



industriales  
etsii

Escuela Técnica  
Superior  
de Ingeniería  
Industrial

# UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

## Programación de trayectorias en 5 ejes para generación de álabes de un compresor centrífugo con interpolación circular

**TRABAJO FIN DE MÁSTER**

MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

**Autor:** Antonio Manuel Almaida Méndez  
**Director:** Óscar de Francisco Ortiz  
**Codirector:** Manuel Estrems Amestoy



Universidad  
Politécnica  
de Cartagena

Cartagena, 8 de septiembre de 2021



## Resumen

En este proyecto se ha implementado el uso de trayectorias de interpolación circular para la generación de los álabes de un compresor centrífugo con limitación de error cordal. Con esto se persigue la simplificación matemática y de programación de estos componentes con la finalidad de reducir el número de instrucciones del programa y estabilizar su mecanizado, logrando así un mejor acabado.

Para ello se ha desarrollado un programa en Visual Basic que genera, a partir de una nube de puntos que define los caminos de herramienta para el mecanizado de los álabes del rodete, unas trayectorias circulares formadas por arcos unidos bajo continuidad de tipo  $C^1$ , que difieren de los puntos originales una distancia determinada.

Una vez generadas las instrucciones para efectuar el mecanizado del rodete se pudo observar que el número de líneas de código necesarias para realizar el mecanizado se había reducido de forma considerable. Con el fin de examinar las diferencias existentes entre las trayectorias rectilíneas y circulares en el mecanizado, se fabricaron dos rodetes idénticos con los mismos parámetros de mecanizado de modo que las diferencias que pudieran observarse en estos fuesen debidas únicamente a las trayectorias del husillo.

En el rodete mecanizado con trayectorias rectilíneas se pudo observar un acabado más rugoso con efecto escamoso debido al avance intermitente del husillo. Este efecto es provocado por el elevado número de instrucciones, con pequeños desplazamientos, necesarias para lograr el mecanizado del álabe. En cambio, en el rodete mecanizado con trayectorias circulares el acabado era más refinado puesto que el husillo recorría las trayectorias de una forma mucho más fluida.

Además, se tomaron medidas del tiempo de mecanizado de uno de los álabes principales del rodete y se pudo observar que el tiempo necesario para mecanizar el álabe con las trayectorias circulares era un 2,5% menor respecto a las trayectorias rectilíneas.

Por ende, se pudo verificar que las ventajas teóricas ofrecidas por las trayectorias de interpolación circular se cumplían y, por tanto, el uso de este tipo de trayectorias supone una mejora en los procesos de mecanizado.



## Índice

1.	Estado del arte .....	1
2.	Introducción .....	5
2.1.	Centro de fresado multieje .....	5
2.1.1.	Fresadora de 5 ejes .....	5
2.2.	Control Numérico .....	7
2.2.1.	Código G .....	7
2.3.	Interpolación circular .....	8
2.4.	Grado de continuidad de una curva .....	10
3.	Metodología y aplicación .....	11
3.1.	Trayectoria de los álabes del rodete .....	11
3.2.	Programa aproximación circular .....	13
3.2.1.	Descripción .....	13
3.2.1.1.	Continuidad C1 .....	14
3.2.1.2.	Distancia punto – arco .....	15
3.2.1.3.	Error cordal .....	16
3.2.2.	Programación .....	17
3.2.3.	Funcionamiento .....	17
4.	Pruebas de laboratorio .....	21
4.1.	Elaboración del Código G .....	21
4.1.1.	Anverso álabe principal .....	21
4.1.2.	Reverso álabe principal .....	22
4.1.3.	Anverso álabe intermedio .....	22
4.1.4.	Reverso álabe intermedio .....	23
4.2.	Mecanizado .....	24
4.2.1.	Interpolación lineal .....	25
4.2.2.	Interpolación circular .....	25
5.	Análisis de resultados .....	27
5.1.	Acabado superficial .....	27
5.2.	Tiempo .....	29
	Conclusiones .....	31
	Bibliografía .....	33
	Anexos .....	35
	Anexo I. Trayectorias discretizadas de los caminos de herramienta. ....	37
	Anexo II. Código Visual Basic .....	39
	Anexo III. Código G para el mecanizado de los álabes. ....	55
	Anexo IV. Representación de las trayectorias generadas. ....	57



## Índice de figuras

Figura 1.1: Algoritmo para aproximación de arcos circulares.....	1
Figura 1.2: Algoritmo para aproximación de arcos circulares.....	2
Figura 2.1: Sentido de G2 y G3 para diferentes planos de trabajo .....	8
Figura 2.2: Continuidad C0.....	10
Figura 2.3: Continuidad C1.....	10
Figura 3.1: Círculos de Villarceau .....	11
Figura 3.2: Plano bitangente al toroide que pasa por su centro.....	11
Figura 3.3: Dimensiones de la probeta a mecanizar. ....	11
Figura 3.4. Probeta a mecanizar.....	12
Figura 3.5: Desviación del camino de herramienta con respecto a la fibra media.....	12
Figura 3.6: Anverso de los álabes.....	13
Figura 3.7: Reverso de los álabes .....	13
Figura 3.8: Unión de dos arcos con continuidad de clase C0.....	14
Figura 3.9: Unión de dos arcos con continuidad de clase C1 con cambio de concavidad.....	14
Figura 3.10: Unión de dos arcos con continuidad de clase C1 sin cambio de concavidad .....	15
Figura 3.11: Distancia de un punto a un arco .....	16
Figura 3.12: Error cordal para un segmento .....	16
Figura 3.13: Trazo de la curva aleatoria .....	18
Figura 3.14: Comparación de trayectorias lineales y circulares para la curva aleatoria.....	20
Figura 4.1: Cero pieza I.....	25
Figura 4.2: Cero pieza II.....	25
Figura 4.3: Mecanizado del rodete .....	25
Figura 4.4: Situación de la herramienta en la máquina CNC Lagun L650 .....	26
Figura 4.5: Orientaciones positivas de la herramienta .....	26
Figura 5.1: Localización muestras microscopio.....	27
Figura 5.2: Imagen reglada del microscopio de la zona A para interpolación lineal .....	28
Figura 5.3: Imagen reglada del microscopio de la zona A para aproximación circular.....	28
Figura 5.4: Imagen del microscopio de la zona B para interpolación lineal.....	29
Figura 5.5: Imagen del microscopio de la zona B para aproximación circular .....	29
Figura 5.6: Imagen reglada del microscopio de la zona B para interpolación lineal.....	29
Figura 5.7: Imagen reglada del microscopio de la zona B para aproximación circular .....	29
Figura Anexo IV. 1: Aproximación circular del camino de herramienta para álabe principal.....	57
Figura Anexo IV. 2: Aproximación circular del camino de herramienta para álabe intermedio.	57



## *Índice de tablas*

Tabla 3.1: Puntos de la curva aleatoria .....	18
Tabla 3.2: Arcos generados para la curva aleatoria .....	19
Tabla 3.3: Distancia punto – arco para la curva aleatoria.....	19
Tabla 4.1: Resultados de la aproximación circular para el anverso del álabe principal.....	21
Tabla 4.2: Arcos generados para el anverso del álabe principal.....	21
Tabla 4.3: Resultados de la aproximación circular para el reverso del álabe principal .....	22
Tabla 4.4: Arcos generados para el reverso del álabe principal .....	22
Tabla 4.5: Resultados de la aproximación circular para el anverso del álabe intermedio.....	23
Tabla 4.6: Arco generado para el anverso del álabe intermedio .....	23
Tabla 4.7: Resultados de la aproximación circular para el reverso del álabe intermedio .....	23
Tabla 4.8: Arco generado para el reverso del álabe intermedio.....	23
Tabla Anexo I: Nube de puntos de los caminos de herramienta .....	37



## 1. Estado del arte

La aproximación de trayectorias mediante arcos circulares es una técnica utilizada en el área de fabricación que persigue la reducción del volumen de datos de control numérico, así como la reducción del tiempo y la estabilización del proceso de fabricación. Su principal campo de aplicación es el mecanizado por fresado, aunque también se emplea en otros ámbitos como la fabricación aditiva.

Esta técnica persigue sustituir las trayectorias rectas que unen los diferentes puntos que componen una nube de puntos por trayectorias curvas, de modo que la distancia existente entre ambas trayectorias sea inferior a la deseada.

En el estudio realizado por Estrems et al. [1] se desarrolla un nuevo método para aproximar trayectorias circulares entre puntos. Tras mecanizar una superficie toroidal elíptica con el método desarrollado y compararla con el mecanizado propuesto por un software CAM (PTC Creo Parametric), se pudo observar que el nuevo método desarrollado proporcionaba una mejora considerable de la rugosidad superficial.

El procedimiento que desarrollaron para unir dos puntos  $P_1$  y  $P_2$  mediante un arco en el espacio 3D se describe a continuación. Las nomenclaturas están representadas en la Figura 1.1.

1. El vector  $\mathbf{l}$  une los puntos  $P_1$  y  $P_2$ .
2. Se calcula el punto medio  $M_1$  de  $\mathbf{l}$ .
3. Se obtiene el punto  $P_3$  como la intersección de la mediatriz del vector  $\mathbf{l}$  y la superficie  $\mathbf{X}(u, v)$ . Se usa el método de Newton-Raphson en dos dimensiones para obtener  $u$  y  $v$  del punto  $P_3$ .
4. Los puntos  $P_1, P_2$  y  $P_3$  son coplanarios con el centro del arco  $C$ . En este plano se calcula la mediatriz entre los puntos  $P_1$  y  $P_3$ , y también entre los puntos  $P_2$  y  $P_3$ , obteniendo los puntos  $M_2$  y  $M_3$  respectivamente. Siguiendo el procedimiento del punto 3 se calculan los puntos  $P_4$  y  $P_5$ . Calculando las intersecciones de las mediatrices se calcula el centro de dicho arco  $C$ .
5. Los errores cordales se calculan como las distancias existentes entre los puntos  $M_2$  y  $M_3$ , y los puntos  $P_4$  y  $P_5$  respectivamente. Así, el error cordal del arco será el mayor de esos dos valores.

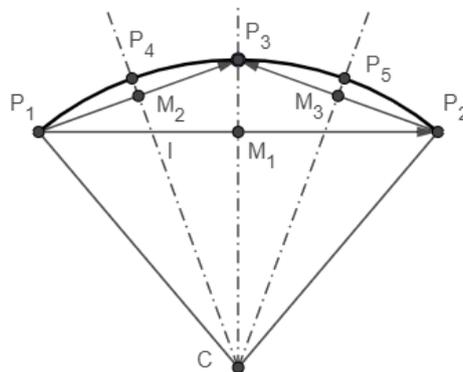


Figura 1.1: Algoritmo para aproximación de arcos circulares. Estrems et al. [1]

Otro estudio en este campo es el realizado por Kawasaki et al. [2] donde se pretende reducir el volumen de datos de control numérico y acortar el tiempo de mecanizado de las máquinas herramienta aplicando la aproximación de arcos circulares. Para ello, se desarrolla un algoritmo, mostrado en la Figura 1.2, que permite obtener dichas trayectorias circulares partiendo de una nube de puntos e indicándole la tolerancia máxima admisible y el número máximo de puntos que pueden componer cada arco.

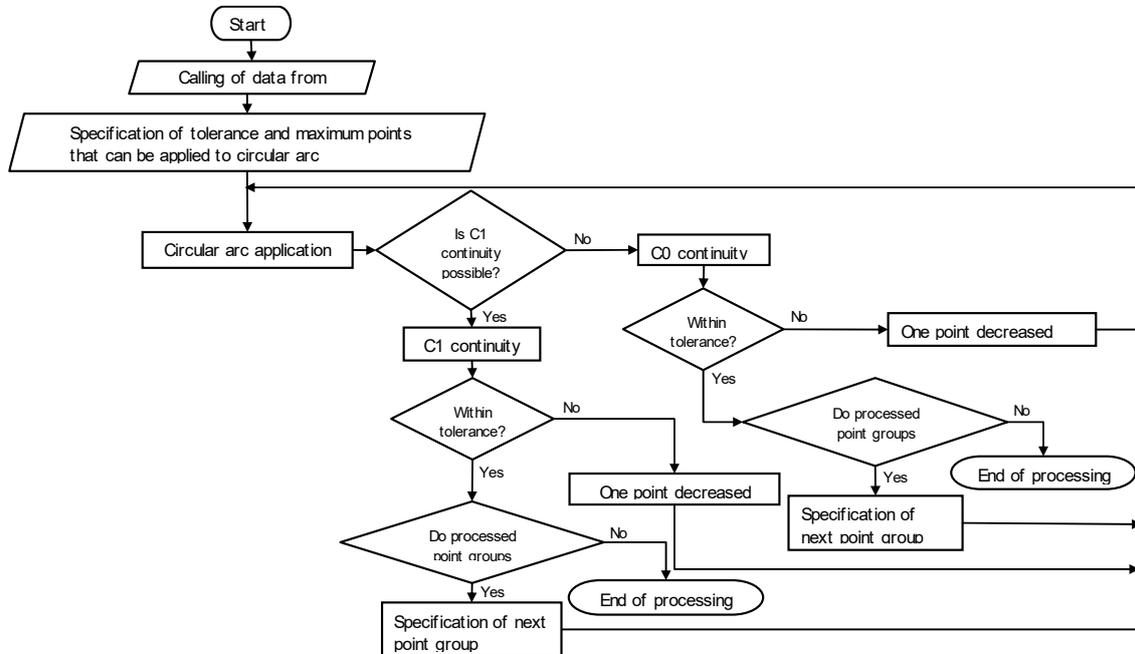


Figura 1.2: Algoritmo para aproximación de arcos circulares. Kawasaki et al. [2]

Tras comparar los resultados obtenidos mediante aproximación lineal y circular se pudo observar que el grupo de puntos pudo reducirse a 1/6, y el volumen de datos a 1/5. En relación con acabado del mecanizado no se pudo observar ningún cambio significativo.

También con el objetivo de reducir el tiempo de mecanizado se presenta el estudio elaborado por G. Vickers y C. Bradley [3], donde se desarrolla un algoritmo que maximiza el número de movimientos circulares para la generación de superficies. Al realizar movimientos cortos interpolados linealmente se observó que solo durante el 10% del tiempo de mecanizado se alcanzaba la velocidad máxima de avance, el resto del tiempo se empleaba en acelerar, desacelerar o realizar pausas entre instrucciones. Finalmente, se consiguió reducir el tiempo de mecanizado en 5 veces el tiempo mediante interpolación lineal.

Sobre esta línea de trabajo se presenta también el estudio realizado por P. Kaewsaiha y N. Dejduong [4] donde se analiza la velocidad de avance para diferentes tipos de trayectorias: interpolación lineal, interpolación circular y las trayectorias generadas por el método desarrollado en dicho estudio que combina la interpolación lineal e interpolación circular para generar curvas Bézier. Los resultados experimentales mostraron un avance más suave y constante de las trayectorias generadas por el algoritmo propuesto, seguido de las trayectorias circulares.

Otro estudio, esta vez enfocado en la fabricación aditiva, es el desarrollado por Hochgesang [5]. En dicho estudio se desarrolla un software con el objetivo de sustituir las instrucciones G0 y G1 (movimientos lineales) por G2 y G3 (movimientos circulares). Con esto persigue comprimir los archivos GCode y reducir el tartamudeo causado por la falta de búfer en el controlador de la impresora 3D. A continuación, se describe el algoritmo desarrollado:

1. Se encuentran tres puntos consecutivos y se genera un arco que pasa a través de dichos puntos. El arco generado siempre pasa por el primer y último punto, pero no por los intermedios.
2. Se comprueba que la desviación máxima es inferior a la preestablecida.
3. Si el error cometido es inferior al valor de referencia, se añade el siguiente punto al arco y se vuelve a comprobar la desviación máxima.
4. Se repite el paso anterior hasta que la desviación máxima es superior al valor referencia. Cuando esto ocurre, se elimina el último punto añadido y se genera el arco final.
5. Una vez se ha completado un arco, se crea un nuevo arco repitiendo el bucle descrito.

Este software se encuentra actualmente en desarrollo puesto que presenta algunos errores. Además, la unión entre los diferentes arcos la realiza únicamente bajo continuidad  $C^0$ , lo que puede comprometer el acabado final de la pieza a fabricar, así como la estabilidad en un proceso de mecanizado. En el capítulo 2.4 se define el grado de continuidad de una curva y se muestran unos ejemplos representativos de las diferentes tipologías.

También se han realizado otros trabajos teóricos sobre el desarrollo de algoritmos para la obtención de trayectorias por aproximación de arcos, si bien no se han ejecutado estas trayectorias generadas en máquinas herramientas para observar las mejoras que puedan darse. Algunos ejemplos de este tipo de trabajos son los estudios realizados por X.Yang [6] y por D. Meek y D. Walton [7], donde se desarrollan algoritmos para aproximar curvas planas mediante arcos con continuidad  $C^1$ .

Otro estudio similar es el llevado a cabo por Qiu et al. [8], en el que se desarrolla un algoritmo de interpolación de arcos circulares con tres características fundamentales:

1. El error de interpolación satisface la precisión deseada en cualquier segmento de la curva.
2. Los parámetros de los arcos circulares dependen de la precisión deseada y el número de arcos es el menor.
3. Los arcos son compatibles con la curva en concavidad y convexidad, y tienen casi el mismo radio de curvatura.

Finalmente, L. Piegl y W. Tiller [9] proponen un algoritmo para generar dichas trayectorias mediante biarcos, cumpliendo también con la continuidad  $C^1$ .



## 2. Introducción

### 2.1. Centro de fresado multieje

El mecanizado multieje es un proceso de mecanizado que surge de forma posterior al mecanizado tradicional de tres ejes, en el que herramientas controladas por un ordenador de control numérico (CN) se trasladan a lo largo de tres ejes para producir piezas de diferentes materiales.

En el mecanizado multieje se consigue que dichas herramientas controladas por un ordenador de control numérico se muevan con cuatro o más grados de libertad, realizando movimientos angulares alrededor de uno o varios ejes. En consecuencia, consiguen mejorar las características del mecanizado convencional como pueden ser:

- Disminución del tiempo de trabajo ya que las piezas no tienen que ser giradas manualmente por un operario.
- Mejor acabado superficial debido a las trayectorias tangenciales que realiza la herramienta durante el mecanizado.
- Fabricación de piezas más complejas como consecuencia del mayor número de grados de libertad.

El mecanizado multieje engloba diferentes operaciones como puede ser corte con láser, corte con chorro de agua o fresado, que es el más frecuente.

#### 2.1.1. Fresadora de 5 ejes

Este tipo de máquinas poseen el número de grados de libertad suficientes para mecanizar piezas complejas con superficies curvas. El mecanizado de cinco ejes implica el movimiento de la pieza y/o herramienta a lo largo de cinco ejes de forma simultánea, es decir, con cinco grados de libertad. Estos cinco grados de libertad se consiguen con la traslación a lo largo de tres ejes ortogonales y la rotación alrededor de dos de estos ejes.

Para lograr los cinco grados de libertad existen multitud de combinaciones diferentes de ejes y movimientos, pero todos ellos se pueden agrupar en tres grandes grupos:

- Mesa-Mesa. Pueden ser horizontales o verticales. En este tipo de fresadoras todos los movimientos rotatorios, excepto el del husillo, se realizan a través de las mesas. La mesa principal soporta una segunda mesa rotatoria, con un movimiento de rotación independiente al de la mesa principal, donde se fija la pieza que va a ser mecanizada. En la Figura 2.1 se puede observar con mayor detalle la arquitectura constructiva de dichas máquinas herramientas.

Este tipo de fresadoras es el más simple y extendido ya que se puede obtener añadiendo las mesas rotatorias a una máquina de mecanizado convencional de tres ejes.

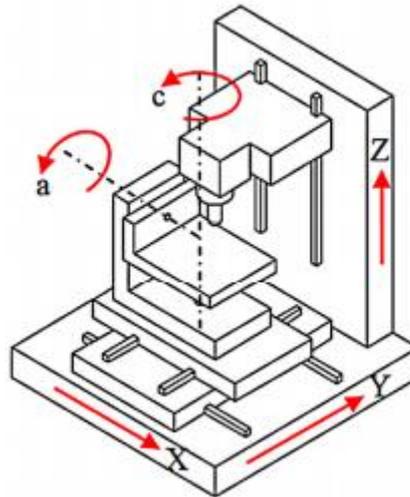


Figura 2.1: Fresadora de 5 ejes Mesa – Mesa. Farouki et al. [10]

- Mesa-Cabezal. Como su propio nombre indica este tipo de fresadoras poseen una mesa rotatoria y un cabezal basculante, que es el que porta la herramienta. En consecuencia, es el encargado de soportar las fuerzas de corte propias de cada proceso. En la Figura 2.2 se muestra un esquema donde se indican los giros realizados por cada uno de los componentes.

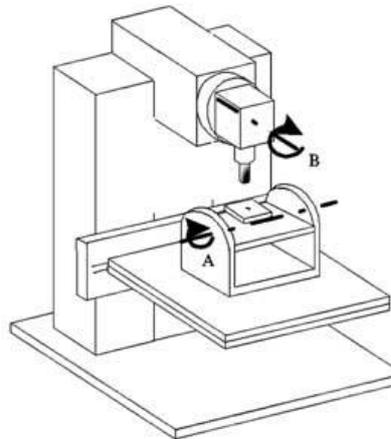


Figura 2.2: Fresadora de 5 ejes Mesa – Cabezal. Farouki et al. [10]

Este tipo de fresadoras es el más versátil ya que puede mecanizar piezas grandes y pesadas.

- Cabezal-Cabezal. Pueden ser horizontales o verticales. En este tipo de fresadoras todos los movimientos rotatorios son realizados por el cabezal de la máquina, consiguiendo así una completa orientación de la herramienta. En algunos casos puede llegar a cambiarse el cabezal durante el mecanizado, no solo la herramienta. Aunque pueden llegar a mecanizar geometrías complejas, su movimiento está limitado.

A continuación, se muestra una imagen representativa de este tipo de máquinas.

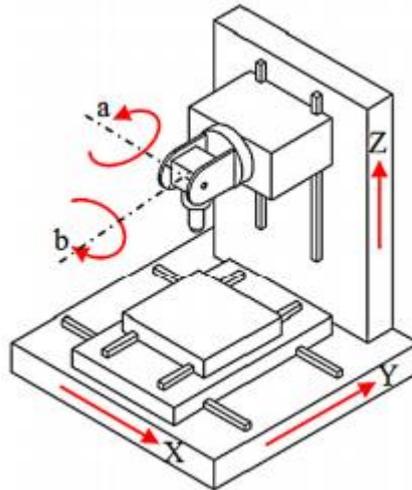


Figura 2.3: Fresadora de 5 ejes Cabezal – Cabezal. Farouki et al. [10]

## 2.2. Control Numérico

El control numérico CN es una forma de automatización programable donde las acciones mecánicas de una máquina herramienta o la posición de un elemento físico, normalmente una herramienta, son controladas por un programa que contiene datos alfanuméricos codificados.

Existen dos métodos para la elaboración de los programas que contienen toda la información necesaria para realizar el mecanizado completo de una pieza:

- Programación manual. Un operario es el encargado de elaborar cada una de las instrucciones necesarias para el mecanizado de una pieza por medio de razonamientos y cálculos.
- Programación automática. Las instrucciones son elaboradas por un ordenador. Se parte del diseño CAD (Computer-Aided Design) de una pieza y mediante un software de CAM (Computer-Aided Manufacturing) se genera el plan de operación y los caminos de herramienta. Finalmente se obtiene, tras un postprocesado, la lista de instrucciones necesarias para el mecanizado de una pieza.

Para fabricar una pieza con el uso del control numérico se ejecuta el programa, que contiene las instrucciones necesarias para el mecanizado de la misma, en el software informático de un ordenador que controla los movimientos de la herramienta en relación a los ejes de coordenadas de la máquina. De esta forma se consigue incrementar la producción, mejorar la precisión de las piezas finales, fabricar piezas con geometrías complejas, etc.

### 2.2.1. Código G

El código G, también conocido como RS-274, es el lenguaje de programación más usado en control numérico. Se emplea para ordenar acciones específicas a la máquina herramienta, desde movimientos simples hasta características más complejas como el cambio de herramienta.

Los caracteres más usados comúnmente, regidos bajo la norma DIN 66024 y 66025, son los siguientes:

- N: dirección correspondiente al número de bloque o secuencia.
- X, Y, Z: direcciones correspondientes a dichos ejes de la máquina herramienta.
- G: dirección correspondiente a funciones preparatorias.
- M: dirección correspondiente a las funciones auxiliares o complementarias.
- F: dirección correspondiente a la velocidad de avance.
- S: dirección correspondiente a la velocidad de rotación del husillo principal.
- I, J, K: direcciones utilizadas para programar arcos de circunferencia.
- T: dirección correspondiente al número de herramienta.
- A: ángulo
- R: radio
- P: programa

### 2.3. Interpolación circular

Se conoce como interpolación circular al movimiento curvo entre dos puntos, es decir, al desplazamiento desde un punto inicial hasta un punto final siguiendo una trayectoria circular.

La interpolación circular requiere principalmente de 5 datos: el punto final, el radio de curvatura, el centro, la velocidad de avance y una dirección de movimiento; y puede dar lugar a un movimiento circular completo o parcial.

Este tipo de movimientos son más exigentes en las máquinas CNC (Control Numérico por Computadora) ya que hay que coordinar dos ejes con gran precisión y exactitud, y se inicia a través de los códigos modales G2 y G3.

Ambos comandos indican interpolación circular. La diferencia reside en el sentido que se recorre el arco (sentido horario u antihorario). Este sentido depende del plano de trabajo (XY, XZ, YZ) tal y como se muestra en la figura siguiente:

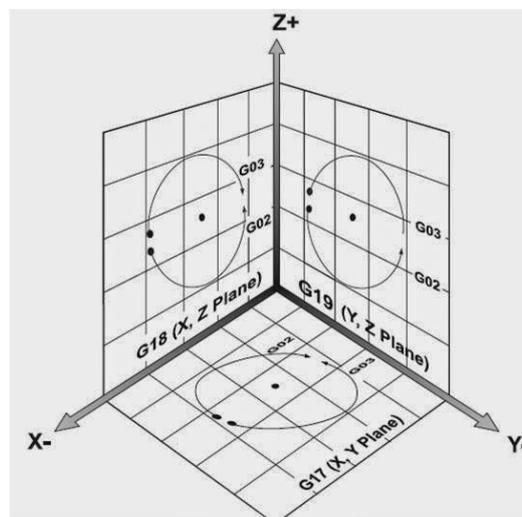


Figura 2.4: Sentido de G2 y G3 para diferentes planos de trabajo [11]

Existen tres formas de programar el código G2 y G3 para interpolaciones circulares:

- Método I, J, K. Este es el único método que puede ser usado para programar una trayectoria circular completa. Durante la programación del mismo solamente se utilizarán dos de los tres caracteres nombrados en función del plano de trabajo seleccionado.
  - G17 (plano XY): utiliza I y J.
  - G18 (plano XZ): utiliza I y K.
  - G19 (plano YZ): utiliza J y K.

Los caracteres I, J y K indican la distancia existente entre el punto inicial del arco y el centro del mismo en cada uno de los ejes del plano de trabajo. El punto inicial no se indica en la instrucción ya que la máquina herramienta considera como punto de inicio la ubicación del husillo cuando se ejecuta la instrucción.

- Método R. Este método solo puede ser usado para generar arcos con menos de 360 grados. Es más simple que el método comentado con anterioridad, pues solo hay que indicar el punto final del arco y el radio del mismo.
- Método de coordenadas polares. En este método se indica con la letra R la distancia entre el punto final del arco y el origen polar, y con el carácter Q se indica el ángulo formado por la recta horizontal que pasa por el origen polar y la recta que une el origen polar con el punto final del arco. Además, será necesario indicar el centro del arco con los caracteres I, J, K comentados anteriormente.

En este método resulta de gran utilidad la función G30 que permite preseleccionar cualquier punto del plano de trabajo como nuevo origen de coordenadas polares. Si no se selecciona, se asume como origen polar el origen del sistema de referencia activo (cero pieza).

A continuación, se muestran ejemplos de los tipos de programación explicados anteriormente para un arco cualquiera. Así, para generar el arco que se muestra en la Figura 2.5 serán necesarios los códigos de la Tabla 2.1 en función de la tipología de programación empleada.

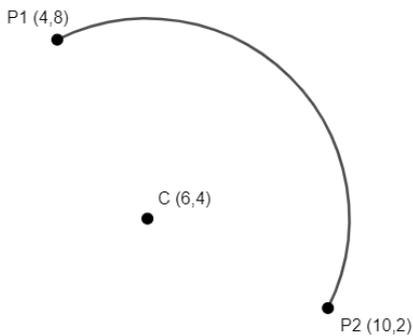


Figura 2.5: Arco para ejemplo de programación instrucción G2

Tabla 2.1: Programación del arco ejemplo

Método I, J, K	N10 G1 G90 X4 Y8 N20 G1 Z-2 N30 G2 X10 Y2 I2 J-4
Método R	N10 G1 G90 X4 Y8 N20 G1 Z-2 N30 G2 X10 Y2 R4.472
Coordenadas Polares	N10 G1 G90 X4 Y8 N20 G1 Z-2 N30 G30 N40 G2 R8.485 Q315 I2 J-4

## 2.4. Grado de continuidad de una curva

Se entiende como grado de continuidad de una curva al número de veces que su expresión puede ser derivada obteniendo una función continua. Una curva con continuidad  $C^0$ , como la mostrada en la Figura 2.6, es continua, pero no su pendiente. Si la continuidad de la curva es  $C^1$ , como la mostrada en la Figura 2.7, tanto la curva como su pendiente serán continuas.

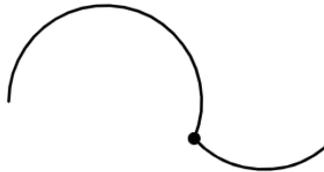


Figura 2.6: Continuidad  $C^0$

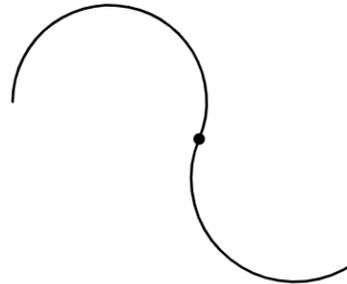


Figura 2.7: Continuidad  $C^1$

Para que una curva sea continua en un punto, los vectores tangentes a izquierda y derecha de dicho punto deben coincidir. La magnitud de los vectores tangentes no afecta a la apreciación que se puede hacer de continuidad en dicho punto. Por tanto, se suele hablar de continuidad geométrica, denotada con  $C$ , cuando los vectores tangentes a ambos lados del punto tienen la misma dirección, independientemente del tipo de magnitud. En consecuencia, la continuidad matemática (vectores tangentes con la misma dirección y magnitud) implica continuidad geométrica.

### 3. Metodología y aplicación

#### 3.1. Trayectoria de los álabes del rodete

Los puntos que describen la trayectoria de la fibra media de un álabe se han obtenido a través de los círculos de Villarceau. Estos círculos, mostrados Figura 3.1, se obtienen como resultado de cortar de forma oblicua un toroide por un plano que pasa por el centro del mismo y que es doblemente tangente a dicho toroide, tal y como se puede observar en la Figura 3.2.

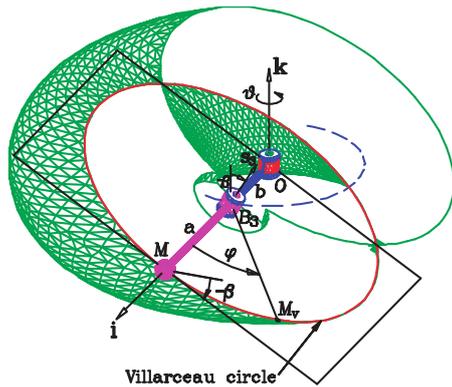


Figura 3.1: Círculos de Villarceau [12]

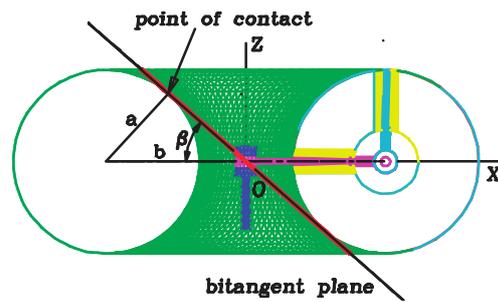


Figura 3.2: Plano bitangente al toroide que pasa por su centro [12]

Las dimensiones del toroide que se han empleado en este estudio para obtener la nube de puntos que definen la trayectoria de la fibra media de cada uno de los álabes se indican a continuación. La nomenclatura hace referencia a la Figura 3.2.

$$a = 16 \text{ mm}$$

$$b = 25 \text{ mm}$$

Estas medidas están condicionadas por la probeta sobre la que se van a mecanizar los álabes del rodete. En la Figura 3.3 se muestran las medidas de dicha probeta, que se corresponde con el centro del toroide, Figura 3.4.

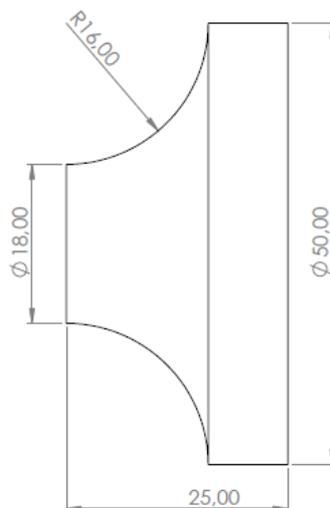


Figura 3.3: Dimensiones de la probeta a mecanizar.

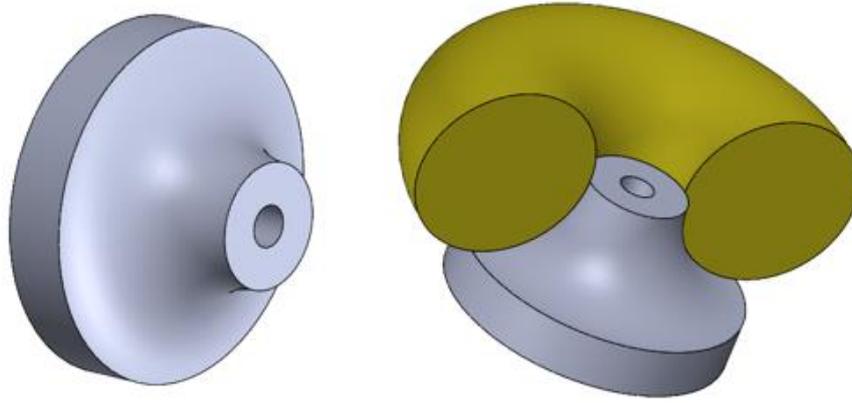


Figura 3.4. Probeta a mecanizar

Conocidos estos puntos es necesario obtener los puntos que definen los caminos de herramienta, es decir, las coordenadas que hay que introducir en el control numérico para mecanizar el álabe. Para obtenerlos es necesario tener en cuenta parámetros como el espesor del propio álabe, el diámetro de la herramienta, la profundidad del álabe en la entrada y salida del rodete, etc.

A continuación, se detalla a modo de ejemplo la influencia del espesor del álabe y el diámetro de la herramienta. Tal y como se observa en la Figura 3.5, la trayectoria del camino de herramienta (trazo discontinuo) para el mecanizado del álabe debe alejarse de la fibra media (trazo continuo de color rojo) una distancia igual a la suma del semi espesor del álabe y el radio de la herramienta para conseguir el mecanizado de este.

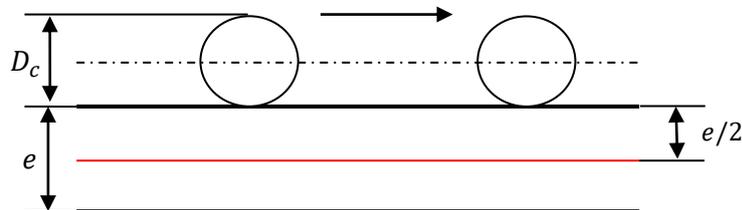


Figura 3.5: Desviación del camino de herramienta con respecto a la fibra media

En consecuencia, para el mecanizado de cada álabe tendremos dos trayectorias. Una de ellas será el camino de herramienta para el mecanizado del anverso, Figura 3.6, y la otra lo será para el mecanizado del reverso, Figura 3.7. Además, el rodete estará compuesto por dos tipos de álabes: álabes principales y álabes intermedios. Por tanto, se tienen 4 caminos de herramienta diferentes para mecanizar el rodete del compresor centrífugo.

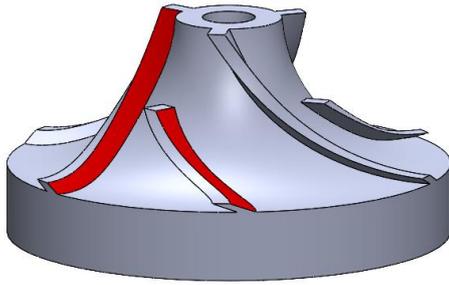


Figura 3.6: Anverso de los álabes

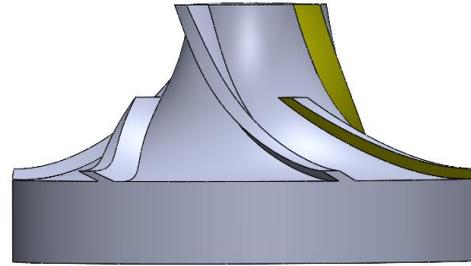


Figura 3.7: Reverso de los álabes

En el Anexo I se muestran los puntos que discretizan dichas trayectorias para el mecanizado de un rodete que tiene las siguientes características:

- Espesor del álabe: 2mm
- Profundidad del álabe en la entrada: 3mm
- Profundidad del álabe en la salida: 1mm

## 3.2. Programa aproximación circular

### 3.2.1. Descripción

Para trazar curvas circulares a través de los puntos que definen las trayectorias de los caminos de herramienta para el mecanizado de los álabes del rodete del compresor centrífugo se va desarrollar un programa en Visual Basic, cuyo código se puede leer en el Anexo II, que cumpla con los siguientes requisitos:

- En la unión entre los diferentes arcos circulares debe darse una continuidad de tipo  $C^1$ .
- La distancia entre los puntos y la trayectoria curva debe ser inferior al valor fijado.
- El error cordal, entendiéndose como este la mayor diferencia entre las trayectorias rectas que unen los diferentes puntos y las trayectorias curvilíneas, debe ser inferior al valor de referencia.

Destacar que esta desviación no es un error cordal en su propia definición ya que se están comparando dos trayectorias que son aproximaciones de una curva discretizada.

Una vez definidas las trayectorias circulares, el programa generará las instrucciones en GCode para exportarlas al fichero que contiene el resto de instrucciones, necesarias para el mecanizado de los álabes del rodete del compresor centrífugo en la fresadora de 5 ejes. Además, el programa generará el código de Matlab necesario para la representación gráfica de las trayectorias rectilíneas y las trayectorias curvas generadas para poder compararlas y verificar que no se ha producido ningún error.

A continuación, se muestra el fundamento teórico existente detrás de cada uno de los requisitos expuestos anteriormente.

### 3.2.1.1. Continuidad $C^1$

Para que dos arcos se unan bajo continuidad  $C^1$  debe darse, como precondition, una continuidad  $C^0$ . Como ya se introdujo en el Capítulo 2.4, la continuidad  $C^0$  para dos arcos se produce cuando el punto final del primer arco coincide con el punto inicial del segundo, tal y como se muestra en la Figura 3.8. Esta condición se expresa matemáticamente con la ecuación (1).

$$(x_0 - C_x)^2 + (y_0 - C_y)^2 = r^2 \quad (1)$$

Donde  $(x_0, y_0)$  son las coordenadas del punto de unión de ambos arcos,  $(C_x, C_y)$  son las coordenadas del centro del nuevo arco y  $r$  es el valor del radio del segundo arco. Estas notaciones quedan reflejadas en la figura siguiente.

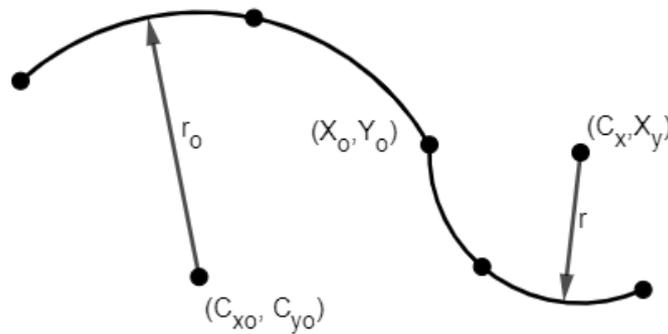


Figura 3.8: Unión de dos arcos con continuidad de clase  $C^0$

Para que ambos arcos se unan bajo continuidad de tipo  $C^1$ , geoméricamente debe darse la condición de que los centros de ambos arcos estén situados en la recta que une el centro del primer arco con el punto de unión de ambos arcos, representada con la letra  $s$ . La Figura 3.9 y Figura 3.10 muestran las dos posibles soluciones para la unión de dos arcos bajo continuidad  $C^1$ .

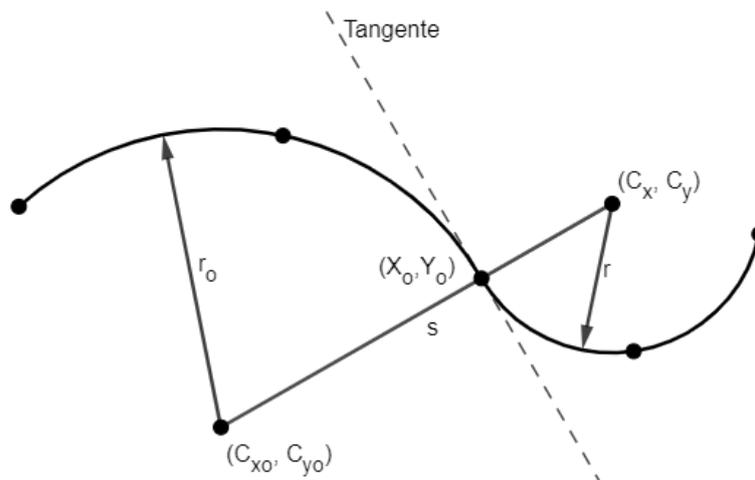


Figura 3.9: Unión de dos arcos con continuidad de clase  $C^1$  con cambio de concavidad

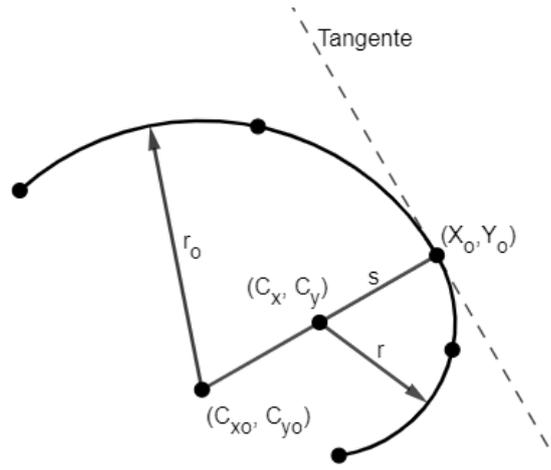


Figura 3.10: Unión de dos arcos con continuidad de clase  $C^1$  sin cambio de concavidad

Esta condición se puede expresar matemáticamente con las ecuaciones (2) y (3), que definen las coordenadas del centro del nuevo arco para que se cumpla la continuidad tipo  $C^1$ .

$$C_x = a \cdot (x_0 - C_{x0}) + C_{x0} \quad (2)$$

$$C_y = a \cdot (y_0 - C_{y0}) + C_{y0} \quad (3)$$

Donde  $(C_{x0}, C_{y0})$  son las coordenadas del centro del primer arco y  $(C_x, C_y)$  son las coordenadas del centro del segundo arco. El valor de  $a$  representa la distancia existente entre los centros de ambos arcos, expresada en múltiplos del radio del primer arco,  $r_0$ , tal y como se muestra en las ecuaciones (4) y (5).

$$a = \frac{r_0 + r}{r_0} \quad (4)$$

$$a = \frac{r_0 - r}{r_0} \quad (5)$$

El uso de una u otra expresión tendrá como resultado arcos diferentes. Así, el empleo de la expresión (4) generará un segundo arco con cambio de concavidad. En consecuencia, el uso de la expresión (5) dará lugar a un segundo arco con la misma concavidad.

### 3.2.1.2. Distancia punto – arco

Para calcular cada una de las distancias existentes entre los diferentes puntos que discretizan la curva teórica y las trayectorias resultantes de las aproximaciones circulares se empleará la ecuación (6).

Gráficamente, dicha distancia se obtiene trazando una circunferencia con el mismo centro que el arco y que pase por el punto en cuestión. Así, la resta de ambos radios será la distancia buscada, denotada con la letra  $d$ . Matemáticamente se expresa con la siguiente ecuación.

$$d = \left| \sqrt{(x_p - C_x)^2 + (y_p - C_y)^2} - r \right| \quad (6)$$

Donde  $(x_p, y_p)$  son las coordenadas del punto P,  $(C_x, C_y)$  son las coordenadas del centro del arco y  $r$  es el radio de este, tal y como se observa en la figura siguiente.

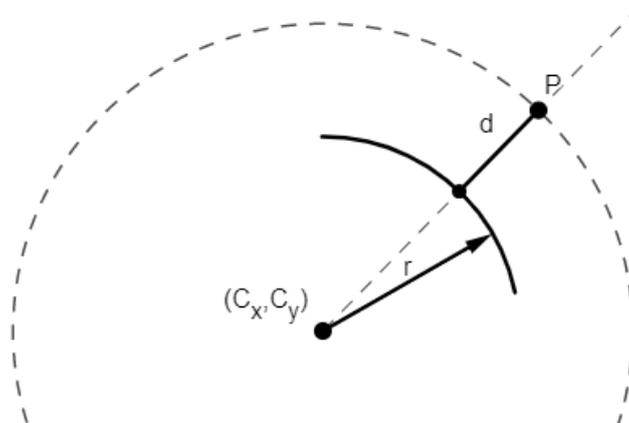


Figura 3.11: Distancia de un punto a un arco

### 3.2.1.3. Error cordal

El error cordal de los arcos generados respecto a cada uno de los segmentos que unen los distintos puntos se calcula de forma similar al apartado anterior. En este caso, el punto P se define como la mediatriz de cada uno de los segmentos. Por tanto, el error cordal para cada segmento vendrá dado por la siguiente expresión.

$$e = r - d_{cm} \quad (7)$$

Donde  $r$  es el radio del arco y  $d_{cm}$  es la distancia entre el punto P y el centro del arco, tal y como se puede observar en la figura siguiente.

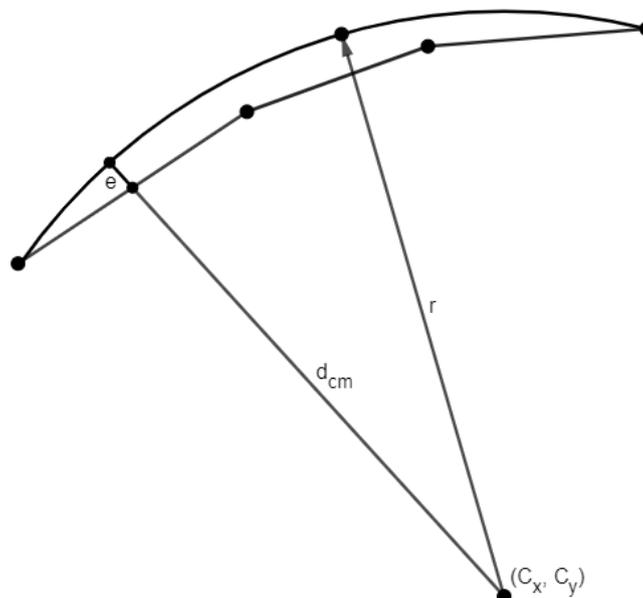


Figura 3.12: Error cordal para un segmento

### 3.2.2. Programación

En este apartado se van a detallar las diferentes fases de desarrollo del algoritmo que permite obtener las trayectorias circulares.

En primer lugar, se planteó un algoritmo en el cual el usuario elegía el máximo número de puntos que podía abarcar cada arco de la nube de puntos y el máximo error punto - arco. Dicho algoritmo se puede dividir en dos bloques, un bloque para generar el primer arco y otro para generar el resto de arcos.

El desarrollo del primer bloque comenzó con la elaboración del código que permitía trazar el primer arco. Para ello, la condición que se impuso por motivos de diseño fue que el arco pasara por el primer y último punto. Por ejemplo, si se definía que los arcos debían estar formados por 5 puntos, este primer arco pasaría por el primer punto de la nube y por el quinto. El siguiente paso fue aplicar la limitación de error punto – arco, empleando un bucle iterativo que modificaba el centro y el radio del arco hasta conseguir que la distancia entre cada uno de los puntos de la nube y el arco fuese inferior a la determinada por el usuario.

Una vez que se consiguió lo descrito en el párrafo anterior se procedió a desarrollar el segundo bloque del algoritmo. Para ello se comenzó a elaborar el código que permitía generar el segundo arco con continuidad de clase  $C^1$ , abarcando el número de puntos determinado. Sin embargo, al aplicar la limitación de error punto – arco aparecía una incongruencia. En múltiples ocasiones, el arco con continuidad de clase  $C^1$  no puede abarcar tantos puntos como se le indica cumpliendo con el máximo error permitido para cada uno de los puntos. Siguiendo con el ejemplo anterior, el segundo arco podría ser tangente al anterior y cumplir con el error punto – arco en los puntos 5, 6 y 7 de la nube de puntos, pero no con los puntos 8, 9 y 10 debido a un cambio de concavidad en el punto 7.

Para solventar esto se produjo un cambio en el planteamiento del algoritmo, permitiendo que el algoritmo decidiese de forma autónoma el número de puntos que debe abarcar cada arco. Este planteamiento permite, además, obtener la trayectoria curvilínea con el menor número de arcos.

En el capítulo siguiente se explica en detalle el funcionamiento de la versión final del algoritmo.

### 3.2.3. Funcionamiento

En este apartado se va a describir el funcionamiento del programa desarrollado usando el ejemplo de una nube de puntos que define la trayectoria de una curva aleatoria, mostrada en la Figura 3.13.

Para iniciar el programa es necesario introducir las coordenadas de los puntos que definen la trayectoria sobre la que se quiere aplicar el método de interpolación circular, mostrados en la Tabla 3.1.

Tabla 3.1: Puntos de la curva aleatoria

Coordenadas de los puntos	
X [mm]	Y [mm]
0.72	1.85
1.44	1.09
2.92	0.51
4.3	0.47
5.7	0.89
7.34	2.05
7.88	3.21
8.3	4.31
9.34	4.91
10.4	4.99
11.64	4.91
12.44	4.55
13.86	3.45
14.62	2.43
15.18	1.01

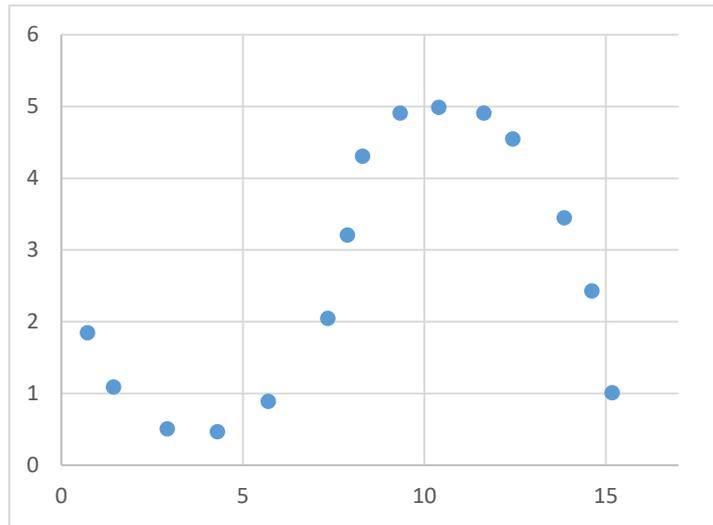


Figura 3.13: Trazo de la curva aleatoria

Una vez definidos los puntos, se establece la distancia máxima permitida entre cada uno de los puntos y los arcos que generará el programa. Para este ejemplo, la distancia máxima permitida será de 0,05mm. En consecuencia, si el primer arco engloba 4 puntos, la distancia de los 4 primeros puntos de la Tabla 3.1 con respecto al primer arco deberá de ser igual o menor a 0,05mm.

Una vez definida dicha distancia, se ejecuta el algoritmo que se describe a continuación.

1. Se genera un arco que pasa por el primer y tercer punto.
2. Si la distancia entre cada uno de los puntos y el arco es superior a la distancia máxima permitida, el programa entra en un bucle iterativo para reducir dicha distancia modificando el radio y centro del arco generado hasta que se cumpla la restricción impuesta.
3. Si se consigue que la distancia entre cada uno de los puntos y el arco sea inferior al valor de referencia, se añade un punto y se vuelve al paso 2.
4. Cuando el programa no consigue disminuir dicha distancia al valor de referencia, se elimina el último punto y se genera el arco.
5. A partir del punto final del arco anterior se genera un nuevo arco con continuidad de tipo  $C^1$  con respecto al anterior empleando las ecuaciones (2) y (3), y añade el siguiente punto de la nube.
6. Vuelta al punto 2.

Cabe destacar que los arcos no tienen por qué pasar por los puntos que definen la trayectoria. Únicamente el primer arco tiene la restricción, por razones de diseño del programa, de pasar por el primer y último punto que compone el mismo tal y como se puede observar en la Tabla 3.3. En este caso, el primer arco engloba 4 puntos de modo que el error de los puntos 1 y 4 son cero. Sin embargo, el arco 5 engloba también 4 puntos y ninguno de estos 4 puntos pasa por el arco.

Una vez finalizado el algoritmo se representan los parámetros que definen cada arco, mostrados en la Tabla 3.2. Además, se indican las distancias entre cada uno de los puntos y los arcos correspondientes, Tabla 3.3, y el mayor error cordal, que para este ejemplo toma el valor de 0,124mm.

Tabla 3.2: Arcos generados para la curva aleatoria

Arco	Nº Puntos	Datos del arco			Punto final del arco		Ángulos	
		C <sub>x</sub> [mm]	C <sub>y</sub> [mm]	r [mm]	X [mm]	Z [mm]	Inicio	Final
1	4	3.589	3.960	3.562	4.300	0.470	3.776	4.913
2	3	3.103	6.349	6.000	7.314	2.076	4.913	5.491
3	2	6.051	3.358	1.800	7.845	3.213	5.491	6.202
4	3	9.739	3.060	1.900	9.339	4.917	3.061	1.783
5	4	10.371	0.127	4.900	12.447	4.565	1.783	1.133
6	3	10.583	0.580	4.400	14.583	2.413	1.133	0.430
7	2	3.038	-2.878	12.700	15.133	0.995	0.430	0.310

Tabla 3.3: Distancia punto – arco para la curva aleatoria

Arco	e <sub>1</sub>	e <sub>2</sub>	e <sub>3</sub>	e <sub>4</sub>
1	0.000	0.024	0.047	0.000
2	0.000	0.046	0.036	
3	0.036	0.035		
4	0.035	0.006	0.007	
5	0.007	0.037	0.048	0.017
6	0.017	0.044	0.041	
7	0.041	0.050		

Por último, se muestra la Figura 3.14 comparando las trayectorias rectilíneas que recorrería la herramienta en la máquina de control numérico y la nueva trayectoria generada mediante arcos.

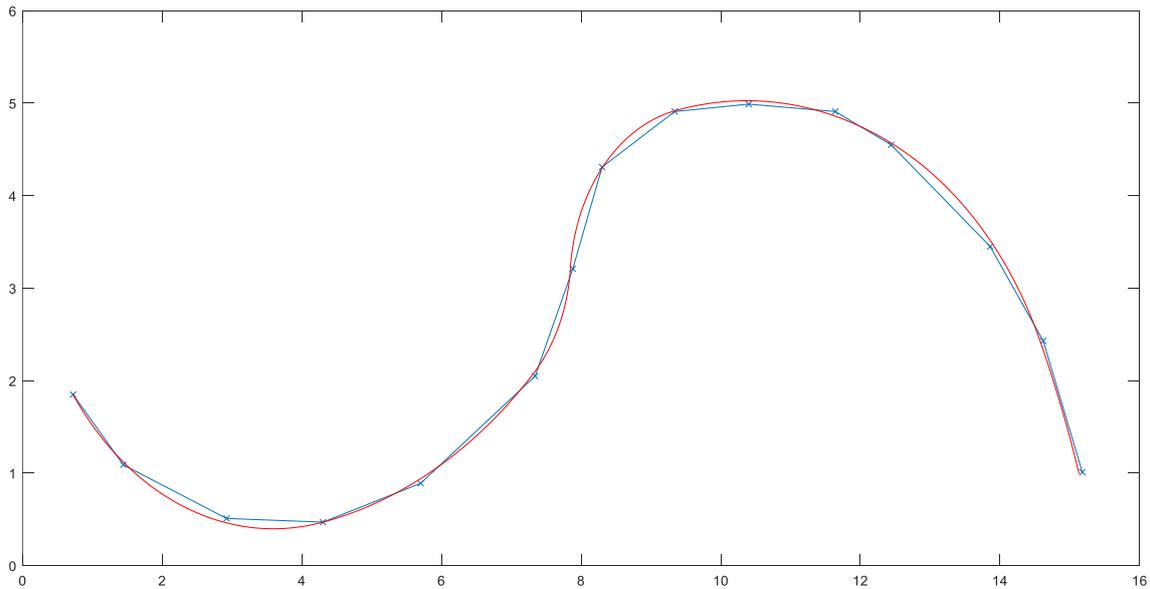


Figura 3.14: Comparación de trayectorias lineales y circulares para la curva aleatoria

Para realizar el mecanizado de esta trayectoria serían necesarias 14 instrucciones de tipo G1 en el control numérico. Con los resultados de la aproximación circular se sustituirían dichas instrucciones por 7 de tipo G2 y/o G3. Además, se puede observar que la trayectoria compuesta por arcos es más suave que la trayectoria compuesta por segmentos rectos. Con esto se persigue estabilizar el mecanizado con el objetivo de conseguir un mejor acabado de la pieza.

En los siguientes capítulos se va a aplicar este método a las trayectorias que definen los caminos de herramienta que permiten mecanizar los álabes del rodete del compresor centrífugo y se realizará el mecanizado del mismo para comprobar si en la práctica se consigue un mejor acabado superficial.

## 4. Pruebas de laboratorio

### 4.1. Elaboración del Código G

El primer paso para realizar el mecanizado del rodete será generar el código G de las trayectorias circulares que definen los caminos de herramienta a partir de las nubes de puntos pertinentes, mostradas en la Tabla Anexo I. Para ello se utilizará el programa desarrollado cuyo funcionamiento ha sido explicado en el capítulo anterior.

Como ya se ha comentado en el apartado 3.1, en el proceso de fabricación del rodete se requieren 4 trayectorias diferentes para lograr el mecanizado completo de los álabes principales e intermedios. En consecuencia, serán 4 las nubes de puntos que habrá que introducir en el programa.

En los puntos siguientes se muestran los resultados obtenidos para cada uno de los caminos de herramienta. Destacar que para cada nube de puntos se han generado diferentes soluciones, y para seleccionar la aproximación circular de cada nube de puntos se ha tenido en cuenta el número de arcos generados y el error cordal, buscando el balance óptimo entre el menor número de arcos y un error cordal del orden de 5 ó 6 centésimas de milímetro.

#### 4.1.1. Anverso álabe principal

La trayectoria que define el camino de herramienta para el mecanizado del anverso de los álabes principales consta de 41 puntos, lo que se traduce en 40 movimientos G1. Tras aplicar la aproximación circular a dicha trayectoria se obtienen los resultados mostrados en la siguiente tabla.

Tabla 4.1: Resultados de la aproximación circular para el anverso del álabe principal

Error Punto – Arco [mm]	Número de arcos	Error cordal [mm]
0,1	1	0,146
0,05	2	0,096
0,01	2	0,056
0,008	2	0,054
0,005	3	0,051
0,001	18	0,048

Los resultados que se han elegido para esta trayectoria son los de la fila 4. En la Tabla 4.2 se detallan las características de los dos arcos generados.

Tabla 4.2: Arcos generados para el anverso del álabe principal

Arco	Nº Puntos	Datos del arco			Punto final del arco		Ángulos	
		C <sub>x</sub> [mm]	C <sub>y</sub> [mm]	r [mm]	X [mm]	Z [mm]	Inicio [rad]	Final [rad]
1	23	-0.927	-259.712	244.131	-140.385	-59.334	3.039	2.179
2	19	-1.752	-258.526	242.686	21.926	-16.998	2.179	1.473

Conocidos los puntos finales de cada arco y el radio se genera de forma automática el código G de las instrucciones necesarias para el mecanizado del reverso del álabe principal.

N10 G3 X-140.385 Z-59.334 R244.131 B40 CXXX

N20 G3 X21.926 Z-16.998 R242.686 B0 CXXX

Los valores del giro en C no se indican ya que varían para cada álabe puesto que están desfasados 120 grados. Como se puede observar, para este camino de herramienta se ha reducido en 38 el número de instrucciones necesarias para el mecanizado.

#### 4.1.2. Reverso álabe principal

La trayectoria que define el camino de herramienta para el mecanizado del reverso de los álaves principales consta de 41 puntos, lo que se traduce en 40 movimientos G1. Tras aplicar la aproximación circular a dicha trayectoria se obtienen los resultados mostrados en la siguiente tabla.

Tabla 4.3: Resultados de la aproximación circular para el reverso del álabe principal

Error Punto – Arco [mm]	Número de arcos	Error cordal [mm]
0,1	1	0,145
0,05	2	0,096
0,021	2	0,067
0,01	5	0,057
0,005	5	0,052
0,001	11	0,048

Los resultados que se han elegido para esta trayectoria son los de la fila 3. En la Tabla 4.4 se detallan las características de los dos arcos generados.

Tabla 4.4: Arcos generados para el reverso del álabe principal

Arco	Nº Puntos	Datos del arco			Punto final del arco		Ángulos	
		C <sub>x</sub> [mm]	C <sub>y</sub> [mm]	r [mm]	X [mm]	Z [mm]	Inicio [rad]	Final [rad]
1	17	4.445	-258.832	242.984	-189.095	-111.919	1.473	2.492
2	25	6.053	-260.053	245.003	-237.635	-234.699	2.492	3.038

Conocidos los puntos finales de cada arco y el radio se genera de forma automática el código G de las instrucciones necesarias para el mecanizado del anverso del álabe principal.

```
N10 G2 X-189.095 Z-111.919 R242.984 B58.5 CXXX
N20 G2 X-237.635 Z-234.699 R245.003 B90 CXXX
```

Como se puede observar, para este camino de herramienta también se ha reducido en 38 el número de instrucciones necesarias para el mecanizado.

#### 4.1.3. Anverso álabe intermedio

La trayectoria que define el camino de herramienta para el mecanizado del anverso de los álaves intermedios consta de 22 puntos, lo que se traduce en 21 movimientos G1. Tras aplicar la aproximación circular a dicha trayectoria se obtienen los resultados mostrados en la siguiente tabla.

Tabla 4.5: Resultados de la aproximación circular para el anverso del álabe intermedio

Error Punto – Arco [mm]	Número de arcos	Error cordal [mm]
0,1	1	0,146
0,05	1	0,096
0,015	1	0,061
0,01	2	0,056
0,005	2	0,052
0,001	3	0,048

Los resultados que se han elegido para esta trayectoria son los de la fila 3. En la Tabla 4.6 se detallan las características de los dos arcos generados.

Tabla 4.6: Arco generado para el anverso del álabe intermedio

Arco	Nº Puntos	Datos del arco			Punto final del arco		Ángulos	
		C <sub>x</sub> [mm]	C <sub>y</sub> [mm]	r [mm]	X [mm]	Z [mm]	Inicio [rad]	Final [rad]
1	22	-1.770	-258.543	242.703	21.926	-17.000	2.297	1.473

Conocido el punto final del arco y el radio se genera de forma automática el código G de la instrucción necesaria para el mecanizado del reverso del álabe intermedio.

*N10 G3 X21.926 Z-17 R242.703 B0 CXXX*

Como se puede observar, para este camino de herramienta se ha reducido en 20 el número de instrucciones necesarias para el mecanizado.

#### 4.1.4. Reverso álabe intermedio

La trayectoria que define el camino de herramienta para el mecanizado del reverso de los álabes intermedios consta de 26 puntos, lo que se traduce en 25 movimientos G1. Tras aplicar la aproximación circular a dicha trayectoria se obtiene los resultados mostrados en la siguiente tabla.

Tabla 4.7: Resultados de la aproximación circular para el reverso del álabe intermedio

Error Punto – Arco [mm]	Número de arcos	Error cordal [mm]
0,1	1	0,146
0,05	1	0,096
0,02	1	0,067
0,01	2	0,057
0,005	2	0,052
0,001	5	0,048

Los resultados que se han elegido para esta trayectoria son los de la fila 3. En la Tabla 4.8 se detallan las características de los dos arcos generados.

Tabla 4.8: Arco generado para el reverso del álabe intermedio

Arco	Nº Puntos	Datos del arco			Punto final del arco		Ángulos	
		C <sub>x</sub> [mm]	C <sub>y</sub> [mm]	r [mm]	X [mm]	Z [mm]	Inicio [rad]	Final [rad]
1	22	-1.770	-258.543	242.703	21.926	-17.000	2.297	1.473

Conocido el punto final del arco y el radio se genera de forma automática el código G de la instrucción necesaria para el mecanizado del reverso del álabe intermedio.

*N10 G3 X-183.172 Z-104.438 R242.921 B56.25 CXXX*

Como se puede observar, para este camino de herramienta se ha reducido en 24 el número de instrucciones necesarias para el mecanizado.

Una vez se tienen las instrucciones para cada uno de los caminos de herramienta se unen en un archivo de texto junto con unas instrucciones que posicionan la herramienta para la entrada de cada uno de los mecanizados. Como resultado se ha generado un código que permite mecanizar un rodete completo con 360 líneas de código menos que en el caso inicial.

En el Anexo III se muestra el código G para el mecanizado de un álabe principal y de un álabe secundario, mientras que en el Anexo IV se representan las trayectorias generadas.

## 4.2. Mecanizado

Con el fin de poder analizar las mejoras llevadas a cabo por la aproximación circular se va a mecanizar el rodete de dos formas diferentes. En primer lugar, se mecanizará empleando únicamente instrucciones de tipo G0 y G1, es decir, movimientos rectilíneos. Acto seguido se mecanizará el rodete sustituyendo los movimientos rectilíneos por las trayectorias circulares generadas con el programa desarrollado.

Para que los mecanizados sean lo más homogéneos posibles y que las diferencias sean únicamente debidas a las diferencias en el código, los parámetros del mecanizado serán los mismos para ambos casos.

- Velocidad de avance: 300 mm/min.
- Velocidad de giro: 1300 rpm.
- Fresa cilíndrica de radio 6 mm.

El primer paso, común a ambos mecanizados, será tomar el cero pieza. Este está situado en el centro de la cara superior de la probeta, tal y como se muestra en la Figura 4.1. La posición correspondiente a B0 hará referencia a la posición vertical de la probeta, donde el eje de revolución de la probeta coincide con el eje de revolución de la herramienta, mostrado en la Figura 4.2.

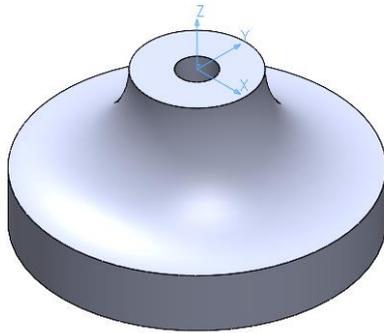


Figura 4.1: Cero pieza I



Figura 4.2: Cero pieza II

#### 4.2.1. Interpolación lineal

El mecanizado de los álabes del compresor centrífugo se efectuó sin ningún inconveniente con el código compuesto únicamente por instrucciones G0 y G1, es decir, movimientos rectilíneos entre los diferentes puntos que determinan la trayectoria del camino de herramienta.

Durante el mecanizado del rodete se tomó medida del tiempo de mecanizado de un álabe principal con la finalidad de buscar diferencias con las nuevas trayectorias circulares generadas. A continuación, se muestra una imagen tomada durante el mecanizado.



Figura 4.3: Mecanizado del rodete

#### 4.2.2. Interpolación circular

Antes de mecanizar el rodete con las nuevas trayectorias generadas mediante aproximación circular se realizó una prueba en vacío del código G generado. Durante esta prueba se pudo observar la existencia de una incidencia relacionada con los planos de trabajo.

Para mecanizar el rodete con las trayectorias circulares generadas (movimientos de tipo G2 y G3) es necesario trabajar en el plano XZ por motivos de fabricación, plano que se selecciona a través de la instrucción G18.

En la máquina CNC Lagun L650 la herramienta está colocada en el eje Z, tal y como se puede observar en la Figura 4.4. Sin embargo, al usar la instrucción G18 la máquina entiende que el eje en el que se encuentra la herramienta pasa a ser el eje Y, es decir, la máquina herramienta interpreta que la herramienta siempre está situada en el eje perpendicular al plano de trabajo. Sin embargo, esto no es correcto ya que físicamente no se modifica la geometría de la máquina y, en consecuencia, hay una incongruencia entre la interpretación del software de la máquina y la posición real de la herramienta.

Por ende, cuando se ejecutó el programa en vacío se pudo observar que la herramienta se situaba en puntos con coordenadas incorrectas.

Para solventar esta incidencia se buscó en el manual de programación del CNC 8065 la instrucción que permite seleccionar como nuevo eje de herramienta cualquier eje longitudinal de la máquina. La instrucción empleada fue `#TOOL AX [Z+]`, la cual permite trabajar en el plano XZ con la herramienta situada en el eje Z, tal y como se muestra en la Figura 4.5.

Tras este cambio en el código se efectuó el mecanizado del rodete sin ninguna incidencia.



Figura 4.4: Situación de la herramienta en la máquina CNC Lagun L650

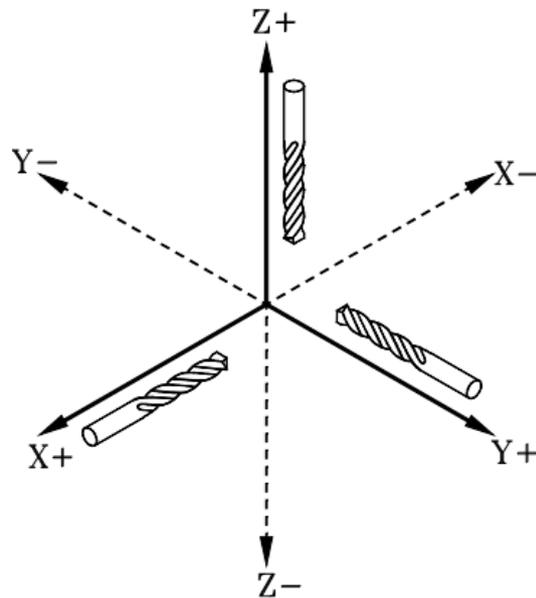


Figura 4.5: Orientaciones positivas de la herramienta [13]

## 5. Análisis de resultados

### 5.1. Acabado superficial

Con el fin observar las diferencias existentes en el acabado superficial de los rodetes mecanizados con los diferentes programas, interpolación lineal e interpolación circular, se va a emplear un microscopio digital para analizar la superficie de estos. Para ello, se han tomado muestras en el reverso del álabe principal (A) y en el anverso del mismo (B). En la Figura 5.1 se indican los puntos exactos donde se han fotografiado las muestras.

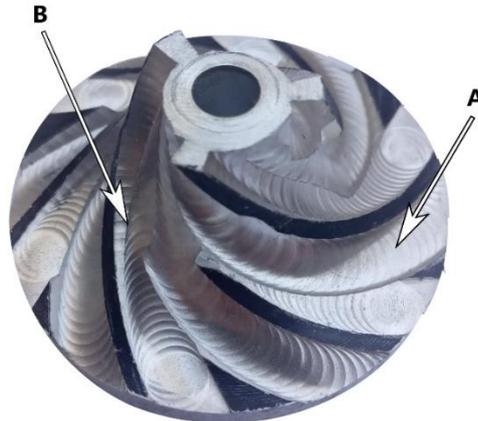


Figura 5.1: Localización muestras microscopio.

Las imágenes han sido tomadas con el microscopio digital Caidna 40-1000X, mostrado en la Figura 5.2. Además, se ha empleado una regla calibrada, Figura 5.3, que sirve de escala en las muestras tomadas con la finalidad de tener una mejor percepción de los resultados obtenidos.



Figura 5.2: Microscopio digital Caidna 40-1000X.

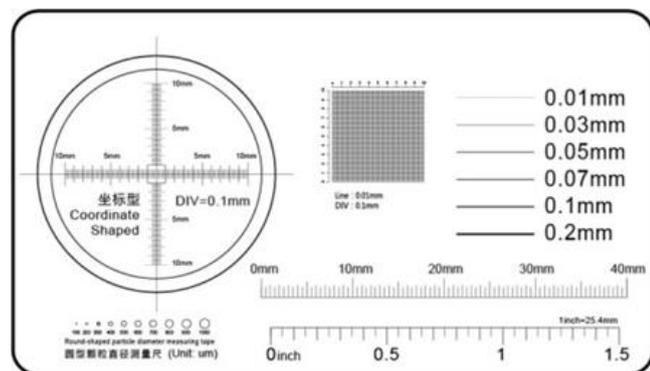


Figura 5.3: Regla calibrada microscopio digital

Así pues, cada uno de los espacios comprendidos entre dos rayas de la regla calibrada que aparece en las muestras se corresponde con 0,05 mm.

Analizando las imágenes del reverso del álabe principal se puede observar, en la Figura 5.4, que durante el mecanizado del rodete se han generado marcas en la superficie que le dan un aspecto escamoso.

Este efecto se debe a los pequeños desplazamientos lineales que realiza la herramienta para completar el camino de herramienta. Estos pequeños movimientos hacen que el recorrido que realiza el husillo por la curva teórica no se produzca con fluidez, pues no existe una continuidad en el camino de herramienta ya que está discretizado en 40 movimientos lineales. Como consecuencia, se generan esas marcas en la superficie mecanizada, de modo que cada “escama” se corresponde con el avance de la herramienta para cada instrucción G1.

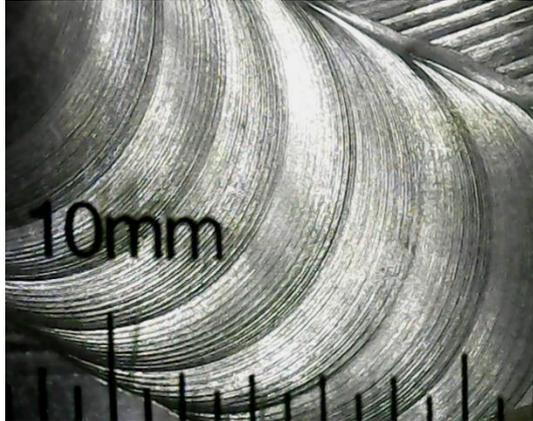


Figura 5.4: Imagen reglada del microscopio de la zona A para interpolación lineal

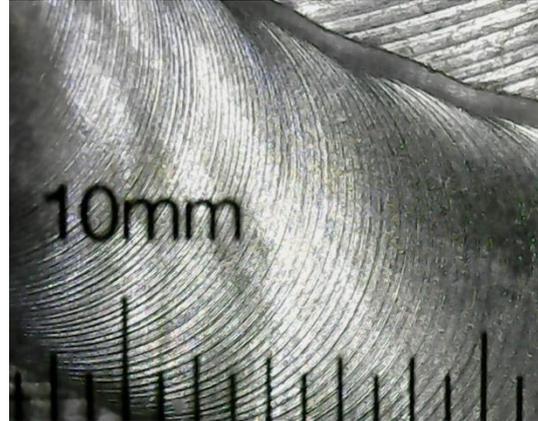


Figura 5.5: Imagen reglada del microscopio de la zona A para aproximación circular

En contraposición, la Figura 5.5 muestra una superficie con un acabado más refinado donde no aparecen esas escamas. Esto se debe a que el reverso del álabe ha sido mecanizado únicamente con una instrucción G2, de modo que el avance de la herramienta ha sido más constante durante el mecanizado.

La ventaja que proporciona realizar el mecanizado con una única instrucción reside en que una vez la herramienta ha comenzado a recorrer el camino de herramienta no se detiene hasta llegar al final, mientras que en el caso inicial realiza 40 inicios y paradas durante el recorrido, lo que repercute de forma considerable en el acabado final.

En el anverso ocurre algo similar. La superficie mecanizada empleando movimientos lineales, Figura 5.6, presenta ese acabado escamoso que da lugar a una superficie rugosa. Sin embargo, el uso de una trayectoria compuesta por arcos que presentan una continuidad de tipo  $C^1$  da lugar a una trayectoria suave, sin cambios pronunciados de dirección, que genera una superficie con un mejor acabado.



Figura 5.6: Imagen del microscopio de la zona B para interpolación lineal



Figura 5.7: Imagen del microscopio de la zona B para aproximación circular



Figura 5.8: Imagen reglada del microscopio de la zona B para interpolación lineal



Figura 5.9: Imagen reglada del microscopio de la zona B para aproximación circular

## 5.2. Tiempo

Durante las pruebas de mecanizado se tomaron medidas del tiempo de mecanizado completo de un álabe principal, es decir, mecanizado del anverso y reverso del mismo.

El tiempo empleado en el mecanizado del álabe con el código compuesto por trayectorias rectilíneas fue de 2 minutos y 40 segundos. Con las nuevas trayectorias generadas mediante aproximación circular se redujo el tiempo de mecanizado a 2 minutos 36 segundos.

Esto supone una disminución del 2,56% del tiempo de mecanizado, que puede deberse a las pequeñas deceleraciones y aceleraciones que se producen entre las numerosas instrucciones de tipo G1. Así, si el camino de herramienta se recorre únicamente con una instrucción G2 o G3, la velocidad de avance es más estable a lo largo del mecanizado consiguiendo así reducir el tiempo de fabricación.



## Conclusiones

En el presente proyecto se ha implementado el uso de trayectorias de interpolación circular para realizar el mecanizado de los álabes de un compresor centrífugo con la finalidad de comprobar si las ventajas que ofrecen este tipo de trayectorias respecto a las trayectorias rectilíneas se verificaban en la práctica.

Para automatizar el proceso de generación de dichas trayectorias circulares se ha desarrollado un programa en Visual Basic que permite obtener, a partir de una nube de puntos, una trayectoria formada por arcos con continuidad de clase  $C^1$  que difieren de esos puntos una distancia determinada.

Tras comprobar el correcto funcionamiento del mismo y generar los caminos de herramienta necesarios para el mecanizado del rodete, se efectuó el mecanizado de los rodetes en el Edificio de Laboratorios de Investigación (ELDI) de la Universidad Politécnica de Cartagena.

Acto seguido se estudiaron las superficies mecanizadas de los rodetes mecanizados empleando trayectorias rectilíneas y circulares en el microscopio, pudiendo observar que el acabado proporcionado por las nuevas trayectorias generadas es más refinado.

Además, el tiempo de mecanizado necesario para fabricar los rodetes con trayectorias circulares es un 2,5% inferior al tiempo de mecanizado con trayectorias rectilíneas.

Con estos resultados se puede afirmar que el uso de trayectorias circulares supone un avance en los procesos de mecanizado, pues además de reducir el número de líneas de código, lograr un mejor acabado superficial y disminuir el tiempo de mecanizado, se consigue estabilizar el proceso de mecanizado debido a la suavidad de los movimientos, lo que se traduce en una disminución de ruidos y vibraciones.

Por todo esto, se puede concluir que el proyecto ha cumplido con las expectativas marcadas inicialmente tras verificar en el laboratorio las mejoras propuestas por este tipo de trayectorias circulares, y dando como resultado un programa de Visual Basic que permite generar trayectorias de aproximación circular para una nube de puntos dada.



## Bibliografía

- [1] M. Estrems *et al.*, “Trajectory generation in 5-axis milling of freeform surfaces using circular arc approximation and its influence in surface roughness,” *Procedia Manuf.*, vol. 41, pp. 208–215, 2019, doi: <https://doi.org/10.1016/j.promfg.2019.07.048>.
- [2] K. Kawasaki, T. Murata, Y. Kobayashi, and K. Shirai, “Reduction of nc machining data volume and evaluation of machined surface by circular ARC approximation of discrete points,” in *Proceedings - ASPE 2010 Annual Meeting*, 2010, vol. 50, pp. 176–179.
- [3] G. W. Vickers and C. Bradley, “Curved surface machining through circular arc interpolation,” *Comput. Ind.*, vol. 19, no. 3, pp. 329–337, 1992, doi: [https://doi.org/10.1016/0166-3615\(92\)90070-4](https://doi.org/10.1016/0166-3615(92)90070-4).
- [4] P. Kaewsaiha and N. Dejdumrong, “A combined linear and circular-arc approximation of curves for feedrate smoothing of CNC machining,” in *2013 10th International Conference Computer Graphics, Imaging and Visualization*, 2013, pp. 23–26.
- [5] B. Hochgesang, “Arc Welder: Anti Stutter and GCode Compression,” 2020. [https://plugins.octoprint.org/plugins/arc\\_welder/](https://plugins.octoprint.org/plugins/arc_welder/).
- [6] X. Yang, “Efficient circular arc interpolation based on active tolerance control,” *Comput. Des.*, vol. 34, no. 13, pp. 1037–1046, 2002, doi: [https://doi.org/10.1016/S0010-4485\(01\)00164-6](https://doi.org/10.1016/S0010-4485(01)00164-6).
- [7] D. S. Meek and D. J. Walton, “Approximation of discrete data by G1 arc splines,” *Comput. Des.*, vol. 24, no. 6, pp. 301–306, 1992, doi: [https://doi.org/10.1016/0010-4485\(92\)90047-E](https://doi.org/10.1016/0010-4485(92)90047-E).
- [8] H. Qiu, C. Kai, and L. Yan, “Optimal circular arc interpolation for NC tool path generation in curve contour manufacturing,” *Comput. Des.*, vol. 29, no. 11, pp. 751–760, 1997, doi: [https://doi.org/10.1016/S0010-4485\(97\)00019-5](https://doi.org/10.1016/S0010-4485(97)00019-5).
- [9] L. A. Piegl and W. Tiller, “Data approximation using biarcs,” *Eng. Comput.*, vol. 18, no. 1, pp. 59–65, 2002.
- [10] R. T. Farouki, C. Y. Han, and S. Li, “Inverse kinematics for optimal tool orientation control in 5-axis CNC machining,” *Comput. Aided Geom. Des.*, vol. 31, no. 1, pp. 13–26, 2014, doi: <https://doi.org/10.1016/j.cagd.2013.11.002>.
- [11] “Learning CNC Strategies and Tips.” [www.cnc-programming-tips.blogspot.com](http://www.cnc-programming-tips.blogspot.com).
- [12] C.-C. Lee and J. M. Hervé, “Oblique circular torus, Villarceau circles, and four types of Bennett linkages,” *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, vol. 228, no. 4, pp. 742–752, 2014.
- [13] Fagor, *Cnc 8065. Manual de programación*. 2013.





industriales  
etsii

Escuela Técnica  
Superior  
de Ingeniería  
Industrial

# UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

## Anexos



Universidad  
Politécnica  
de Cartagena



## Anexo I. Trayectorias discretizadas de los caminos de herramienta.

Tabla Anexo I: Nube de puntos de los caminos de herramienta

Anverso Principal		Reverso Principal		Anverso Intermedio		Reverso Intermedio	
X [mm]	Z [mm]	X [mm]	Z [mm]	X [mm]	Z [mm]	X [mm]	Z [mm]
-242.606	-225.219	18.604	-16.254	-162.855	-77.005	28.073	-17
-241.07	-215.792	9.111	-15.881	-155.596	-70.822	18.604	-16.254
-239.167	-206.432	-0.391	-15.88	-148.103	-64.929	9.111	-15.881
-236.899	-197.154	-9.888	-16.253	-140.385	-59.334	-0.391	-15.88
-234.271	-187.972	-19.365	-16.998	-132.456	-54.047	-9.888	-16.253
-231.287	-178.9	-28.808	-18.114	-124.328	-49.075	-19.365	-16.998
-227.95	-169.953	-38.201	-19.601	-116.013	-44.426	-28.808	-18.114
-224.267	-161.143	-47.53	-21.455	-107.523	-40.107	-38.201	-19.601
-220.243	-152.485	-56.782	-23.674	-98.872	-36.125	-47.53	-21.455
-215.884	-143.991	-65.941	-26.255	-90.074	-32.485	-56.782	-23.674
-211.197	-135.675	-74.994	-29.193	-81.141	-29.193	-65.941	-26.255
-206.189	-127.549	-83.927	-32.485	-72.088	-26.255	-74.994	-29.193
-200.868	-119.626	-92.725	-36.125	-62.929	-23.674	-83.927	-32.485
-195.242	-111.919	-101.376	-40.107	-53.677	-21.455	-92.725	-36.125
-189.319	-104.438	-109.866	-44.426	-44.348	-19.601	-101.376	-40.107
-183.109	-97.195	-118.181	-49.075	-34.955	-18.114	-109.866	-44.426
-176.622	-90.201	-126.309	-54.047	-25.512	-16.998	-118.181	-49.075
-169.867	-83.468	-134.238	-59.334	-16.035	-16.253	-126.309	-54.047
-162.855	-77.005	-141.956	-64.929	-6.538	-15.88	-134.238	-59.334
-155.596	-70.822	-149.449	-70.822	2.964	-15.881	-141.956	-64.929
-148.103	-64.929	-156.708	-77.005	12.457	-16.254	-149.449	-70.822
-140.385	-59.334	-163.72	-83.468	21.926	-17	-156.708	-77.005
-132.456	-54.047	-170.475	-90.201			-163.72	-83.468
-124.328	-49.075	-176.962	-97.195			-170.475	-90.201
-116.013	-44.426	-183.172	-104.438			-176.962	-97.195
-107.523	-40.107	-189.095	-111.919			-183.172	-104.438
-98.872	-36.125	-194.721	-119.626				
-90.074	-32.485	-200.042	-127.549				
-81.141	-29.193	-205.05	-135.675				
-72.088	-26.255	-209.737	-143.991				
-62.929	-23.674	-214.096	-152.485				
-53.677	-21.455	-218.12	-161.143				
-44.348	-19.601	-221.803	-169.953				
-34.955	-18.114	-225.14	-178.9				
-25.512	-16.998	-228.124	-187.972				
-16.035	-16.253	-230.752	-197.154				
-6.538	-15.88	-233.02	-206.432				
2.964	-15.881	-234.923	-215.792				
12.457	-16.254	-236.459	-225.219				
21.926	-17	-237.627	-234.7				



## Anexo II. Código Visual Basic.

### Sub Arcos()

```

'////DECLARACIÓN DE VARIABLES////
Dim x(1 To 100) As Single 'Coordenada X de los puntos
Dim y(1 To 100) As Single 'Coordenada Y de los puntos
Dim Xc(1 To 50) As Single 'Coordenada X del centro del arco provisional
Dim Yc(1 To 50) As Single 'Coordenada Y del centro del arco provisional
Dim Xcf(1 To 50) As Single 'Coordenada X del centro del arco definitivo
Dim Ycf(1 To 50) As Single 'Coordenada Y del centro del arco definitivo
Dim Xm As Single 'Coordenada X de la mediatriz
Dim Ym As Single 'Coordenada Y de la mediatriz
Dim X0(1 To 50) As Single 'Coordenada X del punto de conexión de los arcos
Dim Y0(1 To 50) As Single 'Coordenada Y del punto de conexión de los arcos
Dim d(50, 50) As Single 'Matriz errores Punto - Arco provisionales
Dim df(50, 50) As Single 'Matriz errores Punto - Arco definitivos
Dim dmax As Single 'Error Punto - Arco máximo
Dim e As Single 'Error Punto - Arco permitido
Dim r(1 To 50) As Single 'Radio arcos provisional
Dim rf(1 To 50) As Single 'Radio arcos definitivo
Dim i As Integer 'Contador 1
Dim j As Integer 'Contador 2
Dim k As Integer 'Contador 3
Dim iteraciones As Long 'Contador 4
Dim m As Integer 'Contador 5
Dim a As Integer 'Número arco actual
Dim ap(1 To 50) As Integer 'Número de puntos de cada arco
Dim p As Integer 'Número de puntos que compone la nube de puntos
Dim n As Single 'Número de puntos a agrupar
Dim t As Single 'Testigo
Dim alpha_cos As Single 'Valor coseno
Dim alpha_sin As Single 'Valor seno
Dim alpha_ini(1 To 50) As Single 'Ángulo punto inicio del arco
Dim alpha_fin(1 To 50) As Single 'Ángulo punto final del arco
Dim G As Integer 'Indica si se trata de un arco G2 o G3
Dim Pi As Double 'Valor de pi
Pi = WorksheetFunction.Pi() 'Valor de pi
Dim dcm As Single 'Distancia centro arco - mediatriz segmento
Dim err(1 To 50) As Single 'Error cordal de cada segmento
Dim merr As Single 'Error cordal máximo
Dim itmax As Long 'Número de iteraciones máximas
Dim incr As Double 'Incremento entre iteraciones

```

```
'////AJUSTE DEL PROGRAMA////'
```

```
p = Cells(4, 7)
```

```
e = Cells(8, 7)
```

```
itmax = 200000
```

```
incr = 0.005
```

```
'Bucle para rellenar el array de X e Y con los datos de los puntos
```

```
For i = 1 To p
```

```
    x(i) = Cells(i + 3, 3)   'Se suma 2 porque los datos empiezan en la fila 3
```

```
    y(i) = Cells(i + 3, 4)   'Se suma 2 porque los datos empiezan en la fila 3
```

```
Next i
```

```
'////PRIMER ARCO////'
```

```
'PARA FORMAR EL PRIMER ÁRCO CON Y-
```

```
'Arco número 1
```

```
a = 1
```

```
n = 3
```

```
While dmax <= e
```

```
    'Datos necesarios para hacer el primer arco
```

```
    Xm = (x(1) + x(n)) / 2
```

```
    Ym = (y(1) + y(n)) / 2
```

```
    'Centro inicial del arco
```

```
    Xc(a) = Xm
```

```
    Yc(a) = Ym
```

```
    'Radio inicial del arco
```

```
    r(a) = Sqr((x(1) - Xc(a)) ^ 2 + (y(1) - Yc(a)) ^ 2)
```

```
    'Errores punto-arco iniciales
```

```
    For j = 1 To n
```

```
        d(a, j) = Abs(Sqr((x(j) - Xc(a)) ^ 2 + (y(j) - Yc(a)) ^ 2) - r(a))
```

```
    Next j
```

```
    'Error punto-arco máximo
```

```
    dmax = d(a, 1)
```

```
    j = 2
```

```
    For j = 2 To n
```

```
        If (dmax < d(a, j)) Then
```

```
            dmax = d(a, j)
```

```
        End If
```

```
    Next j
```

'Para grabar el arco

**If** dmax <= e **Then**

**For** j = 1 **To** n

df(a, j) = d(a, j)

Xcf(a) = Xc(a)

Ycf(a) = Yc(a)

rf(a) = r(a)

ap(a) = n

**Next** j

**End If**

'Bucle para disminuir el error

iteraciones = 0

**While** ((dmax > e) And iteraciones < itmax)

'Relocalización del centro del arco

Yc(a) = Yc(a) - incr

$Xc(a) = \frac{(y(n) ^ 2 + x(n) ^ 2 - y(1) ^ 2 - x(1) ^ 2) / (2 * (x(n) - x(1))) - (Yc(1) * (y(n) - y(1))) / (x(n) - x(1))}{}$

r(a) = **Sqr**((x(1) - Xc(a)) ^ 2 + (y(1) - Yc(a)) ^ 2)

'Recálculo de los errores punto-arco

**For** j = 1 **To** n

d(a, j) = **Abs**(**Sqr**((x(j) - Xc(a)) ^ 2 + (y(j) - Yc(a)) ^ 2) - r(a))

**Next** j

'Recalculo del error punto-arco máximo

dmax = d(a, 1)

**For** j = 2 **To** n

**If** (dmax < d(a, j)) **Then**

dmax = d(a, j)

**End If**

**Next**

iteraciones = iteraciones + 1

**Wend**

'Para grabar el arco

**If** dmax <= e **Then**

**For** j = 1 **To** n

df(a, j) = d(a, j)

Xcf(a) = Xc(a)

Ycf(a) = Yc(a)

rf(a) = r(a)

ap(a) = n

**Next** j

**End If**

n = n + 1

## Wend

'PARA FORMAR EL PRIMER ÁNGULO CON Y+

If n = 4 And dmax > e Then

'Arco número 1

a = 1

n = 3

dmax = 0

**While** dmax <= e

'Datos necesarios para hacer el primer arco

$X_m = (x(1) + x(n)) / 2$

$Y_m = (y(1) + y(n)) / 2$

'Centro inicial del arco

$X_c(a) = X_m$

$Y_c(a) = Y_m$

'Radio inicial del arco

$r(a) = \text{Sqr}((x(1) - X_c(a))^2 + (y(1) - Y_c(a))^2)$

'Errores punto-arco iniciales

**For** j = 1 **To** n

$d(a, j) = \text{Abs}(\text{Sqr}((x(j) - X_c(a))^2 + (y(j) - Y_c(a))^2) - r(a))$

**Next** j

'Error punto-arco máximo

dmax = d(a, 1)

j = 2

**For** j = 2 **To** n

**If** (dmax < d(a, j)) **Then**

dmax = d(a, j)

**End If**

**Next** j

'Bucle para disminuir el error

iteraciones = 0

**While** ((dmax > e) And iteraciones < itmax)

'Relocalización del centro del arco

$Y_c(a) = Y_c(a) + \text{incr}$

$X_c(a) = (y(n)^2 + x(n)^2 - y(1)^2 - x(1)^2) / (2 * (x(n) - x(1))) - (Y_c(1) * (y(n) - y(1))) / (x(n) - x(1))$

$r(a) = \text{Sqr}((x(1) - X_c(a))^2 + (y(1) - Y_c(a))^2)$

'Recálculo de los errores punto-arco

**For** j = 1 **To** n

$d(a, j) = \text{Abs}(\text{Sqr}((x(j) - X_c(a))^2 + (y(j) - Y_c(a))^2) - r(a))$

**Next** j

'Recalculo error punto-arco máximo

dmax = d(a, 1)

**For** j = 2 **To** n

**If** (dmax < d(a, j)) **Then**

dmax = d(a, j)

**End If**

**Next** j

iteraciones = iteraciones + 1

**Wend**

'Para grabar el arco

**If** dmax <= e **Then**

**For** j = 1 **To** n

df(a, j) = d(a, j)

Xcf(a) = Xc(a)

Ycf(a) = Yc(a)

rf(a) = r(a)

ap(a) = n

**Next** j

**End If**

n = n + 1

**Wend**

**End If**

'PUNTO DE CONEXIÓN ARCO 1 Y ARCO 2

n = n - 2

X0(a) = x(n)

Y0(a) = y(n)

'Ángulo del punto inicial

alpha\_cos = ArcCos((x(1) - Xcf(a)) / (Sqr(Abs(x(1) - Xcf(a)) ^ 2 + Abs(y(1) - Ycf(a)) ^ 2)))

alpha\_sin = ArcSin((y(1) - Ycf(a)) / (Sqr(Abs(x(1) - Xcf(a)) ^ 2 + Abs(y(1) - Ycf(a)) ^ 2)))

'Para saber en qué cuadrante se encuentra

**If** alpha\_sin > 0 **Then**

alpha\_ini(a) = alpha\_cos

**End If**

**If** alpha\_sin < 0 **Then**

alpha\_ini(a) = 2 \* Pi - alpha\_cos

**End If**

'Ángulo del punto final

alpha\_cos = ArcCos((x(n) - Xcf(a)) / (Sqr(Abs(x(n) - Xcf(a)) ^ 2 + Abs(y(n) - Ycf(a)) ^ 2)))

alpha\_sin = ArcSin((y(n) - Ycf(a)) / (Sqr(Abs(x(n) - Xcf(a)) ^ 2 + Abs(y(n) - Ycf(a)) ^ 2)))

'Para saber en qué cuadrante se encuentra

**If** alpha\_sin > 0 **Then**

alpha\_fin(a) = alpha\_cos

**End If**

**If** alpha\_sin < 0 **Then**

alpha\_fin(a) = 2 \* Pi - alpha\_cos

**End If**

'DEFINIR SI ES G2 O G3 EL PRIMER ARCO

G = 0

Xm = (x(1) + x(n)) / 2

Ym = (y(1) + y(n)) / 2

**If** x(1) = x(n) **Then**

**If** (y(1) < y(n) And Xcf(a) > Xm) **Then**

G = 2

**End If**

**If** (y(1) > y(n) And Xcf(a) < Xm) **Then**

G = 2

**End If**

**If** (y(1) < y(n) And Xcf(a) < Xm) **Then**

G = 3

**End If**

**If** (y(1) > y(n) And Xcf(a) > Xm) **Then**

G = 3

**End If**

**End If**

**If** x(1) < x(n) **Then**

**If** (y(1) < y(n) And Xcf(a) > Xm) **Then**

G = 2

**End If**

**If** (y(1) = y(n) And Ycf(a) < Ym) **Then**

G = 2

**End If**

**If** (y(1) > y(n) And Xcf(a) < Xm) **Then**

G = 2

**End If**

**If** (y(1) < y(n) And Xcf(a) < Xm) **Then**

G = 3

**End If**

**If**  $(y(1) = y(n) \text{ And } Ycf(a) > Ym)$  **Then**

$G = 3$

**End If**

**If**  $(y(1) > y(n) \text{ And } Xcf(a) > Xm)$  **Then**

$G = 3$

**End If**

**End If**

**If**  $x(1) > x(n)$  **Then**

**If**  $(y(1) > y(n) \text{ And } Xcf(a) > Xm)$  **Then**

$G = 3$

**End If**

**If**  $(y(1) = y(n) \text{ And } Ycf(a) < Ym)$  **Then**

$G = 3$

**End If**

**If**  $(y(1) < y(n) \text{ And } Xcf(a) < Xm)$  **Then**

$G = 3$

**End If**

**If**  $(y(1) > y(n) \text{ And } Xcf(a) < Xm)$  **Then**

$G = 2$

**End If**

**If**  $(y(1) = y(n) \text{ And } Ycf(a) > Ym)$  **Then**

$G = 2$

**End If**

**If**  $(y(1) < y(n) \text{ And } Xcf(a) > Xm)$  **Then**

$G = 2$

**End If**

**End If**

'DATOS DEL PRIMER ARCO

Cells(4, 12) = n

Cells(4, 13) = Xcf(a)

Cells(4, 14) = Ycf(a)

Cells(4, 15) = rf(a)

Cells(4, 16) = X0(a)

Cells(4, 17) = Y0(a)

Cells(3 + a, 18) = alpha\_ini(a)

Cells(3 + a, 19) = alpha\_fin(a)

Cells(3 + a, 20) = G

j = 1

**While** (j <= n)

Cells(4, 20 + j) = df(a, j)

j = j + 1

**Wend**

'////RESTO DE ARCOS////'

'Arco número 2

a = 2

**While** n < p And a <= 50

'Punto inicio arco

k = n

'Reset dmax

dmax = 0

**While** dmax <= e And n < p

'Punto siguiente

n = n + 1

'Radio inicial arco

r(a) = 1

'Centro inicial arco

$Xc(a) = (rf(a - 1) - r(a)) / rf(a - 1) * (X0(a - 1) - Xcf(a - 1)) + Xcf(a - 1)$

$Yc(a) = (rf(a - 1) - r(a)) / rf(a - 1) * (Y0(a - 1) - Ycf(a - 1)) + Ycf(a - 1)$

'Errores punto-arco iniciales

j = 1

**For** i = k **To** n

d(a, j) = Abs(Sqr((x(i) - Xc(a)) ^ 2 + (y(i) - Yc(a)) ^ 2) - r(a))

j = j + 1

**Next** i

'Error punto-arco máximo

dmax = d(a, 1)

**For** i = 2 **To** j

**If** (dmax < d(a, i)) **Then**

dmax = d(a, i)

**End If**

**Next** i

'Para grabar el arco

**If** dmax <= e **Then**

**For** i = 1 **To** j

    df(a, i) = d(a, i)

    Xcf(a) = Xc(a)

    Ycf(a) = Yc(a)

    rf(a) = r(a)

    ap(a) = n

    m = j - 1

**Next** i

**End If**

'Bucle para disminuir el error

iteraciones = 0

**While** ((dmax > e) And iteraciones < itmax)

'Nueva situación del centro y radio

r(a) = r(a) + incr

Xc(a) = (rf(a - 1) - r(a)) / rf(a - 1) \* (X0(a - 1) - Xcf(a - 1)) + Xcf(a - 1)

Yc(a) = (rf(a - 1) - r(a)) / rf(a - 1) \* (Y0(a - 1) - Ycf(a - 1)) + Ycf(a - 1)

'Errores punto-arco iniciales

j = 1

**For** i = k **To** n

  d(a, j) = Abs(**Sqr**((x(i) - Xc(a)) ^ 2 + (y(i) - Yc(a)) ^ 2) - r(a))

  j = j + 1

**Next** i

'Error punto-arco máximo

dmax = d(a, 1)

**For** i = 2 **To** j

**If** (dmax < d(a, i)) **Then**

    dmax = d(a, i)

**End If**

**Next** i

'Para grabar el arco

**If** dmax <= e **Then**

**For** i = 1 **To** j

    df(a, i) = d(a, i)

    Xcf(a) = Xc(a)

    Ycf(a) = Yc(a)

    rf(a) = r(a)

    ap(a) = n

    m = j - 1

**Next** i

**End If**

iteraciones = iteraciones + 1

**Wend**

**Wend**

'PARA CAMBIAR LA CURVATURA

**If** n = k + 1 **And** iteraciones = itmax **Then**

'Reset dmax

dmax = 0

**While** dmax <= e

'Radio inicial arco

r(a) = 1

'Centro inicial arco

$Xc(a) = (rf(a - 1) + r(a)) / rf(a - 1) * (X0(a - 1) - Xcf(a - 1)) + Xcf(a - 1)$

$Yc(a) = (rf(a - 1) + r(a)) / rf(a - 1) * (Y0(a - 1) - Ycf(a - 1)) + Ycf(a - 1)$

'Errores punto-arco iniciales

j = 1

**For** i = k **To** n

$d(a, j) = \text{Abs}(\text{Sqr}((x(i) - Xc(a)) ^ 2 + (y(i) - Yc(a)) ^ 2) - r(a))$

j = j + 1

**Next** i

'Error punto-arco máximo

dmax = d(a, 1)

**For** i = 2 **To** j

**If** (dmax < d(a, i)) **Then**

dmax = d(a, i)

**End If**

**Next** i

'Para grabar el arco

**If** dmax <= e **Then**

**For** i = 1 **To** j

df(a, i) = d(a, i)

Xcf(a) = Xc(a)

Ycf(a) = Yc(a)

rf(a) = r(a)

ap(a) = n

m = j - 1

**Next** i

**End If**

'Bucle para disminuir el error

iteraciones = 0

**While** ((dmax > e) And iteraciones < itmax)

'Nueva situación del centro y radio

$r(a) = r(a) + \text{incr}$

$Xc(a) = (rf(a - 1) + r(a)) / rf(a - 1) * (X0(a - 1) - Xcf(a - 1)) + Xcf(a - 1)$

$Yc(a) = (rf(a - 1) + r(a)) / rf(a - 1) * (Y0(a - 1) - Ycf(a - 1)) + Ycf(a - 1)$

'Errores punto-arco iniciales

$j = 1$

**For** i = k **To** n

$d(a, j) = \text{Abs}(\text{Sqr}((x(i) - Xc(a)) ^ 2 + (y(i) - Yc(a)) ^ 2) - r(a))$

$j = j + 1$

**Next** i

'Error punto-arco máximo

$dmax = d(a, 1)$

**For** i = 2 **To** j

**If** (dmax < d(a, i)) **Then**

$dmax = d(a, i)$

**End If**

**Next** i

'Para grabar el arco

**If** dmax <= e **Then**

**For** i = 1 **To** j

$df(a, i) = d(a, i)$

$Xcf(a) = Xc(a)$

$Ycf(a) = Yc(a)$

$rf(a) = r(a)$

$ap(a) = n$

$m = j - 1$

**Next** i

**End If**

iteraciones = iteraciones + 1

**Wend**

'Punto siguiente

$n = n + 1$

**Wend**

**If** iteraciones = itmax **Then**

$n = n - 1$

**End If**

**End If**

'INSTRUCCIÓN QUE SE EJECUTA CUANDO HAY UN ARCO QUE YA COGE EL ÚLTIMO PUNTO Y CONSIGUE HACER CONINUIDAD C1

**If** n = p And iteraciones < itmax **Then**

n = n + 1

**End If**

'PUNTOS DE CONEXIÓN

'Angulo punto inicial

alpha\_cos = ArcCos((x(k) - Xcf(a)) / (Sqr(Abs(x(k) - Xcf(a)) ^ 2 + Abs(y(k) - Ycf(a)) ^ 2)))

alpha\_sin = ArcSin((y(k) - Ycf(a)) / (Sqr(Abs(x(k) - Xcf(a)) ^ 2 + Abs(y(k) - Ycf(a)) ^ 2)))

'Para saber en qué cuadrante se encuentra

**If** alpha\_sin > 0 **Then**

alpha\_ini(a) = alpha\_cos

**End If**

**If** alpha\_sin < 0 **Then**

alpha\_ini(a) = 2 \* Pi - alpha\_cos

**End If**

'Angulo punto final

n = n - 1

alpha\_cos = ArcCos((x(n) - Xcf(a)) / (Sqr(Abs(x(n) - Xcf(a)) ^ 2 + Abs(y(n) - Ycf(a)) ^ 2)))

alpha\_sin = ArcSin((y(n) - Ycf(a)) / (Sqr(Abs(x(n) - Xcf(a)) ^ 2 + Abs(y(n) - Ycf(a)) ^ 2)))

'Para saber en qué cuadrante se encuentra

**If** alpha\_sin > 0 **Then**

X0(a) = rf(a) \* Cos(alpha\_cos) + Xcf(a)

Y0(a) = rf(a) \* Sin(alpha\_cos) + Ycf(a)

alpha\_fin(a) = alpha\_cos

**End If**

**If** alpha\_sin < 0 **Then**

X0(a) = rf(a) \* Cos(-alpha\_cos) + Xcf(a)

Y0(a) = rf(a) \* Sin(-alpha\_cos) + Ycf(a)

alpha\_fin(a) = 2 \* Pi - alpha\_cos

**End If**

'DEFINIR SI ES G2 O G3 EL RESTO DE ARCOS

G = 0

Xm = (x(k) + x(n)) / 2

Ym = (y(k) + y(n)) / 2

**If** x(1) = x(n) **Then**

**If** (y(k) < y(n) And Xcf(a) > Xm) **Then**

G = 2

**End If**

**If**  $(y(k) > y(n) \text{ And } Xcf(a) < X_m)$  **Then**

$G = 2$

**End If**

**If**  $(y(k) < y(n) \text{ And } Xcf(a) < X_m)$  **Then**

$G = 3$

**End If**

**If**  $(y(k) > y(n) \text{ And } Xcf(a) > X_m)$  **Then**

$G = 3$

**End If**

**End If**

**If**  $x(k) < x(n)$  **Then**

**If**  $(y(k) < y(n) \text{ And } Xcf(a) > X_m)$  **Then**

$G = 2$

**End If**

**If**  $(y(k) = y(n) \text{ And } Ycf(a) < Y_m)$  **Then**

$G = 2$

**End If**

**If**  $(y(k) > y(n) \text{ And } Xcf(a) < X_m)$  **Then**

$G = 2$

**End If**

**If**  $(y(k) < y(n) \text{ And } Xcf(a) < X_m)$  **Then**

$G = 3$

**End If**

**If**  $(y(k) = y(n) \text{ And } Ycf(a) > Y_m)$  **Then**

$G = 3$

**End If**

**If**  $(y(k) > y(n) \text{ And } Xcf(a) > X_m)$  **Then**

$G = 3$

**End If**

**End If**

**If**  $x(k) > x(n)$  **Then**

**If**  $(y(k) > y(n) \text{ And } Xcf(a) > X_m)$  **Then**

$G = 3$

**End If**

**If**  $(y(k) = y(n) \text{ And } Ycf(a) < Ym)$  **Then**

$G = 3$

**End If**

**If**  $(y(k) < y(n) \text{ And } Xcf(a) < Xm)$  **Then**

$G = 3$

**End If**

**If**  $(y(k) > y(n) \text{ And } Xcf(a) < Xm)$  **Then**

$G = 2$

**End If**

**If**  $(y(k) = y(n) \text{ And } Ycf(a) > Ym)$  **Then**

$G = 2$

**End If**

**If**  $(y(k) < y(n) \text{ And } Xcf(a) > Xm)$  **Then**

$G = 2$

**End If**

**End If**

### 'REPRESENTAR RESULTADOS

Cells(3 + a, 12) = n

Cells(3 + a, 13) = Xcf(a)

Cells(3 + a, 14) = Ycf(a)

Cells(3 + a, 15) = rf(a)

Cells(3 + a, 16) = X0(a)

Cells(3 + a, 17) = Y0(a)

Cells(3 + a, 18) = alpha\_ini(a)

Cells(3 + a, 19) = alpha\_fin(a)

Cells(3 + a, 20) = G

**For** j = 1 **To** m

Cells(3 + a, 20 + j) = df(a, j)

**Next** j

a = a + 1

**Wend**

'CALCULAR ERRORES CORDALES

a = a - 1

j = 1

k = 1

**For** j = 1 **To** a

i = ap(j)

**While** k < i

$X_m = (x(k) + x(k + 1)) / 2$

$Y_m = (y(k) + y(k + 1)) / 2$

dcm = **Sqr**((Xm - Xcf(j)) ^ 2 + (Ym - Ycf(j)) ^ 2)

**err**(k) = Abs(dcm - rf(j))

k = k + 1

**Wend**

**Next** j

'CALCULAR ERROR CORDAL MÁXIMO

k = k - 1

merr = **err**(1)

**For** j = 2 **To** k

**If** **err**(j) > merr **Then**

merr = **err**(j)

**End If**

**Next** j

Cells(12, 7) = merr

**End Sub**



### Anexo III. Código G para el mecanizado de los álabes.

#### **Código para álabe principal.**

N10 G1 X-248.7 Y-3.355 Z-234.7 B90 C-270 (posicionamiento para anverso)  
N20 G1 X-243.774 Y2.56 (entrada para anverso)  
N30 G3 X-140.385 Z-59.334 R244.131 B40.5 C-180.926 (mecanizado anverso)  
N31 G3 X21.926 Z-16.998 R242.686 B0 C-140.208 (mecanizado anverso)  
N40 G1 X33 Y15.855 (salida después de anverso)  
N50 G1 Y3.355 (posicionamiento para reverso)  
N60 G1 X28.073 Y-2.56 (entrada para reverso)  
N70 G2 X-189.095 Z-111.919 R242.984 B58.5 C-207.908 (mecanizado reverso)  
N71 G2 X-237.635 Z-234.699 R245.003 B90 C-270 (mecanizado reverso)  
N80 G1 X-248.7 Y-15.855 (salida después de reverso)  
N90 G1 Y-3.355 (posicionamiento para anverso)  
N100 G1 C-150 (rotación para mecanizar el siguiente álabe)

#### **Código para álabe intermedio.**

N310 G1 X-248.7 Y0 Z-234.7 B90 C-90 (posicionamiento para anverso)  
N320 G1 X-240.7 Y0 Z-234.7 B90 C-90 (entrada para anverso)  
N330 G1 X-239.533 Y0 Z-225.219 B87.75 C-85.2  
N340 G1 X-237.997 Y0 Z-215.792 B85.5 C-80.413  
N350 G1 X-236.094 Y0.001 Z-206.432 B83.25 C-75.652  
N360 G1 X-233.83 Y0.003 Z-197.154 B81 C-70.929  
N370 G1 X-231.207 Y0.008 Z-187.972 B78.75 C-66.257  
N380 G1 X-228.234 Y0.017 Z-178.9 B76.5 C-61.645  
N390 G1 X-224.918 Y0.034 Z-169.953 B74.25 C-57.105  
N400 G1 X-221.267 Y0.061 Z-161.143 B72 C-52.644  
N410 G1 X-217.294 Y0.103 Z-152.485 B69.75 C-48.271  
N420 G1 X-213.011 Y0.167 Z-143.991 B67.5 C-43.992  
N430 G1 X-208.432 Y0.257 Z-135.675 B65.25 C-39.813  
N440 G1 X-203.576 Y0.384 Z-127.549 B63 C-35.737  
N450 G1 X-198.462 Y0.556 Z-119.626 B60.75 C-31.768  
N460 G1 X-193.111 Y0.785 Z-111.919 B58.5 C-27.908  
N470 G1 X-187.547 Y1.084 Z-104.438 B56.25 C-24.157  
N480 G1 X-181.798 Y1.468 Z-97.195 B54 C-20.517  
N490 G1 X-175.894 Y1.954 Z-90.201 B51.75 C-16.987  
N500 G1 X-169.867 Y2.56 Z-83.468 B49.5 C-13.565  
N510 G1 X-162.855 Y2.56 Z-77.005 B47.25 C-10.251  
N520 G3 X21.926 Z-17 R242.703 B0 C39.792 (mecanizado anverso)  
N530 G1 X33 Y15.855 (salida después de anverso)  
N540 G1 Y3.355 (posicionamiento para reverso)  
N550 G1 X28.073 Y-2.56 (entrada para reverso)  
N560 G2 X-183.172 Z-104.438 R242.921 B56.25 C-24.157 (mecanizado reverso)  
N570 G1 X-189.942 Y-1.854 Z-111.919 B58.5 C-27.908  
N580 G1 X-196.218 Y-1.313 Z-119.626 B60.75 C-31.768  
N590 G1 X-202.027 Y-0.906 Z-127.549 B63 C-35.737  
N600 G1 X-207.394 Y-0.608 Z-135.675 B65.25 C-39.813  
N610 G1 X-212.339 Y-0.393 Z-143.991 B67.5 C-43.992  
N620 G1 X-216.877 Y-0.244 Z-152.485 B69.75 C-48.271  
N630 G1 X-221.021 Y-0.144 Z-161.143 B72 C-52.644  
N640 G1 X-224.781 Y-0.08 Z-169.953 B74.25 C-57.105  
N650 G1 X-228.164 Y-0.041 Z-178.9 B76.5 C-61.645

N660 G1 X-231.175 Y-0.019 Z-187.972 B78.75 C-66.257

N670 G1 X-233.817 Y-0.008 Z-197.154 B81 C-70.929

N680 G1 X-236.09 Y-0.002 Z-206.432 B83.25 C-75.652

N690 G1 X-237.996 Y-0.001 Z-215.792 B85.5 C-80.413

N700 G1 X-239.533 Y0 Z-225.219 B87.75 C-85.2

N710 G1 X-240.7 Y0 Z-234.7 B90 C-90

N720 G1 X-248.7 (salida después de reverso y posicionamiento para anverso)

N730 G1 C30 (giro para mecanizar el siguiente álabe)

## Anexo IV. Representación de las trayectorias generadas.

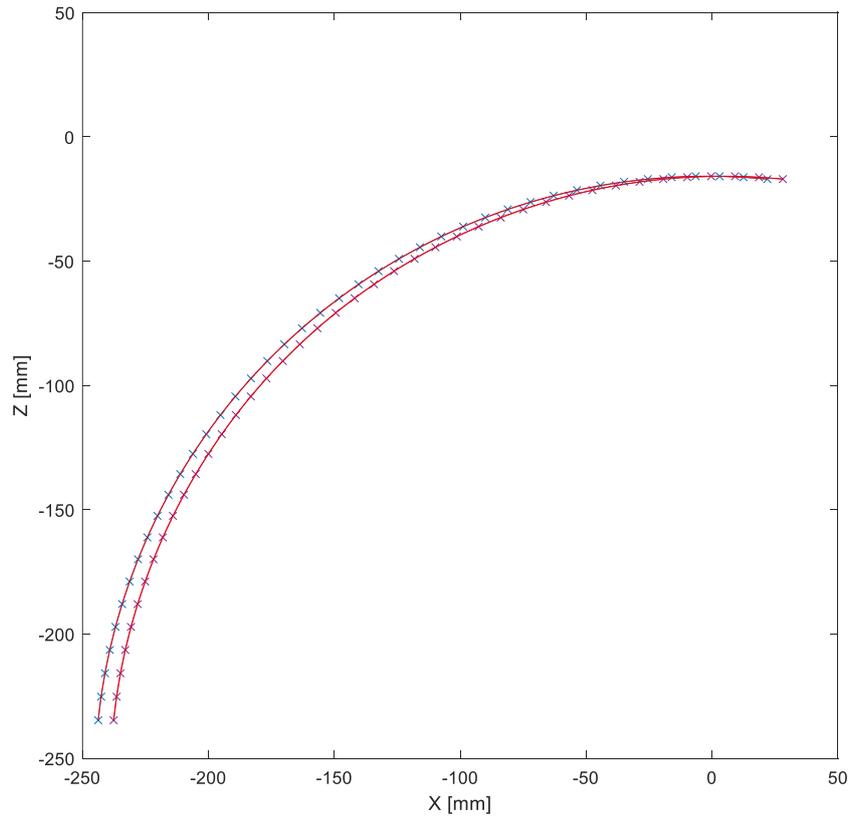


Figura Anexo IV. 1: Aproximación circular del camino de herramienta para álabe principal.

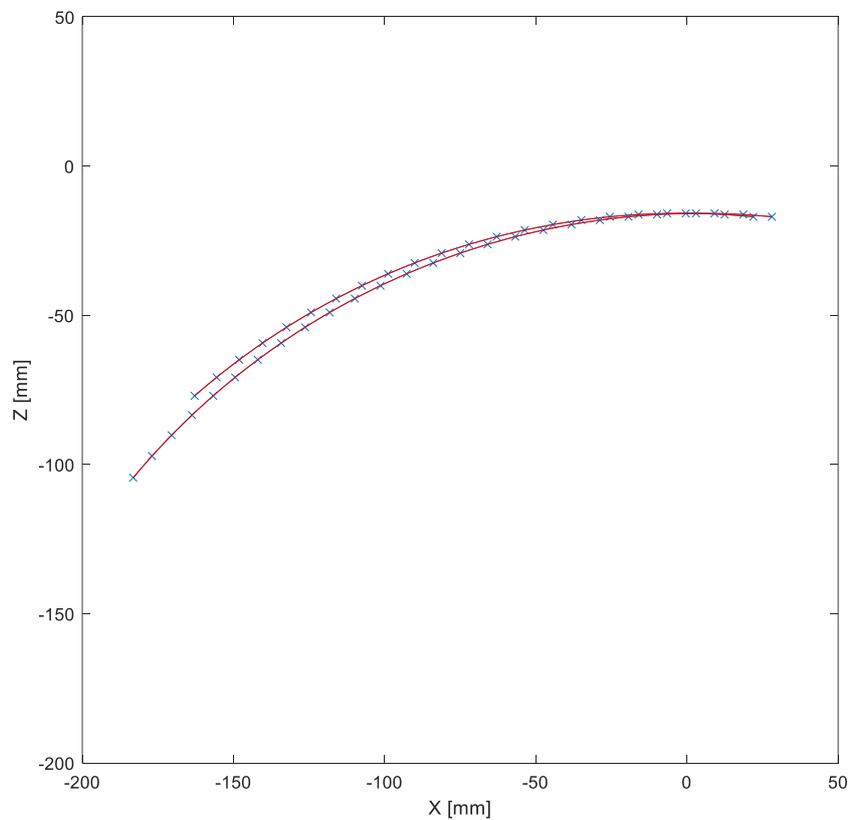


Figura Anexo IV. 2: Aproximación circular del camino de herramienta para álabe intermedio.