

# **CONTROL NUMÉRICO COMPUTARIZADO UTILIZANDO INTERPOLACIÓN LINEAL PARA AUTONIVELAR LA SUPERFICIE DE TRABAJO EN UNA CNC**

*COMPUTERIZED NUMERICAL CONTROL USING LINEAR INTERPOLATION TO SELF-LEVEL THE WORKING SURFACE IN A CNC*

***Miguel Antonio Lázaro Santiago***

Universidad Autónoma del Carmen  
*miguel\_antonio98@hotmail.com*

***David Inocente Santiago Casanova***

Universidad Autónoma del Carmen

***José Luis Vázquez Ávila***

Universidad Autónoma del Carmen  
*jvazquez@pampano.unacar.mx*

***Walter Ariel Silva Martínez***

Universidad Autónoma del Carmen

***Marco Antonio Rodríguez Blanco***

Universidad Autónoma del Carmen

***Horacio Alvarado Vigil***

Universidad Autónoma del Carmen

***Homero Toral Cruz***

Universidad de Quintana Roo  
*htoral@uqroo.edu.mx*

## **Resumen**

En este trabajo se presenta la implementación de una máquina CNC (Control Numérico Computarizado, por sus siglas en español) de bajo costo que utiliza un algoritmo de autonivelación para mejorar el desempeño del maquinado. Mediante el algoritmo se puede corregir por software el desnivel de la cama de fresado independientemente de las imperfecciones del material que se va a maquinar. Básicamente lo que hace el software es deformar el dibujo que se imprimirá de tal

manera que este compense la deformación del material. Se utiliza una tarjeta de desarrollo "Arduino UNO" para capturar los datos de una sonda de contacto, posteriormente la información obtenida será procesada por una computadora, que a su vez enviará los datos procesados al Arduino para que este accione los motores. Se utilizó NetBeans para el desarrollo de la plataforma del usuario y el firmware GRBL (para Arduino) como lenguaje de programación, ambos de código libre, además el software Matlab es utilizado para realizar las simulaciones del código. Los resultados muestran que el uso del algoritmo de autonivelación efectivamente mejora el proceso de maquinado.

**Palabras Claves:** Auto nivelación, Arduino, CNC, deformación, GRBL, interpolación lineal.

## **Abstract**

*This paper presents the implementation of a CNC machine (Computerized Numerical Control, for its acronym in Spanish) of low cost that uses a self-leveling algorithm to improve the machining performance. By means of the algorithm, the unevenness of the milling bed can be corrected, by software, independently of the imperfections of the material to be machined. Basically, what the software does is to deform the drawing that will be printed in such a way that it compensates for the deformation of the material. An "Arduino UNO" board is used to capture the data from a contact probe, then the information obtained will be processed by a computer, which in turn will send the processed data to the Arduino, so that it drives the motors. NetBeans was used for the development of the user platform and the GRBL firmware (for Arduino) as programming language, both of them are free code, in addition the Matlab software is used to perform the code simulations. The results show that the use of the self-leveling algorithm improves the machining process.*

**Keywords:** Arduino, CNC, GRBL, lineal interpolation, self-leveling.

## **1. Introducción**

Uno de los grandes problemas al momento de fresar con una máquina CNC es que la superficie del material o la cama del CNC no son lo suficientemente planas,

por lo tanto, al momento de cortar en estas condiciones perdemos precisión o el producto final tiende a deformarse. Una solución muy común que se le dio a este problema es el de rectificarla con una fresa [Luis, 2015]. En la literatura existen diversos trabajos que abordan este tema, para CNCs de diferentes aplicaciones. En [De Lica, 2009] y [Kaushik, 2012] se usa una técnica de puntos de referencia para el diseño de placas PCB (*printed circuit board*, por sus siglas en inglés) implementadas en CNC. Dicha técnica permite localizar de manera más precisa los CI (circuitos integrados) en la placa. En [De Lica, 2009] se utilizan técnicas de procesamiento de imágenes para disminuir los desperfectos en la impresión de PCBs para el diseño de circuitos. En [Barahona, 2013] se implementa un sistema de posicionamiento para orientar la impresión en una impresora 3D. Por otro lado, en [Shaowei, 2012] se presenta un método para modelar, identificar y compensar los errores producidos al momento de maquilar con máquinas CNC, mediante un interferómetro láser. En [Altintas, 2012] se presenta un algoritmo, basado en el uso de filtros, para la compensación de los errores de contorno producidos por las vibraciones de la máquina.

En este trabajo se propone un algoritmo que utiliza interpolación lineal para resolver el problema de la deformación de los materiales. El algoritmo permite escanear la deformación de la superficie del material o cama de fresado por medio de una sonda de contacto. En teoría si conocemos la inclinación que hay en diferentes puntos del material, podemos deducir que tanto se tiene que deformar el dibujo para compensar el desnivel. En la siguiente sección se presenta la metodología para llevar a cabo la implementación.

## **2. Métodos**

La figura 1 muestra el diagrama a bloques de las partes del CNC y el orden de los procesos de software.

### **Sistema de sondeo**

El sistema de sondeo se basa en el principio de los sistemas digitales (0 y 1). Para esto se elaboró una sonda de contacto. Se utiliza un PCB en blanco como una

placa conductora a la cual se le solda un cable que va conectado al Arduino, y otro cable que va del Arduino a la broca del enrutador, se utiliza un shield CNC V3 para arduino UNO, como se muestra en las figuras 2 y 3. Ambos funcionarán como un interruptor ya que la unión de estos dos se comportará como se ve en figura 4.

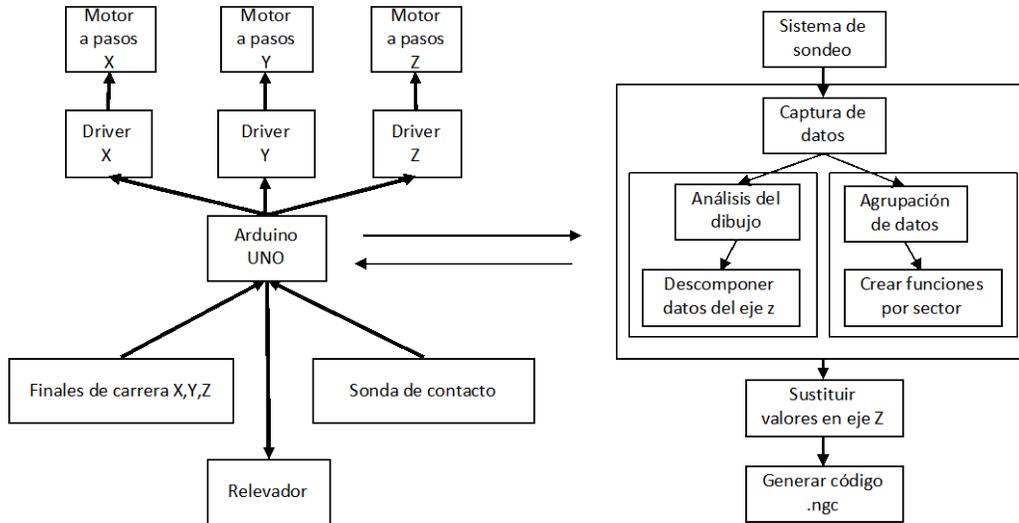


Figura 1 Diagrama de bloques del modelo experimental.



Figura 2 PCB y broca.

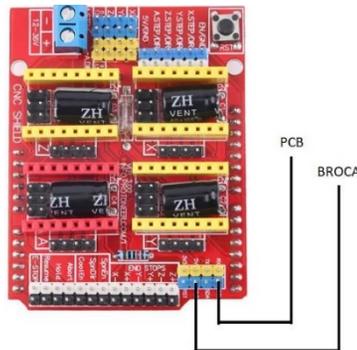


Figura 3 Se usaron los pines SCL Y GND de la shield CNC V3.

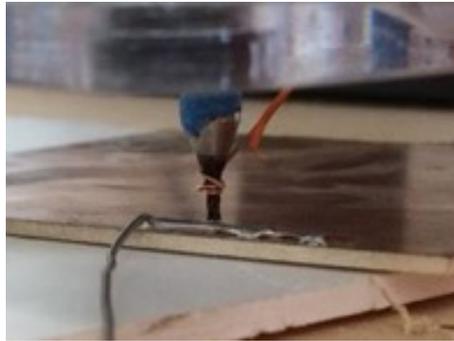


Figura 4 Momento en el que se cierra el circuito.

El funcionamiento de este sistema consiste en establecer las dimensiones del plano en milímetros, este dato es utilizado para limitar el área a escanear (figura 5). Como segundo paso, es necesario colocar puntos de referencia a una distancia específica en cada uno de los ejes (x, y, z.). A esta distancia le llamamos “resolución”. Mientras más grande sea la distancia entre cada punto mayor será el rango de error que tendrá la sonda al momento de escanear, por otro lado, mientras más pequeña sea esta distancia, se tendrá más precisión al momento de capturar los datos, aunque su proceso será más lento.

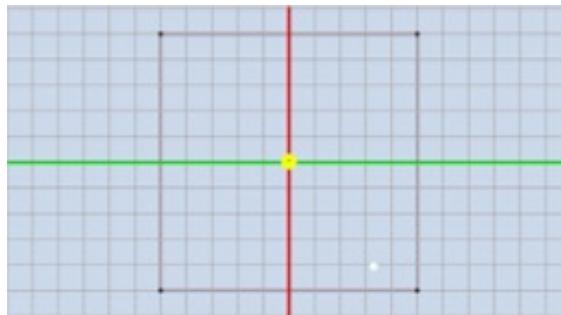


Figura 5 Establece dimensiones del plano ejemplo 10mmx10mm.

Para elegir la resolución es necesario considerar que tan desnivelado o deformado está el material, mientras más desnivel o curvatura tenga el material se requiera de una resolución más pequeña, es decir que la separación entre cada punto sea más corta. Por otro lado, si el material es plano o tiene una inclinación pequeña se puede optar por utilizar una resolución mayor, es decir que la distancia entre cada punto sea más distante, como se muestra en la figura 6.

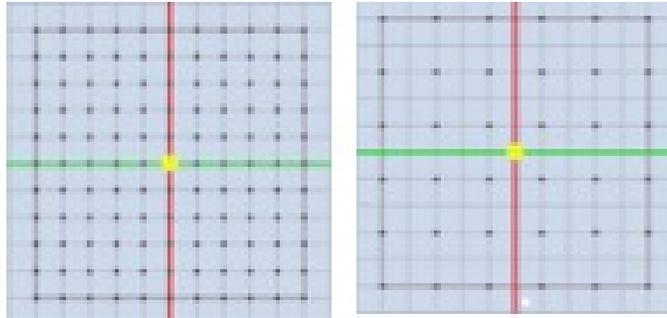


Figura 6 Izquierda con resolución 1 y derecha con resolución 2

### Captura de datos

Este proceso se divide en 2 subrutinas y 4 subprocesos: Se crean 3 matrices de ceros de  $n \times m$ , donde  $n$  es el número de puntos seccionados respecto al eje “X” y  $m$  es el número de puntos seccionados respecto al eje “Y”. Una vez creadas las matrices se realiza el sondeo del material. Este proceso consiste en hacer bajar el eje Z en cada uno de los puntos (guardados en las matrices X y Y) e ir guardando la posición. Cuando la broca (situada en el eje Z) toque la placa conductora se cerrará el circuito indicándole al Arduino que ha llegado al límite por lo tanto la posición Z será guardada en la tercera matriz, como se muestra en la figura 7.

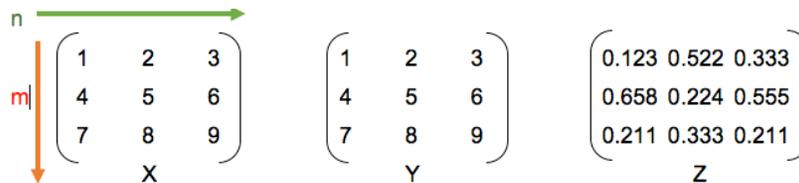


Figura 7 La matriz X y Y muestran las posiciones en las que se obtuvo Z.

### Agrupación de datos

Una vez capturado todos los datos se agrupan los datos de las matrices X y Y en submatrices de  $2 \times 2$ , las cuales son llamadas sectores, figura 8. El número de matrices se puede deducir del producto de  $n-1$  con  $m-1$ .

### Calcular funciones por sector

Para este subproceso se calcula las pendientes de los puntos: AB, AC, CD y DB. Usando la ecuación para la pendiente dados dos puntos,  $m=(y_2-y_1)/(x_2-x_1)$  y la



con la expectativa de que la herramienta esté cortando. Cuando se encuentra en el rango de G1 a G0, la máquina está realizando algún corte, y de G0 a G1, la máquina está localizando, no está cortando. Mientras que “X” y “Y” muestran la posición a la que se tienen que desplazar, por lo que hay que interpolar el eje “Z” entre cada desplazamiento del eje “X” y “Y”. La letra F M3 se utiliza para activar el relevador y por lo tanto el encendido del router y M2 el apagado del router que indica al usuario que se ha terminado el trabajo. Antes de cortar con el CNC el dibujo se capturan previamente los datos del desnivel de la cama de fresado con la sonda con una resolución de 1 muestra por cada 375 mm por lo que resulta una matriz de 3 x 3. La tabla 1 muestra el desarrollo del procedimiento para el ejemplo de la figura 9. Para saber el valor real que tendrá Z, para el primer corte, primero se localiza en que cuadrante esta respecto a X,Y (n,m). De la tabla se observa que el primer corte respecto a ‘Y’ tiene su origen en 25 y finaliza en 75, mientras que el eje ‘X’ se mantiene en 25 ya que se trata de una línea recta. En la tabla de datos capturados por la sonda se busca un sector que esté en el rango de 25-75.

Tabla 1 Datos obtenidos del dibujo.

Datos Cuadrado de 10mm x 10mm x 3mm					
	Origen del corte 1	Corte 1	Corte 2	Corte 3	Corte 4
Eje Z	25	-3	-3	-3	-3
Eje Y	25	75	75	25	25
Eje X	25	25	75	75	25

La tabla 2 muestra los datos capturados por la sonda, estos datos también se muestran en la figura 10. De la figura 10 se observa que el cuadrante abarca desde ‘Y0.0’ (m3, n1) hasta ‘Y375’ (m2, n1), así que se procede a calcular la función de la sonda respecto a Z, esta función corresponde a la ecuación general de la recta para dos puntos. La función resultante se usa para buscar el valor que tendrá ‘Z’ en el origen y el final del corte; al resultado será sumado o restado al valor original del dibujo en Z este proceso se repetirá en cada un de las líneas de código. La idea básica es conectar los 2 puntos dados en X y Y. La función interpolada es una línea recta entre los dos puntos que se interceptarán con el eje Z. El penúltimo proceso

consiste en remplazar los valores de Z por los nuevos valores obtenidos por la interpolación. Para esto se agregaron nuevas líneas de código en el vector. Por cada dirección de X y Y se agregará una nueva dirección en Z. La figura 11 muestra un ejemplo del fragmento de código, esta vez con los nuevos valores de Z.

Tabla 2 Datos capturados por la sonda.

Datos Capturados por la Sonda									
Cuadrante	X	Y	Z	X	Y	Z	X	Y	Z
n*m	n1			n2			n3		
m1	0	750	1.7372	375	750	1.3794	750	750	0.4138
m2	0	375	1.9912	375	375	0.9124	750	375	1.0594
m3	0	0	1.8712	375	0	0.9555	750	0	0.9207

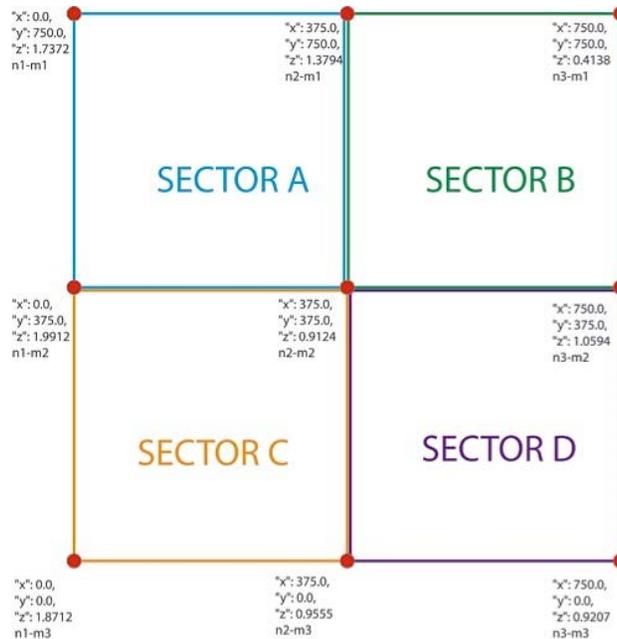


Figura 10 Ejemplo división en sectores para realizar los cortes, respecto a figuras 8 y 9.

Como se puede observar en la figura 11, ahora por cada línea de código o dirección se le anexa el inicio y fin de cada interpolación. Este proceso se hará en todo el dibujo dando como resultado un nuevo dibujo deformado, pero que a su vez encaja casi perfectamente con la deformación del material. Este algoritmo que entra en la categoría plugin se agregó al núcleo del software “Universal G-Code Sender” (UGS) por medio de la plataforma Netbeans [Shapeoko, 2018]. Otras de las modificaciones

que se realizó fue traducir parcialmente el programa, puesto que está en inglés, todo esto con la intención de dar mayor comodidad al usuario final. Una aproximación del algoritmo se presenta en diagrama de flujo de la figura 12 y 13.

T1	
G17	
G21	
G0Z21.8712	
G0X0.0000Y0.0000S1.2000M3	Rango de interpolación 1
G0X25.0000Y25.0000Z5.0000	para buscar el origen
G1Z-2.2398F508	
G1Y75.0000F1270.0	Rango de interpolación 2, Corte 1
G1Z-2.8712	
G1X75.0000	Rango de interpolación 3, Corte 2
G1Z-2.9555	
G1Y25.0000	Rango de interpolación 4, Corte 3
G1Z-2.9124	
G1X25.0000	Rango de interpolación 5, Corte 4
G0Z-2.2398	
G0X150.0000Y150.0000Z.25.2398	Rango de interpolación 6, para localizar una nueva posición

Figura 11 Nueva interpolación en los ejes.

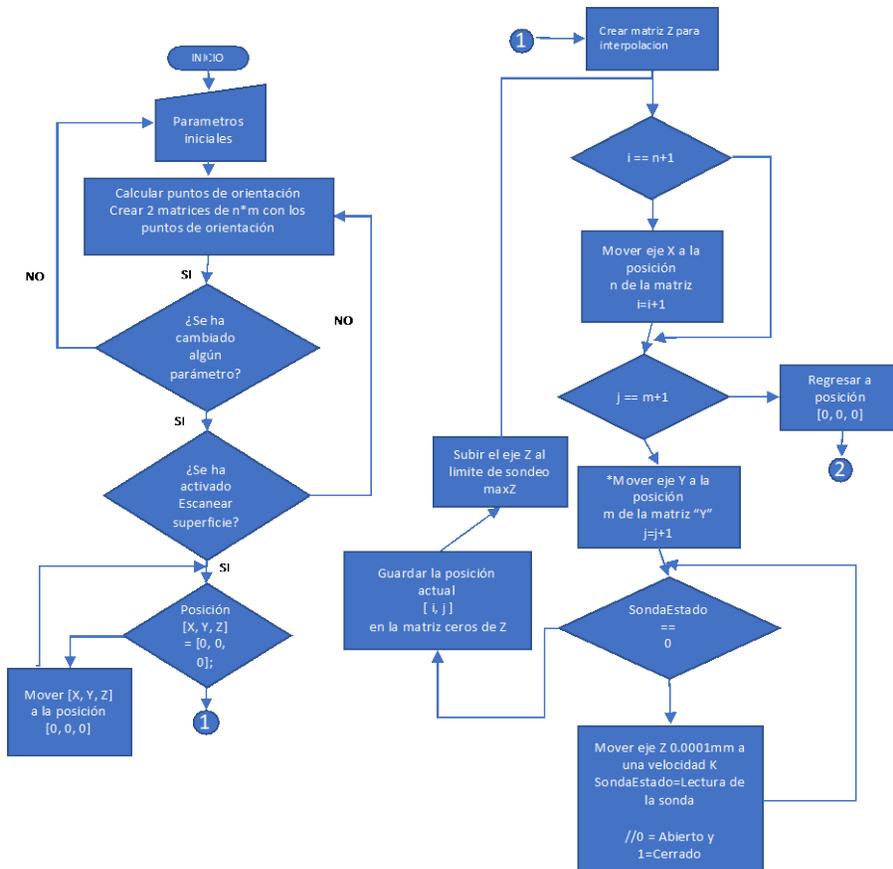


Figura 12 Diagrama de flujo del algoritmo empleado parte 1.

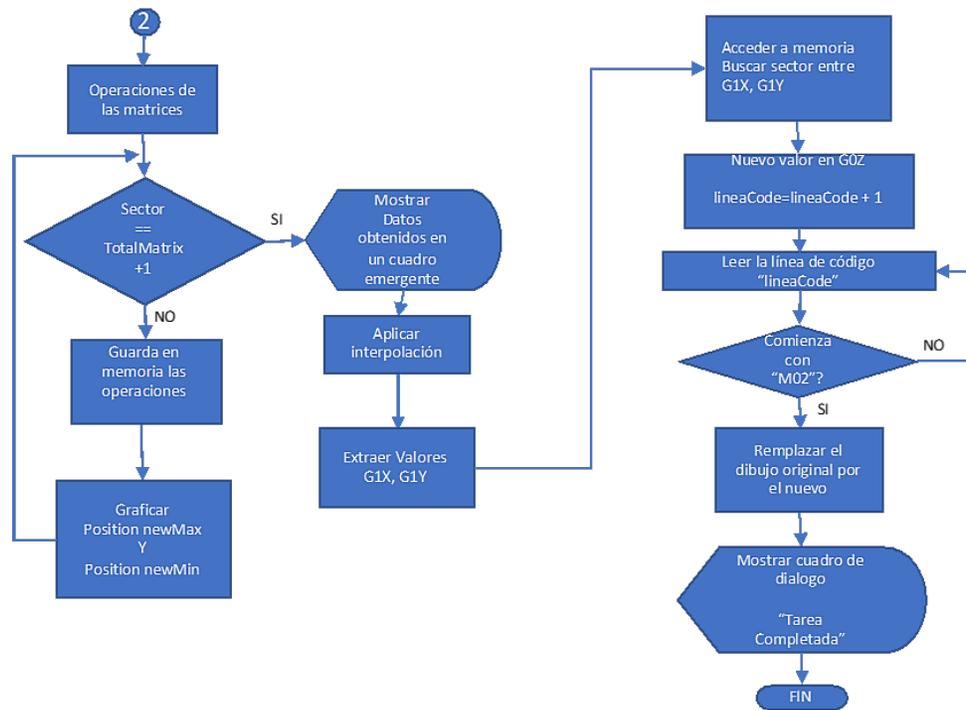


Figura 13 Diagrama de flujo del algoritmo empleado parte 2.

### 3. Resultados

Para llevar a cabo el experimento se diseñó un “PCB” y un dibujo para ser grabado en madera con una deformación exagerada, para esto se usó el software “Inkscape” [Norton, 1999]. Posteriormente se importó al “UGS” y se abrió la extensión que se añadió. En el caso del “PCB” se puede usar directamente la superficie de la baquelita para la sonda de contacto.

En la figura 14 se puede observar el dibujo original y la forma en la que se secciona el dibujo. Los puntos negros representan las muestras capturadas por la sonda sobre el material. En la figura 15 se muestra el resultado obtenido después de sondear toda la superficie del material. Como se puede observar en los colores, el color verde indica que el rango de deformación es mínimo, casi insignificante, mientras que el rojo indica que la superficie cambia bruscamente, también muestra un mensaje con los datos obtenidos del sondeo. Por último, la figura 16 muestra como se ve el dibujo una vez aplicado el algoritmo.

De las figuras 15 y 16 se puede observar que algunas letras como la ‘A’, ‘C’ y ‘N’ tienen una ligera deformación, mientras que las curvas del logo se ven como si

tuvieran irregularidades. También se observa que los círculos o las curvas se deforman con líneas rectas, esto es debido a que se usó interpolación lineal en vez de interpolación polinomial.

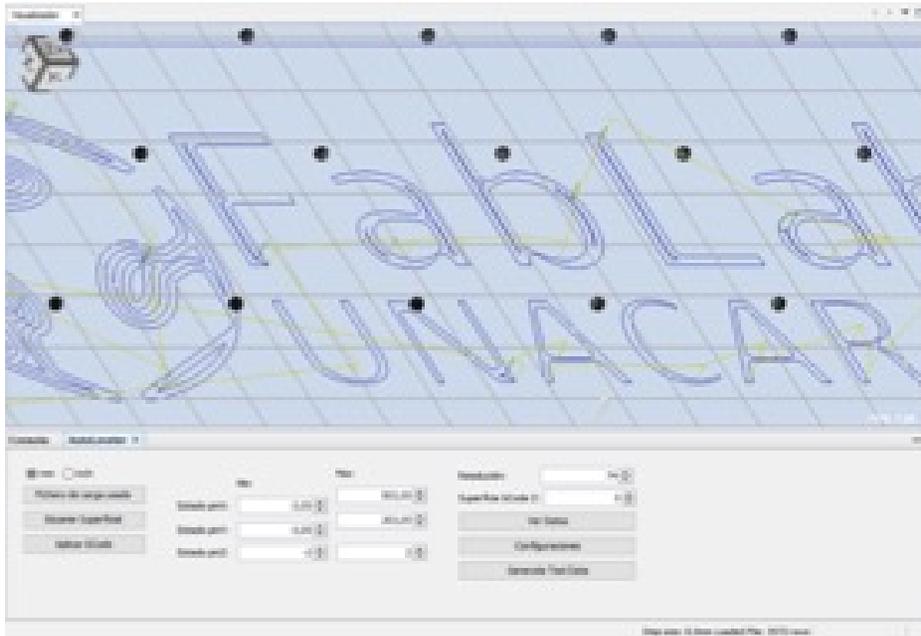


Figura 14 Dibujo original con una resolución de 74.

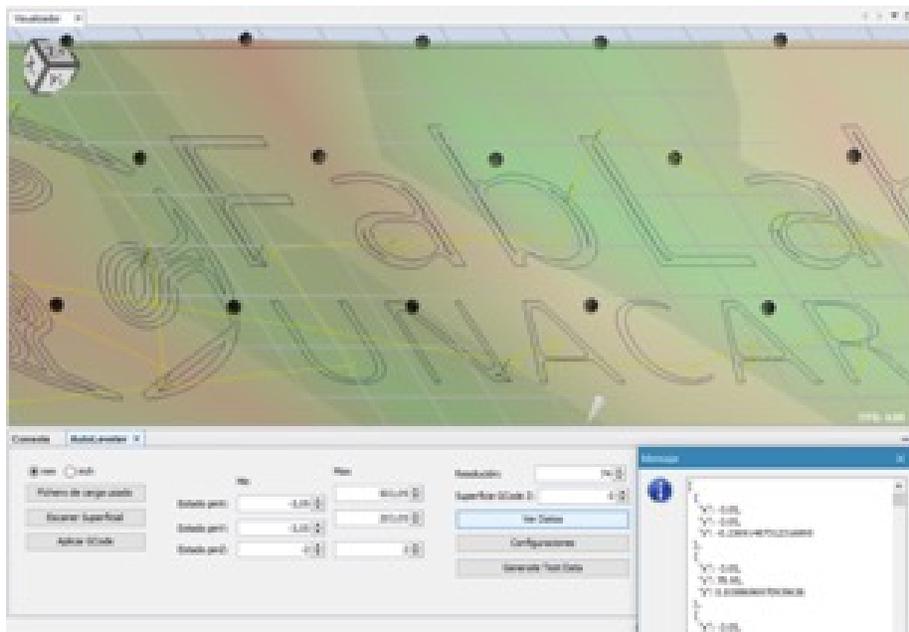


Figura 15 Recopilación de datos de áreas verdes= deformación mínima, áreas rojas = deformación más alta.

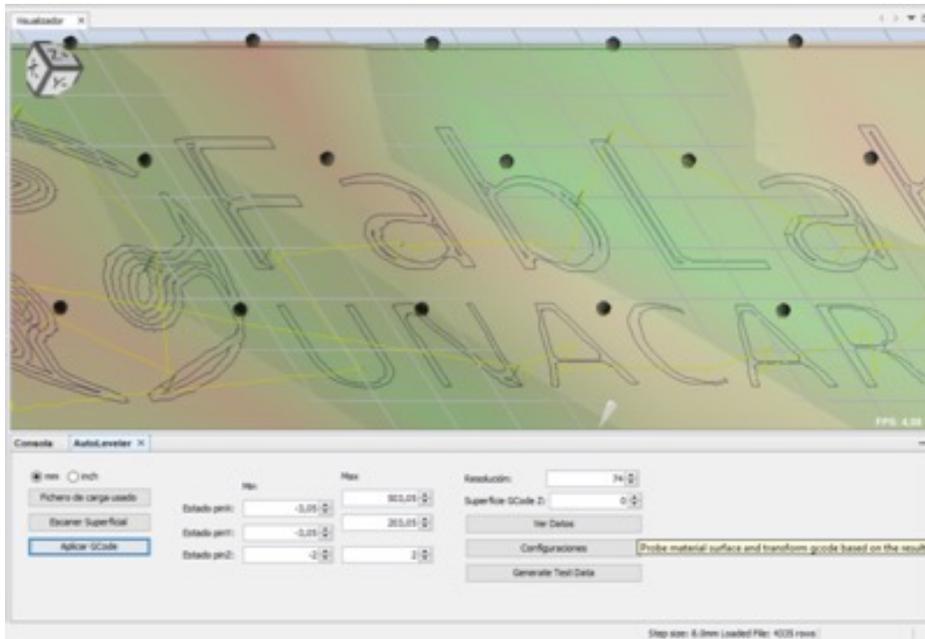


Figura 16 Aplicando interpolación.

Adicionalmente, se realizó el mismo proceso para un PCB. Primero se realizó una prueba de control, sin el auto nivel, como se muestra en la figura 17. Se puede observar que los cortes son irregulares y con mala definición. Posteriormente se sondea la superficie de la baquelita aprovechando el área de cobre donde se solda un cable directamente al Arduino. Este último resultado, utilizando el algoritmo de interpolación lineal, se muestra en la figura 18. Como se puede observar la precisión del PCB de la figura 18, con respecto al PCB de la figura 17, es ampliamente superior y limpia.

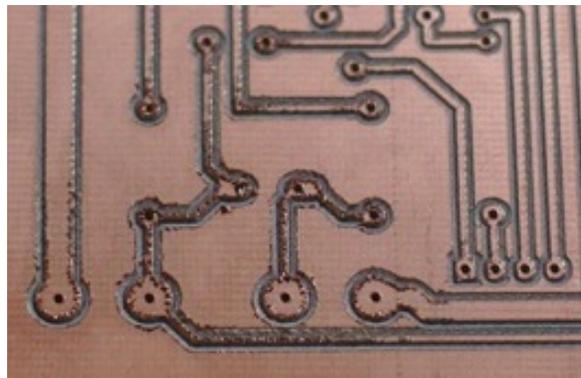


Figura 17 Implementación de un PCB sin interpolación.

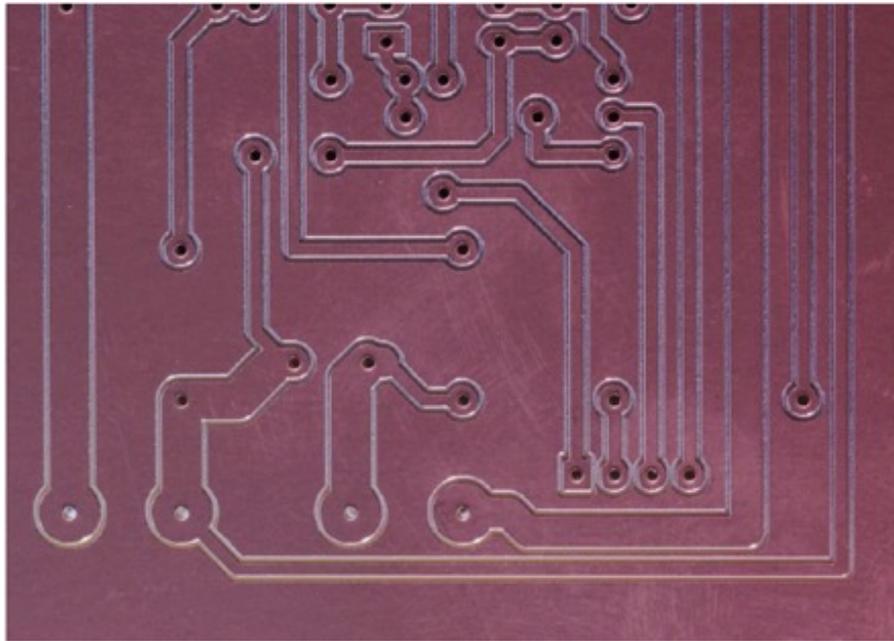


Figura 18 Implementación de un PCB con interpolación.

#### 4. Discusión

Existen algunos inconvenientes con este tipo de implementación, por ejemplo: cuando el número de líneas de código es grande, se requiere que el tamaño del buffer para la comunicación con el Arduino sea más grande; además, con este tipo de implementación el eje Z trabajará más, sin mencionar el esfuerzo que hace cada que sube y baja el router, por lo tanto se calentará más, sin embargo, esto es fácil de solucionar, basta con que en lugar de trabajo tenga una buena ventilación, o no se le exija trabajos muy largos al CNC. Por otro lado, para la fabricación de circuitos impresos los resultados fueron sorprendentes, el tiempo de maquinado de un PCB es mucho menor que cuando se utiliza cloruro férrico, además es más preciso al momento de hacer pistas estrechas. Este algoritmo funciona únicamente para la extensión.gnc de la distribución de “*ShapeOko*”.

#### 5. Conclusiones

En este trabajo se presentó una alternativa de mejora como software, para la implementación de una máquina CNC de código abierto, el cual fue modificado de acuerdo a las necesidades del presente trabajo. Al tener una “CNC Casera”,

fabricada con materiales reciclados y madera muchas veces no se obtienen trabajos de buena calidad, sin embargo, el uso del software de autonivelación resuelve muchas de las imperfecciones al momento de la implementación. Adicionalmente al auto nivelar la superficie de la cama de trabajo de la CNC puede servir como un límite por software en caso de no tener finales de carrera físico y así evitar que la máquina se termine destruyendo por la fuerza de los motores a paso o del router de fresado. Los resultados obtenidos son satisfactorios, mostrando que el algoritmo utilizado es eficiente para los casos aquí planteados.

## **6. Bibliografía y Referencias**

- [1] Altintas Y. and Khoshdarregi M. R.. Contour error control of CNC machine tools with vibration avoidance, *CIRP Annals*, Elsevier, 61(1), 335-338, 2012.
- [2] Barahona E. A. T., León-Medina J. X. and Díaz E. T. Sistema de posicionamiento aplicado a la técnica de impresión 3d modelado por deposición fundida. *Revista de investigación, desarrollo e innovación*, 3(1), 25-32, 2013.
- [3] De Luca Penaccia, A., De la Fraga, L. G. (2009). A tool for errors detection in printed circuit boards production. *Journal of Applied Research and Technology*, 7(1), 41-50.
- [4] Luis S. (2015). Desnivel en la mesa de trabajo: <https://frs-cnc.com/foro1/mach/25243-desnivel-en-la-mesa-de-trabajo>, mayo de 2018.
- [5] Kaushik S. and Ashraf J. Automatic PCB defect detection using image subtraction method. *International Journal of Computer Science and Network*, 2012.
- [6] Norton R. Diseño de máquinas, AlfaOmega. 1999.
- [7] Shaowei Zhua, Guofu Dinga, Shengfeng Qinb, Jiang Leiab, Li Zhuanga and Kaiyin Yan. Integrated geometric error modeling, identification and compensation of CNC machine tools, *International Journal of Machine Tools and Manufacture*, Elsevier, 52(1), 24-29, 2012.
- [8] ShapeOko. (2018). G-Code de shapeoko: <https://www.shapeoko.com/wiki/index.php/G-Code>, mayo de 2018.