

UNIVERSITAT POLITÈCNICA DE VALÈNCIA  
DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

TFM Máster Universitario en Automática Industrial y  
Automática:

# MEJORA DEL SISTEMA DE CONTROL PARA IMPRESORAS 3D

Autor: Abel Martínez Martínez

---

Director:  
Ángel Valera Fernández

Codirector:  
Miguel Sánchez López

Septiembre 2015



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



departamento de ingeniería  
de sistemas y automática



# Índice general

<b>1. Introducción</b>	<b>2</b>
<b>2. Desarrollo teórico</b>	<b>5</b>
2.1. Impresoras 3D . . . . .	5
2.1.1. Estado del arte . . . . .	5
2.1.2. Proyecto RepRap . . . . .	9
2.1.3. Esquema de funcionamiento de la impresión 3D . . . . .	10
2.2. Modelo 3D . . . . .	11
2.2.1. Archivos de modelado. . . . .	12
2.3. Programa rebanado . . . . .	13
2.4. Controlador máquina . . . . .	14
2.5. Software de control de trayectorias . . . . .	15
2.6. Esquema hardware de impresora 3d . . . . .	17
2.7. Hardware control de la máquina . . . . .	17
2.8. Etapa de potencia . . . . .	20
2.9. Actuadores . . . . .	21
2.10. Lenguajes de programación de M-H . . . . .	24
2.10.1. Código ISO . . . . .	24
2.10.2. Programación APT . . . . .	27
2.10.3. Lenguajes propietarios . . . . .	27
2.11. Generadores de código . . . . .	27
2.11.1. GRBL . . . . .	27
2.11.2. linuxCNC . . . . .	28
2.12. Identificación y control de procesos . . . . .	30
<b>3. Desarrollo práctico</b>	<b>33</b>
3.1. Impresora Prusa I3 . . . . .	33
3.1.1. Descripción impresora. . . . .	33
3.1.2. Software. . . . .	37
3.2. Modificación de la impresora (hardware) . . . . .	40
3.3. Identificación y diseño de controladores para motores CC . . . . .	43

3.4. Programación unidad control . . . . .	50
3.5. Resultados . . . . .	51
<b>4. Conclusiones y trabajos futuros</b>	<b>53</b>
4.1. Trabajos futuros . . . . .	53

## Resumen

En el presente trabajo se realiza un estudio sobre la situación actual de la impresión 3d. A continuación se analizan los problemas a los que se enfrentan los usuarios cuando necesitan fabricar una pieza con su máquina. Se plantea una alternativa al modelo de funcionamiento actual el lazo abierto utilizando motores paso a paso. En su lugar se implementa un sistema de control en lazo cerrado utilizando motores de continua, encoders, una etapa de potencia y un microcontrolador. Se imprimen piezas con la configuración original para compararlas más tarde con las que se fabricarán con los motores de continua. Se instalan los motores en la máquina y se obtiene el modelo de funcionamiento de los mismos. Con ayuda de un simulador se obtienen los parámetros del controlador y se verifica que el funcionamiento en simulación es el adecuado. Se implementa en un microcontrolador el programa de control de los motores y se configura para que funcione correctamente. Tras realizar varias pruebas de funcionamiento se imprimen piezas utilizando los motores de continua.

**Palabras clave:** Impresora 3d, control de motores, FDM, máquina CNC.

## Resum

En el present treball es realitza un estudi sobre la situació actual de la impressió 3d. A continuació s'analitzen els problemes a què s'enfronten els usuaris quan necessiten fabricar una peça amb la seva màquina. Es planteja una alternativa al model de funcionament actual amb llaç obert utilitzant motors pas a pas. En el seu lloc s'implementa un sistema de control amb llaç tancat utilitzant motors de contínua, encoders, una etapa de potència i un microcontrolador. S'imprimeixen peces amb la configuració original per comparar més tard amb les que es fabricaran amb els motors de contínua. S'instal·len els motors a la màquina i s'obté el model de funcionament dels mateixos. Amb ajuda d'un simulador s'obtenen els paràmetres del controlador i es verifica que el funcionament en simulació és l'adequat. S'implementa en un microcontrolador el programa de control dels motors i es configura perquè funcioni correctament. Després de realitzar diverses proves de funcionament s'imprimeixen peces utilitzant els motors de contínua.

**Palaules clau:** Impresora 3d, control de motors, FDM, màquina CNC

## Abstract

In this paper it is presented a study on the current status of 3D printing. Here are the issues that users face when they need to make a piece with the machine are analyzed. It poses an alternative to the current operating model using open loop stepper motors. Instead a system of closed loop control using DC motors, encoders, a power amplifier and a microcontroller is implemented. Pieces are printed with the original settings for later comparison with those with dc motor manufactured. Motors are installed on the machine and the operating model of those is obtained. Using a simulator controller parameters are obtained and verified that the performance simulation is right. Program motor control is implemented on a microcontroller and configured to work properly. After doing several test runs are printed pieces using DC motors.

**Keywords:** 3d printer, dc motor control, FDM, CNC machine.

# Capítulo 1

## Introducción

Las impresoras 3d pueden utilizarse para la obtención de prototipos así como para fabricación de piezas. Históricamente se han empleado máquinas para estos mismo fines, como por ejemplo la fresadora o el torno. Utilizando estas máquinas, se obtiene la pieza requerida mediante la eliminación del material sobrante, un proceso denominado también fabricación sustractiva. Las máquinas han utilizado las tecnologías disponibles con el fin de obtener piezas de mayor calidad y más complejas. Se controlaban inicialmente mecánicamente, utilizando sincronizaciones mecánicas, como conexiones husillo-tuerca o mecanismos de levas por ejemplo. Las primeras máquinas guiadas por ordenador hacían uso de tarjetas perforadas para su control, y posteriormente han pasado a controlarse plenamente por ordenador. Al mismo tiempo, la idea de fabricar piezas de forma aditiva requiere superar dos grandes desafíos. El primero es depositar el material en la posición que se desea y el segundo es conseguir la tecnología necesaria para añadir el material de forma progresiva y de este modo ir conformando la pieza. Las tecnologías de impresión 3d actuales se patentaron entre 1986 y 1992. La primera de ellas fue inventada en 1983 por Chuck Hull y fue denominada como estereolitografía. Consiste en aplicar patrones de luz a un líquido fotosensible que solidifica capa a capa, de este modo se puede ir fabricando la pieza con la geometría deseada. Más tarde fue el estudiante Carl Deckard quien ideó un sistema que fue denominado sinterizado selectivo por láser. En este caso se unían partículas de polvo mediante la acción calorífica del láser. La tercera tecnología que se desarrolla es la técnica de depósito de filamento fundido, inventada por S. Scott Crump en 1989, esta idea consiste en formar la pieza aplicando filamentos de termoplástico, que se pegan unos a otros y también entre capas. Durante muchos años la utilización de la impresión 3d se ha reservado a sectores muy concretos de la industria, como el sector aeroespacial o del automovil. Probablemente esto se debía al alto precio de esta tecnología

que estaba protegida por ley gracias a las patentes de que disponían las empresas que la explotaban. En 2005 el Dr Bowyer de la Universidad de Bath construye la primera impresora autoreplicante RepRap y con ella impulsa en gran medida la idea de la democratización de la impresión 3d. La idea principal de este proyecto es la de fabricar máquinas capaces de fabricar las piezas que las componen, de este modo es posible para el propietario de una de estas máquinas fabricar una máquina para un amigo. Otros participantes del mundo de la impresión 3d son los portales de intercambio de modelos 3d, como por ejemplo thingiverse.com. que pone a disposición del usuario miles de modelos diseñados por otras personas que pueden utilizarse o editarse al gusto.

## Objetivos

El objetivo principal de este trabajo es el de mejorar la calidad de las piezas obtenidas utilizando una impresora de 3D de código abierto. Uno de los problemas que presentan estas máquinas es que su sistema de control funciona en bucle abierto utilizando motores paso a paso. El problema es que si por algún motivo, el motor que controla la posición de un eje no puede girar hasta la posición generada por el generador de trayectorias, lo que sucederá es que el motor perderá pasos, sin que el control reciba una retroalimentación que lo informe de la situación acaecida. Esta situación puede generar que las piezas no tengan la calidad esperada, o bien que no resulten útiles para lo que se habían diseñado. Con el fin de conseguir el objetivo principal mencionado anteriormente, se plantean los siguientes subobjetivos:

- Analizar la problemática relacionada con los sistemas de fabricación asistida por ordenador.
- Estudiar diferentes tecnologías de impresoras 3D.
- Analizar diferentes lenguajes de programación de M-H.
- Conocer soluciones de generadores de código para impresión 3D.
- Analizar la arquitectura hardware y software de una impresora 3D.
- Sustituir las etapas relacionadas con el sistema de actuación de la impresora 3D.
- Caracterizar los motores de corriente continua que se utilizarán en la impresora 3D.

- Realizar la programación de los algoritmos de control de los motores en la arquitectura software de la máquina.
- Verificar el correcto funcionamiento del trabajo desarrollado.

## Justificación

El desarrollo de este trabajo está justificado debido al aumento de la cantidad de seguidores de la impresión 3D, con una demanda constante de mejora de sus propias máquinas. Existen servosistemas industriales que pueden controlar con gran precisión la posición de los motores, pero a pequeña escala y para el ámbito doméstico, no hay sistemas capaces de controlar los ejes de una máquina guiada por ordenador a precios razonables. Por otra parte cuando una persona dedica tiempo (horas) y dinero (material) en obtener una pieza, lo último que quiere es obtener una pieza defectuosa que no le sirve para nada. Desde el punto de vista económico, si el sistema funciona y los consumidores empiezan a demandar controladores con motores de continua para la impresión 3D, supondrá un abaratamiento del equipo y quedará disponible para la elaboración de otros proyectos, como brazos biónicos por ejemplo.

La realización de este trabajo es para mí más que un requisito necesario para obtener el título de Máster en Automática e Informática Industrial, supone la posibilidad de poner en práctica los conocimientos adquiridos durante el máster y también el grado. Puesto que el trabajo planteado en este documento integra diferentes disciplinas como son informática, control, electrónica y mecánica. En el máster se imparten asignaturas relacionadas con informática, por ejemplo Computadores y redes, Sistemas distribuidos o Sistemas empotrados. Las asignaturas de control que se imparten en el máster son Implementación de sistemas de control, Instrumentación y control, Control y programación de robots, Planificación y control dinámico de robots. En referencia a la parte de electrónica y mecánica, se han visto de forma casual en el máster pero en el grado en electrónica industrial y automática si que se estudió más en profundidad.

# Capítulo 2

## Desarrollo teórico

### 2.1. Impresoras 3D

En el siguiente apartado se explicarán brevemente algunas de las tecnologías de impresión más importantes que se conocen hasta el momento. A continuación se estudiarán las estructuras que se utilizan para construir las impresoras 3d. En el siguiente apartado se presentará uno de los proyectos que más ha contribuido al avance de la impresión 3d doméstica. En el siguiente punto se introducen los elementos necesarios para llevar a cabo una impresión 3d, entrando a detallar elementos como el modelo 3d, el programa de rebanado o el programa de control de los motores entre otros.

#### 2.1.1. Estado del arte

Tradicionalmente uno de los métodos más utilizados en la industria a la hora de fabricar prototipos era la fabricación por eliminación de material a partir de una pieza de mayor tamaño. De esta forma trabajan por ejemplo los tornos, las fresadoras, las máquinas de electroerosión o las limadoras, entre otras. Pero el modo de funcionamiento utilizado en la impresión 3D es el contrario, en este caso la fabricación de las piezas se basa en depositar material en las zonas de la pieza que lo requieran, dejando los orificios de esta sin aportar material. En la actualidad es posible imprimir utilizando diferentes técnicas, pero las más conocidas son:

- Estereolitografía (SLA)
- Sinterización Selectiva por Láser (SLS)
- Deposición de Material Fundido (FDM)

**Estereolitografía (SLA)** Es la primera técnica de impresión 3D que apareció y sus siglas derivan del inglés (Stereolithography). Consiste en la aplicación de luz ultravioleta a una resina líquida sensible a la luz (Ver Figura 2.1). De este modo se va formando la pieza que se desea imprimir capa a capa. La calidad y el nivel de detalle de las piezas obtenidas con este método es muy alto, pero no es común su uso doméstico debido al alto coste de adquisición.

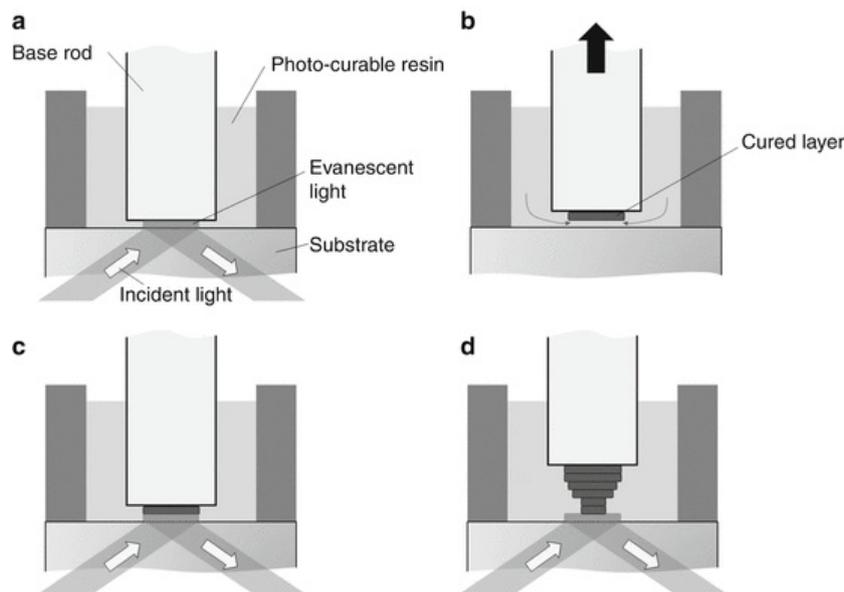


Figura 2.1: Funcionamiento estereolitografía.

**Sinterización Selectiva por Láser (SLS)** Con esta técnica se pueden imprimir multitud de materiales en polvo. Su base de funcionamiento es que un láser impacta en una capa de polvo, en ese momento el material se funde y se solidifica. De este modo se va conformando la pieza. Al igual que la técnica de SLA permite obtener piezas de gran precisión y su utilización se concentra en el ámbito industrial y de investigación.

**Deposición de Material Fundido (FDM)** La tecnología de funcionamiento de estas impresoras es la de formar la pieza capa a capa depositando finos hilos de material fundido con el que se van formando las piezas (Ver Figura 2.3). Es posible imprimir diferentes tipos de plástico. Ha sido posible la expansión de la impresión 3D a los hogares en gran medida gracias a la aparición de proyectos de código abierto importantes, como por ejemplo RepRap. A nivel particular, la tecnología de impresión más extendida entre los entusiastas es la de FDM.

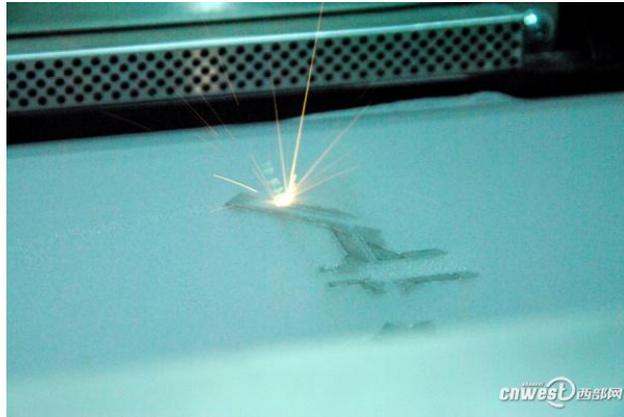


Figura 2.2: Impresora 3D láser.

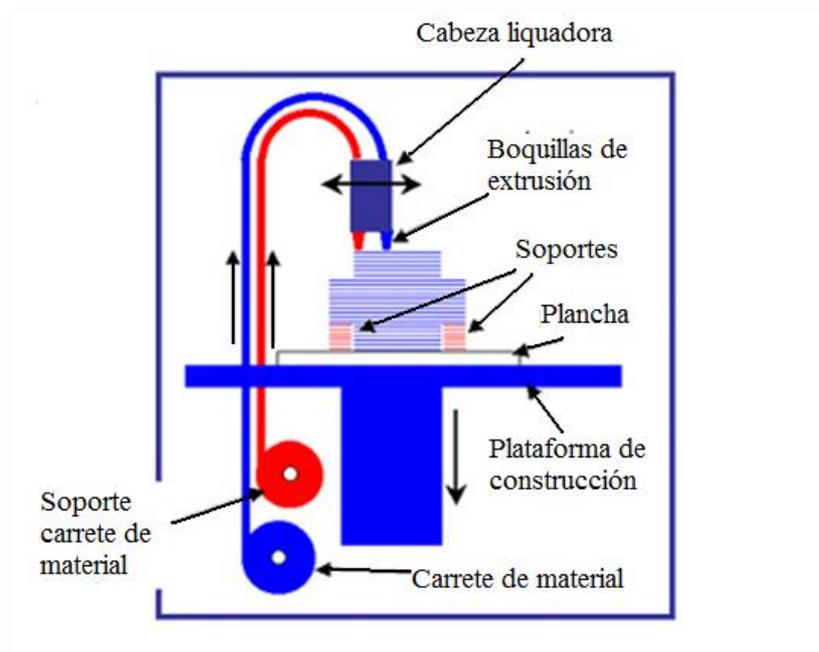


Figura 2.3: Funcionamiento FDM.

### **Tipos de impresoras**

A continuación se clasifican las impresoras 3D por deposición de material fundido según la construcción de la máquina.

- **Impresoras delta:** Las impresoras delta están construidas utilizando un conjunto de cadenas cinemáticas de 4 barras. Dispone de 3 brazos que unen el extrusor al chasis de la máquina y la principal ventaja

que presentan con respecto a las impresoras cartesianas es que en estas máquinas, la pieza no efectúa ningún movimiento. En las Figuras 2.4 se pueden ver una imagen de impresora delta.

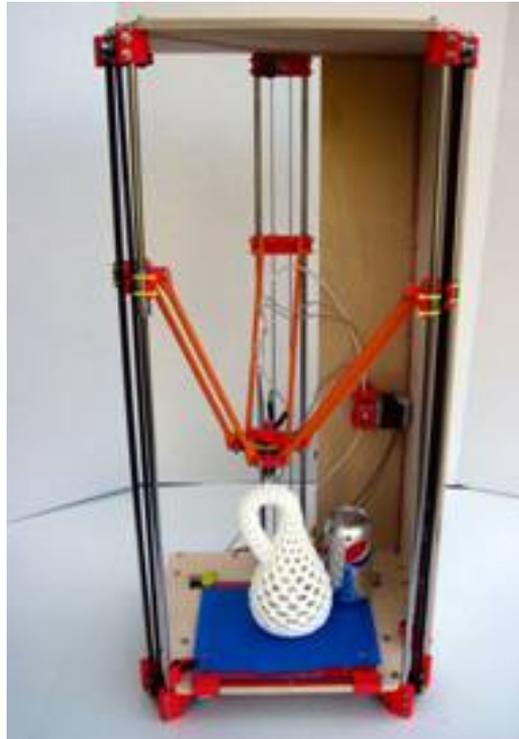


Figura 2.4: Impresora delta.

- **Impresora cartesiana:** Este modelo de impresora dispone de 3 ejes dispuestos de forma ortogonal, lo que le permite posicionarse en todas las posiciones dentro de su campo de trabajo. El movimiento puede aplicarse al extrusor, a la pieza o repartirse entre ambos. Normalmente los movimientos X y Z suelen aplicarse sobre el extrusor, mientras que el Y se le aplica a la pieza, aunque existen modelos en los que la altitud Z también se aplica sobre la pieza. La Figura 2.5 muestra una imagen de una impresora cartesiana.
- **H-belt o coreXY:** Las impresoras con este tipo de mesa son muy similares a las impresoras cartesianas, pero tienen la diferencia en los ejes X e Y. Gracias a su diseño particular, se controlan los ejes X e Y con una sola correa. La Figura 2.6 muestra una imagen de una impresora Hbelt. Los motores de estos ejes se sitúan en la parte fija de la máquina, por lo que las piezas en movimiento son más ligeras

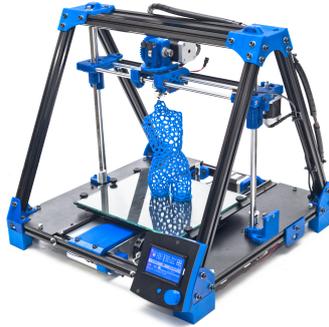


Figura 2.5: Impresora cartesiana.

y se pueden alcanzar mayores velocidades. La configuración H-belt y coreXY no son idénticas, pero su dinámica de funcionamiento sí es la misma.

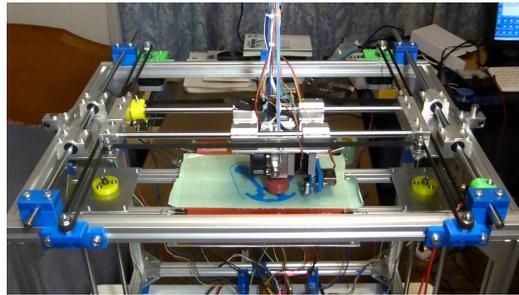


Figura 2.6: Impresora con mesa Hbelt.

### 2.1.2. Proyecto RepRap

En el 2004 Adrian Bowyer se dió cuenta de que la impresión 3D era una tecnología versátil mediante la cual debía de ser posible diseñar una máquina de impresión 3D de fabricación de filamento fundido FFF que podría fabricar una parte importante de sus propias piezas. La industria convencional encuentra poca utilidad en esta idea. ¿Para qué vender máquinas a los clientes cuando nunca van a comprar otra máquina, ni recambios o incluso les permite la fabricación de máquinas en competencia directa con el vendedor? Pero poseer una impresora 3D puede tener ventajas para las personas en general. Cualquiera persona que disponga de una puede hacer piezas, y también puede fabricar otra máquina para darla a un amigo. Este es un caso interesante de fallo del mercado, en el que la máquina auto-replicante es un

objeto que las personas podrían querer, pero en el que nadie está interesado en vender. Por estas razones se decidió diseñar y fabricar una máquina y facilitar todos los diseños bajo la licencia GNU General Public Licence en la web. Este fue el inicio del proyecto RepRap. RepRap es la abreviación de Replicating Rapid-prototyper. RepRap está teniendo un éxito significativo, y actualmente se encuentra en su segunda versión. Desde el principio RepRap fue concebida como una máquina que podría ser utilizada por personas para hacer cosas en sus casas, así como en la industria. El coste de los materiales necesarios para fabricar una RepRap es de alrededor de 400€, permitiendo que se ajuste al presupuesto de individuos, así como pequeñas comunidades en el mundo desarrollado. RepRap fabrica objetos de una calidad ligeramente inferior que las máquinas comerciales, pero a un precio del 1 por ciento de su coste.

Cada desarrollo o mejora en el diseño del software o la electrónica de RepRap viene a través de las iniciativas de sus propios usuarios. No existe una dirección central dando órdenes. Los propios usuarios invierten su tiempo en la evolución del diseño de las RepRap.

### **2.1.3. Esquema de funcionamiento de la impresión 3D**

Se presenta el esquema de funcionamiento como el conjunto de herramientas tanto software como hardware que se deben disponer para poder proceder a una impresión 3D. En primer lugar se encuentra la obtención del modelo para imprimir. Esta operación se puede realizar a partir de una pieza existente, utilizando un escáner o una cámara 3D. También es posible obtener el modelo con ayuda de programas informáticos en los que, a través de herramientas gráficas, el usuario dibuja la pieza con todas las especificaciones que desea. Una vez generado el modelo, este se envía a un programa informático que convertirá el modelo, en formato de gráficos en 3 dimensiones a un programa que indica las trayectorias que debe recorrer la máquina para obtener la pieza deseada. Cuando se tiene el programa de trayectorias, es necesario disponer de un software que establezca la comunicación entre el ordenador y la máquina. Este software se encarga del control de la temperatura de la cama calentadora, la temperatura del extrusor o el envío de funciones que debe realizar la máquina. Ya en la máquina, se utiliza un software para interpretar las funciones de trayectoria enviadas por el ordenador para convertirlas en señales de control que se envían a los actuadores. Como norma general, estos actuadores no pueden ser controlados directamente por el microcontrolador disponible en la máquina, por tanto se utilizan etapas de potencia para controlar los motores. La Figura 2.7 muestra un esquema de los pasos lógicos necesarios para llevar a cabo una impresión 3D. A continuación se explicarán

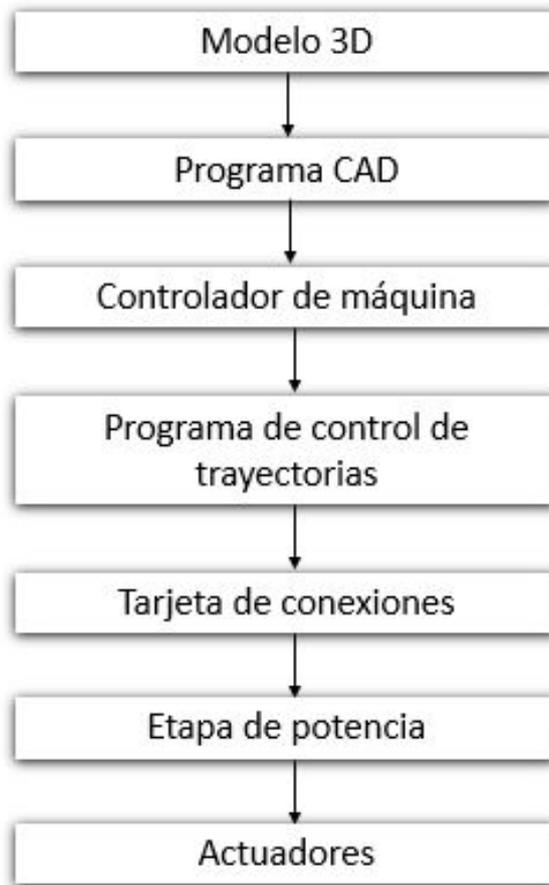


Figura 2.7: Esquema de impresión 3D.

las posibilidades existentes en cada uno de los pasos definidos en el esquema de la Figura 2.7.

## 2.2. Modelo 3D

Los modelos 3d son la representación del objeto 3d. Para ello se utiliza una colección de puntos en el espacio dentro de un espacio tridimensional. Estos puntos se pueden conectar por varias entidades geométricas tales como triángulos, líneas, superficies curvas, etc. La obtención del modelo es el primer paso que debemos realizar para la impresión de una pieza en 3D. Se puede obtener el modelo utilizando programas de edición 3D, escaneando objetos reales o bien descargando el modelo de repositorios de internet en los cuales

los usuarios comparten sus modelos.

- **Diseño 3D** Se utilizan programas informáticos para generar el modelo conforme a nuestras existencias. En la actualidad los programas de diseño 3d son sencillos de utilizar para la mayoría de los usuarios y tras un pequeño periodo de aprendizaje son capaces de diseñar sus propios modelos 3d. Existen variedad de programas tanto de pago como gratuitos con los que podemos obtener nuestros modelos.
- **Escaneado 3D** La obtención del modelo mediante escaneado 3d resulta útil cuando se dispone del objeto del que se quiere obtener el modelo. Consiste en obtener mediante algún tipo de sensor la geometría del modelo. Algunas de las tecnologías de escaneado existentes son de contacto mecánico, de tiempo de vuelo, de triangulación, diferencia de fase, holografía conoscópica, luz estructurada, luz modulada, estereoscópicos o silueta. A menudo estos modelos no se pueden imprimir directamente, por lo que requieren de algún tipo de postprocesado.
- **Utilizar modelos compartidos** Esta puede ser la opción más utilizada por la mayoría de aficionados a la impresión 3d. De este modo, se puede reutilizar el modelo diseñado por otra persona y que ha decidido compartir. Es útil siempre y cuando se encuentre el objeto que se desea imprimir y encaje con nuestro propósito. Algunos de los modelos compartidos disponen de algunos parámetros de impresión, esto significa que es posible editar el texto del objeto o modificar alguna medida fácilmente antes de imprimir nuestra propia copia del objeto. Compartir los modelos a través de la red es un acto altruista que sin embargo puede generar beneficios, como por ejemplo la realimentación de otras personas que dan su punto de vista sobre el diseño.

### 2.2.1. Archivos de modelado.

Los archivos de modelado son creados por programas de modelado 3d. Contienen toda la información referente al objeto que se desea imprimir. En el caso de los archivos para impresión 3d, estos deben generar superficies cerradas que no se entrecrucen. A continuación se citan algunos de los tipos de archivos más utilizados junto con algunas de sus características principales.

**STL.** Es un formato de archivo de diseño asistido por computador (CAD) que define la geometría de objetos 3d. En este tipo de archivo

no se incluyen datos sobre el color o la textura del modelo. Es el formato estándar para las tecnologías de fabricación aditiva. Utiliza una malla de triángulos cerrada para definir la geometría de un objeto. La cantidad de triángulos se puede elegir por el diseñador en el momento de generar el fichero. Una cantidad demasiado elevada generará un archivo de gran tamaño. Por el contrario un archivo con pocos triángulos generará un fichero con una resolución demasiado pequeña.

**3MF.** El formato 3MF tiene capacidad para definir completamente un modelo, incluyendo información interna, sobre el color y otras características. Es un modelo ampliable a nuevas innovaciones de la impresión 3D. Es un formato diseñado para la fabricación aditiva. Actualmente se trata de un formato de código abierto donado por Microsoft.

**IGES.** Graphics International Exchange Standard es un estándar internacional utilizado por la mayoría de los vendedores de CAD. El formato es compatible para permitir el intercambio de archivos entre diferentes proveedores.

## 2.3. Programa rebanado

El programa de rebanado es el que se encarga de procesar el modelo a imprimir para obtener un archivo que la máquina pueda entender. Generalmente las máquinas de impresión 3d utilizan archivos escritos en código G. El programa de rebanado obtiene la sección de la pieza a fabricar en cada una de las capas que conformarán la pieza. A continuación se generan las trayectorias en función de los parámetros introducidos en el programa, que pueden ser altura de capa, porcentaje de relleno y tipo de relleno (cuadrado, colmena de abeja). Algunos programas para rebanado son:

- MatterSlice. Es una aplicación de consola C# que genera código G para imprimir en 3d. Es una mejora importante de CuraEngine, el cual es una traducción a c++ de Skienforge. MatterSlice tiene mejoras significativas en la legibilidad del código, estabilidad numérica y generación de características.
- Skeinforge. Es una cadena de herramientas compuesta por scripts de Python que convierten un modelo 3d en código G para impresoras RepRap.
- Slic3r. El proyecto nació en 2011 dentro de la comunidad Reprap. Es capaz de obtener el código G para imprimir un modelo hasta 100 ve-

ces antes que el programa Skeinforge. Es compatible con computación paralela multihilo.

- Cura. Es un programa de rebanado que destaca por su simplicidad, motivo por el cual está cautivando a muchos aficionados. Actualmente es el programa que utilizan las máquinas Ultimaker.
- Kisslicer. A diferencia de los programas anteriores de código abierto, este programa dispone de una versión gratuita y otra de pago. Las dos versiones son el mismo programa, pero la versión de pago ofrece más funcionalidades. Este programa funciona en Windows, OSX, Linux, RaspBerry Pi y FreeBSD.
- RepSnapper. Está escrito completamente en C++ y por ello debe ser más rápido que los rebanadores que funcionan con scripts. Dispone de algunas características únicas.
- Preform. Este programa prepara los modelos para imprimir con impresoras del fabricante Formlabs.

## 2.4. Controlador máquina

Es el programa interfaz entre el usuario y la máquina. Se ejecuta en el ordenador y pone a la disposición del usuario los botones para la activación de los actuadores. Su principal función es la de leer y enviar el código G al control de trayectorias para que se ejecute la impresión. Algunos programas ofrecen utilidades añadidas, como ejecución de programas de rebanado. Algunos de los programas que controlan la máquina son:

- PronterFace. Este proyecto de código abierto ofrece soluciones para enviar el código G a la máquina utilizando un interfaz gráfico o de consola. Ofrece también a los usuarios la posibilidad de crear botones personalizados y macros. Funciona en Windows, Linux y OSX.
- Repetier. Es un software de gestión de impresoras sencillo de utilizar. Es compatible con la mayoría de los firmwares de las máquinas. En su origen tenía licencia Apache V2 hasta la versión 0.91. Las versiones posteriores a la 0.91 son de código cerrado.
- MatterControl. Este proyecto también es gratuito y abierto. Su principal característica es la sencillez con la que se realizan las impresiones, el propio programa se encarga de realizar el rebanado, con el programa

que elijamos. También es posible poner diferentes piezas en cola para imprimir una después de la otra.

- ReplicatorG. Este software compatible con las impresoras Makerbot replicator, Thing-O-Matic, CupCake CNC, máquinas RepRap y otras máquinas genéricas. Se le puede introducir programas escritos en código G o archivos en formato .stl. Es un programa multiplataforma basado en el entorno Arduino/Processing.
- RepRaptor. Este programa está hecho con Qt con la intención de resultar un programa rápido y de presencia minimalista.
- RepRap Host Software. Disponible para Linux, Windows y Mac, este software implementa el rebanador y el enviador de códigos G a la máquina.
- send.py. Esta aplicación tiene la utilidad de enviar el código G a las impresoras basadas en Arduino. Es una aplicación que funciona en consola y muestra por pantalla comentarios sobre la evolución de la fabricación.
- X2sw. Para hacer este software se han utilizado algunos de los programas conocidos de impresión 3d, como Slic3r, Skeinforge para rebanar las piezas y printrun para enviar los códigos G a la impresora. Está optimizado para utilizarse con extrusión doble, para imprimir en varios colores, varios materiales o material de soporte.
- Simplify3D. Este software comercial para impresión 3d ofrece la posibilidad imprimir desde archivos .stl, encargándose el mismo del rebanado. Destaca por su compatibilidad con impresoras de dos extrusores y por la rapidez con la que realiza la operación de rebanado. Es una herramienta multiplataforma que funciona en Linux, Windows y Mac.
- 3Delta. Este software está especialmente diseñado para funcionar con impresoras tipo delta. Tiene una apariencia similar al Pronterface y ofrece utilidades como ayuda para calibrar la impresora delta.

## 2.5. Software de control de trayectorias

Se trata del software encargado de controlar a más bajo nivel los diferentes actuadores de la máquina. Tiene la misión de controlar la temperatura del extrusor y de la cama calentadora. También interpreta los comandos escritos en código G y genera las trayectorias adecuadas para enviar a cada motor

los pasos necesarios para obtener la pieza correctamente. Algunos de los programas de control de trayectorias son:

- Marlin. Es un firmware para impresoras RepRap. Según las necesidades del usuario, puede configurarse para que funcione correctamente en diferentes tipos de máquina. Soporta hasta 4 extrusores. También es compatible con pantallas LCD y tarjetas SD. Da soporte a impresoras cartesianas, delta y scara. Es un proyecto de código abierto.
- GRBL. Es un controlador de trayectorias basado en Arduino. Está diseñado para utilizarse en máquinas CNC pero no en impresoras, por lo que no puede ser utilizado para imprimir. Pero su generador de trayectorias ha servido como motor para el firmware Marlin.
- Sprinter. Firmware diseñado para la tarjeta RAMPS y otras tarjetas monoprocesador. Soporta impresión con tarjeta SD y activa la cama calentadora.
- SJFW. Este firmware es compatible con la electrónica de la impresora Makerbot. También ofrece posibilidad de imprimir desde tarjeta SD y pantalla LCD. Al igual que Marlin es capaz de leer bloques de código en adelante para ejecutarlos más rápidamente.
- SailFish. Es un firmware diseñado para la impresora Makerbot. Dispone de características indicadas para usuarios experimentados. Es compatible con mesas coreXY y H-belt. También permite modificar la velocidad de desplazamiento y la temperatura del extrusor mientras imprime.
- Repetier. La ventaja que ofrece este sistema es que dispone de un sistema de control de datos con comprobación de suma que permite una gran fiabilidad. Ofrece la posibilidad de controlar el puerto I2C en la tarjeta Sanguino para utilizar la pantalla LCD.
- Aprinter. Este programa tiene de particular que es compatible con multitud de microcontroladores, por ejemplo diferentes modelos AVR, AT91SAM, STM32F4 y Freescale MK20. Ofrece soporte para arquitecturas tipo delta, delta rotativo y CoreXY. Se pueden utilizar otras configuraciones si se implementan las transformaciones de coordenadas directa e inversa, siempre que el microcontrolador disponga de suficiente memoria RAM. Utiliza temporizadores hardware separados para cada motor, por lo que tiene una alta precisión.

- Smoothie. Este firmware está diseñado para el microcontrolador LPC17xx (Arquitectura ARM cortex M3). Este es un potente microcontrolador de 32 bits. Tiene en su generador de trayectorias una parte del GRBL. Asociado a este firmware está la placa de control Smoothie-board.

## 2.6. Esquema hardware de impresora 3d

Las diferentes posibilidades de impresora 3d tienen en común la presencia una serie de elementos hardware. La Figura 2.8 muestra la relación que tienen los diferentes elementos hardware. En lo más alto se encuentra el hardware de

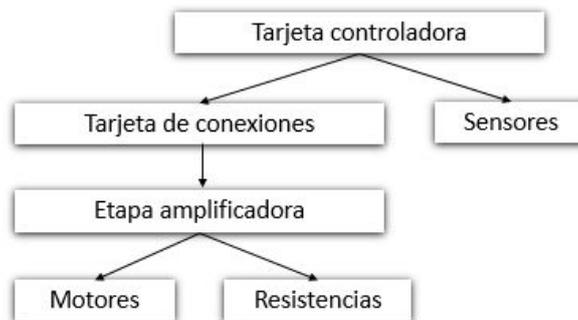


Figura 2.8: Esquema de elementos hardware.

control de la máquina. Conectado a este está la tarjeta de conexiones. A esta tarjeta se conectan los sensores de contacto, los sensores de temperatura y las etapas de potencia de los actuadores. A las etapas de potencia se conectan los actuadores.

## 2.7. Hardware control de la máquina

Es un dispositivo de control, generalmente un microprocesador o microcontrolador que dispone de los periféricos necesarios para acceder a los datos de la impresión, lectura de tarjetas o comunicación USB y también salidas oportunas para el control de los actuadores, resistencias y motores. Algunas de las tarjetas utilizadas en impresión 3d son:

- Arduino Mega + RAMPS. Es una tarjeta microcontroladora basada en el integrado ATmega1280. Dispone de 54 pines de entrada salida, entre los cuales 14 se pueden utilizar como salidas de ancho de pulso

modulado (PWM), 16 se pueden usar como entradas analógicas. También tiene 4 puertos serie. Dispone de un oscilador de cristal de 16 megahercios, conexión USB y se puede alimentar con un conector tipo «Jack». RAMPs. Está diseñada para encajar todos los componentes

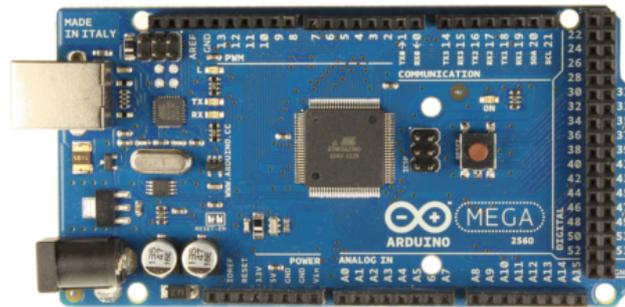


Figura 2.9: Microcontrolador arduino mega.

necesarios para una impresora Reprap en un pequeño paquete de bajo coste. Ramps se conecta a la tarjeta Arduino Mega y dispone de mucho espacio para expansión. El diseño modular incluye conectores para la etapa de potencia Pololu, explicada más adelante y también para el control del extrusor. Se pueden utilizar otras placas de expansión de Arduino siempre que la tarjeta Ramps quede en la parte superior.

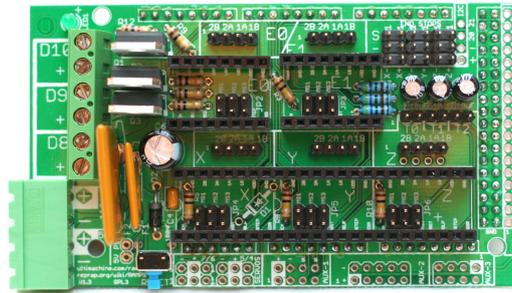


Figura 2.10: Tarjeta de control de la impresora 3D RAMPS.

- Sanguinololu. Es una tarjeta microcontroladora que integra las conexiones para la etapa de potencia. Está diseñada para ser utilizada en impresoras RepRap y otros dispositivos CNC. Utiliza el microcontrolador ATmega644P, aunque también es posible utilizar el ATmega1284. Tiene 4 salidas a etapa de potencia con los pines compatibles con la etapa de amplificación Pololu, que se explicará más adelante. Esta tarjeta cuenta con 14 pines que se pueden utilizar como entrada y salida

estándar. Estos pines se pueden configurar como puertos de comunicaciones I2C, SPI y serie. También dispone de algunos conversores de entrada analógicos. Destaca de esta placa el amplio rango de tensiones de entrada, de 7 a 30 V, según la web del proyecto [8] sirve para facilitar la alimentación de la placa en función de la fuente de alimentación de que dispone el usuario.

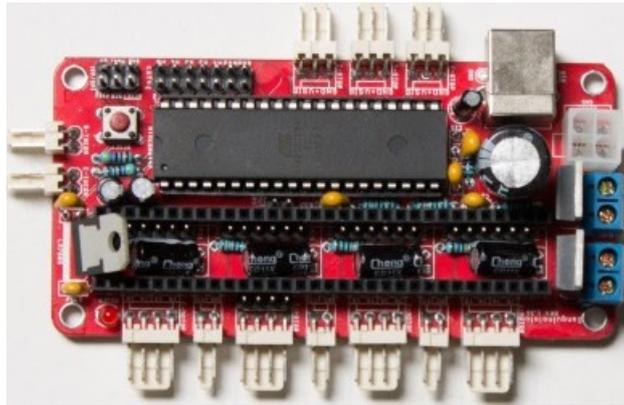


Figura 2.11: Microcontrolador Sanguinololu.

- Gen7. Esta tarjeta microcontroladora es una opción económica para el control de una impresora RepRap o otra máquina CNC. Dispone de 4 salidas para etapa de potencia. Es posible utilizar el microcontrolador ATmega1284P-PU o bien el ATmega644-20PU.

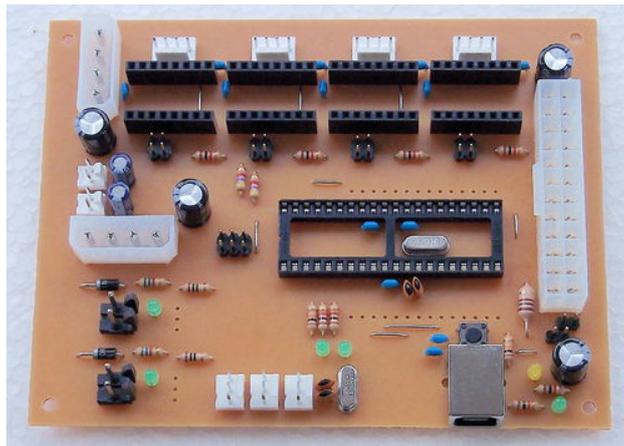


Figura 2.12: Microcontrolador Gen7.

- Smoothieware. Es una tarjeta todo en uno que cuenta con el microcontrolador, conexiones necesarias para finales de carrera, sensores de temperatura y etapas de potencia para la conexión de motores. Esta tarjeta microcontroladora está diseñada para funcionar con el circuito integrado cortex M3 LPC1769 o el LPC1768. Se trata de una tarjeta de código abierto y está pensada para funcionar con el firmware con el mismo nombre. Puede utilizarse en impresoras 3d, cortadoras láser, fresadoras de control numérico, o pequeñas máquinas de control numérico. Se puede utilizar en máquinas de mayores dimensiones utilizando amplificadores externos. Tiene salida para 5 motores de pasos y se puede acceder a señales de interés, como la señal de paso y dirección que actúan sobre los motores. Puede controlar motores de potencia importante, hasta 35V y 2A por motor bipolar. Dispone también de alojamiento para tarjetas SD y conexión USB. Como diferencia en respecto a otras etapas de amplificación, dispone de control digital de corriente independiente para cada motor, en lugar de utilizar potenciómetros.

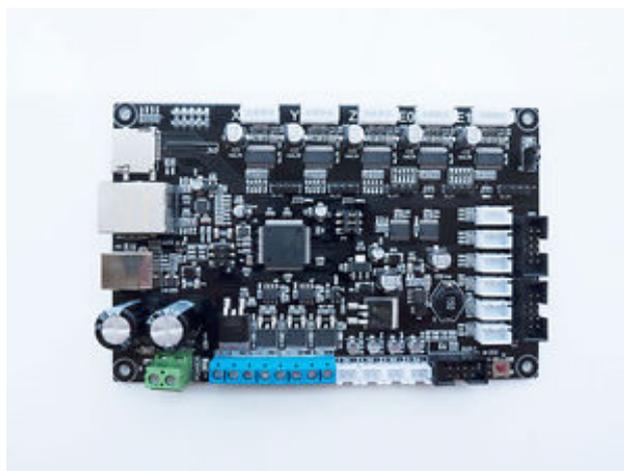


Figura 2.13: Tarjeta de control de la impresora Smoothieware.

## 2.8. Etapa de potencia

Es habitual que los controladores, tipo Arduino Mega, RaspBerry Pi o STM32F4 discovery dispongan de gran cantidad de pines de entrada salida. Pero estos pines tienen una tensión de salida de 3.3 a 5V, en función del modelo y una corriente de salida máxima de 4 a 20mA. Por ello, para controlar motores con tensiones de alimentación de 12 o 24V e intensidades de 1 o 2A

es necesaria la utilización de etapas de potencia. A continuación se presentan algunas de las etapas de potencia que se pueden utilizar para el control de motores.

**Controlador EasyDriver:** Esta etapa de potencia puede funcionar con corrientes de hasta 750mA por fase. Por defecto funciona con 8 micropasos por paso, parámetro que puede modificarse conectando los pines MS1 y/o MS2 a masa. Esta etapa de amplificación se basa en el integrado Allegro A3967. Su voltaje máximo de funcionamiento es de 30 Voltios. Las tarjeta de control de motores Easydriver funciona con señales de paso y dirección. **PaperStepper(L293D):** Esta etapa de amplificación cuenta con un inte-

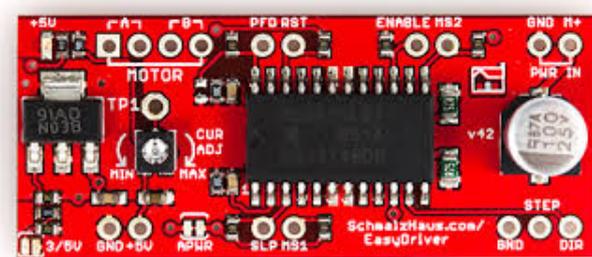


Figura 2.14: Tarjeta amplificadora Easydriver.

grado L293D capaz de controlar motores de hasta 600mA por fase. Funciona con tensiones de hasta 36V y a diferencia de circuitos anteriores, este debe ser accionado con las señales de los motores directamente.

**Pololu A4988:** Este dispositivo es bastante utilizado en control de motores para impresión 3d. Está basado en el circuito integrado Allegro A4988. Dispone de limitadores de corriente y también protección térmica. Su resolución de trabajo es de hasta 16 micropasos por paso y funciona con señales de paso y dirección. Puede controlar motores con hasta 1A por fase si el dispositivo no tiene disipador y hasta 2A por fase si cuenta con disipador y una buena refrigeración. Se puede limitar la corriente máxima mediante el potenciómetro de la placa.

## 2.9. Actuadores

Los actuadores son todos aquellos componentes hardware que son capaces de interactuar con el medio físico. Son ejemplos de actuadores una bomba de agua, un ventilador, un altavoz o un motor. A continuación se describen diferentes tipos de motores eléctricos, así como los extrusores y camas calentadoras que se utilizan en la impresión 3d.

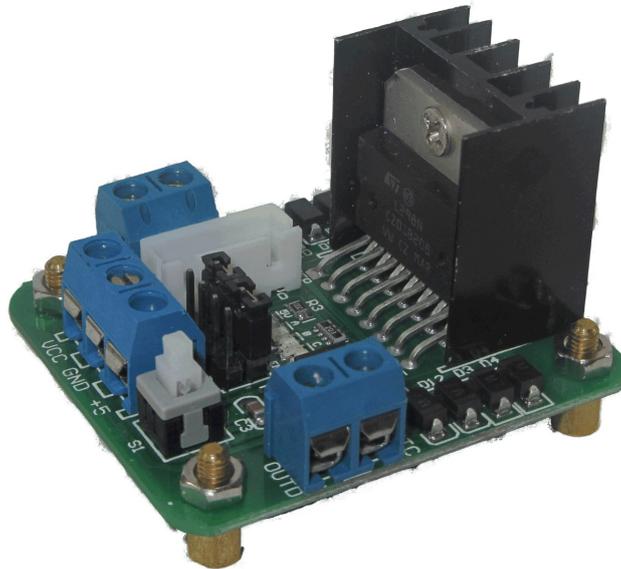


Figura 2.15: Tarjeta amplificadora Paperstepper.



Figura 2.16: Tarjeta amplificadora Pololu.

**Motores paso a paso.** En este tipo de motores cuando se recibe un impulso de corriente se transforma en un movimiento preciso del eje, conocido como paso. Según la secuencia de pasos empleada, el motor puede girar en un sentido o en el otro. La velocidad de giro del motor depende de la frecuencia en la que se le aplican los impulsos. Las ventajas de estos motores son:

- Son compatibles con controladoras digitales.
- Resultan ideales para trabajar en lazo abierto.
- Posicionamiento preciso de entre 3-5 % con buena repetibilidad.
- Bajo mantenimiento, no tienen escobillas.

- Los circuitos de control y excitación son sencillos.

Por contra tienen la desventaja de presentar resonancias en su funcionamiento, especialmente si no se controlan adecuadamente. Otra desventaja que tienen es que es difícil operar a altas revoluciones, porque se pierden los pasos. Estos motores se utilizan en robótica, automatización, pequeños electrodomésticos, telecomunicaciones, etc.

**Motores corriente continua con escobillas.** Los motores de corriente continua son máquinas eléctricas que convierten la energía eléctrica en mecánica, provocando el giro del eje del motor, gracias a los pares generados por el campo magnético. Un motor de corriente continua consta principalmente de dos partes. El estator da soporte mecánico al motor y contiene los devanados principales de la máquina, conocidos también como polos. Los polos pueden ser imanes permanentes o devanados con hilo de cobre sobre núcleo de hierro. El rotor es generalmente con forma cilíndrica, también devanado y con núcleo. El motor se alimenta a través de las escobillas de forma directa.

**Motores corriente continua sin escobillas.** Los motores de corriente continua sin escobillas tienen la ventaja de que al no tener escobillas, no hay que cambiarlas, por lo que la duración de estos motores es mayor. Son más silenciosos que los motores convencionales y pueden girar a mayor velocidad. Su control es complejo, hay que enviar señales de conmutación perfectamente sincronizadas, pero gracias a la electrónica de control disponible en la actualidad no es difícil controlar este tipo de motores.

**Extrusores.** El extrusor es el elemento que transforma el filamento de 1.75 o 3mm en un filamento de 0.2 a 0.4mm en función del extrusor. Es el filamento de menor tamaño el que se deposita capa a capa para conformar la pieza. Los extrusores se calientan con ayuda de resistencias eléctricas. El voltaje de alimentación de los extrusores es de alrededor de 12V y su potencia es de unos 40W. Los extrusores de impresora 3d disponen de un orificio en el que se introduce un sensor de temperatura (termistor) que se utiliza para mantener la temperatura controlada.

**Cama calentadora.** La cama calentadora consiste en un circuito impreso que tiene una resistencia, cuando se le hace pasar una corriente, la resistencia se calienta y desprende calor, según el efecto Joule. La necesidad de tener una superficie caliente viene dada de que algunos materiales, como el PLA, se pegan mejor a una superficie caliente que a una que está a temperatura ambiente. La corriente que circula por una cama caliente puede llegar a los 10A.

## 2.10. Lenguajes de programación de M-H

El lenguaje de programación de la máquina-herramienta es la herramienta que permite al usuario programar la máquina. Utilizando estos lenguajes, el usuario puede programar una secuencia de trayectorias, conectar o desconectar un ventilador, o decidir con que extrusor quiere imprimir. A continuación se muestran algunos de los modelos de programación existentes.

### 2.10.1. Código ISO

El código ISO surge a los inicios de la invención de máquinas herramientas controladas por ordenador. En estos momentos los fabricantes de máquinas-herramientas conocido como G y M, que son las siglas en inglés «Genuine and Miscelaneous». El problema que existía es que al no haber un código estandarizado, cada fabricante codificaba su máquina a su manera, por lo que era muy complicado utilizar el programa de una máquina en otra sin tener que reescribirlo prácticamente por completo. A partir de la implantación del código ISO la mayoría de fabricantes implementan en sus máquinas la programación ISO, y ya es posible pasar un programa de una máquina a otra directamente o con ligeras modificaciones.

Los programa escritos en ISO se estructuran en bloques. Cada bloque puede contener uno o más de los siguientes comandos.

N G X Y Z F S T M
-------------------

- «N»: Son etiquetas que se pueden utilizar para navegar por el programa o repetir parte del código.
- «G»: Los comandos G son funciones generales directamente relacionadas en la programación de la máquina. Entre ellas encontramos por ejemplo G00 que indica desplazamiento rápido, G01, G02 y G03 que son desplazamiento rectilíneo y curvo. Las funciones G40, G41, G42 y G43 tienen que ver con la compensación de la herramienta en altura y en diámetro. La función G53 indica que las coordenadas introducidas a continuación se encuentran en función del sistema de referencias de la máquina y las funciones G54 a G59 son sistemas de coordenadas definidos por el usuario. Se introduce la función G90 cuando las coordenadas se introducen de forma absoluta y la función G91 cuando se introducen de forma incremental.

Tabla 2.1: My caption

Función	Descripción
G00	Posicionamiento rápido
G01	Interpolación lineal
G02	Interpolación Circular en sentido horario
G03	Interpolación Circular en sentido antihorario
G04	Temporización
G10	Anulación imagen espejo
G11	Imagen espejo en eje X
G12	Imagen espejo en eje Y
G13	Imagen espejo en eje Z
G17	Plano XY
G18	Plano XZ
G19	Plano YZ
G31	Guardar origen de coordenadas
G32	Recuperar origen de coordenadas
G40	Anulación de compensación de radio
G41	Compensación de radio a la izquierda
G42	Compensación de radio a la derecha
G43	Compensación de longitud
G44	Anulación de compensación de longitud
G53	Traslado de origen a coordenadas máquina
G54	Traslado de origen
G70	Programación en pulgadas
G71	Programación en milímetros
G73	Giro de sistema de coordenadas
G74	Búsqueda de cero máquina
G90	Programación en absolutas
G91	Programación en incrementales
G93	Coordenadas polares
G94	Avance en mm/min
G95	Avance en mm/rev

- «X Y Z»: son las coordenadas destino de la herramienta, también pueden introducirse otras coordenadas, como A y B si la máquina dispone de más ejes o coordenadas auxiliares como I J o K para definir el centro de las trayectorias curvas.
- «F»: Es el avance de trabajo de la máquina. Se puede introducir en

milímetros por minuto o en milímetros por revolución del cabezal en función del comando G introducido, milímetros por minuto si es G98 y milímetros por revolución si se trata de G99. También se puede programar la máquina en pulgadas si se introduce la función G20. Para programarla en sistema métrico se introduce la función G21.

- «S»: Indica la velocidad de giro del cabezal. Existen dos posibilidades para introducir la velocidad de giro, puede hacerse en metros por minuto si previamente se ha introducido la función G96. También es posible introducir la velocidad de giro en revoluciones por minuto si previamente se ha introducido la función G97.
- «T»: Indica el número de herramienta, puede utilizarse para cambiar la herramienta si la máquina dispone de cambiador automático de herramientas y también para llamar a la herramienta siguiente si la máquina dispone de esta funcionalidad.
- «M»: Se denominan funciones auxiliares y realizan acciones diversas no incluidas en ninguno de los bloques anteriores. Por ejemplo las funciones M3 y M4 gestionan el giro del cabezal a izquierda y a derecha. Las funciones M8 y M9 inician y detienen la bomba de refrigerante. Para finalizar un programa se suele utilizar la función M2 o M30.

Tabla 2.2: My caption

Función	Descripción
M00	Parada de programa
M01	Parada condicional del programa
M02	Final del programa
M03	Arranque del husillo en sentido horario
M04	Arranque del husillo en sentido anti-horario
M05	Parada del husillo
M06	Cambio de herramienta
M08	Arranque bomba de refrigerante
M09	Paro bomba de refrigerante
M30	Final de programa con reseteo de variables

En este modelo de programación, el operario debe introducir cada movimiento a realizar por la máquina, por lo que es un sistema de bajo nivel.

## 2.10.2. Programación APT

APT se trata de un lenguaje de alto nivel para programar máquinas de control numérico [1]. Se encarga de calcular la trayectoria que debe seguir la herramienta para obtener la pieza deseada. APT fue creado entre los años 1950 y 1960 para simplificar la tarea de calcular la geometría de puntos requeridos en la industria aeroespacial. Fue el resultado directo de la nueva tecnología CNC que aparecía en aquel tiempo y solucionaba en gran medida el desafío de calcular los movimientos que debían hacer las máquinas para obtener piezas complejas. APT fue creado antes de la aparición de los interfaces gráficos, por lo que se programa en texto plano para especificar la geometría y los caminos de la herramienta. Algunas características sorprendentes del lenguaje APT es que es capaz de controlar máquinas de hasta 5 ejes de forma continua permitiendo a la herramienta seguir la trayectoria en todo momento con la inclinación deseada. Permite definir 7 tipos de forma de herramienta, como esférico, toroidal, con forma de radio, cola de milano o de achafanar entre otras.

## 2.10.3. Lenguajes propietarios

Existen multitud de lenguajes propietarios, por lo general cada fabricante de controladores implementa el suyo propio, aunque también suelen funcionar con el código ISO. La principal ventaja de programar en código propietario suele ser que los fabricantes ofrecen facilidades para obtener las trayectoria de las piezas, como son por ejemplo ciclos de mecanizado estándar. Algunos de los lenguajes de programación propietarios existentes en el mercado son:

- Fagor.
- Fanuc.
- Heidenhain.
- Siemens.

## 2.11. Generadores de código

### 2.11.1. GRBL

GRBL [2] es un programa informático gratuito de código abierto de alto rendimiento para controlar el movimiento de máquinas que trasladan o fabrican cosas y funciona directamente sobre una placa Arduino. La mayoría de impresoras 3D de código abierto funcionan gracias a GRBL.

GRBL está optimizado para funcionar con las funciones generadas por los generadores de código G tipo Cad cam, por lo tanto no incluye la posibilidad de programar utilizando variables, ciclos de mecanizado o bases de datos de herramientas. Actualmente GRBL presenta su versión 0.9, la cual incluye mejoras y características profesionales, como son por ejemplo:

- Algoritmo de pasos más suave.
- Habilita una salida PWM para controlar la velocidad del husillo a través del parámetro S del código G.
- La aceleración de los ejes puede ser introducida independientemente y GRBL calculará automáticamente la aceleración máxima y velocidad en la trayectoria en función de la dirección solicitada.
- GRBL comprueba si algún bloque de programación excede los límites de trabajo de la máquina antes de ejecutar el bloque, mostrando una señal de alarma en caso de tener programada alguna coordenada fuera de la máquina.
- El programa controla las trayectorias circulares más rápido cuanto mayor es el radio de estas, de este modo escala la longitud del segmento para que el máximo error radial sea siempre menor que la tolerancia establecida, que por defecto es 0.002mm.
- Incorpora un simulador para ejecutar el código G en un ordenador sin necesidad de arduino y obtener la respuesta como si se dispusiese de uno.
- Ofrece la posibilidad de trabajar con la estructura llamada coreXY. Esta configuración de mesa de trabajo tiene la ventaja de ofrecer mayor velocidad del actuador, gracias a una disminución de las inercias del conjunto de piezas en movimiento.
- Interruptores de seguridad para puertas, GRBL detiene el avance de la máquina, la rotación del husillo y el refrigerante hasta que la puerta se cierra de nuevo y el operario rearma la operación.

### **2.11.2. linuxCNC**

LinuxCNC [3] es un programa informático de código abierto. Las versiones actuales de LinuxCNC se distribuyen bajo la licencia «GNU General Public Licence» y «Lesser GNU General Public License». LinuxCNC es un programa

informático para utilizar ordenadores como si se trataran de fresadoras o tornos. LinuxCNC ofrece:

- Diferentes interfaces gráficos de usuario, entre ellos uno para pantallas táctiles.
- Un intérprete para «G-code», el lenguaje de programación para máquina herramienta RS-274.
- Un sistema de planificación de movimientos en tiempo real con sistema «Look ahead», que permite leer un determinado número de bloques por delante del que se está ejecutando. De este modo el controlador es capaz de cambiar la ganancia del controlador de posición y velocidad para ejecutar una trayectoria que sea más favorable a la dinámica de los ejes.
- Operación de sistemas electrónicos de bajo nivel, como sensores o drivers de motores.
- Una capa de conexión sencilla de utilizar para conseguir rápidamente una configuración única para la máquina.
- Un programa informático PLC programable con diagramas de escalera.
- Fácil instalación con paquetes .deb o live-CD.

LinuxCNC dispone además de las siguientes características:

- Puede controlar simultáneamente hasta 9 ejes y soporta variedad de interfaces.
- El control puede operar tanto con servos analógicos como PWM con el lazo de realimentación cerrado y también con motores de pasos.
- Las características del control de movimiento incluyen compensación de radios y de longitud, desviación de trayectoria limitada a una tolerancia específica, roscado sincronizado para el torno, sincronización de ejes, avance de trabajo adaptativo, control de avance por el operario o control de velocidad constante.
- Soporta sistemas de movimiento no cartesianos a través de módulos cinemáticos personalizados. Dispone de arquitecturas como hexápodos y sistemas con uniones rotativas para controlar el movimiento de robots como el PUMA o el SCARA.

- Utiliza la extensión de tiempo real de Linux y actualmente soporta las versiones 2.4 y 2.6 del kernel de Linux con las extensiones de tiempo real aplicadas por RT-Linux o el parche RTAI.

## 2.12. Identificación y control de procesos

Desde el punto de vista del control, un proceso se define como una operación o conjunto de pasos con una secuencia determinada, que producen una serie de cambios graduales que llevan de un estado a otro, y que tienden a un determinado resultado final. Se denomina proceso a cualquier operación que se vaya a controlar. Ejemplos de ellos son procesos químicos, mecánicos, biológicos, etc.

Los motivos por los cuales se quieren controlar los procesos son muy amplios y pueden variar en función de cada proceso. Por ejemplo nos puede interesar controlar la temperatura de un horno de cerámica para controlar la calidad del producto acabado. En otros casos, se decide controlar un proceso para mejorar la productividad de una planta, o también para evitar que las personas realicen tareas pesadas y repetitivas.

Cuando se trata de procesos bien conocidos, se pueden realizar identificaciones teniendo en cuenta las propiedades físicas del proceso, pero en la mayoría de procesos reales, aparecen conjuntos de procesos con multitud de variables y parámetros que hacen que se utilicen otros métodos para la identificación del proceso. Estos métodos son por ejemplo la obtención del modelo del proceso mediante la utilización de métodos empíricos, como son la aplicación de señales.

- Respuesta de escalón. Es un método de identificación de sistemas experimental mediante el cual se obtiene de forma experimental la respuesta del sistema a una entrada, en este caso de escalón. Observando la salida del sistema y con ayuda de la experiencia se puede determinar el tipo de sistema que se está identificando, así como su ganancia. La Figura 2.17 muestra la respuesta de un sistema de primer orden ante una entrada de escalón. La Figura 2.18 muestra la respuesta de un sistema de primer orden con integrador y la Figura 2.19 muestra la respuesta de un sistema de segundo orden. En el caso del sistema de primer orden,  $\frac{Kg}{as+1}$ , se obtiene el valor de  $Kg$  como la división entre el valor en régimen permanente del sistema y la magnitud de la señal de entrada. La constante de tiempos  $a$  la podemos estimar como el tiempo transcurrido desde que se aplica el escalón de entrada hasta que el valor de salida alcanza el 67 % del valor final. En el caso del sistema de primer orden

con integrador  $\frac{Kg}{s(as+1)}$ , se calcula la ganancia como la división entre la pendiente de salida en régimen permanente y la magnitud de la señal de entrada. La constante de tiempos de este tipo de sistema se calcula midiendo el tiempo transcurrido entre un sistema que se comporta como integrador ideal y el sistema analizado.

- Respuesta sostenida. Es un sistema de ajuste de controladores que se basa en un método experimental. También es conocido como el método de Ziegler-Nichols. El método para obtener los parámetros del controlador es el de montar el sistema en un bucle cerrado en el que se disponga de un amplificador de la señal de error. Para una señal de entrada constante, se incrementará el valor de amplificación hasta llevar al sistema a un comportamiento marginalmente estable. En este momento se toma nota del valor de la ganancia. El otro parámetro necesario para el ajuste del controlador con este método es el periodo de resonancia de la señal de salida en régimen permanente.
- P El control proporcional es el más sencillo de los controladores básicos. Su acción de control es proporcional a la señal de error entre la señal de referencia y la salida. Su ventaja es que es fácil de implementar y sus desventajas que no ofrece mejoras ni en el régimen transitorio ni en el régimen permanente.
- PD El control proporcional-derivativo presenta mejoras en el régimen transitorio con respecto al control proporcional. Su acción de control es proporcional al error y también a la velocidad con la que este cambia. Ventajas de este controlador son un transitorio más amortiguado y rápido. Las desventajas son que pueden convertir el sistema en inestable si existen perturbaciones (vibraciones) en el sistema y que no ofrece cancelación de error frente a entradas finitas..
- PI El control proporcional-integrador presenta mejoras en el régimen permanente del sistema con respecto al controlador proporcional, concretamente elimina el error en régimen permanente.
- PID Este es el controlador que se utiliza cuando es necesario una mejora de la respuesta dinámica del sistema y también de la respuesta estática.
- Control reseteado. El control reseteado es una técnica de control que pretende superar las limitaciones fundamentales de los controladores lineales tradicionales [6]. Lo que se hace en el control reseteado es poner a cero o a cualquier otro valor una parte o por completo el estado.

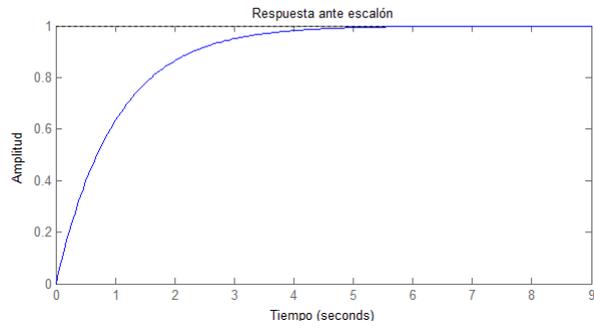


Figura 2.17: Respuesta de sistema de primer orden.

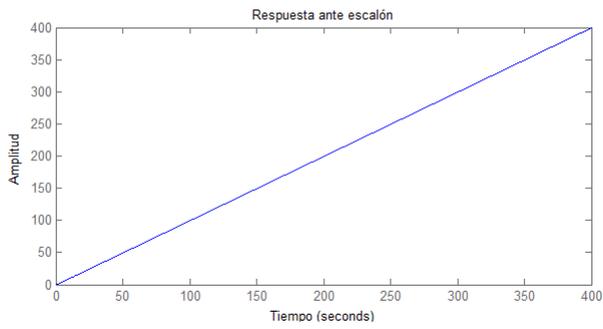


Figura 2.18: Respuesta de sistema de primer orden con integrador.

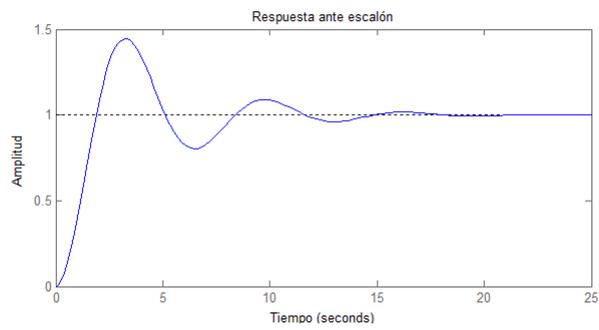


Figura 2.19: Respuesta de sistema de segundo orden.

# Capítulo 3

## Desarrollo práctico

En el presente capítulo se explica la metodología empleada para modificar la impresora para que en lugar de utilizar motores paso a paso para sus movimientos en los ejes X e Y, se utilicen motores de corriente continua. En primer lugar se describe la máquina, una impresora 3d Prusa I3 del proyecto RepRap. A continuación se definen los componentes software y hardware y se justifica el motivo de su elección. Seguidamente se listan los componentes necesarios para la nueva implementación con motores de continua y se justifica el motivo de elegir estos elementos. Montados los motores en la máquina, se explican las herramientas empleadas para la obtención del modelo. A continuación se indica como se ha implementado el control de los motores. En el siguiente apartado se realizan las pruebas del sistema y se muestran los primeros resultados. Estos resultados se analizan y se ajusta de nuevo el control de los motores para obtener un control más acorde a las necesidades del sistema. Por último se muestran las piezas impresas utilizando los motores de continua.

### 3.1. Impresora Prusa I3

#### 3.1.1. Descripción impresora.

La impresora que se va a utilizar para el desarrollo del presente trabajo es la impresora Prusa I3. Esta impresora es la tercera iteración de impresoras desarrolladas por Josef Prusa. La máquina tiene una capacidad de impresión de 200x200x200mm, aunque se trata de una máquina de diseño paramétrico, esto es que está diseñada para que los usuarios puedan fabricarla en función de sus necesidades. El chasis de la máquina está construido con chapa de aluminio de 6mm, varilla roscada de 10mm y piezas impresas con impresora

3d. El tipo de impresora es cartesiano, y cuenta con 5 motores para accionar los movimientos de la máquina. Un motor para el eje X, otro motor para el eje Y, dos motores para el desplazamiento en el eje Z y un motor para extruir el hilo fundido. Tres de los motores se encuentran fijados al chasis de la máquina, 2 del eje Z y uno del eje Y, por lo que el peso de estos motores no afecta a las inercias de las piezas en movimiento. En cambio el motor del eje X se encuentra ubicado en el eje Z, por lo que tiene un movimiento de subida y bajada. La inercia de este motor no debe preocuparnos en exceso, puesto que el eje Z se mueve en incrementos de baja dimensión a poca velocidad. Por último el motor del extrusor se encuentra colocado en el eje X y si que afectará con su peso a todos los movimientos del eje X. Se ha utilizado esta impresora porque es el resultado de un proyecto de código abierto, por lo que se dispone de todos los planos y algoritmos de la máquina. Por otra parte la mecánica de la máquina era la apropiada para el desarrollo del trabajo. Además esta impresora me la podía facilitar mi director de proyecto Miguel Sanchez.

El elemento de control principal de la impresora es una controladora Arduino Mega, la cual tiene acoplada una tarjeta de control de motores RAMPS. Esta tarjeta cuenta con 4 etapas de potencia Pololu A4988 que se utilizan para la alimentación de los 5 motores. Los dos motores que controlan el movimiento del eje Z se accionan desde la misma tarjeta Pololu. Los motores utilizados en la impresora son los NEMA 17, utilizados en frecuencia en pequeñas máquinas de control numérico e impresoras 3d. Estos motores tienen un par a bajas revoluciones de  $0.35\text{N}\cdot\text{m}$  cuando están alimentados a 12V y tienen una corriente media de 2A. La máquina también cuenta con sensores finales de carrera en los 3 ejes.

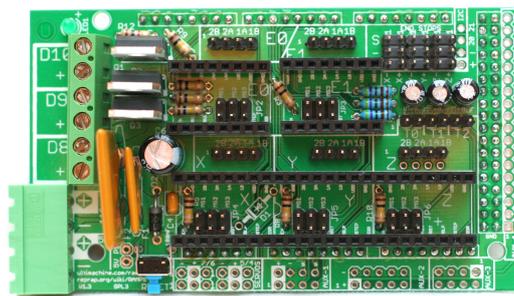


Figura 3.1: Tarjeta de control de la impresora 3D RAMPS.

**RAMPS:** Es la tarjeta que recibe las órdenes de movimiento del PC para controlar las señales que se envían a los motores (ver Figura 3.1). La tarjeta RAMPS dispone de salidas para hasta 5 motores paso a paso. También

tiene salidas para accionar el calentador del extrusor o la cama calentadora, así como su temperatura. La tarjeta RAMPS envía en cada momento las señales correctas a unas pequeñas etapas de potencia para amplificar la señal posteriormente y pilotar los motores paso a paso. Realmente lo que se envía a las etapas de potencia son los pasos que el motor debe rotar, así como el sentido de giro. De este modo, cuando se requiera de una velocidad más alta, lo que se hará es enviar más pasos en menos tiempo para que el motor realice más giros.

**Motores de pasos** Los motores de pasos son transductores electro-mecánicos, dispositivos capaces de transformar energía eléctrica en mecánica. La principal diferencia con otros tipos de motores eléctricos es que el movimiento de giro de su eje es discreto. El giro del eje depende de una sucesión de pasos o saltos discretos. Esta característica hace que este tipo de motores resulte ideal para el control de multitud de dispositivos de bajo coste, puesto que conociendo las señales que se le envían al motor, es posible conocer la orientación del eje.

Tal y como se ha comentado anteriormente en este trabajo, la impresora dispone originalmente de motores paso a paso (Ver Figura 3.2. Este tipo de motores era encontrado con frecuencia en máquinas industriales guiadas por ordenador. Los motivos por los que los diseñadores de las impresoras de 3D los eligen son, probablemente idénticos a los que construyeron máquinas de control numérico en el pasado:



Figura 3.2: Motor de pasos Nema 17.

- Facilidad para realizar el control.

- Realimentación no necesaria.
- Par de rotación máximo a bajas velocidades.

Sin embargo los fabricantes de maquinaria industrial dejaron progresivamente de utilizar motores paso a paso para utilizar servomotores. Los motivos por los que los motores paso a paso dejaron de ser utilizados fueron:

- Pérdida de pasos.
- Generación de vibraciones en la estructura.
- Baja velocidad máxima de rotación.
- Bajo par de rotación a altas revoluciones.

**Motor de corriente continua.** El motor de corriente continua que se ha utilizado en este proyecto es el Mabuchi 370 (Ver Figura 3.3). Este motor tiene un voltaje nominal de 12V y una velocidad nominal de 5600 revoluciones por minuto sin carga. El par de este motor es de 2.48mN·m.



Figura 3.3: Motor de CC Mabuchi 370.

**Microcontrolador Mapple mini.** Como se ha descrito anteriormente, se va a utilizar un microcontrolador para controlar la posición de los motores. Se necesita un microcontrolador con 4 entradas digitales con interrupción hardware para controlar la posición de los dos motores, 2 entradas por encoder. También se necesitan 4 entradas digitales para controlar las señales de paso y dirección de control de los motores, 2 de ellas con interrupciones hardware. Por último se necesitan 4 salidas PWM para controlar los dos motores. El microcontrolador debe ser económico para cumplir el objetivo del proyecto, de ser una solución económica para impresoras 3d domésticas.

Se ha elegido el microcontrolador Mapple mini. Tiene un potente microcontrolador ARM Cortex M3 de 32 bits. Dispone de 34 pines de entrada-salida digital, 9 de ellas con función PWM hardware. Este dispositivo tiene un IDE de programación basado en Arduino, que es compatible con las librerías escritas para él. El microcontrolador Mapple cumple con los requisitos técnicos y también con el requisito económico, pues cuesta unos 4 euros.

### 3.1.2. Software.

El firmware programado en la tarjeta Arduino Mega es el Marlin, que se comunica con el Pronterface instalado en el ordenador. El programa de rebanado utilizado es el Slicer y para la obtención de modelos se utilizarán modelos compartidos en la red.

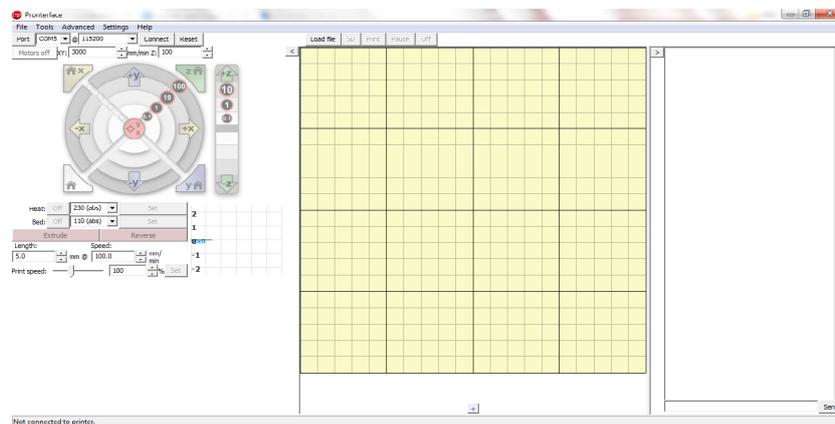


Figura 3.4: Programa informático PronterFace.

**Pronterface:** Es un programa de código abierto creado por Kliment Yanev. Este programa convierte un ordenador en un controlador de máquina CNC. Dispone de una interfaz gráfica, desde la cual es posible operar la máquina y monitorizar el estado de la misma.

**Slic3r:** El proyecto nació en 2011 dentro de la comunidad RepRap de la mano de Alessandro Ranellucci, su autor fundador. Slic3r ofrece al usuario la posibilidad de generar el archivo de configuración de la máquina, en código G, a partir de un archivo en el que se encuentra la geometría de la pieza y los parámetros definidos por el usuario. Pero Slic3r continúa evolucionando gracias al aporte de centenares de usuarios que plantean mejoras y prueban el programa en sus propias máquinas de impresión. Slic3r ha sido pionero en ofrecer algunas herramientas en su programa informático, como son:

- El uso de múltiples extrusores.

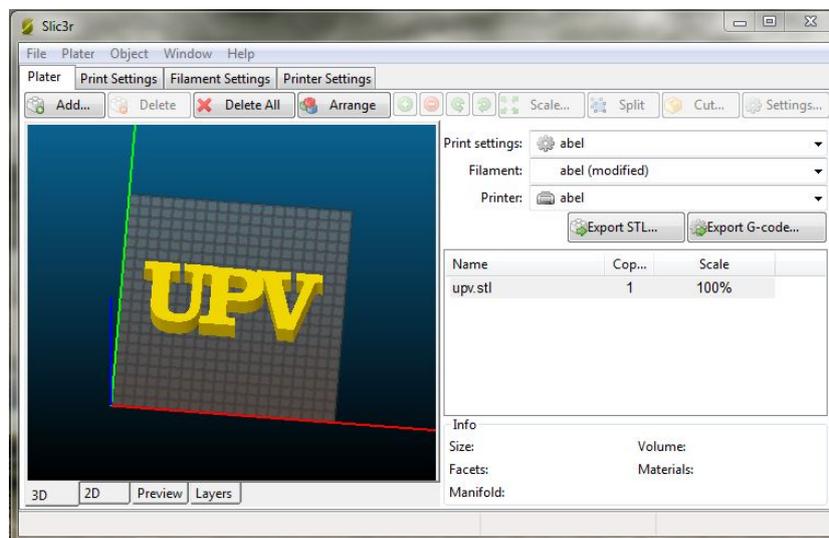


Figura 3.5: Programa informático Slic3r.

- Brim, Expandir unos milímetros la primera capa para mejorar el pegado de la pieza.
- Microcapas, consiste en imprimir las partes exteriores de la pieza con un tamaño de capa menor al del relleno de la pieza.
- Detección de puentes y voladizos.
- Aplicación de consola.
- Capas de altura variable.
- Impresión secuencial, un objeto cada vez.
- Relleno de colmena de abeja.
- Corte y pegado de modelos 3d.
- Seccionado de modelos en partes.
- Admite archivos en formato AMF (Additive Manufacturing File)
- Impresión con extrusores de diferentes tamaños.
- Otros.

Las características mencionadas anteriormente están ahora presentes en la mayoría de programas de impresión comerciales.

**Marlin** El software Marlin es un programa para controlar impresoras Reprap utilizando controladores mononúcleo. Marlin soporta impresión desde archivos y desde tarjeta SD y tiene planificación de trayectoria «look-ahead», que significa que lee algunos bloques por delante del que está ejecutando. Este proyecto tiene la licencia GNU GPL v3. Está basado en Sprinter y en Grbl, aunque se le han añadido muchas funcionalidades. Alguna de sus características son:

- Movimiento basado en interrupciones con aceleración lineal real.
- Generación de pasos a alta velocidad.
- «Look ahead», lee bloques por adelantado, permitiendo alta velocidad de lectura y alta velocidad en las esquinas.
- Interrupción en caso de sobrecalentamiento, ideal para capas con poco material fundido.
- Soporte para finales de carrera.
- Soporte para SD y sistema de archivos en SD.
- Soporta pantalla LCD, idealmente 20 x 4 líneas o 128 x 64 pixeles.
- Menú de impresión autónoma desde LCD con encoder de selección y botón.
- Almacenamiento de parámetros de la máquina en memoria EEPROM.
- Ajuste dinámico de la temperatura para impresión a velocidades variables.
- Envío al ordenador de las señales de fin de carrera y de la temperatura del extrusor y cama calentadora.
- Permite trayectorias circulares, lo cual disminuye la cantidad de bloques a enviar.

## 3.2. Modificación de la impresora (hardware)

A continuación se detallan las modificaciones que se van a hacer sobre la máquina.

**Modificaciones mecánicas:** Se desmontan los motores de pasos de los ejes X e Y y se instalan los nuevos motores de continua, utilizando unos soportes impresos en 3d fabricados anteriormente para este propósito. Se ajustará la tensión de las correas si es necesario.

**Modificaciones electrónicas:** Los motores de pasos se controlan a través de impulsos eléctricos, pero estos impulsos no son generados directamente por la tarjeta de control Arduino, en su lugar, se envían las señales de paso y dirección a la etapa de potencia Pololu. Es el Pololu el que convierte las señales de paso y dirección en los impulsos para controlar los motores de pasos. En este proyecto se retiran los Pololus que controlan los motores X e Y. Se toman las señales de paso y dirección de ambos motores y se envían a un microcontrolador. En el microcontrolador se programa un bucle de control que tiene como referencia las señales de paso y dirección recibidas. La acción de control de este microcontrolador se envía por PWM a la etapa de potencia que controla los motores de continua.

**Modificaciones firmware:** Los parámetros del firmware Marlin pueden modificarse en tiempo de ejecución, en el parámetro pulsos por milímetro definido para los motores paso a paso se introducen los pulsos generados por el encoder cuando el carro avanza 1 milímetro. El código para el microcontrolador que controla los motores de continua se escribe para que registre los pasos enviados por el Arduino Mega y envíe las señales de control a los motores de CC.

La sustitución de los motores originales paso a paso por los nuevos motores de continua requiere la utilización de técnicas mecánicas, electrónicas, informáticas y de control automático. Cuando se retiran los motores de pasos se puede ver que no tienen la misma forma y dimensiones que los motores de continua. Por lo tanto es necesario diseñar y fabricar soportes para mantener en su lugar los nuevos motores. En este caso no ha sido necesario diseñar y fabricar estos soportes porque mi tutor Miguel Sánchez ya disponía de soportes fabricados para experimentos anteriores, y me los ha proporcionado. Se montaron entonces los nuevos motores de continua a los soportes fabricados con otra impresora 3D y estos se atornillaron a la máquina. Una vez realizado este paso, puede ser necesario tensar las correas dentadas que transmiten el movimiento rotatorio del eje del motor al movimiento lineal de la mesa de trabajo. Esto se debe bien a que la nueva polea tiene un diámetro diferente al anterior y/o a que la posición del nuevo motor no es la misma que la del antiguo motor. Esto es lo que ocurrió en el eje Y, donde la posición

del motor no coincidía con la del motor anterior, por lo que fue necesario desmontar el cierre de la correa para tensarlo correctamente. En la Figura 3.6 se puede ver como queda el motor que acciona el eje X de la impresora. En la Figura 3.7 se aprecia el montaje final del motor que acciona el eje Y de la máquina. Para realizar el control de posición de los ejes, es necesario

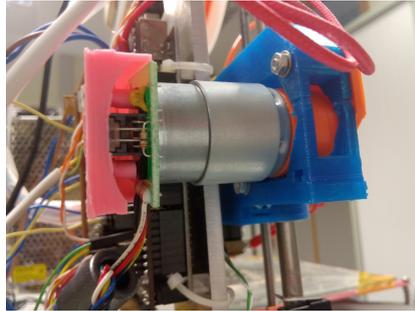


Figura 3.6: Montaje del motor del eje X.

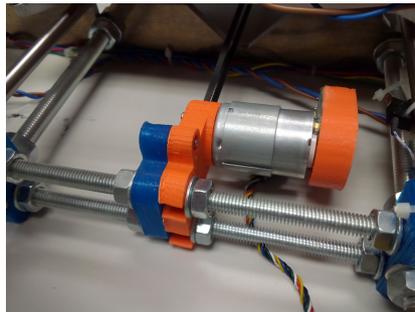


Figura 3.7: Montaje del motor del eje Y.

obtener las señales de paso y dirección que proporciona la tarjeta RAMPS. Para ello se retira la tarjeta de control de motores paso a paso Pololu y se obtienen mediante cables las señales de paso y dirección de los pines 7 y 8. La conexión entre la tarjeta RAMPS y el microcontrolador Mapple no puede realizarse directamente porque las tensiones de funcionamiento no coinciden. La tarjeta RAMPS funciona con tensiones de entre 0 y 5V mientras que el microcontrolador Maple funciona con tensiones de entre 0 y 3.3V. Para solucionar este problema se ha utilizado un divisor resistivo entre las dos tarjetas para las señales de paso y dirección de los dos motores. Para asegurar una correcta comunicación entre controladores es necesario conectar las masas de ambos dispositivos. La masa se encuentra en la tarjeta RAMPS en el pin 9. Con el fin de conocer la posición de los motores de continua, se conectan la

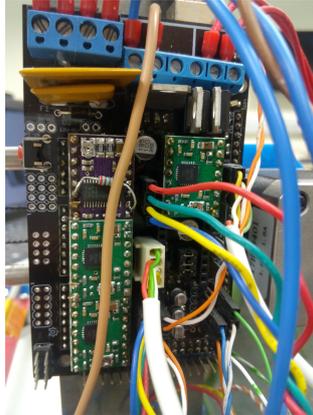


Figura 3.8: Conexión del microcontrolador a tarjeta RAMPS.

salida de los encoders a la entrada del microcontrolador. El microcontrolador Mapple, conectado convenientemente a la tarjeta de la impresora RAMPS, no es capaz por sí mismo de proveer la intensidad necesaria para accionar los motores de continua, por lo tanto es necesaria la utilización de una etapa de potencia. En este caso se ha utilizado una tarjeta que contiene un circuito integrado L298N. Este integrado tiene en su interior dos puentes en H que son capaces de controlar el sentido y velocidad de 2 motores de continua. Se conectan 4 salidas PWM (2 para cada motor) a las entradas de señal de la etapa de potencia. También se conectan las masas y finalmente se alimenta la etapa de potencia a través de una fuente de alimentación externa. La salida de esta tarjeta se conecta a los motores de continua. Al cambiar los motores, también es necesario revisar la configuración de la tarjeta RAMPS sobre los pasos que debe realizar para avanzar 1mm en cada uno de los ejes. Abriendo el programa Pronterface se puede ver al conectar la máquina como la configuración anterior era de 211.93 pulsos por milímetro en el eje X y 105.74 pulsos por milímetro en el eje Y. Aunque ahora no se van a instalar motores de pasos, se utilizará la lectura de los encoders para determinar la posición de referencia del motor. En este caso, los dos canales del motor generan 4 flancos por cada uno de los 448 pulsos que disponen. También se sabe que cada revolución hace avanzar el eje el número de dientes de la polea multiplicado por el paso de la correa. Por lo tanto:

$$448\text{pulsos} * 4\text{flancos} = 14\text{dientes} * 2\text{mm/diente}$$

$$\text{pulsos/diente} = 64$$

Se modifican los parámetros de la impresora introduciendo la función M92 X64 Y64. De este modo la controladora enviará 64 pasos por cada milímetro

de avance de la máquina requerido.

### 3.3. Identificación y diseño de controladores para motores CC

Con el fin de modelar el funcionamiento del motor de corriente continua, se monta el motor en un eje lineal que simulará el funcionamiento de un eje de la impresora 3D. El eje está fabricado con un perfil de aluminio ISO. La parte móvil del eje está impresa en 3d, así como el soporte del motor. El motor es el mismo que se utilizará más tarde para posicionar uno de los ejes de la impresora. El motor está controlado por un microcontrolador Maple mini (Fig. 3.9). El microcontrolador tiene a su salida una señal de modulación de



Figura 3.9: Microcontrolador Maple.

ancho de pulso (PWM) que acciona la etapa de potencia que a su vez controla el motor de continua. Se introducen distintas acciones de control al motor de continua y se lee a través del microcontrolador la posición del motor para estudiar su respuesta temporal (Ver Figura 3.10). Se espera que la respuesta del sistema se corresponda a la de un sistema de primer orden con integrador  $Sis = \frac{K_g}{s(s/a+1)}$ , donde  $K_g$  está asociado a la pendiente del sistema y  $a$  es la constante de tiempo del sistema. En la Figura 3.11 se puede ver las gráficas empleadas para el cálculo de la ganancia y de la constante de tiempo del sistema. Se puede calcular la ganancia del sistema calculando la pendiente de la rampa obtenida a la salida. Para obtener la constante de tiempos del sistema, dibujamos en la misma gráfica de la salida del sistema la respuesta de un integrador de pendiente igual al sistema. El valor de  $a$  del sistema equivale a la distancia en el eje de tiempos entre las dos señales. En esta etapa de

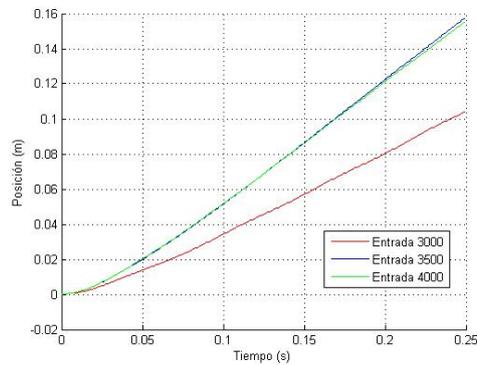


Figura 3.10: Respuesta del motor de continua.

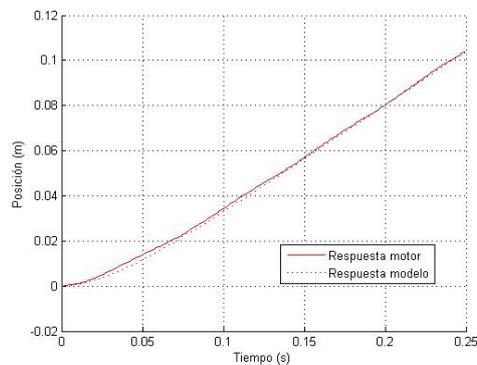


Figura 3.11: Modelo continuo del eje lineal.

la caracterización del motor se observa que el eje no empieza a moverse hasta que no se le aplica un determinado nivel de señal de entrada. Este fenómeno es conocido como zona muerta y está asociado a la fricción y otros fenómenos físicos. También se observa que el motor responde con la misma respuesta para entradas a partir de cierto nivel. Este comportamiento es conocido como saturación. A partir de los parámetros del sistema se simula un sistema para comprobar que la respuesta es similar a la obtenida en el eje. A continuación se obtiene el modelo discreto utilizando el método de discretización retenedor de orden cero (ZOH). Cuando se obtiene el modelo discreto de la planta se compara con los resultados obtenidos de la misma. Seguidamente se calculan los parámetros del controlador utilizando el método de Ziegler y Nichols en Lazo Cerrado o de la Oscilaciones sostenidas. Según este método el sistema debe configurarse en lazo cerrado e incrementar la ganancia progresivamente hasta que el sistema se encuentra en una situación marginalmente estable.

En la Figura 3.12 puede observarse el sistema en el momento en que la ganancia se ha establecido para que la salida se encuentre oscilando. En este

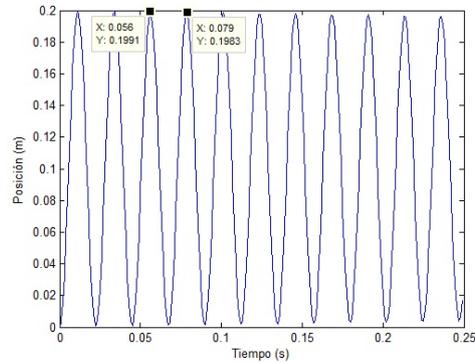


Figura 3.12: Salida del sistema en situación oscilatoria.

momento se toma el valor de la ganancia del controlador ( $K_{cu}$ ) y se mide el tiempo que transcurre entre cada oscilación ( $T_c$ ). Dichos valores se utilizan para sintonizar el valor del controlador que se desea utilizar, que puede ser P, PI, PD o PID. En la Tabla 3.1 se muestran los valores de sintonización a partir de los parámetros obtenidos  $K_{cu}$  y  $T_c$ . A partir de los datos de la

Controlador	$K_c$	$T_i$	$T_d$
P	$K_{cu}/2$	$\infty$	0
PD	$K_{cu}/1.6666$	$\infty$	$T_c/8$
PI	$K_{cu}/2.2$	$T_c/2.2$	0
PID	$K_{cu}/1.7$	$T_c/2$	$T_c/8$

Tabla 3.1: Sintonización parámetros Ziegler-Nichols

tabla anterior, se configuran controladores P, PD, PI y PID para observar el resultado del sistema modelado. La Figura 3.13 muestra el resultado de aplicar los valores de Ziegler-Nichols y la salida después de ajustar los niveles de ganancia. A continuación se implementa un controlador P en el eje y se introducen en primer lugar los parámetros obtenidos con la Tabla 3.1. La Figura 3.14 muestra en color azul la respuesta obtenida. La salida presenta una sobreoscilación del 20 %, una respuesta inaceptable para el control de una impresora 3d. La gráfica de color rojo de la misma Figura muestra la respuesta del sistema cuando se introducen los parámetros ajustados para el modelo simulado. En este caso se elimina la sobreoscilación pero el error de posición permanente es de casi el 90 %, totalmente inaceptable para los requisitos del presente trabajo. Tras comprobar el funcionamiento del contro-

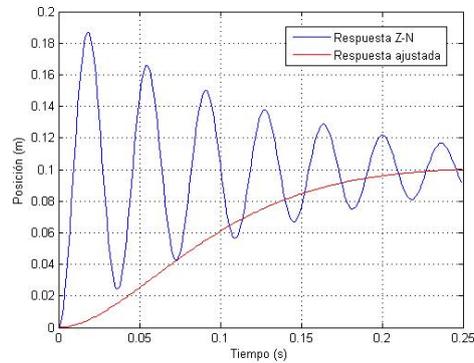


Figura 3.13: Respuesta simulada del controlador P.

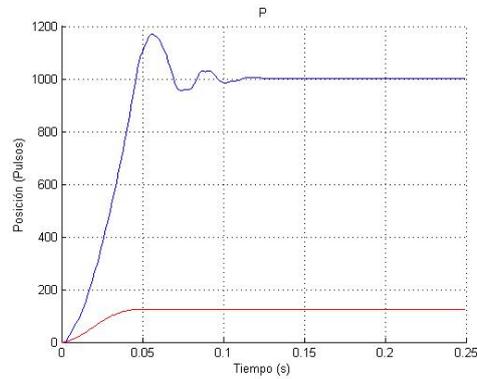


Figura 3.14: Respuesta real del controlador P.

lador P, se configura ahora un controlador PD y se introducen los parámetros obtenidos según la metodología Ziegler-Nichols. Para la señal de referencia de 1000 pulsos, podemos ver la respuesta en color azul en la gráfica de la Figura 3.15. El componente derivativo ha mejorado la respuesta sobreoscilatoria del controlador P, pero todavía presenta una sobreoscilación del 12%. Se introducen ahora los valores ajustados y se obtiene la gráfica de color rojo de la Figura. Esta respuesta no presenta sobreoscilación, pero tiene un error permanente del 30%. La respuesta es mucho mejor que la del regulador proporcional, pero todavía no se puede aceptar. Se plantea a continuación la implementación de un controlador PI. La Figura 3.16 muestra en color azul la respuesta correspondiente a los parámetros obtenidos a la simulación y en color rojo la respuesta una vez ajustados estos parámetros. Con los parámetros de Ziegler-Nichols el sistema presenta una sobreoscilación del 18% y error de posición nulo. Por el contrario cuando se ajustan los parámetros se elimina

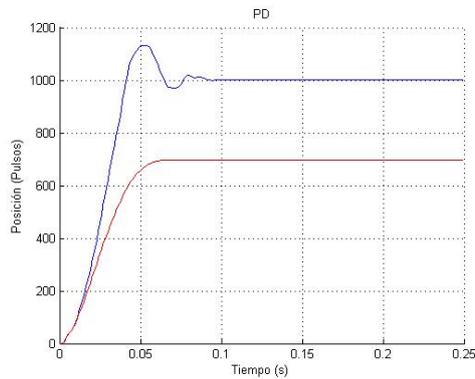


Figura 3.15: Respuesta real del controlador PD.

la sobreoscilación pero el sistema tiene un error transcurridos 0.25s del 75 %. Se deberá ajustar más este control si se quiere imprimir con la impresora utilizando motores de continua. A continuación se implementa un controlador

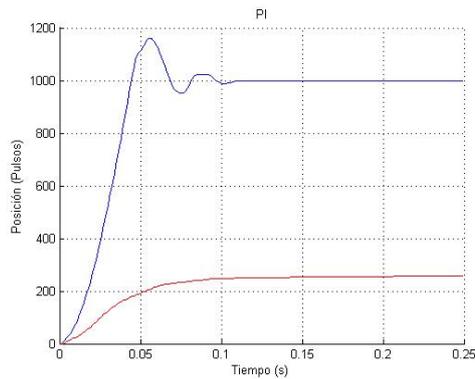


Figura 3.16: Respuesta real del controlador PI.

PID. Se introducen en primer lugar los parámetros obtenidos empíricamente utilizando la metodología Ziegler-Nichols y la respuesta se puede ver en color azul en la Figura 3.17. La sobreoscilación es de alrededor del 18 % y el error de posición es cero. Cuando se introducen los parámetros ajustados del PID, la respuesta no presenta sobreoscilación pero si tiene error de posición del 20 %. Se puede ver esta respuesta en color azul en la Figura 3.17.

Una vez que se ha controlado el motor de continua en un eje lineal, se montan los motores en la impresora para obtener el modelo del sistema, y proceder al control del mismo. Al igual que en el modelado del eje lineal, lo primero que se hace es observar la respuesta del sistema ante entradas de escalón. La Figura 3.18 muestra la respuesta del sistema ante entradas de

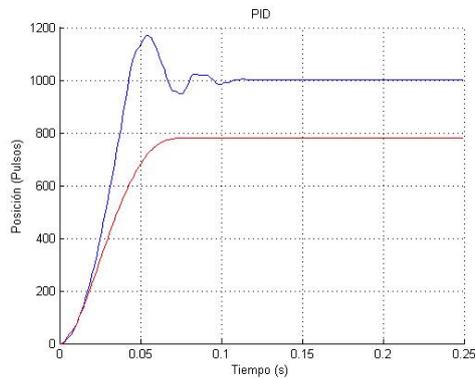


Figura 3.17: Respuesta real del controlador PID.

valor 3000 de los dos ejes. A partir de los datos de estas gráficas, obtenemos los parámetros correspondientes a la ganancia y la constante de tiempo del sistema en los dos ejes. La Figura 3.19 muestra los modelos de los diferentes ejes obtenidos. A continuación se calcula el modelo discreto del sistema

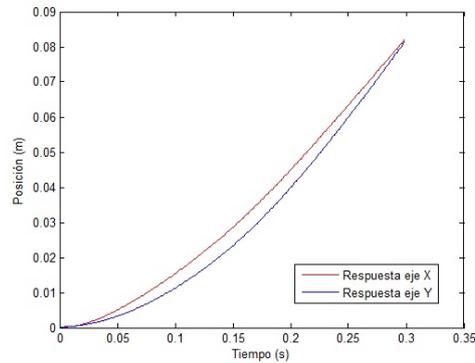


Figura 3.18: Respuesta de los ejes X e Y de la impresora.

utilizando el método de discretización mantenedor de orden cero (ZOH). En la Figura 3.20 se puede ver la coincidencia del modelo discreto con el modelo continuo obtenido anteriormente. En este punto se estiman los valores de los controladores utilizando el método de las oscilaciones mantenidas. Para ello se utiliza el modelo discreto añadiendo un control proporcional y un bucle de realimentación.

Por último se va a proceder a estimar el valor de zona muerta que tienen los motores de la impresora. Para ello se genera una señal senoidal de amplitud ligeramente superior al que se espera que tengan los motores. De este modo lo que veremos es a partir de que valor, el eje inicia su movimiento, la

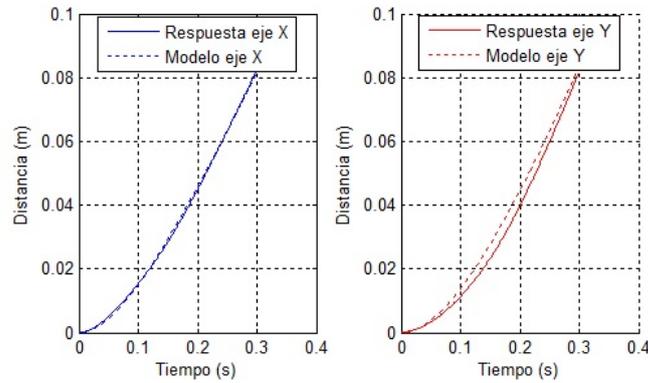


Figura 3.19: Modelo de los ejes X e Y de la impresora.

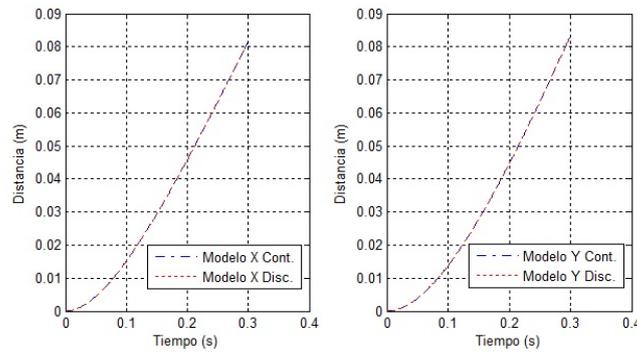


Figura 3.20: Modelo discreto de los ejes X e Y de la impresora.

Figura 3.21 muestra la señal senoidal de entrada y la gráfica de posición de salida. El valor de la zona muerta puede ser o no simétrico, por ello observaremos cuando inicia el movimiento en el sentido positivo y cuando lo hace en el sentido negativo. Primero se observa el momento en que se inicia el movimiento, y se busca el valor que tiene la señal senoidal en ese mismo tiempo. A continuación se marca el momento en que se inicia el movimiento en el otro sentido, y se marca también el valor de la señal senoidal en ese momento. Se repite en varias ocasiones para comprobar que se ha hecho correctamente. Se observa que en el sentido positivo los valores de entrada en que empieza el movimiento son 1740, 1875 y 1711. En cambio los valores del desplazamiento negativo son algo menores, de 1373, 1653 y 1532. Con esta serie de valores, queda a criterio del autor elegir el valor más alto, el valor más bajo o el valor promedio. Pero teniendo en cuenta el motivo de estar buscando el valor de la zona muerta, que es conseguir eliminar el error en régimen permanente, si se eligen los valores mínimos o promedio quedarán casos en lo que el eje no

se moverá aún siendo necesario. Por lo que se establece el valor de la zona muerta el mayor de los leídos, o incluso algo superior para evitar que el control no llegue a la referencia cuando pase el tiempo y la máquina empiece a tener desgaste.

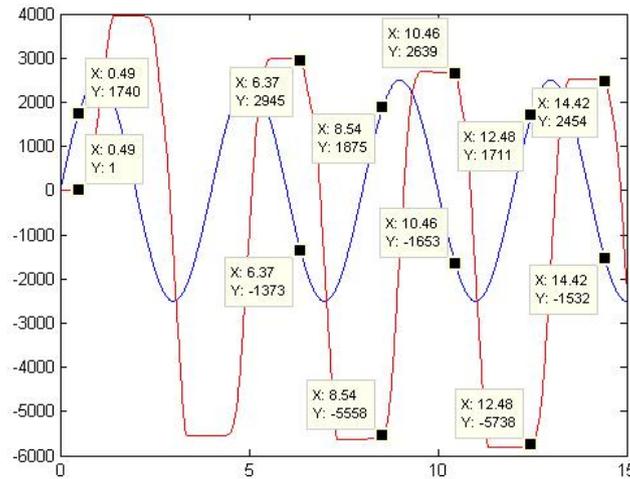


Figura 3.21: Cálculo de la zona muerta del eje X.

### 3.4. Programación unidad control

En primer lugar, se programa el microcontrolador para que cada milise-gundo realice las tareas de cálculo de la acción de control, en función de la referencia y de la posición del motor. Esto se realiza con ayuda de la librería de control PID disponible para Arduino. Se implementan controles P, PI, PD y PID. A pesar de que la respuesta es similar a la obtenida previamente en simulación, se da la circunstancia de que el error permanente es demasiado elevado, especialmente para pequeños cambios en la referencia.

Se opta entonces por compensar la acción de la zona muerta. Al valor calculado por la función de control, se le añade ahora un valor igual al calculado para la zona muerta. Esta acción de control ayuda al seguimiento de la señal, pero todavía no se consigue eliminar completamente el error de posición.

Se añade entonces una acción integral para eliminar el error en el régimen permanente y se satura la salida para que solo entregue valores dentro del rango de los motores. Con esta acción se consigue error permanente cero, pero aparece una sobreoscilación por lo que será necesario tratarla.

Se implementa entonces una condición en la que se resetea el valor del integrador en el momento en que el error está dentro de una franja de entre 3 y -3 pulsos.

Con estos ajustes, el controlador de los motores queda en condiciones de imprimir las piezas que nos sean necesarias.

### 3.5. Resultados

Finalmente se han impreso piezas controlando los ejes X e Y con motores de continua. En las Figuras 3.22 y 3.23 se pueden ver la pieza impresa con motores de pasos a la izquierda y la pieza impresa con motores de continua es la de la derecha. Las piezas impresas con los motores de continua tienen una



Figura 3.22: Piezas impresas.



Figura 3.23: Piezas impresas.

calidad igual o superior a las impresas utilizando los motores paso a paso. Se

pueden observar algunas imperfecciones en las piezas fabricadas antes de la modificación que se solucionan en la versión final. Las Figuras 3.24 y 3.25



Figura 3.24: Piezas impresas.



Figura 3.25: Piezas impresas.

muestran otro ejemplo de piezas impresas antes y después de implementar los motores de continua. A la izquierda de las imágenes se puede ver la pieza impresa con los motores paso a paso y a la derecha se encuentra la pieza impresa utilizando los motores de continua. Al igual que con la pieza del anillo, la calidad de la pieza impresa con los motores de continua es igual o superior a la impresa antes de realizar la modificación.

# Capítulo 4

## Conclusiones y trabajos futuros

Conforme a los objetivos planteados inicialmente para este trabajo, se ha conseguido implementar unos motores de continua para su uso en una máquina de impresión 3D. Gracias al control desarrollado ha sido posible la impresión de piezas de una calidad similar a la obtenida con motores paso a paso. Tras un análisis teórico del sistema, se ha diseñado y simulado un control, que posteriormente se ha implementado en una máquina real. A pesar de que el resultado era similar al obtenido en simulación, no resultaba funcional para seguir la secuencia de órdenes generada por el control de la impresora. Por lo que fue necesario ajustar los parámetros, e incluso modificar el control para conseguir un compromiso entre la respuesta dinámica del sistema y el error de posición.

En este proyecto he podido profundizar en el estudio de la tecnología de impresión 3D y particularmente en el control de motores de corriente continua. Pero al mismo tiempo me ha permitido conocer un poco más la filosofía de trabajo de código libre, en la que muchas personas trabajan de forma altruista para compartir experiencias y conocimiento al resto de las personas.

### 4.1. Trabajos futuros

En este trabajo se ha demostrado prácticamente que es posible la implementación de un control de motores de continua para imprimir piezas de 3 dimensiones utilizando una impresora 3D. Se proponen como trabajos futuros la utilización de motores de continua más potentes, con los que el rango de funcionamiento sea mayor para un control más eficaz de los ejes. A continuación se propone la investigación de procesos de identificación de la máquina, para su ajuste automático o semiautomático, ya que la mayoría

de estas máquinas son operada por entusiastas que conocen bien el mundo de la impresión 3D, pero en cambio sus conocimientos de control automático son bajos o nulos. Otra línea de investigación que resultaría interesante estudiar es la incorporación de un segundo bucle de control y referencia, que se encargue de controlar la velocidad de desplazamiento del eje, para atenuar las posibles vibraciones que puede hacer el sistema cuando llega a las sucesivas referencias. Para ello podría ser necesaria la instalación de sensores de velocidad.

# Bibliografía

- [1] Lenguaje de programación APT, consultado el 11/09/2015, <http://aptos.sourceforge.net/WhatIsApt.html>
- [2] Generador de trayectorias y control CNC, consultado el 10/09/2015, [www.bengler.no](http://www.bengler.no)
- [3] Generador de trayectorias y control CNC, consultado el 10/09/2015, [www.linuxcnc.org](http://www.linuxcnc.org)
- [4] S Bradshaw, A Bowyer and P Haufe, "The Intellectual Property Implications of Low-Cost 3D Printing", (2010) 7:1 SCRIPTed 5, <http://www.law.ed.ac.uk/ahrc/script-ed/vol7-1/bradshaw.asp>
- [5] [http://www-app.etsit.upm.es/departamentos/teat/asignaturas/lab-ingel/motores%20paso%20a%20paso\\_1def.pdf](http://www-app.etsit.upm.es/departamentos/teat/asignaturas/lab-ingel/motores%20paso%20a%20paso_1def.pdf)
- [6] Barreiro, A., Baños, A. (2012). Sistemas de Control basados en Reset. Revista Iberoamericana de Automática e Informática Industrial RIAI, 9(4), 329-346.
- [7] B. Evans, Practical 3D Printers: The Science and Art of 3D Printing.
- [8] Proyecto Sanguinololu, <http://reprap.org/wiki/Sanguinololu/es>, consultado el 10/09/2015