

la temperatura durante la etapa de *Regeneración*, lo cual es normal y no indica un mal funcionamiento.

Así, a modo de ejemplo, una vez encendida la computadora, y tras pulsar el botón físico verde **Marcha** en la estación de trabajo, que enciende la impresora, tendremos que abrir el navegador e introducir la dirección IP y la contraseña requerida. A partir de ahí, podemos encender el calentamiento de la cama de impresión mediante la Macro **Mesa Calefactada-Calentamiento_Mesa_ON** en el lado derecho de la pantalla, como se observa en la Figura 3.12.

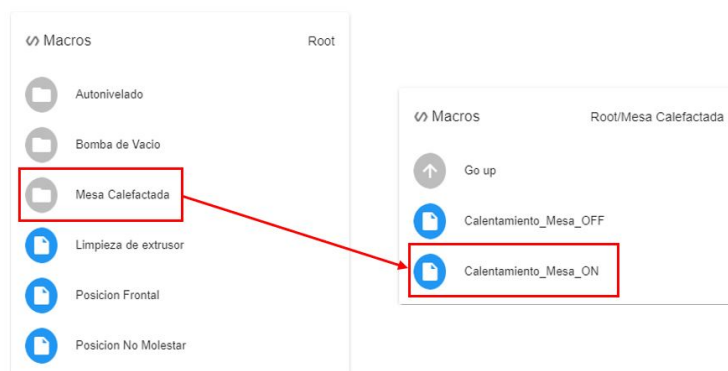


Figura 3.12: Macros para encender el calentamiento de la cama de impresión.

La temperatura de la cama de impresión aumenta aproximadamente 3°C por minuto. La temperatura se alcanza cuando los números mostrados en el panel de control son iguales, como se puede ver en la Figura 3.13. Para imprimir con ABS que contiene fibra de carbono, la temperatura de la mesa debería ajustarse a 90°C .

Tras esto, se debe presionar **Inicio X**, luego **Inicio Y** y después **Inicio Z** en el software de control. El panel de control estándar para manejar la impresora se puede ver en la Figura 3.13. La impresora establece sus ejes uno después del otro. El homing del eje Z

debe repetirse más adelante en el proceso, pero sin un homing inicial, la impresora no se puede mover a lo largo del eje Z para los siguientes pasos.

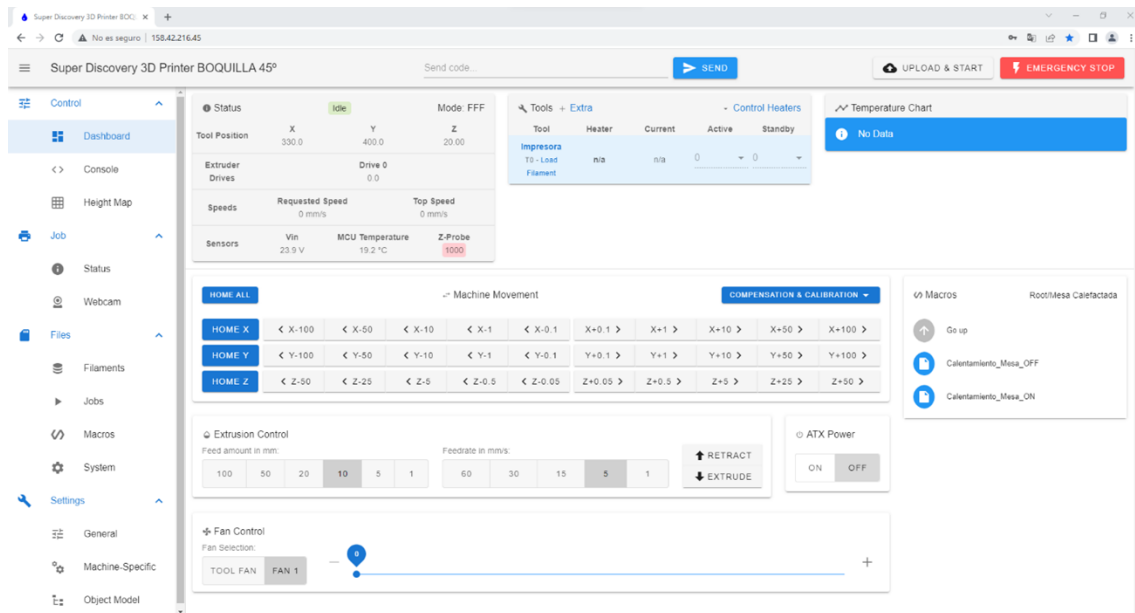


Figura 3.13: Software de control de la impresora.

Antes del primer trabajo de impresión del día, se debe crear un mapa de altura de la superficie de la cama de impresión. Esto debe hacerse después de calentar la cama de impresión y activar el vacío.

Pulsando **Home Z** en el software de control se puede ajustar el cabezal de impresión. Este proceso inicia la creación del mapa de altura a través del Macro **Autonivelado**, permitiendo elegir la zona o combinación de zonas que se utilizarán para la impresión, como se muestra en la Figura 3.14. En general, la información de altura se tiene en cuenta para el movimiento de la impresora y el G-code se modifica automáticamente. La expansión de la cama de impresión y, por lo tanto, la altura depende de varios parámetros, como la temperatura ambiente, y puede variar. Una correcta autonivelación permite mejorar la adherencia de la pieza y obtener una primera capa óptima sobre la que imprimir el resto de capas.

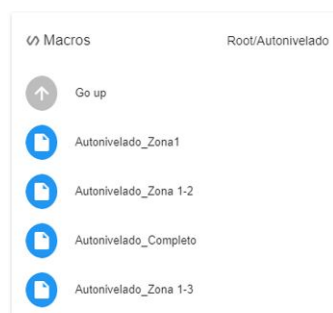


Figura 3.14: Macro para crear un mapa de altura.

En el proceso de extrusión directa de gránulos, el material plástico fundido se expulsa a través de una boquilla a una velocidad uniforme, permitiendo la construcción de la sección transversal del modelo CAD. Al finalizar cada capa, se ajusta la altura de la plataforma y/o se modifica la posición del cabezal de impresión para poder añadir la siguiente

capa. Este proceso se repite sucesivamente hasta completar la fabricación del modelo. De esta forma, se construye de manera progresiva la pieza mediante la superposición de capas. En ciertos casos, es necesario añadir estructuras de soporte adicionales para sostener voladizos u otras partes del objeto que no pueden ser impresas de manera independiente. Estas estructuras de soporte deben ser retiradas posteriormente como parte del proceso de acabado final, lo que conlleva riesgo de dañar la pieza. La estrategia presentada en este TFG evita la necesidad de construir estructuras de soporte, optimizando el proceso de impresión 3D para piezas de gran tamaño.

CAPÍTULO 4

Propuesta de rutina

En este capítulo se muestra el diseño y la implementación en Python de la rutina de postprocesado. La funcionalidad del código debe permitir la impresión de piezas con un plano de deposición de 45° . Para ello, el código deberá tomar como entrada el G-code original generado por el slicer (*Cura* o *Simplify3D*) para una pieza rotada y devolver, como salida, un G-code modificado, que será el que se enviará a la impresora para controlar la impresión de la pieza final. Además, el programa ha de permitir modificar los parámetros necesarios para imprimir, mediante desplazamientos horizontales, la pieza original con una inclinación α del cabezal de impresión. En nuestro caso $\alpha = 45^\circ$, dado que se dispone de una impresora 3D cuyo cabezal se puede inclinar un ángulo de 45 grados, a diferencia de la impresión tradicional, donde el cabezal se sitúa en ángulo recto. En la Figura 4.1 se muestra el esquema general del proceso a seguir.

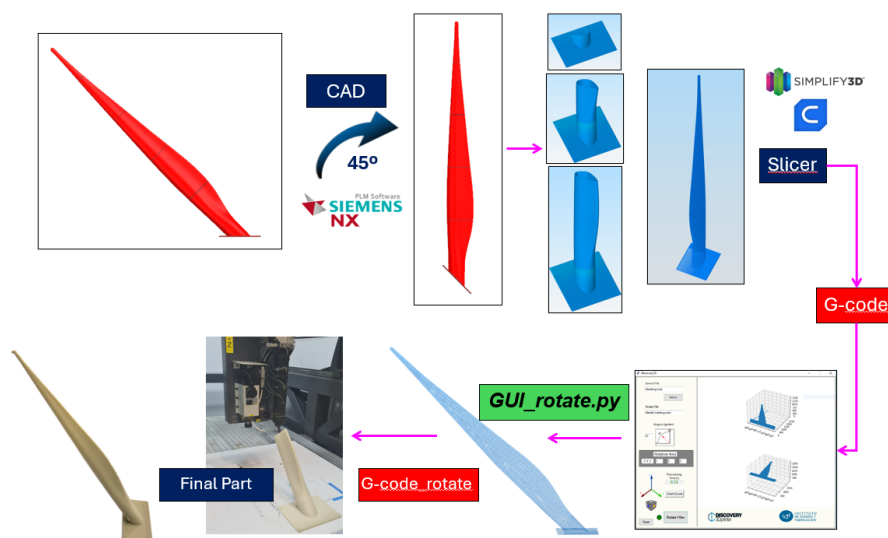


Figura 4.1: Esquema trabajo rutina Python.

Como se aprecia, el primer paso consiste en el diseño de la pieza a imprimir mediante software CAD. A continuación, se introduce el fichero generado (formato STL, en este caso) para la pieza rotada un ángulo de 45° en el slicer (*Cura* o *Simplify3D*), de manera que el G-code original generado por el slicer contiene las trayectorias de deposición a imprimir en el plano XY horizontal para la pieza rotada un ángulo de 45° (*G-code*), no para la pieza que se desea imprimir. La rutina de postprocesamiento implementada debe, por tanto, rotar las coordenadas de las trayectorias, de manera que el resultado final sea un

G-code (*G-code_rotate*) con las instrucciones que debe seguir la impresora para imprimir la pieza original con un plano de deposición de 45° .

En líneas generales, la propuesta de este trabajo se basa en los pasos descritos en la Figura 4.1 para la impresión de la pala de aerogenerador mostrada en la Figura 4.2, donde se pueden observar distintos momentos de la impresión real de la pieza.



Figura 4.2: Impresión de una pala de aerogenerador.

4.1 Requisitos

Con el fin de cumplir con los objetivos propuestos se definen los siguientes requisitos funcionales:

- El sistema deberá tomar como entrada el G-code original generado por el slicer (*Cura* o *Simplify3D*) para una pieza rotada y devolver, como salida, un G-code modificado que permita la impresión de piezas con un plano de deposición de 45 grados.
- El sistema deberá permitir la selección del fichero G-code original y grabar en el mismo directorio el nuevo G-code modificado, con el mismo nombre pero añadiendo la extensión *_rotate*.
- El sistema mostrará dos gráficas 3D, donde se representarán la pieza sin rotar y la pieza rotada, lo que permitirá observar errores en el funcionamiento del programa.
- El usuario podrá modificar el ángulo y el eje de rotación de la pieza.
- El sistema mostrará el tiempo de ejecución del proceso.
- El usuario podrá añadir nuevas instrucciones al G-code inicial y final, así como cambiar la posición de la pieza sobre la mesa, o cambiar la colocación final de la boquilla, pudiendo controlar el inicio y la finalización de la impresión de la pieza.

Por otra parte, el sistema debe satisfacer los siguientes requisitos no funcionales:

- Usabilidad: La interfaz de usuario debe ser intuitiva y de fácil manejo, de manera que el usuario no requiera conocimientos sobre el desarrollo interno del código. El usuario solo tendrá que ejecutar un fichero *.exe* e introducir los datos en el programa.

- Rendimiento: El sistema debe procesar la respuesta en un tiempo razonable (no más de 1 segundo cada 1000 líneas). Hay que tener en cuenta que el tiempo de ejecución dependerá del tamaño del fichero G-code que, en la mayoría de los casos, incluye cientos de miles de líneas de código. La interfaz debe responder de forma rápida y fluida, sin retrasos significativos, en la actualización de la información.
- Compatibilidad: El sistema se podrá ejecutar en un ordenador con sistema operativo Windows o Linux.

El programa está compuesto por dos scripts de Python: **GUI_rotate.py** y **rotate.py** que se ejecutarán, tras lanzar el usuario el ejecutable correspondiente **GUI_rotate.exe**. Los scripts de Python contienen las variables y funciones necesarias para implementar la transformación del G-code que permita la impresión 3D de la pieza original. Para su implementación se ha elegido el entorno de desarrollo integrado IDE *Visual Studio*. El programa se inicia ejecutando el fichero **GUI_rotate.py** que abre la ventana donde seleccionar e introducir los parámetros necesarios para la ejecución del código, mientras que la transformación del G-code se programa en el script **rotate.py**.

4.2 Diseño de la interfaz gráfica de usuario

El diseño de la interfaz gráfica de usuario se ha programado en el script **GUI_rotate.py**. El aspecto de la ventana principal será similar al de la Figura 4.3.

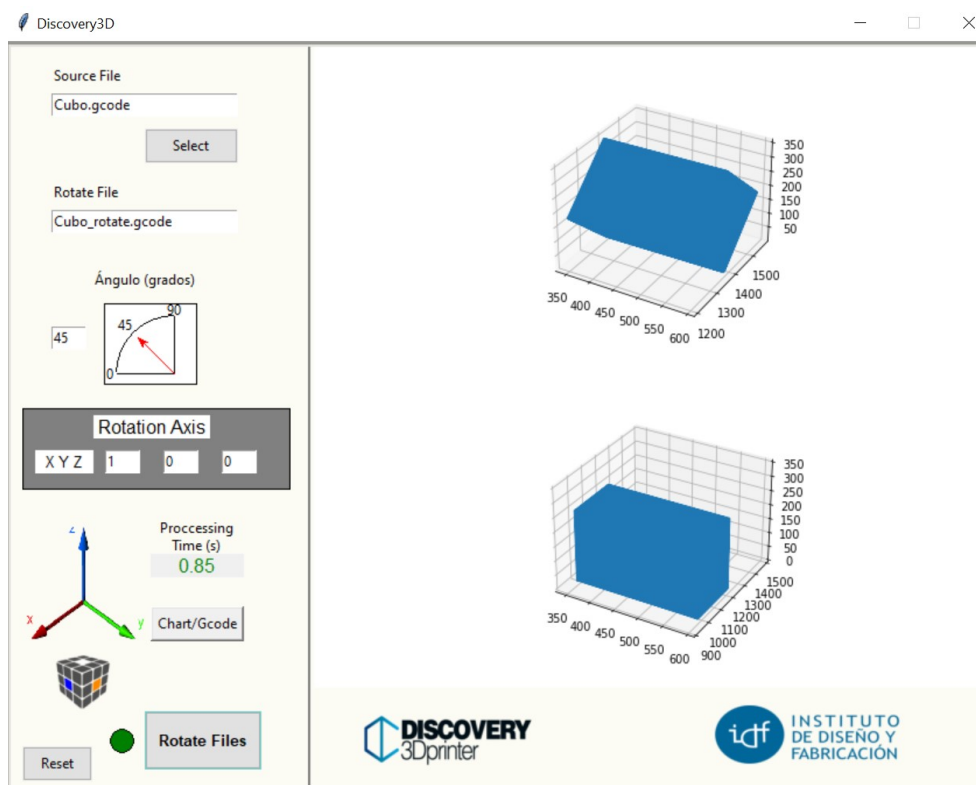


Figura 4.3: Diseño de la interfaz de usuario.

El fichero **GUI_rotate.py** es un script de Python que incluye los widgets Tkinter que formarán la interface gráfica para la visualización en 3D de las geometrías, tanto de la pieza rotada como de la pieza original. Además, genera una interfaz gráfica de usuario

que facilita el manejo de la aplicación y permite seleccionar parámetros básicos de la operación, tales como el ángulo o eje de giro, ficheros de entrada/salida, etc.

Como se puede apreciar, el menú incluye un item *Souce File* en el que se debe introducir el nombre del fichero original G-code, generado por el slicer, correspondiente a la pieza rotada (en este caso, *Cubo.gcode*) y, automáticamente, escribe el nombre del fichero modificado en el campo *Rotate File*, añadiendo la extensión *_rotate* al nombre del fichero original. Se trata del G-code final, denominado *Cubo_rotate.gcode* en el ejemplo, y que será el que incluirá las instrucciones necesarias para imprimir la pieza original. Asimismo, el usuario cuenta con la posibilidad de cambiar la inclinación de rotación de la pieza o también el eje de giro. En el ejemplo, se gira sobre el eje X y se rota un ángulo de 45°. Además, dado que el programa debe procesar, en la mayoría de los casos, ficheros con miles de líneas de código, se ha considerado interesante indicar el tiempo de ejecución del mismo *Processing Time (s)*.

Por otro lado, en la parte derecha de la ventana, se incluyen las dos gráficas de la pieza (original y modificada), lo que permite al usuario verificar si el programa está funcionando adecuadamente. La gráfica superior representa la pieza original rotada diseñada por el slicer (*Cubo.gcode*, en el ejemplo) y que el script debe procesar también para volver a dibujarla tras pulsar el botón *Rotate Files*, dado que existen puntos en el fichero G-code original, generado por el slicer, que no contienen la coordenada Z, por lo que el código deberá añadirla para, posteriormente, representarla en la figura superior derecha; mientras que la gráfica inferior contiene la pieza final a imprimir, *Cubo_rotate.gcode*, generada tras aplicar la rotación. Ambas figuras se obtienen tras pulsar la pestaña *Rotate Files*, que ejecuta la rutina **rotate.py**, donde se implementa la funcionalidad principal que se pretende con este código.

Por último, al pulsar la pestaña *Chart/Gcode* en la Figura 4.3, se abre una nueva ventana, similar a la Figura 4.4, donde es posible añadir instrucciones al inicio y al final del G-code, lo que permite agregar o eliminar comandos específicos del G-code generado por el slicer, modificando el funcionamiento del código, si se desea.



Figura 4.4: Ventana secundaria para modificar el inicio y el final del G-code.

También se pueden modificar, en esta ventana secundaria, las coordenadas de la posición de la pieza sobre la mesa (en la parte inferior izquierda), así como la colocación final de la boquilla (en la parte inferior derecha).

4.3 Implementación

4.3.1. Implementación de la interfaz gráfica de usuario: GUI_rotate.py

Se muestran, a continuación, algunas de las partes del código más importantes para crear la GUI. En primer lugar, se han de importar las librerías necesarias para su implementación, entre las que destacamos *Numpy*, *Matplotlib*, *IPL* y *Tkinter*, que ya se han comentado anteriormente.

La librería Tkinter contiene numerosos widgets que facilitan el diseño y la creación de la interfaz gráfica de usuario. Entre los usados en este trabajo, cabe mencionar *Tk()* para crear la ventana principal, *Frame()* para organizar la GUI, permitiendo agrupar widgets dentro de una ventana, *Text()* para crear un cuadro donde introducir textos que se usarán como parámetros del programa, *Label()* para mostrar texto en la GUI, *Button()* para crear los botones que se usan para lanzar acciones, *Canvas()* para la creación del área de las gráficas o *TopLevel()* para crear nuevas ventanas secundarias. Asimismo, entre las funciones de la librería PIL para abrir y procesar imágenes en diferentes formatos, destacamos las funciones *Image.open()* y *ImageTk.PhotoImage()*, para definir la función que permite redimensionar los gráficos. La última se usa para compatibilizar el formato de la imagen con las funciones de la biblioteca Tkinter.

Respecto al aspecto de la ventana, se han definido una serie de funciones que permiten diseñar la interfaz. No comentaré todas ellas, solo alguna a modo de ejemplo. El diseño se estructura en varias secciones, comenzando con la creación de la ventana principal tal y como se muestra en la Figura 4.5, donde se define la elección del tamaño y color de la ventana, así como la posición y el diseño del primer botón para seleccionar el fichero de entrada.

```

raiz = Tk()

raiz.title("Discovery3D") # Window title
raiz.resizable(0, 0) # Allow resize (0,0)-> not allowed

raiz.config(bg="#f0f0f0") # Window color

# First Section
miFrame = Frame(raiz)
miFrame.pack(fill="both", expand="True")
miFrame.config(bg="#f0f0f0")
miFrame.config(width="850", height="650") #Initial window size
miFrame.config(bd=5) # Window borders
miFrame.config(relief="groove") # Show the border with "groove" style

# Create label above the first text box with beige background
label_source = Label(miFrame, text="Source File", bg="#f0f0f0")
label_source.place(x=35, y=15, width=60, height=20)

# Create first text box (entry)
entry_texto = Entry(miFrame)
entry_texto.place(x=35, y=40, width=80 * 2, height=20)

# Create first button
btn_select1 = ttk.Button(miFrame, text="Select", command=lambda: seleccionar_archivo(entry_texto, entry_rotate, frame_plot_up))
btn_select1.place(x=115, y=70, width=80, height=20 + 10)

```

Figura 4.5: Creación de la ventana principal.

En la Figura 4.6 se muestra el diseño de la sección relacionada con el ángulo de rotación. De igual manera, se han programado el resto de secciones que conforman el diseño de la aplicación, aunque no se muestran en la memoria.

```

def dibujar_flecha(canvas, angulo):
    # Calculate the arrowhead coordinates
    x = 60 - 45 * math.cos(math.radians(angulo))
    y = 60 - 45 * math.sin(math.radians(angulo))

    # Draw arrow
    canvas.create_line(60, 60, x, y, arrow="last", fill="red", width=1, tag="flecha")

def dibujar_medio_circulo(canvas):
    # Drawing the quarter circle
    canvas.create_arc(10, 10, 110, 110, start=90, extent=90, outline="black", width=1)
    # Mark the grades
    for i in range(0, 91, 45):
        x = 60 - 45 * math.cos(math.radians(i))
        y = 60 - 45 * math.sin(math.radians(i))
        # Adjust the coordinates so that the numbers are on the outside of the circle
        if i == 0:
            canvas.create_text(x - 10, y, text=str(i), fill="black")
        elif i == 45:
            canvas.create_text(x - 10, y - 10, text=str(i), fill="black")
        elif i == 90:
            canvas.create_text(x, y - 10, text=str(i), fill="black")

```

Figura 4.6: Diseño de la sección para elegir el ángulo de rotación.

Por otra parte, el programa comienza con la ejecución del fichero **GUI_rotate.py**, que abre una ventana similar a la de la Figura 4.3 y en la cual el usuario deberá seleccionar el fichero G-code a procesar, así como el resto de parámetros, si lo desea. Por defecto, el ángulo está fijado a 45° y el eje de rotación es el eje X. Al pulsar el botón *Select*, el usuario seleccionará el fichero G-code que desea procesar (por defecto, muestra *Cubo.gcode*). Tras esto, se escribirá el nombre del fichero seleccionado y se actualizará el nombre del fichero modificado, añadiendo la extensión *_rotate* (*Cubo_rotate.gcode*, por defecto) para, una vez seleccionado el archivo, representar la gráfica de la pieza original en la parte superior derecha de la ventana. La implementación de esta funcionalidad se detalla en la Figura 4.7.

```

def seleccionar_archivo(entry, entry_rotate, frame_plot_left):
    global filename, filename2
    filename = filedialog.askopenfilename(initialdir="/", title="Seleccionar Archivo", filetypes=(("Archivos GCODE", "*.gcode"), ("Todos los archivos", "*.*")))
    if filename:
        # Gets the name of the selected file
        selected_filename = os.path.basename(filename)
        # Sets the file name in the input text box
        entry.delete(0, tk.END)
        entry.insert(0, selected_filename)
        # Updates the rotation input text box with the selected file name
        entry_rotate.delete(0, tk.END)
        entry_rotate.insert(0, selected_filename.replace('.gcode', '_rotate.gcode'))

    filename2 = entry_rotate.get()
    # Updates the graph according to the selected file
    update_plot_from_gcode(filename, frame_plot_left)

```

Figura 4.7: Selección de fichero G-code.

El trozo de código anterior incluye la función *update_plot_from_gcode* (*filename, frame_plot_left*), cuya definición se muestra en la Figura 4.8 y con la que se debe dibujar la pieza original generada por el slicer. Esta función también sirve para actualizar la gráfica del fichero de entrada tras ejecutar la rutina **rotate.py**, que permite escribir todos los puntos del G-code correspondientes a la pieza, con sus tres coordenadas. Una función de estructura similar (*update_plot_from_gcode2(filename2, frame_plot_right)*) permitirá representar la gráfica de la pieza del G-code modificado, cuyas coordenadas se calculan también tras la ejecución de **rotate.py**.

```

# Update initial graphic
def update_plot_from_gcode(filename, frame_plot_left):
    if filename:
        # Processing Gcode file
        with open(filename, 'r') as file:
            # Read x, y, z coordinates from Gcode file
            coordenadas = []
            for line in file:
                if (line.startswith('G1') or line.startswith('G0')) and 'X' in line and 'Y' in line and 'Z' in line:
                    coords = [0, 0, 0]
                    for coord in line.split():
                        if coord.startswith('X'):
                            coords[0] = float(coord[1:])
                        elif coord.startswith('Y'):
                            coords[1] = float(coord[1:])
                        elif coord.startswith('Z'):
                            coords[2] = float(coord[1:])
                    coordenadas.append(coords)

            if coordenadas:
                # Convert coordinates to numpy array for easy processing
                coordenadas = np.array(coordenadas)
                long = len(coordenadas)
                # Clean the frame to update the chart instead of creating a new one
                for widget in frame_plot_left.wininfo_children():
                    widget.destroy()

                # Create a new figure and axes in the frame
                fig = Figure(figsize=(5, 4), dpi=70)
                ax = fig.add_subplot(111, projection='3d')

                # Plot coordinates in 3D
                ax.plot(coordenadas[:,0], coordenadas[:,1], coordenadas[:,2])
                # ax.plot(coordenadas[1:,0], coordenadas[1:,1], coordenadas[1:,2])

                # Create a graphic canvas in the frame
                canvas = FigureCanvasTkAgg(fig, master=frame_plot_left)
                canvas.draw()
                canvas.get_tk_widget().pack(anchor=tk.W, side=tk.TOP, fill=tk.BOTH, expand=1)
            else:
                for widget in frame_plot_left.wininfo_children():
                    widget.destroy()

```

Figura 4.8: Actualiza la pieza descrita en el G-code_original.

```

def execute_and_change_light():
    iniciar_contador()

    # Calling the function to save text files
    guardar_BeginCode()
    guardar_EndCode()
    guardar_chart2()
    guardar_valores()

    global light_status

    # Change light status to red
    light_status = "red"

    # Execute the file rotate.py
    print("Running rotate.py..")
    result = subprocess.run(["python", "rotate.py"], stdout=subprocess.PIPE, stderr=subprocess.PIPE)

    # Verify process output status
    if result.returncode == 0: # If the process is successfully completed
        print("The execution of rotate.py was successful.")
        light_status = "green" # Change light status to green
    else:
        print("Execution of rotate.py failed.")

    update_light() # Updating the light color
    update_plot_from_gcode(filename, frame_plot_up) # Update first graph
    update_plot_from_gcode2(filename2, frame_plot_down) # Update second graph
    detener_contador()

```

Figura 4.9: Transforma el G-code_original y muestra las gráficas de las piezas original y rotada.

La ejecución del código **rotate.py** se realiza al pulsar el botón *Rotate Files* gracias a la acción definida en la función *execute_and_change_light()*, cuya descripción se puede ver en la Figura 4.9. Al finalizar esta función, se muestra el tiempo de ejecución del código, se actualizan las gráficas, mostrando las piezas original actualizada y la pieza a imprimir ya rotada. Asimismo, se muestra un círculo verde, en caso de que la ejecución haya sido exitosa, y rojo, si ha fallado la ejecución. Todo ello requiere de otras funciones, como las de actualización de gráficas, ya comentadas, y otras como *update_light()*, *iniciar_contador()* y *detener_contador*, necesarias para realizar las acciones descritas.

Como se observa en la Figura 4.9, la acción principal de la función es la ejecución del script **rotate.py**, que implementa la funcionalidad necesaria para llevar a cabo la transformación del *G-code original*.

4.3.2. Implementación del código principal: rotate.py

La Figura 4.10 muestra el flujo del código **rotate.py**. Básicamente, la rutina lee el *G-code* generado por el slicer correspondiente a la pieza rotada un ángulo de 45° , guarda las instrucciones y comandos anteriores y posteriores a las que se identifican con movimientos de los puntos de la pieza, interpreta las trayectorias y realiza una transformación geométrica para girarlas, genera un nuevo *G-code_rotado* con los puntos girados y añade las instrucciones guardadas de manera que la impresión final se corresponda con el de la pieza original.

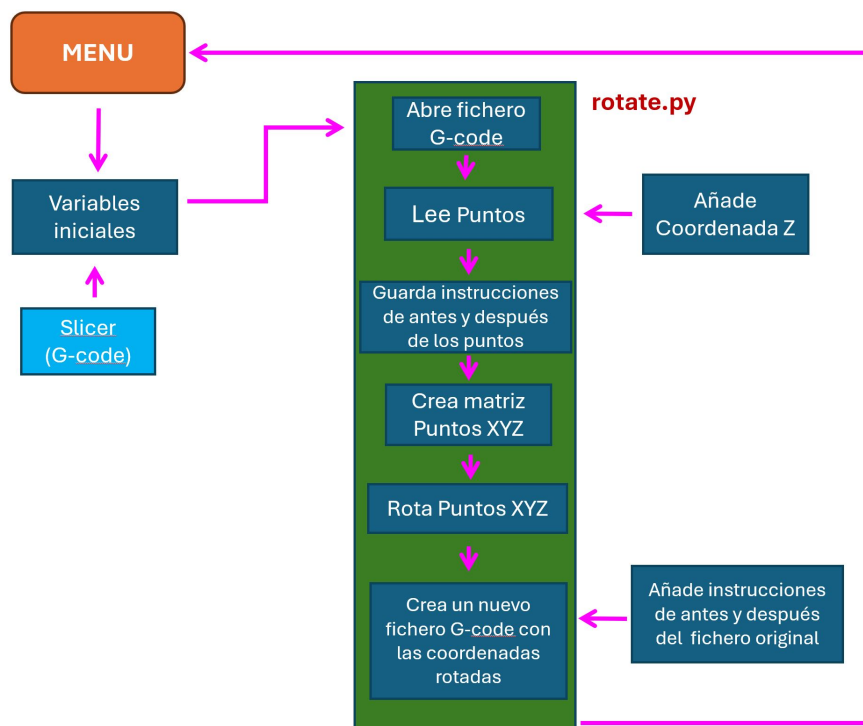


Figura 4.10: Flujo del código **rotate.py**.

Por tanto, una vez generado el *G-code original* correspondiente a la pieza girada mediante el slicer, nuestro script **rotate.py** realiza las siguientes acciones:

1. Abre y lee el *G-code_original* generado por el slicer para la pieza rotada, guardando parámetros importantes, tales como el grosor de la capa o parámetros del *G-code* inicial que no afectan al efecto de la rotación, instrucciones relativas al desplazamiento del extrusor **G92** o la posición de los ejes **G28**, como se ve en la Figura 4.11.

```
# Read the file G-code
file_gcode = open(file, 'r').readlines()
file_gcode = [line.strip() for line in file_gcode if line.strip() != ""]
file_gcode = [";Anulada: " + line if line.startswith("G28 ") else line for line in file_gcode]
```

Figura 4.11: Lectura del fichero G-code.

2. Se recorre el código, leyendo desde la primera capa hasta la última, almacenando los valores de las coordenadas X, Y, Z de cada una de las líneas correspondientes a instrucciones de movimiento, como se aprecia en la Figura 4.12.

```
# Lists for storing coordinates
Mx, My, Mz = [], [], []

# Previous coordinates
prev_x, prev_y, prev_z = 0, 0, 0

# Variable to indicate if we are in the section of interest
in_interesting_section = False
# Processes each line of the file G-Code
for line in file_gcode:
    # Check if we have reached the section of interest
    if line.startswith(";LAYER:0" or "; layer 0" or "; layer 1"):
        in_interesting_section = True
        continue # Skip to the next iteration to avoid processing the ;LAYER:0 line
    # If we are in the section of interest, check if the line contains movement information (G1 or G0)
    if in_interesting_section and (line.startswith('G1') or line.startswith('G0')):
        # Extracts coordinates (X, Y, Z) and the extrusion values (E)
        x = float(re.search(r'X(?:\d+(\.\d+)?)', line).group(1)) if 'X' in line else prev_x
        y = float(re.search(r'Y(?:\d+(\.\d+)?)', line).group(1)) if 'Y' in line else prev_y
        z = float(re.search(r'Z(?:\d+(\.\d+)?)', line).group(1)) if 'Z' in line else prev_z
        # Update previous coordinates
        prev_x, prev_y, prev_z = x, y, z

        # Stores the coordinates in the lists
        Mx.append(x)
        My.append(y)
        Mz.append(z)
```

Figura 4.12: Extrae las coordenadas de los puntos.

3. A continuación, se aplica la matriz de rotación un ángulo α respecto del eje *eje* elegido a la matriz de puntos P , almacenando dichos valores en la matriz rotada R (ver Figura 4.14). Para ello, se tienen en cuenta las matrices de rotación un ángulo α respecto de los tres ejes X, Y, Z, definidas como

$$R = \begin{bmatrix} \cos(\alpha) + n_x^2 v(\alpha) & n_x n_y v(\alpha) - n_z \sin(\alpha) & n_x n_y v(\alpha) + n_y \sin(\alpha) \\ n_y n_x v(\alpha) + n_z \sin(\alpha) & \cos(\alpha) + n_y^2 v(\alpha) & n_y n_z v(\alpha) - n_x \sin(\alpha) \\ n_z n_x v(\alpha) - n_y \sin(\alpha) & n_z n_y v(\alpha) + n_x \sin(\alpha) & \cos(\alpha) + n_z^2 v(\alpha) \end{bmatrix} \quad (4.1)$$

donde

$$v(\alpha) = 1 - \cos(\alpha) \quad (4.2)$$

siendo

$$n_x = \begin{cases} 1 & \text{si eje} = X \\ 0 & \text{si eje} = Y, Z \end{cases}, \quad n_y = \begin{cases} 1 & \text{si eje} = Y \\ 0 & \text{si eje} = X, Z \end{cases}, \quad n_z = \begin{cases} 1 & \text{si eje} = Z \\ 0 & \text{si eje} = X, Y \end{cases} \quad (4.3)$$

La Figura 4.13 muestra la implementación de la matriz de rotación, necesaria para invertir la figura.

```
# Define the transformation matrix
angulo = -Angulo * np.pi / 180 # Convert to radians
M1 = np.array([np.cos(angulo) + Axe[0] ** 2 * (1 - np.cos(angulo)),
               Axe[0] * Axe[1] * (1 - np.cos(angulo)) - Axe[2] * np.sin(angulo),
               Axe[0] * Axe[2] * (1 - np.cos(angulo)) + Axe[1] * np.sin(angulo)])

M2 = np.array([Axe[1] * Axe[0] * (1 - np.cos(angulo)) + Axe[2] * np.sin(angulo),
               np.cos(angulo) + Axe[1] ** 2 * (1 - np.cos(angulo)),
               Axe[1] * Axe[2] * (1 - np.cos(angulo)) - Axe[0] * np.sin(angulo)])

M3 = np.array([Axe[2] * Axe[0] * (1 - np.cos(angulo)) - Axe[1] * np.sin(angulo),
               Axe[2] * Axe[1] * (1 - np.cos(angulo)) + Axe[0] * np.sin(angulo),
               np.cos(angulo) + Axe[2] ** 2 * (1 - np.cos(angulo))])
```

Figura 4.13: Matriz de transformación de coordenadas.

4. Puesto que es frecuente que en el G-code original no esté la coordenada Z, se debe buscar dicha coordenada y añadirla, tal y como se describe en la Figura 4.12. De esta forma, se genera y almacena una matriz de puntos P que contiene todas las posiciones de las trayectorias del *G-code original* (*origined_coordinates*), así como la matriz de los puntos rotados (*transformed_coordinates*). Además, en caso de que la nueva matriz de puntos exceda de los límites de la impresora, será necesario efectuar un desplazamiento de los puntos de la nueva matriz, como se aprecia en la Figura 4.14.

```
# To add Z in the original file
origined_coordinates = []
Id = np.identity(3)
for i in range(len(Mz)):
    new_origined_coordinates = Id[0] * Mx[i] + Id[1] * My[i] + Id[2] * Mz[i]
    origined_coordinates.append(new_origined_coordinates)

# Apply transformation matrix to coordinates
transformed_coordinates = []
for i in range(len(Mz)):
    new_coordinates = M1 * Mx[i] + M2 * My[i] + M3 * Mz[i]
    transformed_coordinates.append(new_coordinates)

# Get the minimum value of Z
min_z = 999999
for coord in transformed_coordinates:
    if coord[2] < min_z:
        min_z = coord[2]

# Adjust all Z coordinates to be positive
adjusted_coordinates = []
for coord in transformed_coordinates:
    adjusted_z = coord[2] - min_z + Altura_capa
    adjusted_coordinates.append([coord[0], coord[1], adjusted_z])
```

Figura 4.14: Cálculo de nuevas coordenadas.

5. Tras leer todo el fichero y una vez rotados los puntos que corresponden a la pieza, se debe generar el G-code rotado. Para ello, se deben invocar dos rutinas que

almacenan todas las instrucciones anteriores a las coordenadas de los puntos que indican movimientos, así como las posteriores. Las instrucciones anteriores y posteriores a las coordenadas de los puntos rotados se obtienen mediante los métodos `buscar_coord_antes(line)` y `buscar_coord(line)`, como se describe en la Figura 4.15.

```
# Function for adding the above
# Search if x exists
def buscar_coord_antes(line):
    if "X" in line:
        match = re.search(r'(.*)\sX\d+(\.\d+)?', line)
        result = match.group(1)
        return result
    # If x exists we write the above
    # If x does not exist search if y exists
    elif "Y" in line:
        match = re.search(r'(.*)\sY\d+(\.\d+)?', line)
        result = match.group(1) if match else ''
        return result
    # If y exists we write the above
    # If neither x exists nor y exists search if z exists
    elif "Z" in line:
        match = re.search(r'(.*)\sZ\d+(\.\d+)?', line)
        result = match.group(1) if match else ''
        return result
    else:
        result = ""
        return result
    # If z exists we write the above

# Search if z exists
def buscar_coord(line):
    if "Z" in line and not line.startswith(";"):
        match = re.search(r'Z\d+(\.\d+)?\s*(.*)', line)
        result = match.group(2) if match else ''
        return result
    # If there is z we write the back side
    # if z does not exist search if y exists
    elif "Y" in line:
        match = re.search(r'Y\d+(\.\d+)?\s*(.*)', line)
        result = match.group(2) if match else ''
        return result
    # If there is y we write the back side
    # If neither z exists nor y exists search if x exists
    elif "X" in line:
        match = re.search(r'X\d+(\.\d+)?\s*(.*)', line)
        result = match.group(2) if match else ''
        return result
    else:
        result = ""
        return result
    # If there is x we write the back side
```

Figura 4.15: Busca instrucciones anteriores y posteriores.

El fichero `G-code_rotado` se generará y grabará como se muestra la Figura 4.16. Una rutina similar genera una actualización del G-code original para añadir la coordenada Z en aquellos puntos en los que no esté detallada en la línea correspondiente.

```
j = 0
# Open the new file in write mode
with open(fileTurn, 'w') as file_turn:
    # Variable to indicate if we are in the section of interest
    in_interesting_section = False
    # Process each line of the original G-Code file
    for line in file_gcode:
        # Check if we have reached the section of interest
        if line.startswith(";LAYER:0" or "; layer 0" or "; layer 1"):
            in_interesting_section = True
            file_turn.write(line + "\n") # Write the line ;LAYER:0 in the new file
            continue # Skip to the next iteration to avoid processing the ;LAYER:0 line
        # If we are in the section of interest, check if the line contains movement information (G1 or G0)
        if in_interesting_section and (line.startswith("G1") or line.startswith("G0")):
            # Before and after search:
            result_pre = buscar_coord_antes(line)
            if result_pre == "" :
                new_line = line
            else:
                new_line_2 = f" X{adjusted_coordinates[j][0]:.2f} Y{adjusted_coordinates[j][1]:.2f} Z{adjusted_coordinates[j][2]:.2f} "
                result = buscar_coord(line)
                new_line = result_pre + new_line_2 + result
            # Write the new line in the new file with a line break
            file_turn.write(new_line + '\n')
            j += 1
        else:
            # If we are not in the section of interest, simply write the line as it is with a line break
            file_turn.write(line + '\n')
```

Figura 4.16: Escribe todas las instrucciones en el fichero G-code rotado.

CAPÍTULO 5

Ejemplos Prácticos

A continuación se van a comentar brevemente las piezas que se han fabricado utilizando el programa desarrollado durante el presente proyecto. Estas piezas han sido propuestas por diferentes empresas, equipos de Generación Espontánea u otros grupos de investigación de la UPV. Se han utilizado dos materiales para la fabricación de las piezas: ABS con carga de fibra de carbono al 20% (ABS/CF) y ABS con carga de fibra de vidrio al 20% (ABS/GF). Las propiedades mecánicas de estos materiales son diferentes y, por supuesto, su coste. Dado que la elección del material o las condiciones del proceso no son objeto de este TFG no se va a entrar en detalle sobre su justificación. A nivel de programación y uso de la aplicación, el tipo de material no afecta.

Durante el desarrollo de este TFG se ha colaborado con los compañeros del equipo Formula Student UPV y se han fabricado diferentes moldes, usando la estrategia propuesta en el TFG. En la parte de la derecha de la Figura 5.1 se pueden ver fabricadas tres de las piezas generadas para el grupo de Generación Espontánea de Formula Student de la UPV usando el G-code generado por el código desarrollado en este trabajo. Se trata de la tapa cover bottle (parte inferior derecha), la tapa dampers (parte intermedia derecha) y el atenuador de impactos (parte superior derecha). A continuación, se muestra cómo funciona la rutina para estos casos, así como para una pala de aerogenerador.

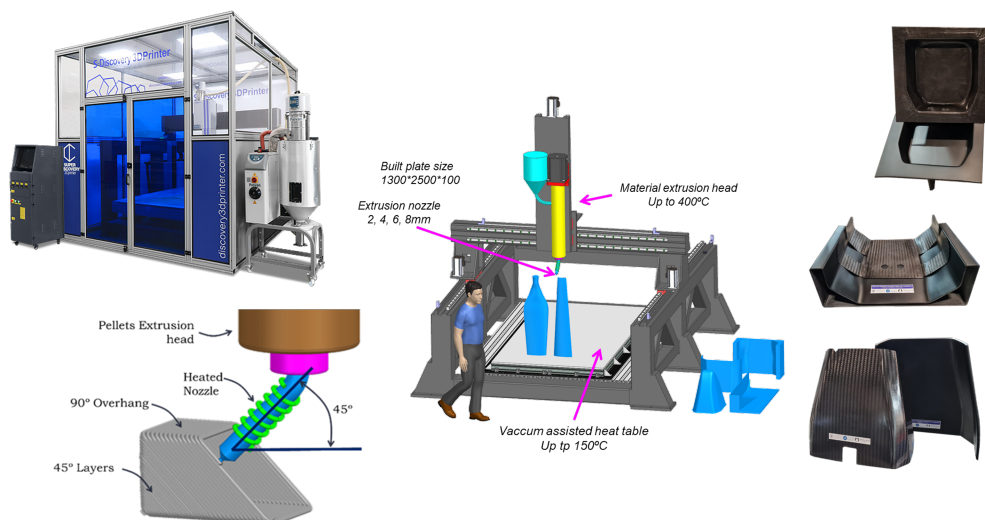


Figura 5.1: Ejemplo componentes FSUPV G-code.

5.1 Pala de aerogenerador

Se trata de una maqueta de una pala de aerogenerador como demostrador del Gel Coat que protege este tipo de componentes. La pieza fue encargada por una empresa de la Comunitat Valenciana para su presentación en la JEC World Conference, celebrada en París en el mes de abril de 2024. La pala tiene un tamaño de 2 metros y fue construida en una única pieza en ABS/GF20. En las Figuras 5.2 y 5.3 se muestran diferentes detalles de la fabricación de la pala.

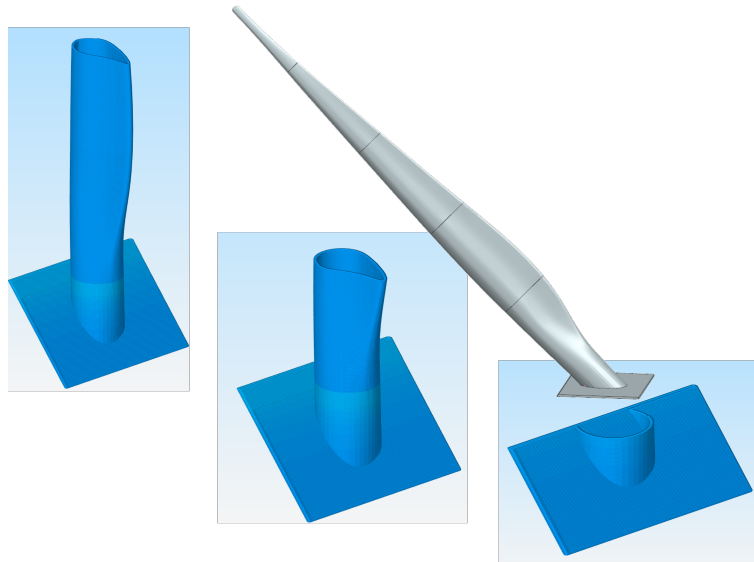


Figura 5.2: Ejemplo diseño álabes aerogenerador.



Figura 5.3: Ejemplo álabes aerogenerador construido.

La Figura 5.4 muestra la conversión del G-code mediante el código desarrollado en este TFG y que permite imprimir la pieza, de grandes dimensiones, inclinada un ángulo de 45 grados, sin necesidad de usar soportes.

El diseño de la pieza fue entregado por la empresa en cinco partes que debían ser ensambladas. El diseño se modificó manteniendo su forma exterior y se hizo la impresión en Vase Mode. Este tipo de impresión permite hacer la totalidad de la pieza sin reposicionamientos del extrusor. La pieza no se podía imprimir verticalmente ya que supera el máximo tamaño de la máquina. Para asegurar que la pieza tuviera suficiente sujeción en la mesa, se modificó el diseño aumentando la superficie de la base.

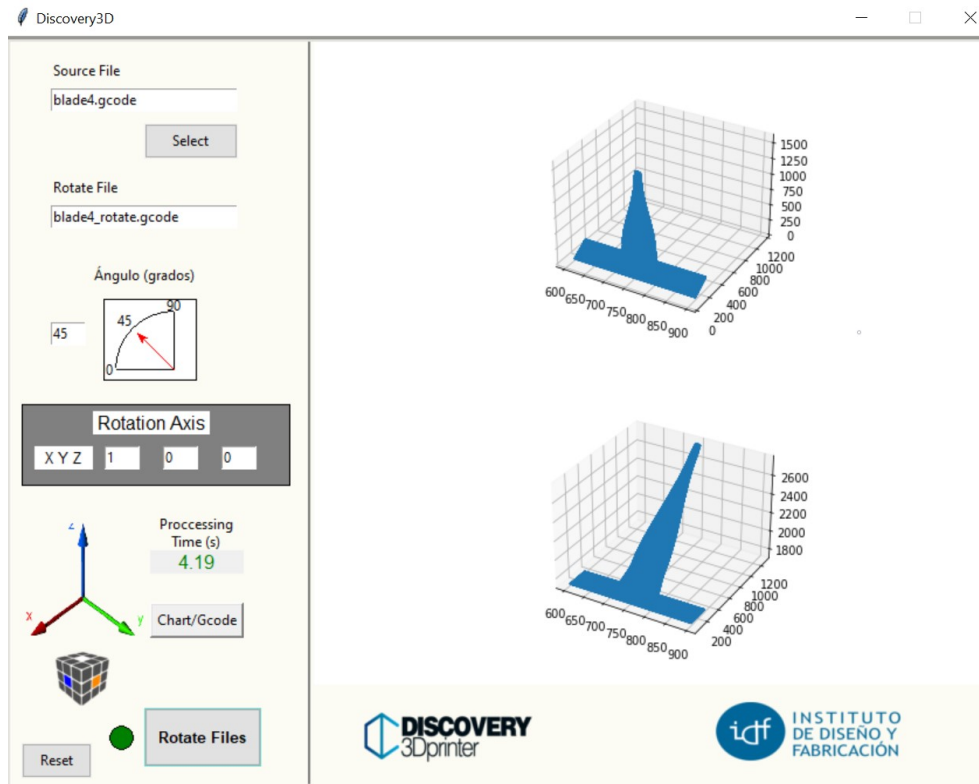


Figura 5.4: Interfaz gráfica para la generación del nuevo G-code de la pala de aerogenerador.

5.2 Tapas FSUPV

En el entorno de la Formula Student, el uso de amortiguadores (*dampers*) es fundamental para el control de la dinámica del vehículo. Los amortiguadores gestionan el movimiento de la suspensión, mejorando la estabilidad, el manejo y la tracción al mantener las ruedas en contacto con el asfalto de manera más efectiva. Estos dampers se cubren con una tapa para mejorar la aerodinámica del coche y su estética. El uso de la fabricación aditiva para fabricar el molde de la tapa de los dampers explota las ventajas de las tecnologías de fabricación modernas, como la personalización, la rapidez en la fabricación o el reciclado de componentes.

Otra pieza importante es la tapa *cover bottle*, cuyo diseño se puede ver en la Figura 5.5. Este molde se ha fabricado a 45 grados para favorecer el alineamiento de las capas con la geometría de la pieza. El tiempo de fabricación de esta pieza fue de dos horas en la máquina de impresión y otras dos horas en operaciones de acabado (lijado y pulido). El proceso final de la fabricación de la pieza, además incluye la colocación de desmoldeante permanente (lámina de teflón), colocación de la fibra de carbono, consumibles (peelply, manta, bolsa de vacío y tackytape). Una vez preparada la pieza en el molde con todos los consumibles, se pone al vacío y se deja unas 5 horas en horno a 80 grados para la polimerización completa del preimpregnado de carbono. La pieza terminada se muestra en la imagen inferior derecha de la Figura 5.1.

La Figura 5.6 muestra la conversión del nuevo G-code mediante el código desarrollado en este TFG, que permite rotar un ángulo de 45 grados la tapa *cover bottle* (pieza representada en la parte inferior derecha de la Figura 5.1). La Figura 5.7 muestra la conversión del G-code para rotar un ángulo de 45 grados la tapa *damper* (pieza representada en la parte intermedia derecha de la Figura 5.1).

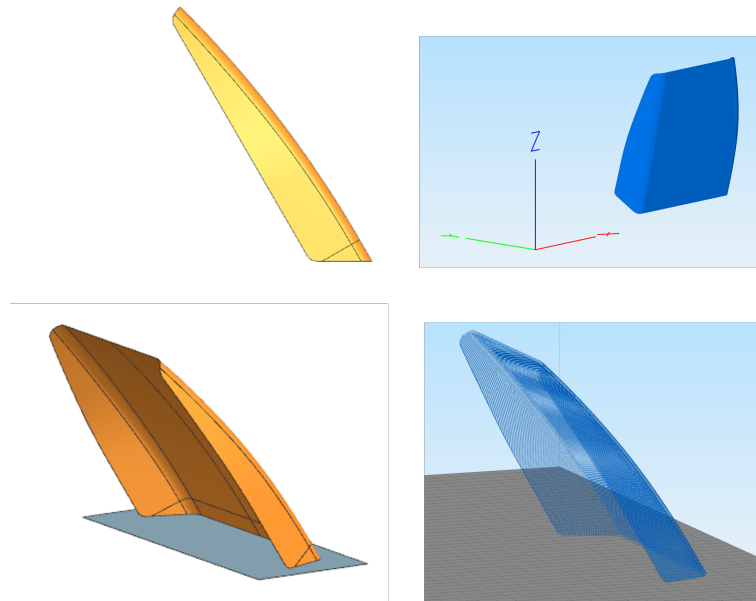


Figura 5.5: Ejemplo tapa cover bottle FSUPV.

5.3 Atenuador impactos FSUPV

Un vehículo Formula Student necesita por normativa un atenuador de impactos en el frontal de coche. Para validar el diseño de este atenuador, es necesario hacer un test en un laboratorio homologado. Esta prueba se realiza para verificar la capacidad del atenuador de absorber y disipar la energía de impacto de manera efectiva, protegiendo tanto al conductor como a la estructura del vehículo. El procedimiento del Test de Impacto es el siguiente:

- El atenuador de impactos se monta en un banco de prueba que simula la configuración del vehículo.
- El vehículo o estructura de prueba es equipado con instrumentación adecuada para medir las fuerzas de impacto y las deformaciones en el atenuador.
- El vehículo (o estructura de prueba) se impulsa hacia un obstáculo fijo a una velocidad especificada, que generalmente ronda los 7 metros por segundo (unos 25 km/h).
- Este impacto simula una colisión frontal, donde el atenuador de impactos debe absorber la energía cinética.

La fuerza de impacto se mide en tiempo real mediante sensores. Se evalúa la deformación o compresión del atenuador para determinar si se comporta dentro de los parámetros esperados.

La Figura 5.8 muestra la conversión del G-code mediante el código desarrollado en este TFG, rotando un ángulo de 45 grados la pieza anterior.

En la Figura 5.9 se muestran diferentes etapas de la fabricación de la bancada del atenuador de impactos para su ensayo en el laboratorio. Como puede verse, es una pieza compleja y gracias a la impresión a 45 grados ha sido posible imprimirla en una sola pieza.

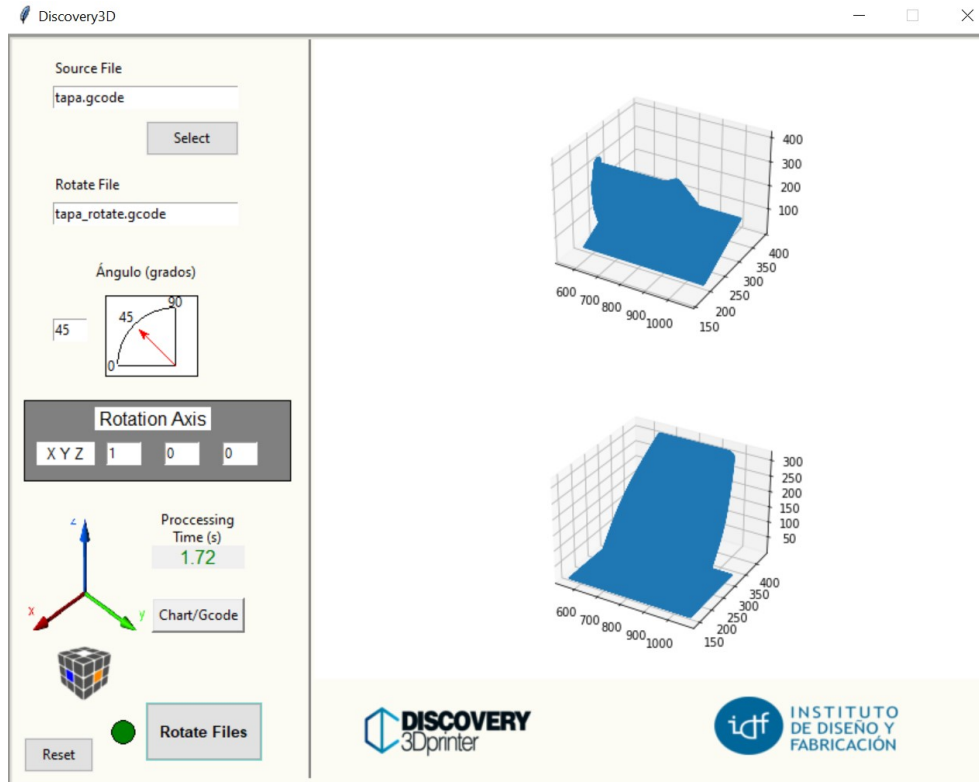


Figura 5.6: Interfaz gráfica para la generación del nuevo G-code de la tapa cover bottle FSUPV.

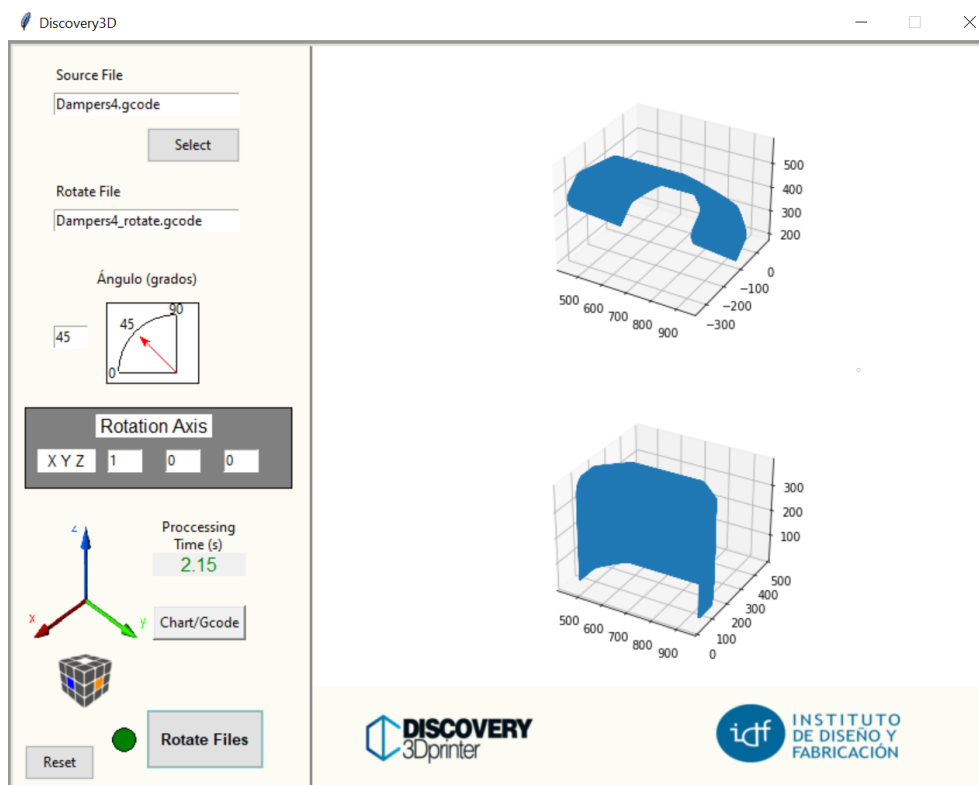


Figura 5.7: Interfaz gráfica para la generación del nuevo G-code de la tapa damper FSUPV.

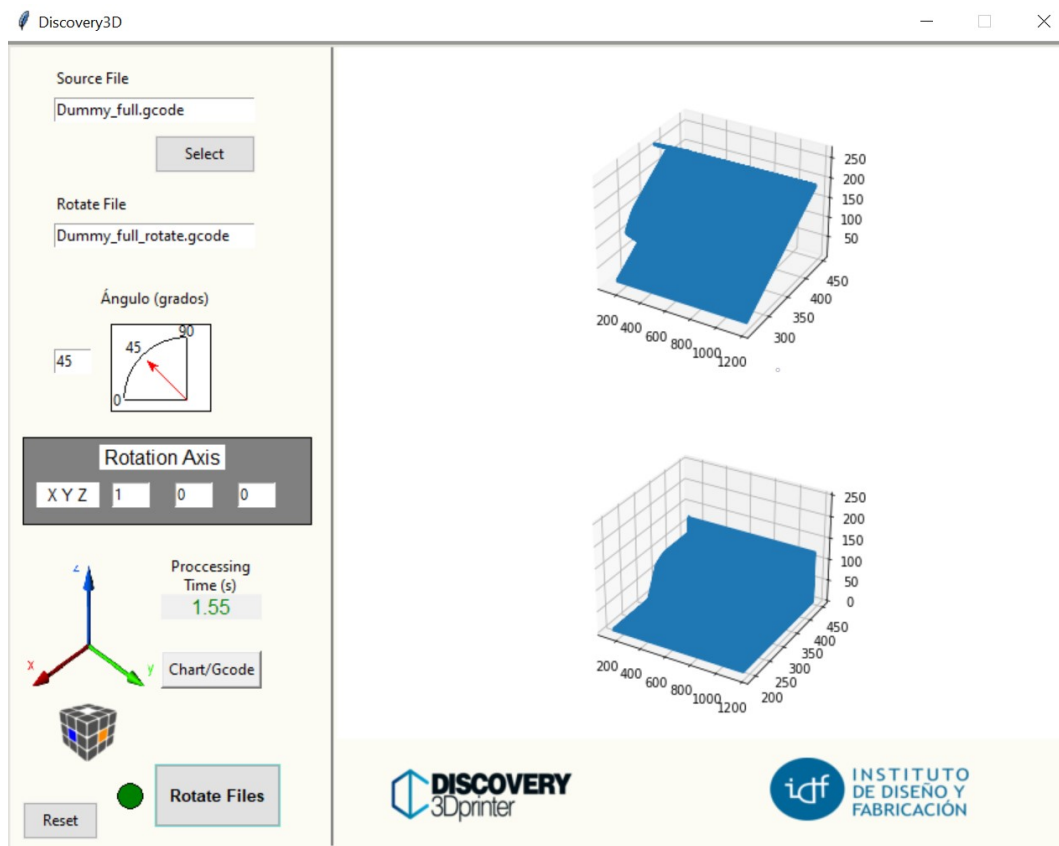


Figura 5.8: Interfaz gráfica para la generación del nuevo G-code del atenuador de impactos.

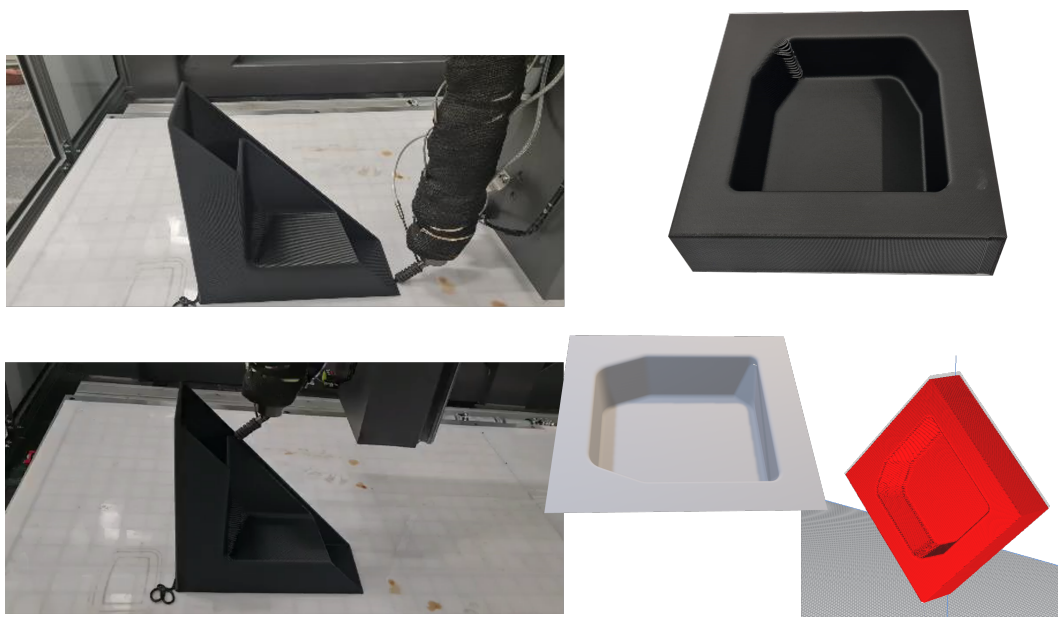


Figura 5.9: Ejemplo impresión atenuador impactos FSUPV.

CAPÍTULO 6

Conclusiones

6.1 Conclusiones del trabajo

En este proyecto se presenta una estrategia que amplía las capacidades de la impresión 3D convencional, limitada al plano XY, y que permite definir trayectorias de deposición en máquinas de Fabricación Aditiva de Gran Formato, con boquillas a 45 grados. La aportación principal es una rutina que modifica el G-code que define las trayectorias de una pieza rotada un ángulo de 45 grados y obtenidas a partir de un programa comercial (*Ultimaker Cura* o *Simplify3D*), devolviendo un G-code con las instrucciones necesarias para poder imprimir la pieza original, usando una impresora 3D con boquilla invertida 45 grados.

El uso de boquillas a 45 grados permite la fabricación de piezas de gran tamaño sin la necesidad de usar soportes. Con la impresión 3D convencional, los voladizos de más de 45 grados deben apoyarse. Con esta propuesta, se evita el uso de apoyos para imprimir grandes piezas. Para ello, el cabezal de impresión de la impresora 3D se gira un ángulo de 45 grados alrededor de un eje horizontal. En este contexto, es necesario una transformación matricial del G-code, generado por un programa comercial (como puede ser *Ultimaker Cura*), ya que en un primer paso se rota la pieza a imprimir y el G-code generado por el slicer corresponde al de la pieza rotada. Al aplicar esta transformación, el G-code final, que representa las instrucciones para los movimientos de la impresora 3D, puede girar en torno a cualquier eje, lo que proporciona una mayor flexibilidad a la hora de colocar el objeto en el lecho de impresión. Este avance abre nuevas posibilidades para imprimir geometrías complejas.

La viabilidad del procedimiento se ha demostrado con la fabricación real de diferentes piezas, tal y como se muestra en el capítulo dedicado a casos prácticos, donde se expone la fabricación real usando el G-code generado por la rutina desarrollada en este trabajo de tres piezas: la pala aerogenerador (expuesto en la JEC World Conference, celebrada en París, abril de 2024), las tapas dampers y cover bottle FSUPV y el atenuador de impactos FSUPV, para el grupo de Generación Espontánea de la Formula Student de la UPV. Entre las ventajas de este diseño están la de mayor velocidad de impresión, mejora de la calidad de la pieza y reducción del coste, al mismo tiempo. Asimismo, nos consta que la rutina está siendo usada por diferentes empresas para la fabricación de piezas, siguiendo la metodología descrita en este trabajo. Entre las empresas o grupos de investigación que recientemente están usando y probando el código para la fabricación de piezas con inclinación de 45 grados, están:

- CNC Bárcenas, empresa con sede en Valdepeñas (Ciudad Real), dedicada a la fabricación de máquinas herramientas, con la que hemos colaborado estrechamente

para depurar la rutina, ya que ha sido la empresa que fabricó la impresora 3D usada en este TFG.

- Sinergia, empresa ubicada en Cartagena, dedicada a la fabricación de embarcaciones, aerogeneradores, etc.
- Studio Artefact, empresa con sede en Montreal (Canada), dedicada a la fabricación de decorados de grandes espectáculos (Circo del Sol).
- Xamk University, en Kouvola (Finland), South-Eastern Finland University of Applied Sciences.



Figura 6.1: Empresas colaboradoras G-code.

Finalmente, a nivel personal, este TFG ha sido para mí una buena oportunidad para profundizar en el lenguaje Python (también Matlab), así como para aprender las bases de la impresión 3D y su aplicación a procesos reales.

6.2 Relación del trabajo con los Estudios cursados

Para la realización de este trabajo he aplicado conocimientos de diferentes asignaturas cursadas durante el Grado. En primer lugar, el cálculo matricial, estudiado en la asignatura de *Álgebra*, me ha servido a la hora de estructurar la rutina que transforma el G-code, donde las coordenadas de los movimientos de los puntos se almacenan en matrices para rotarlos posteriormente, así como también se gestionan matricialmente las instrucciones del G-code anteriores y posteriores a los puntos rotados. Asimismo, las prácticas de esta asignatura fueron con Scilab, un programa que se parece a Matlab, lo que me facilitó los primeros pasos para el diseño de la rutina, que como he dicho, inicialmente comenzó con una primera versión de la rutina programada con Matlab.

Los conocimientos adquiridos de programación en Java en las asignaturas *Introducción a la informática y a la programación* y en *Programación* me han servido a la hora de crear los scripts en Python. Más específicamente, durante mi Erasmus cursé dos asignaturas que también me han ayudado en la programación del código: *Python Programming* y *Matlab Programming*.

Por otra parte, en asignaturas relacionadas con el funcionamiento de los ordenadores (*Fundamentos de Computadores, Tecnologías de Computadores, Estructuras de Computadores*) he estudiado el lenguaje ensamblador para programar y controlar el hardware de manera directa, al igual que lo hace el lenguaje G-code en las máquinas de CNC y, concretamente, en la impresión 3D. Aunque son diferentes lenguajes y el lenguaje ensamblador está más cerca del código máquina, ambos permiten controlar máquinas de manera eficiente y automatizada. El conocimiento de este lenguaje me ha permitido entender mejor la estructura de los ficheros G-code, así como su funcionamiento e importancia para describir los movimientos de la impresora 3D.

Finalmente, para la planificación, gestión y redacción del TFG me ha sido de gran utilidad la asignatura de *Gestión de Proyectos*.

CAPÍTULO 7

Líneas futuras

Como se deduce de las conclusiones del presente TFG, se puede dar por concluida la fase de desarrollo del software. Se ha realizado su verificación experimental en los equipos de la UPV del Instituto de Diseño para la Fabricación (IDF) obteniéndose unos resultados muy satisfactorios. Durante el desarrollo del TFG se han fabricado, utilizando este rutina, más de 10 moldes y piezas para diferentes grupos de investigación y grupos de Generación Espontánea. A continuación se enumeran las principales líneas de trabajo para continuar con esta línea de trabajo:

- **Implantación industrial de la rutina.**

El presente TFG se ha programado de manera íntegra con Python, lo que permite tener un ejecutable para poder ser distribuido y comercializado. Actualmente, como ya se ha comentado en las conclusiones del trabajo, el código se está usando por cuatro empresas, dos nacionales y dos internacionales (ver Figura 6.1). Se propone hacer un seguimiento durante un año de las piezas impresas y recibir el feedback de las empresas para depurar y/o mejorar el código.

- **Desarrollo de trayectorias totalmente tridimensionales (Out of Plane)**

Se ha observado las grandes lagunas en los programas slicers para generar código fuera de plano, lo que abre un gran abanico de líneas de trabajo en el desarrollo de trayectorias de deposición totalmente tridimensionales. En el caso de los equipos de extrusión movidos por un robot manipulador permitirá hacer recubrimientos con trayectorias continuas, evitando escalones entre capas (Figura 2.3).

- **Corte de piezas con superficies**

El programa permite hacer una transformación geométrica de todos los puntos de un G-code. En este TFG sólo se ha experimentado con el giro de los planos, pero nada impide hacer transformaciones más complejas, como escalados o deformaciones. Todas estas transformaciones son posibles, siempre que dispongamos de la ecuación de transformación y de su inversa, ya que los puntos deben ser transformados para llevarlos a *Cura* y retornados a su geometría original para su impresión.

- **Extensión a otros procesos de fabricación**

Se ha propuesto esta técnica para ser aplicada en los procesos de Fabricación Aditiva por MEX al haberse detectado un problema industrial. Es posible hacer la extensión de la técnica a otros procesos de Fabricación Aditiva e incluso a otros procesos de conformado como el corte por láser, el corte por hilo o mecanizado CNC.

Bibliografía

- [1] Additive Manufacturing (Thermwood Corporation, 2024). Consultado en <https://blog.thermwood.com/en-us/thermwood-additive-manufacturing>.
- [2] ASTM-F42.91 Subcommittee. Standard terminology for Additive Manufacturing Technologies. *ASTM Des. 10 (2015) 4 F2792-12a*. 82: C: 1–15, 2016.
- [3] B. Brenken, E. Barocio, A. Favaloro, V. Kunc, R. Byron-Pipes. Development and validation of extrusion deposition additive manufacturing process simulations. *Additive Manufacturing*, 25: 218-226, 2019.
- [4] P. Chesser, B. Post, A. Roschli, C. Carnal, R. Lind, M. Borish, L. Love. Extrusion control for high quality printing on Big Area Additive Manufacturing (BAAM) systems. *Additive Manufacturing*, 28: 445-455, 2019.
- [5] K. Cooper. *Rapid Prototyping—Principles and Applications*. Marcel Dekker, 2001.
- [6] Características de Python, 2022. Consultado en <https://bambu-mobile.com/caracteristicas-de-python/>.
- [7] M. Esteban. *Observatorio para la Industria 4.0*. Universidad de Valladolid, 2017.
- [8] S. Fathizadan, F. Ju, K. Rowe, A. Fiechter, A. Hofmann. Novel Real-Time Thermal Analysis and Layer Time Control Framework for Large-Scale Additive Manufacturing. *ASME. Journal of Manufacturing Science and Engineering*, 1: 143-152, 2021.
- [9] J. Jiang, X. Xu, J. Stringer. Support structures for additive manufacturing: a review. *Journal of Manufacturing and Materials Processing*, 2: 64: 1-23, 2018.
- [10] P. P. Kruth. Material increment manufacturing by rapid prototyping techniques. *CIRP Annals—Manufacturing Technology*, 40: 2: 603–614, 1991.
- [11] Los 9 pilares tecnológicos de la Industria 4.0. *Industria 4.0*. Publicado el 16 de junio de 2015. Consultado en <http://industria-4.blogspot.com/2015/06/los-9-pilares-tecnologicos-de-la.html>.
- [12] L. Love, B. Post, A. Roschli. *Wide and High Area Additive Manufacturing (WHAAM)*. Energy and Transportation Sciences Division Advanced Manufacturing Office, ORNL/TM-2017/167, 2017.
- [13] Juan Luis Martín Morlanes. *Optimización de los procesos de impresión 3D*. TFG, Universidad de Valladolid, 2020.
- [14] Matlab, 2024. Consultado en <https://es.mathworks.com/products/matlab.html>.
- [15] Movimiento RepRap, Wikipedia 2020. Consultado en <http://reprap.org/wiki/RepRap/es>.

- [16] T. D. Ngo, A. Kashani, G. Imbalzano, K. T. Q. Nguyen, and D. Hui. Additive manufacturing (3D printing): A review of materials, methods, applications and challenges. *Composites Part B: Engineering*, 143: 172-196, 2018.
- [17] G. A. Nisja, A. Cao, C. Gao. Short review of nonplanar fused deposition modeling printing. *Material Design & Processing Communications*, 3(4) e221: 1-11, 2021.
- [18] R. Noorani. *Rapid Prototyping — Principles and Applications Rapid Prototyping — Principles and Applications*. JohnWiley & Sons, 2006.
- [19] Numpy, 2023. Consultado en <https://numpy.org>.
- [20] W. Oropallo, L.A. Piegl. Ten challenges in 3D printing. *Engineering with Computers*, 32: 135–148, 2016.
- [21] Patel, A., Taufik, M. Extrusion-Based Technology in Additive Manufacturing: A Comprehensive Review. *Arabian Journal for Science and Engineering* 49, 1309–1342 (2024).
- [22] Python, 2001-2024. Consultado en <https://www.python.org/>.
- [23] PwC. *Industry 4.0: Building the digital enterprise*. Global Industry 4.0 Survey, 2016.
- [24] A. Roschli, K. Gaul, A. M. Boulger, B. Post, P. Chesser, L. Love, F. Blue, M. Borish. Designing for Big Area Additive Manufacturing. *Additive Manufacturing*, 25: 275-285, 2019.
- [25] Siemens Digital Industries Software. Consultado en <https://plm.sw.siemens.com/es-ES/>.
- [26] Software Simplify3D (2024). Consultado en <https://www.simplify3d.com/>.
- [27] Top 5: El mejor software de corte de impresion 3D para impresoras FDM, 2023. Consultado en <https://www.anycubic.es/blogs/guias-de-impresion-3d/top-5-el-mejor-software-de-corte-de-impresion-3d-para-impresoras-fdm>.
- [28] 3Dnatives, el sitio web de la impresión 3D, 2021. Consultado en <https://www.3dnatives.com/es/g-code-proceso-impresion-3d-230920212/>.
- [29] UltiMaker Cura. UltiMaker (2011-2024). Consultado en <https://ultimaker.com/software/ultimakercura/>.
- [30] C.M.S. Vicente, M. Sardinha, L. Reis, et al. Large-format additive manufacturing of polymer extrusion-based deposition systems: review and applications. *Progress in Additive Manufacturing*, 8: 1257–1280, 2023.
- [31] N. Volpato, D. Kretschek, J.A. Foggiatto, C.M. Gomez da Silva Cruz. Experimental analysis of an extrusion system for additive manufacturing based on polymer pellets. *International Journal of Advanced Manufacturing Technology*, 81: 1519–1531, 2015.
- [32] C. Wang, X.P. Tan, S.B. Tor, C.S. Lim. Machine learning in additive manufacturing: State-of-the-art and perspectives. *Additive Manufacturing*, 36: 101538, 2020.
- [33] A. Wedgewood. *Materials Development and Advanced Process Simulation for Additive Manufacturing with Fiber-Reinforced Thermoplastics*. The Composites Institute. IACMI/R003-2020/7.07, 2020.

-
- [34] F. Wulle, D. Coupek, F. Schaffner, A. Verl, F. Oberhofer, T. Maier. Workpiece and machine design in additive manufacturing for multi-axis fused deposition modeling. *Procedia CIRP*, 60: 229–234, 2017.
- [35] M. Wüthrich, J.W. Elspass, P. Bos, S. Holdener. Novel 4-Axis 3D Printing Process to Print Overhangs Without Support Material. *Industrializing Additive Manufacturing*, 130–145, 2021.
- [36] H.M. Zhao, Y. He, J.Z. Fu, J.J. Qiu. Inclined layer printing for fused deposition modeling without assisted supporting structure. *Robotics and Computer-Integrated Manufacturing*, 51: 1-13, 2018.

APÉNDICE A

Presupuesto

Se ha realizado un estudio económico para estimar el coste de realización del proyecto. El presupuesto se ha dividido en dos partes: costes de personal y material, y costes de maquinaria para la realización de las pruebas prácticas para su validación.

La Tabla A.1 incluye una estimación del coste de personal y el material necesario para llevar a cabo el proyecto.

Factor	horas o Kg	coste/hora o coste/kg	coste total
Creación de modelos 3d	20 <i>h</i>	40 €/h	800€
Impresiones	30 <i>h</i>	80 €/h	2400€
Material	50 kg	20 €/kg	1000€
Programación	500 <i>h</i>	40 €/h	20.000€
Coste total			24.200€

Tabla A.1: Costes de personal.

Respecto a los costes en maquinaria se debe tener en cuenta que se realiza una amortización de la maquinaria en 5 años, con lo que el valor de las máquinas se encuentra reducido a una quinta parte de su valor real. Se estima que la máquina trabaja un total de 2.500 horas/año. El coste de la máquina fue de 210.000 €, por lo que el coste horario será $210000 / (5 * 2500) = 16,8$ €/h. Así, en la Tabla A.2 se puede observar el coste de uso de la máquina de impresión durante unas 30 horas.

Factor	unidades	coste/unidad	coste total
Máquina de impresión gran formato	30	16,8€	504€

Tabla A.2: Costes de material.

En conclusión, la suma de ambos costes asciende a 24.704 €, importe al que añadiendo el IVA, resulta en un total de **29.891,84 €**.

APÉNDICE B

Objetivos de Desarrollo Sostenible

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).



El 25 de septiembre de 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. Cada objetivo tiene metas específicas que deben alcanzarse en los próximos 15 años.

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No procede
ODS 1. Fin de la pobreza.				x
ODS 2. Hambre cero.				x
ODS 3. Salud y bienestar.				x
ODS 4. Educación de calidad.				x
ODS 5. Igualdad de género.				x
ODS 6. Agua limpia y saneamiento.				x
ODS 7. Energía asequible y no contaminante.		x		
ODS 8. Trabajo decente y crecimiento económico.		x		
ODS 9. Industria, innovación e infraestructuras.	x			
ODS 10. Reducción de las desigualdades.				x
ODS 11. Ciudades y comunidades sostenibles.	x			
ODS 12. Producción y consumo responsables.	x			
ODS 13. Acción por el clima.	x			
ODS 14. Vida submarina.				x
ODS 15. Vida de ecosistemas terrestres.				x
ODS 16. Paz, justicia e instituciones sólidas.				x
ODS 17. Alianzas para lograr objetivos.		x		

De los anteriores objetivos de desarrollo sostenibles mencionados y, de acuerdo con los grados de relación señalados en la tabla anterior, se puede decir que el proyecto está relacionado con:

- **ODS 7. Energía asequible y no contaminante.** El TFG proporciona una mejora en la fabricación de piezas de diferentes sectores, entre los que destaca el de las energías renovables y, por tanto, contribuye al desarrollo de métodos más eficientes para conseguir energía no contaminante.
- **ODS 8. Trabajo decente y crecimiento económico.** Considero que la aportación de este TFG contribuye a mejorar la fabricación de grandes piezas de diferentes sectores, tan importantes como los de energías renovables, aeroespacial o la construcción y, como consecuencia contribuye a la creación de empleo, así como al crecimiento económico.
- **ODS 9. Industria, innovación e infraestructura.** Como se ha comentado ya, la impresión 3D es uno de los 9 pilares de la Industria 4.0. En particular, la estrategia diseñada en este TFG para imprimir piezas de gran tamaño sin usar soportes supone una innovación importante en los procesos LFAM y contribuye a impulsar la industria, ahorrando tiempo y recursos, y mejorando la infraestructura industrial.
- **ODS 11. Ciudades y comunidades sostenibles.** Uno de los sectores más beneficiados de la impresión 3D de piezas de gran formato es la construcción, por lo que este TFG puede contribuir a la mejora de las infraestructuras en las ciudades. Por otro lado, las tecnologías aditivas se basan en aportar el material en lugar de eliminarlo, lo que hace que los residuos sean muy inferiores y, por lo tanto, mucho más ecológico.
- **ODS 12. Producción y consumo responsables.** Con la técnica de impresión propuesta en este TFG se minimiza la cantidad de material desperdiciado gracias a que se puede evitar el uso de soportes, lo que contribuye a la producción y consumo responsables.
- **ODS 13. Acción por el clima.** Se puede decir que la aportación de este TFG supone una mejora para determinados procesos de fabricación aditiva (ya de por sí más ecológicos que los tradicionales), el uso de menos material, dado que la técnica propuesta no requiere la construcción de soportes para imprimir piezas grandes y, finalmente, aunque no está directamente ligado a la propuesta de este TFG, el hecho de que exista la posibilidad de usar materiales reciclables, contribuyen a la reducción de los gases de efecto invernadero en los procesos productivos, causantes del cambio climático.