

Diseño de un post-procesador para sistemas de modelado por deposición fundida utilizando un brazo robótico Mitsubishi

Design of a post-processor for fused deposition modeling systems using a Mitsubishi robotic arm

Sandino Torres ^{1*}, Santiago C. Gutiérrez ²

¹ Instituto Superior Tecnológico Central Técnico, Quito, Ecuador

² Universitat Politècnica de València, Valencia, España, 46022; scgutier@mcm.upv.es

* Correspondencia: ltorres@istct.edu.ec

Recibido 05 mayo 2020; Aceptado 11 noviembre 2020; Publicado 01 diciembre 2020

Resumen: Este trabajo presenta el desarrollo de un post-procesador para el uso de un robot, Mitsubishi RV-2AJ, en procesos de modelado por deposición fundida (MDF). Para este propósito se diseña y se adapta un sistema de alimentación y arrastre de material a un brazo robótico Mitsubishi. También se ha desarrollado una interfaz para facilitar la comunicación entre el computador y el robot. Para que el robot realice su nueva función, es necesario partir de un modelo digital, compatible con la técnica de impresión 3D, generándose las trayectorias, instrucciones y tareas de trabajo. La información es post-procesada para que sea interpretada en el lenguaje del robot. Para comunicar los diferentes elementos con la controladora del robot se realiza el control y la programación mediante un sistema mínimo de electrónica. Finalmente se desarrolla una aplicación, se estructura el procedimiento y la secuencia necesaria para realizar el proceso de impresión.

Palabras clave: Control de robot, fabricación por deposición fundida, impresión 3D con robot, post-procesador código G, programación de robot.

Abstract: *This work presents a post-processor development for using a robot, Mitsubishi RV-2AJ, in fused deposition modeling (FDM) processes. For this purpose, a material feeding and dragging system are designed and adapted to a Mitsubishi robotic arm. An interface has also been developed to facilitate the communication between the PC and the robot. For the robot to perform its new function, it is necessary to start from a digital model, compatible with the 3D printing technique, generating the paths, instructions, and work tasks. This information is post-processed to be interpreted in the language of the robot. To communicate the different elements with the robot's controller, the control and the programming are carried out through a minimum electronics system. Finally, an application is developed, the procedure and sequence necessary to perform the printing process are structured.*

Keywords: *3D printing with robots, fused filament fabrication, G Code post-processor, robot control, robot programming.*

1 Introducción

El uso de robots industriales está experimentando un crecimiento vertiginoso (Iglesias, Sebastián, & Ares, 2015). Pero en la mayoría de los casos su uso se está relegando a la realización de tareas repetitivas (Keating & Oxman, 2013). En el caso de la industria automotriz son utilizados en diferentes procesos industriales como la soldadura por puntos, por arco eléctrico, pinturas de spray, transporte de materiales, etc. Su uso está destinado a una variedad de industrias

manufactureras, sin embargo, están dirigidos principalmente a cumplir tareas cíclicas como transferencia, carga y descarga de materiales (Ji & Wang, 2019).

Los robots industriales están ampliando su espectro de aplicaciones, siendo utilizados en operaciones de procesamiento relacionadas con el arte o con la arquitectura, como por ejemplo el proyecto MATERIAL (<https://ultimaker.com>), donde se usa un brazo robot de 6 ejes para distribuir material creando objetos en 3D, gracias al uso de materiales

termoestables. De esta manera el concepto de robot industrial se está transformando (Laarman, Jokic, Novikov, Fraguada, & Markopoulou, 2014).

En las aplicaciones relacionadas con el mecanizado de materiales, ha mejorado la precisión, se reducen costes y se tiene una buena integración con los sistemas CAM (Computer Aided Manufacturing).

En el proceso de mecanizado existen muchas ventajas en el uso de los robots con respecto a máquinas herramientas por control numérico, principalmente relacionadas con el coste cuando el tamaño de piezas a mecanizar es grande. Con ellos se puede alcanzar un mayor volumen de trabajo, sobre todo cuando se requieren procesar formas o geometrías complejas debido a que se pueden aumentar ejes adicionales. Además, se puede lograr una flexibilidad en la fabricación con cualquier tipo de operación. Una desventaja es la baja rigidez que tienen los brazos robóticos para estas tareas (Andrés, Gracia, & Tornero, 2009; Pérez, Gutiérrez, & Zotovic, 2018).

El desarrollo de nuevas técnicas y tecnologías de fabricación se encuentra en un gran momento de avance e innovación, por este motivo, los robots industriales están ampliando su utilidad de servicios y se están adaptando a la fabricación a medida, sobre todo en la fabricación digital, que se ha convertido en una potente herramienta para la creación de prototipos (Magnoni, Rebaioli, Fassi, Pedrocchi, & Tosatti, 2017).

Mediante una programación de aplicaciones informáticas se puede conseguir la fabricación de geometrías complejas y de múltiples materiales (Xiao, Eynard, Anwer, Le Duigou, & Durupt, 2016).

Los brazos robots están ganando terreno y están empezando a reemplazar a otras tecnologías de fabricación digital (Wu, Dai, Fang, Liu, & Wang, 2017).

En la técnica de modelado por deposición fundida MDF, en la cual se utiliza una cámara de calentamiento para fundir un polímero, que es alimentado al sistema en forma de filamento, no se han obtenido resultados contundentes, pero se está empezando a experimentar y a buscar utilidades (Gibson, Rosen & Stucker, 2014).

Estos procesos tienen limitaciones en cuanto a dimensiones, velocidad y parámetros del proceso. Usan únicamente un plano para la construcción de capas, lo que afecta a las propiedades finales del objeto construido.

Sin embargo, se han desarrollado plataformas de impresión multiplano usando brazos robóticos de seis grados de libertad como el proyecto Moto Maker. En la cual generan una estructura de celosía 3D, mediante un proceso FDM (Ishak & Larochelle, 2019).

En el presente trabajo se desarrolla un post-procesador para establecer una interfaz entre un PC y un robot Mitsubishi RV-2AJ de cinco grados de libertad, al cual se le adaptó un sistema para el modelado por deposición fundida MDF. Esta configuración permite aplicar las técnicas de prototipado rápido, para la fabricación de productos.

El post-procesador permitirá realizar un proceso MDF, beneficiándose de las ventajas de un robot industrial como su eficiencia, alta flexibilidad y multifuncionalidad y un mayor número de grados de libertad.

Además, el uso de un robot industrial permite acceder a posiciones, que son imposibles con una máquina tipo pórtico, y abarcan mayores espacios de trabajo.

El desarrollo de este proyecto, puede ser aplicado como una fábrica inteligente, el cual permitirá la experimentación de procesos MDF, en el sector académico, mejorando las competencias de los usuarios abordando problemas reales de las industrias y encontrado sus posibles soluciones (Guarín, Baena, Mora, 2019).

2 Metodología

Para el desarrollo de este trabajo se realizó la adaptación de distintos elementos al sistema MDF

En la figura 1, se muestra el diseño realizado para la adaptación del sistema de modelado por deposición fundida al brazo robot Mitsubishi RV-2AJ.



Figura 1: Diseño de la adaptación de MDF al Brazo Robot.

El Robot industrial marca MITSUBISHI RV-2AJ, de 5 grados de libertad, tiene articulaciones, las cuales realizan movimientos rotatorios desarrollados por actuadores, que para este caso son servomotores AC, en los ejes J1 a J3:50W con freno en los ejes J4; J6:15W sin freno, en el eje J5:15 W con freno, unidos a codificadores (encoders) de posición absoluta.

El robot tiene una carga máxima de 2 kg, la longitud de la parte superior del brazo es de 250 mm y la

longitud del antebrazo de 160 mm, un alcance radial máximo de 410 mm, una velocidad máxima de 2100 mm/s, con una repetitividad de ± 0.02 mm. Una de las limitaciones que se consideró para esta adaptación fue la zona muerta del robot, para que su espacio de trabajo no se vea afectado por la estructura del soporte.

Además, se tuvo en cuenta que el soporte sirviera para adaptarse a distintas configuraciones de diferentes proveedores de materiales. Dependiendo del fabricante, el material puede venir en forma de carrete o en forma de caja. Con el diseño realizado para este dispositivo, ambas formas y distintos tamaños pueden ser adaptados.

En la figura 2, se puede ver el rango de operación del robot, en donde se puede observar que hay una zona que está invadida por los elementos adicionales colocados al brazo robot, por lo que se debe inutilizar esta zona para evitar conflictos o colisiones por parte del robot.

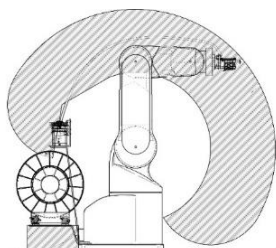


Figura 2: Zona de trabajo del Robot.

El controlador del Robot es el CR1-571, su lenguaje de programación es Melfa Basic (IV), cuenta con 16 entradas y 16 salidas, y con conexión RS-232.

Para adaptar el sistema de modelado por deposición fundida, MDF, se usó un extrusor de marca Rapman, un sensor Honeywell-135-104LAG-J01 y una unidad de arrastre del fabricante de impresoras BCN3D. La conexión de extrusión usada es del tipo Bowden, ya que se necesita reducir masa en el extremo del brazo robótico para garantizar movimientos más eficientes con un menor uso de energía y menos agitación del brazo robótico.

Además, se realizaron varios elementos adicionales para ensamblar el sistema al Robot y un soporte para unir el cabezal de extrusión a la muñeca del Robot, fabricado con un material aislante para evitar que el flujo de calor afecte al brazo robótico.

Así también, se acopló la electrónica al sistema, que será la encargada de suministrar la energía necesaria a los distintos componentes conectados al brazo robótico (la unidad de arrastre, la resistencia térmica, termistores, entre otras).

En la figura 3, se puede apreciar el esquema para la adaptación del sistema al brazo Robot.

2.1 Adaptación del software

Las etapas para llevar a cabo la fabricación de piezas mediante MDF a través de un brazo Robot Mitsubishi RV-2AJ, se resumen en la figura 4.

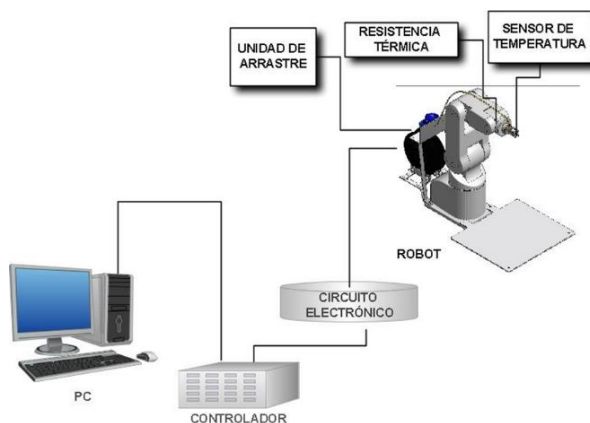


Figura 3: Conjunto del sistema MDF ensamblado al Robot.

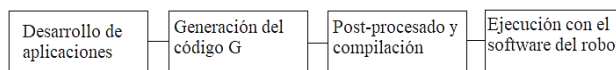


Figura 4: Esquema del proceso de fabricación.

Las instrucciones se realizan a través del software para impresión en 3D, CURA (Ultimaker, Waltham, MA 02451 EE.UU.), el cual genera el fichero en código G. Este código es muy similar al lenguaje de programación utilizado para la programación de máquinas de control numérico, el estándar ISO 6983.

Posteriormente, se deberá interpretar y transformar estas instrucciones al lenguaje del robot, y de esta manera poder controlar los parámetros necesarios para realizar el proceso de fabricación utilizando el brazo. Para obtener la pieza en 3D, es necesario diseñarla previamente mediante cualquier software CAD que exista en el mercado. Los programas de impresión 3D reciben la geometría de la pieza a fabricar en formato "stl".

Esta geometría la descomponen en capas, convirtiendo cada capa en movimientos que realiza la máquina para ir depositando plástico fundido y conseguir el "crecimiento" del modelo.

En la tabla 1, se muestra un ejemplo del tipo de instrucciones generadas por código G.

2.2 Interfaz de la programación

El Robot Mitsubishi RV-2AJ se controla mediante el desarrollo de una interfaz personalizada utilizando Java SE 6.0, en un entorno de programación NetBeans 8.0.2, como se muestra en la figura 5. Esta interfaz se encarga de convertir los datos obtenidos en código G

por el software de impresión, al lenguaje Melfa Basic del robot Mitsubishi. Este lenguaje es utilizado por el controlador del robot para la realización de programas. La sintaxis es de tipo BASIC, y la programación se hace mediante el empleo de sentencias de alto nivel. El controlador utilizado en este trabajo es el CR1-571, el cual soporta la versión IV. El lenguaje de programación se estructura como un conjunto de instrucciones, y su compilación se realiza en lenguaje BASIC estándar.

2.3 Post-Procesado

En esta etapa se realiza el sistema de post-procesado general, que acepta el formato del código G procedente del programa de impresión 3D CURA, y lo traduce al lenguaje Melfa Basic del robot Mitsubishi RV-2AJ. En la etapa de post-procesado se realizan principalmente las siguientes tareas:

- Convertir el lenguaje de la aplicación usada para realizar la impresión 3D al lenguaje del robot.
- Compilación del código post-procesado en el software del robot.

Tabla 1: Descripción de código G.

Código	Descripción
M92E800.00	Establece los pasos por unidad de los ejes
M140S120.00	Fija a 120 °C la temperatura de la cama
M109S230.00	Fija y espera que es extrusor llegue a los 230 °C
G21	Selecciona mm como unidad
G90	Coordenadas absolutas
M107	Arranca con el ventilador apagado si lo hubiera
G28 X0 Y0	Busca finales de carrera X e Y para hacer el 0
G28 Z0	Busca finales de carrera Z
G92 X0.0	Define la posición de los ejes como 0
YO.0 Z0.0	
E0.0	
G1 Z5.0 F60	Movimiento en Z hasta 5.0 acelerado de 0 a 60 mm/min
G92 E0	Define la posición del extrusor como 0

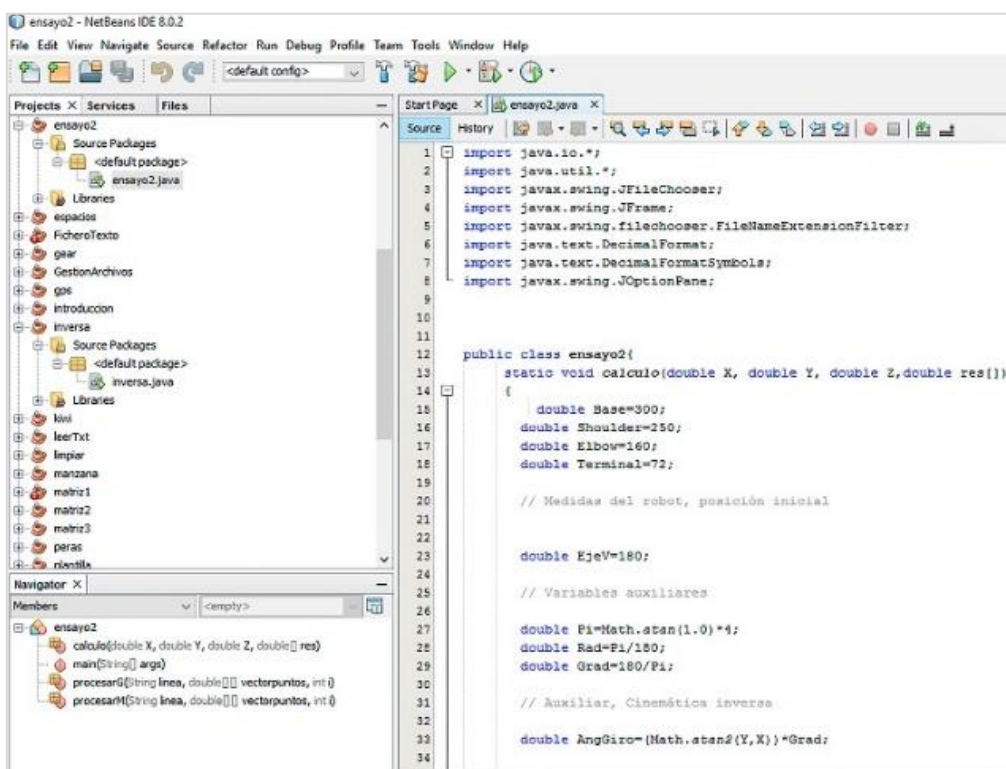


Figura 5: Desarrollo en NetBeans.

El fichero transformado tiene una estructura similar a la del ejemplo que se muestra a continuación:

```
5 TOOL (0,0,174.25,0,0,0)
15 OVRD 5
25 M2=20
35 M3=-100
45 M4=30
55 M5=350
65 P2.Z=0
75 IF P1.X>M5 GOTO 165
```

```
85 MVS P1
95 MVS P2
105 P2.Z=P2.Z-M2
115 IF P2.Z<M3 GOTO 125 ELSE GOTO 85
125 MVS P1
135 P1.X=P1.X+M4
145 P2.X=P2.X+M4
155 GOTO 65
165 HLT
P1=(226.17,71.150,100.00,13.51,180.00,0,0,0,0,0)(6,0)
P2=(226.170,-71.150,0,00,13.510,180,0,0,0,0,0)(6,0)
```

Este código nos muestra información de las acciones y movimientos del brazo robot. En las primeras líneas se establece la posición inicial y las velocidades para los desplazamientos, luego se establecen las variables de posición. Además, se establecen funciones de BASIC estándar que sirven para transferir el control del programa a líneas determinadas.

Al final del programa se incluyen los puntos de destino, (para este ejemplo P1, P2) donde se moverá el extremo del robot de acuerdo a cada instrucción. Estas líneas tienen la información de los puntos a los que se debe mover el efector final del robot, referenciado en el sistema de coordenadas X, Y, Z. Además de la configuración que debe tener el robot para orientarse en la posición deseada.

Teniendo en cuenta estas consideraciones y especificaciones, hacemos un análisis del archivo fuente para convertirlo en el archivo destino. Para este proceso es necesario identificar cada una de las partes de una instrucción e ir las descomponiendo. De esta manera vamos transformando y agrupando el nuevo código en el archivo de destino final.

Tabla 2: Reconocimiento de variables.

Código G	Melfa Basic
M	IN/OUT
G	MOV/MVS
E	IN/OUT
F	SPD
X	P#(X, ,)
Y	P#(, Y, ,)
Z	P#(, , Z,)
S	No aplicable

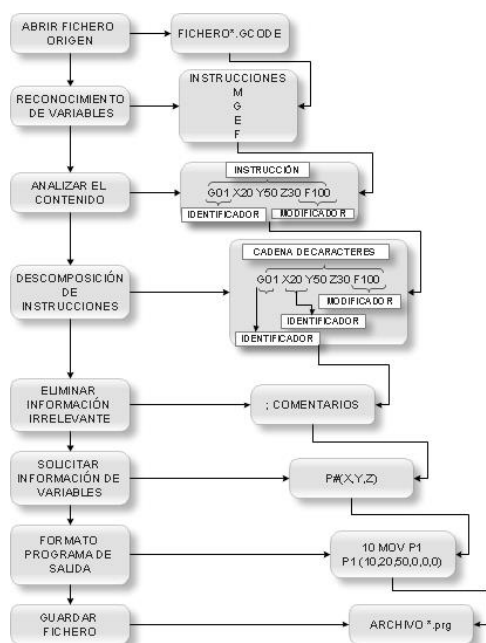


Figura 6: Etapa de post-procesado.

2.4 Obtención del código en lenguaje robot

La tarea principal que se debe conseguir en la etapa de post-procesado es obtener el código en el lenguaje del robot, para esto se debe construir un programa estructurado. A continuación, se detalla cada una de las operaciones representadas en la figura 6, para la obtención del código en lenguaje del robot.

2.4.1 Abrir fichero de origen

El fichero que se generará en CURA es *.GCODE, su formato es ASCII, por lo que se procede a cambiarle la extensión, a *.TXT, de esta forma se crea una instrucción para abrir ficheros de este tipo.

2.4.2 Reconocimiento de variables

Las variables que se debe tener en cuenta para la transformación del código son las que se muestran en la tabla 2.

2.4.3 Analizar el contenido carácter a carácter (Identificar)

Una vez identificadas las variables, es necesario descomponer cada instrucción, analizarla y establecer coherencias de acuerdo a su tipo y a sus parámetros, y de esta manera establecer las condiciones necesarias para cada nueva instrucción en Melfa Basic. Esta evaluación se realiza analizando cada carácter, en concordancia:

- Toda instrucción empieza por el identificador G o M
- El conjunto de identificadores posibles {G,M,E,F,S,X,Y,Z}

Se requieren algunas condiciones, las cuáles se describen en la tabla 3.

2.4.4 Estructura del código en lenguaje Melfa Basic

Para obtener el código en el lenguaje del Robot Mitsubishi, se debe descomponer las instrucciones hasta obtener el formato en lenguaje de programación Melfa Basic, para lo cual primero se eliminan los comentarios e información irrelevante, luego se definen los parámetros de salida, de acuerdo con las condiciones de la tabla 4.

Una vez obtenido este formato con las variables del robot se guarda el fichero con el código de destino en formato ASCII, pero con la extensión “*.PRG”, que es el tipo de fichero que lee el programa del robot.

Una vez creado el programa informático, se realiza la compilación para revisar la sintaxis y semántica del programa, y se obtiene el código. Este código debe ser verificado en el simulador del robot. Si existieran inconsistencias, se deben identificar, y mediante el código de programación solventarlas.

2.4.5 Ejecución con el software del robot

Para la ejecución de programas se utilizará el software *RT ToolBox2*. Este software, junto con MELFSOFT, permite el envío, recepción y comunicación de programas, así como simulaciones complejas. Los programas pueden someterse a diversas pruebas de aplicación.

Para suministrar la energía necesaria a los distintos componentes acoplados al brazo robot, se utiliza una fuente de alimentación externa. Las distintas señales que intervienen en el proceso se controlan a través de las entradas y salidas de la unidad de control del robot. Los elementos que interviene en el proceso son, la unidad de arrastre, donde se encuentra el motor encargado de alimentar el material, el “*hot-end*”, para controlar la temperatura de calentamiento, y el sensor de temperatura.

Tabla 3: Condiciones para código G.

No.	Condición
1	Después de cada identificador se encuentra un valor numérico entero o real.
2	Tras el identificador ‘G’, pueden aparecer los siguientes valores: {0,1,21,28,90,91,92}
3	Tras el identificador ‘M’, pueden aparecer los siguientes valores: {82,84,104,106,107,109,190}
4	Tras el identificador ‘F’ ha de incluirse un entero positivo entre 0 y 9000
5	Tras el identificador ‘S’ ha de incluirse un entero positivo entre 150 y 500. (Este identificador no se toma en cuenta para el post procesado, es decir se elimina ya que esta función no se aplica en las operaciones del robot)
6	La instrucción G0 recibirá como parámetro, al menos un ítem del conjunto {X<x>, Y<y>, Z<z>}, y opcionalmente el modificador ‘F<f>’
7	La instrucción G01 recibirá como parámetro, al menos un ítem del conjunto {X<x>, Y<y>, Z<z>}, y opcionalmente los modificadores ‘F<f>’, ‘S<s>’
8	Únicamente serán válidos los siguientes ítems: {X<x>, Y<y>, Z<z>, S<s>, F<f>, E<e>, G1, G1, G90, G91, G28, G90, G91, G92, M190, M109, M82, M107, M106, M104, M140, M84}
9	Si la instrucción G0, no encuentra uno de los ítems del conjunto {X<x>, Y<y>, Z<z>}, se debe completar el parámetro que falta con el valor de la instrucción inmediata anterior, y si no existiera, su valor será 0.
10	Si la instrucción G1, no encuentra uno de los ítems del conjunto {X<x>, Y<y>, Z<z>}, y el modificador ‘F<f>’, se debe completar el parámetro que falta con el valor de la instrucción inmediata anterior, y si no existiera su valor será 0.

Tabla 4: Definición de parámetros de salida.

No.	Condición
1	Las instrucciones del conjunto {X<x>, Y<y> y Z<z>}, deben ser empleadas para la generación de puntos, {P#, P#+1, etc.}.
2	Con la instrucción ‘G0’, se crea una variable de tipo MOV.
3	Con la instrucción ‘G1’, se crea una variable de tipo MVS. (Movimiento en interpolación lineal).
4	La instrucción ‘F’ crea un parámetro SPD. (Designa la velocidad en mm/s de interpolación), junto con el valor numérico que lo acompaña.
5	La instrucción ‘E’, llama a un programa de subrutina, y se emplea, junto a él, el valor numérico que lo acompaña. A su vez activa una <variable>=M_IN(<bit>). Se reserva un bit para esta operación, y queda deshabilitado para otra operación
6	La instrucción ‘M’, llama a un programa de subrutina, y se emplea, junto a él, el valor numérico que lo acompaña. A su vez activa una <variable>=M_IN(<bit>). Se reserva un bit para esta operación, y queda deshabilitado para otra operación

3 Resultados

Mediante la aplicación que se ha desarrollado, se establece la interfaz, de esta manera el brazo robot realiza las tareas descritas en el código desarrollado, para finalmente obtener piezas mediante la técnica MDF. En la figura 7 se detallan las etapas necesarias para un correcto funcionamiento, utilizando la aplicación desarrollada.

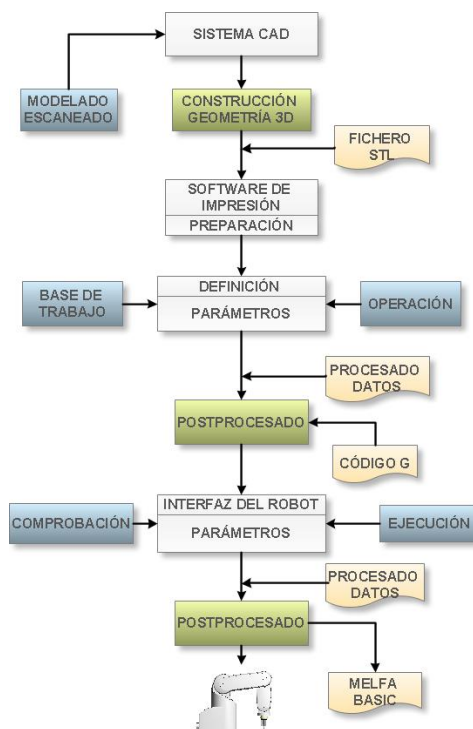


Figura 7: Funcionamiento de la aplicación.



Figura 8: Ícono para ejecutar la aplicación.

Para ejecutar la aplicación desarrollada se creó el icono (ver figura 8), que es el que inicia el proceso de post-procesado. Una vez arrancada la aplicación, se debe establecer la posición inicial de trabajo, y la posición de origen de la pieza, que es donde inicia el proceso de deposición fundida.

Este valor va referenciado a la mesa de trabajo. Debe tenerse en cuenta la longitud de nuestro efector, en este caso el “hot-end”. La medida de esta longitud se ha calculado hasta obtener la posición correcta de trabajo, como se aprecia en la figura 9.

Estos datos sirven para verificar que el movimiento inicial del brazo robot se encuentre dentro de la zona de trabajo, y que el proceso se realice enteramente dentro del área de trabajo.

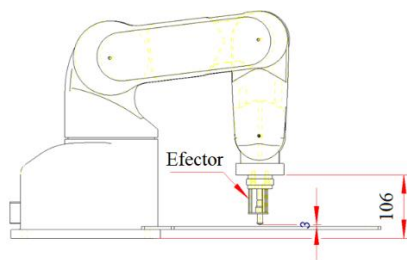


Figura 9: Posicionamiento del brazo robot.

Estas posiciones se calculan mediante cinemática inversa, donde relacionamos las coordenadas articulares en función de las coordenadas cartesianas y la orientación de la herramienta del extremo final del brazo robótico.

La cinemática directa, que es una función vectorial $f_R(l_i, q)$ que vincula coordenadas articulares (q_1, q_2, \dots, q_n) , con propiedades geométricas del sistema mecánico l_i , con la posición y orientación del extremo del robot $[x, y, z, \theta, \phi, \psi]$.

$$\begin{bmatrix} x \\ y \\ z \\ \theta \\ \phi \\ \psi \end{bmatrix} = f_R(l_i, q) \quad (1)$$

$$q = f_R^{-1}(x, y, z, l_i, \theta, \phi, \psi) \quad (2)$$

Donde $q = f_R^{-1}(x, y, z, l_i, \theta, \phi, \psi)$ es la función inversa de la ecuación (1).

θ, ϕ, ψ . Corresponden a los ejes de rotación en cada uno de los ejes lineales x, y, z .

Estas ecuaciones nos permitirán calcular y establecer la posición del efector, ubicado al final del brazo robot. De esta manera podemos ubicar en la posición que consideremos conveniente, al “hot-end” y disponer rápidamente de una posición concreta para iniciar el proceso de deposición fundida del material. Con lo cual, todas las articulaciones del brazo robot se posicionan y orientan según una determinada localización establecida.

Los cálculos están incluidos dentro de la aplicación creada para el interfaz con el brazo robótico. El resultado de los cálculos se muestra en el encabezado del fichero que se crea. A continuación, puede verse un ejemplo de sus valores.

- La posición de origen es: J1= -59.5 J2= 82.6 J3= 72.4 J5= 25 J6= -59.5
- El origen de la pieza es: J1= -35.8 J2= 85.4 J3= 71.9 J5= 22.7 J6= -35.8

El valor de J, corresponde a cada articulación del robot, y es la posición en la que se ubicará; el brazo robot es de cinco grados de libertad. Los cinco grados de libertad se corresponden con J1, J2, J3, J5, J6. El brazo robot no posee el movimiento correspondiente a la articulación J4, motivo por el cual esta no es calculada.

3.1 Inicialización en Melfa Basic

El software Melfa Basic es inicializado y el fichero generado con las secuencias del programa es cargado.

Una vez cargado el programa, se puede realizar la simulación con el comando “Simulation – Automatic operation”. De no existir errores, se guarda el programa en la memoria del robot para cargarlo desde la controladora y ejecutarlo.

Una vez que el programa es inicializado, se establece la comunicación entre el robot y el ordenador, de esta manera se podrá controlar al brazo robótico desde el ordenador y los procedimientos de comprobación. Se deberá seguir el flujo de trabajo mostrado en la figura 7, lo que permitirá el desarrollo del proceso de forma correcta.

3.2 Implementación

Una vez obtenido y compilado el programa, que es la última etapa de la aplicación, y luego de post procesar la información, se obtiene el código en el lenguaje del robot, que para este caso es el Melfa Basic, tal como se puede apreciar en el ejemplo de la figura 10, una vez obtenido el código se procede a cargarlo en el control del brazo robótico.

```

VER2.prg.txt: Bloc de notas
Archivo Edición Formato Ver Ayuda
|La posición de origen es J1= 120.7 J2= 68 J3= 81.1 J5= 30.9 J6= 120.7
El origen de la pieza es J1= 100.7 J2= 79.3 J3= 51.7 J5= 49 J6= 100.7
10 DEF INTE PASOS
20 DEF INTE LONGITUD
30 PASOS=2*LONGITUD
40 DEF INTE ESPERA
50 GOTO 310
60 ESPERA=0.05
70 FOR T=0 TO PASOS
80 m_out(0)=1
90 m_out(1)=0
100 m_out(2)=1
120 m_out(3)=0
130 DLY ESPERA
140 m_out(0)=0
150 m_out(1)=1
160 m_out(2)=0
170 m_out(3)=1
180 DLY ESPERA
190 m_out(0)=0
200 m_out(1)=1
210 m_out(2)=1
220 m_out(3)=0
230 DLY ESPERA
240 m_out(0)=0
250 m_out(1)=0
260 m_out(2)=1
270 m_out(3)=1
280 DLY ESPERA
290 NEXT T
300 RETURN
310 BASE (160.0,-270.0,140.0,0.0,0.0,0.0)
320 SPD 9000.0
330 MOV P1
340 SPD 200.0
350 LONGITUD=3.0
360 GOSUB *MOTOR
370 MOV P2
    
```

Figura 10: Fichero post-procesado en *.PRG.

La estructura del código está compuesta por bloques identificados, los bloques inician con un número que está secuenciado de 10 en 10. La primera parte del código está compuesta por la cabecera de programa, se establece la posición inicial a la que se ubicará el robot, seguidamente de la posición desde donde empieza a trabajar.

Posteriormente se establece la secuencia de movimientos, entre ellos, aquellos aplicados para

obtener la geometría de la pieza. También se activan las señales de salida hacia el control del robot para realizar las funciones previamente establecidas.

3.3 Ensayos

Con el propósito de verificar que el robot cumple con las trayectorias y operaciones generadas mediante la interfaz desarrollada, se establecen diferentes ensayos para comprobar que ejecuta sus acciones con seguridad. En un primer ensayo son comprobados los datos obtenidos para la posición de origen.

En primera instancia se genera el punto inicial de trabajo, esta posición será el nuevo origen. El robot se moverá tomando como referencia ese punto y el nuevo sistema de coordenadas.

El punto de origen de los ejes se puede modificar cuando se requiera definir la nueva base de trabajo.

En figura 11, se muestran las coordenadas en XYZ, referenciadas a la base del brazo robótico, y que, para este ejemplo, será su posición de origen. Así también se muestra la posición del extremo del robot, que para este ejemplo es X=199.996, Y=-199.996 y Z=175.076. Mediante estos datos la aplicación genera los valores de J1, J2, J3, J5, J6, que son las coordenadas articulares para aquella posición.

A partir de esta referencia, el post-procesador establece todos los movimientos, las trayectorias y las funciones que realiza el brazo robótico. Las coordenadas de trabajo a la que debe desplazarse el robot vienen definidas desde un software de impresión 3D, el modelo de ensayo se muestra en la figura 12.

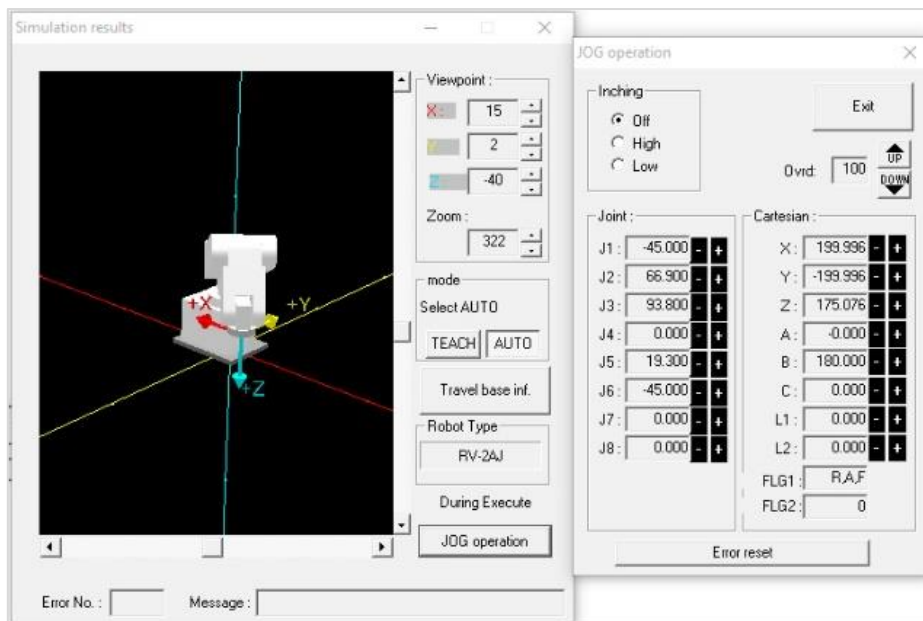


Figura 11: Posición inicial del brazo robótico.

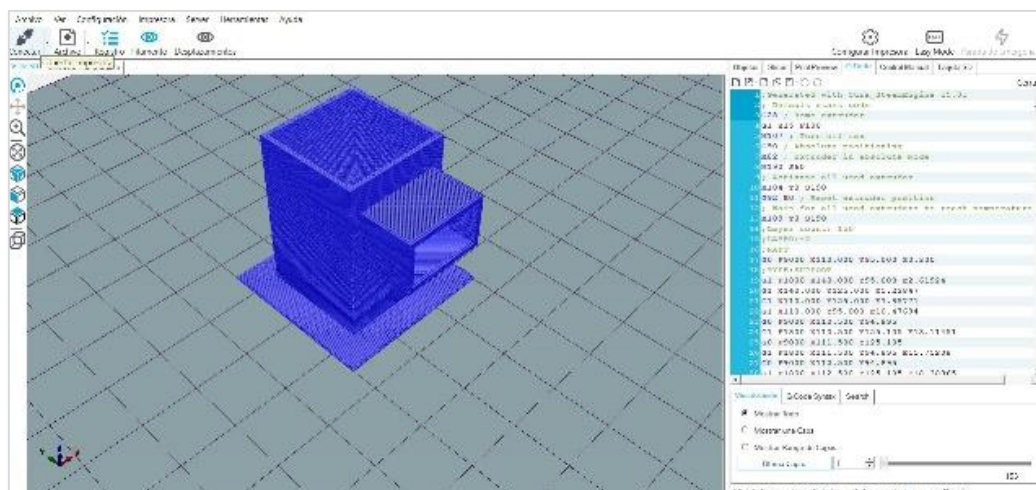


Figura 12: Modelo a obtener mediante MDF.

3.4 Pruebas del robot

Una vez procesada la información, cargamos el programa en la memoria del robot, en su control, e iniciamos la secuencia y comprobamos que es la misma que se obtuvo mediante la simulación gráfica a partir del modelamiento tridimensional.

En la figura 13, se muestra la interfaz del hardware realizada entre el PC y el robot, y el diseño final, ya implementado, de la adaptación de sistema MDF al brazo robot. Además, se muestra la posición inicial del robot una vez que se ejecuta el programa obtenido mediante el post-procesador.

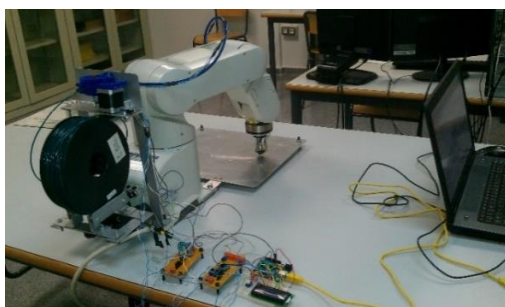


Figura 13: Pruebas con el robot.

4 Discusión

Se realizó un estudio preliminar para establecer una adecuada integración de elementos electrónicos. El circuito electrónico permitió comunicar los diferentes elementos con la controladora del brazo robótico, ofreciendo la potencia necesaria para dar movimiento al sistema. Además, el circuito electrónico permitió suministrar la energía necesaria a los distintos componentes acoplados al brazo robótico (la unidad de arrastre, la resistencia térmica, termistores, entre otros).

Se estableció un protocolo de comunicación, entre el software de impresión 3D y el software del robot, y se generó una interfaz, para realizar la fabricación de un producto, mediante un brazo robótico, a partir de un modelo digital. Una vez adaptado el sistema, fueron transferidas todas las bondades del brazo robótico al proceso. En la tabla 5, se muestra una comparación de los principales parámetros del proceso desarrollado mediante el robot industrial RV-2AJ y máquinas más comunes para impresión 3D.

Tabla 5: Comparación de impresoras 3D y robot adaptado para MDF.

Indicador	impresora 3D	robot industrial
Precisión(mm)	-0.1 – 1.0	-0.1 – 1.0
Espacio de trabajo	Limitado	Amplio
Número de ejes	2 ½	5
Complejidad de la trayectoria	Apto para máquinas de 2 ½ ejes	Cualquier trayectoria compleja
Lenguaje de programación	Código G	Melfa Basic
Manufactura flexible	Operaciones similares	Cualquier tipo de operación
Producción	Pequeñas	Medianas
Precio	Económico	Alto

En la comparación mostrada se puede evidenciar que un brazo robótico pueda aportar varios beneficios al proceso de MDF ya que es posible implementar más ejes de trabajo para la fabricación de piezas, como por ejemplo piezas con voladizos, sin la necesidad de soportes, permitiendo además una amplia superficie de operación. El desarrollo de este post-procesador es aplicable a cualquier otro método de fabricación, siempre y cuando se realice una reconfiguración del efector o cabezal de trabajo. El procedimiento en base a la metodología desarrollada muestra versatilidad y permite dar fiabilidad a todo el proceso, potenciando al proceso MDF con las ventajas que posee un brazo robot industrial.

5 Conclusiones

Se desarrolló un método de fabricación aditivo adaptado a un brazo robótico. Surgió de la idea de fusionar e integrar estas dos tecnologías y experimentar con nuevas técnicas de fabricación. Debido a esto se pueden mencionar las siguientes conclusiones:

Después de evaluar e identificar todos los parámetros necesarios para migrar el sistema de fabricación aditiva de impresoras 3D hacia el brazo robótico industrial, se ha podido experimentar, evaluar y mostrar las nuevas oportunidades que presenta esta propuesta. La fabricación de productos a gran escala es un ejemplo de aplicación que beneficiaría al sector mobiliario, la arquitectura, y el arte.

Las aplicaciones de brazos robóticos en manufactura digital están creciendo y continuarán creciendo, debido a sus ventajas inherentes de flexibilidad. Estas aplicaciones tienen un beneficio potencial si se integran todas las funciones de fabricación en una sola máquina. A través del uso de efectores finales se alcanzan reconfiguraciones flexibles en operaciones de medición sobre la pieza impresa (mediante el acople de un palpador), de mecanizado (mediante el acople de una herramienta rotativa), es decir un sistema multifuncional.

Para la ejecución de varias tareas proyectadas a través de este trabajo, sería interesante desarrollar un programa informático “multitarea”, que permita integrar todos estos procesos de fabricación, sin la necesidad de usar distintas máquinas y diferentes programas.

Conflicto de Interés

Los autores declaran que no existe conflicto de intereses de naturaleza alguna con la presente investigación.

Agradecimientos

Los autores desean expresar su gratitud al Departamento de Ingeniería Mecánica y Materiales de la Universitat Politècnica de València por proporcionar los recursos necesarios para el desarrollo de este trabajo.

Referencias

Andres, J., Gracia, L., & Tornero, J. (2009, April). Inverse kinematics of a redundant manipulator for CAM

integration. An industrial perspective of implementation. In: *2009 IEEE International Conference on Mechatronics* (pp. 1-6). IEEE.

Gibson, I., Rosen, D. W., & Stucker, B. (2014). *Additive manufacturing technologies* (Vol. 17). Springer. New York. EE. UU. <https://doi.org/10.1007/978-1-4939-2113-3>.

Guarín, A., Baena, F., & Mora, J. (2017). *Fábrica de Aprendizaje: Nuevo modelo de enseñanza productiva*. First South American International Conference on Industrial Engineering and Operations Management, Bogota, Colombia,

Iglesias, I., Sebastián, M. A., & Ares, J. E. (2015). Overview of the State of Robotic Machining: Current Situation and Future Potential. *Procedia Engineering*, *132*, 911–917. <https://doi.org/10.1016/j.proeng.2015.12.577>

Ishak, I. B., & Larochelle, P. (2019). MotoMaker: a robot FDM platform for multi-plane and 3D lattice structure printing. *Mechanics Based Design of Structures and Machines*, *47*(6), 703-720.

Ji, W., & Wang, L. (2019). Industrial robotic machining: a review. *International Journal of Advanced Manufacturing Technology*, *1239–1255*. <https://doi.org/10.1007/s00170-019-03403-z>

Keating, S., & Oxman, N. (2013). Compound fabrication: A multi-functional robotic platform for digital design and fabrication. *Robotics and Computer-Integrated Manufacturing*, *29*(6), 439-448.

Laarman, J., Jokic, S., Novikov, P., Fraguada, L. E., & Markopoulou, A. (2014). Anti-gravity additive manufacturing. *Fabricate 2014: Negotiating design & making*, 191-197.

Magnoni, P., Rebaioli, L., Fassi, I., Pedrocchi, N., & Tosatti, L. M. (2017). Robotic AM system for plastic materials: tuning and on-line adjustment of process parameters. *Procedia Manufacturing*, *11*, 346-354.

Pérez, R., Gutiérrez, S. C., & Zotovic, R. (2018). A study on robot arm machining: Advance and future challenges. *Annals of DAAAM & Proceedings*, *29*. <https://doi.org/10.2507/29th.daaam.proceedings.134>

Wu, C., Dai, C., Fang, G., Liu, Y. J., & Wang, C. C. (2017, May). RoboFDM: A robotic system for support-free fabrication using FDM. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 1175-1180). IEEE.

Xiao, J., Eynard, B., Anwer, N., Le Duigou, J., & Durupt, A. (2016). Geometric Models and Standards for Additive Manufacturing: a Preliminary Survey. *2016 Virtual Concept International Workshop - Major Trends in Product Design*, Bordeaux, France, (March 2016), 1–4.