

# Sistema de tiempo real embebido para la conversión de máquinas herramienta manuales a máquinas de control numérico industriales

*Ing. Edgardo Gho, Ing. Martin Ferreyra, Sr. Federico García, Ing. Fernando Szklanny*

*Departamento de Ingeniería e Investigaciones tecnológicas*

*Universidad Nacional de La Matanza*

*egho@unlam.edu.ar; mferreyra@unlam.edu.ar; garcia\_federico@hotmail.com.ar;  
fszklanny@unlam.edu.ar*

## **Resumen**

*En el presente trabajo se describe el diseño e implementación de un sistema en tiempo real diseñado para ser utilizado en una placa de desarrollo embebido (en este caso Raspberry PI 3B), con el fin de automatizar una máquina herramienta manual. Este tipo de máquina herramienta se encuentra comúnmente en ámbitos industriales no actualizados tecnológicamente o en pequeños talleres que no pueden afrontar el costo de la automatización. Mediante este sistema de control se podrá aumentar la precisión de dichas máquinas, superando los problemas mecánicos de los sistemas manuales y obteniendo un grado de precisión en el maquinado comparable con máquinas herramientas de avanzada, partiendo de precisiones un orden de magnitud menor.*

*La solución propuesta es apta para trabajar con sistemas a lazo abierto (sin retroalimentación) o a lazo cerrado (con retroalimentación de posición o velocidad), lo que lo transforma en un diseño muy versátil a la hora de ser implementado independientemente del tipo de máquinas mencionadas, como puede ser el caso de impresoras 3D, tornos, fresadoras o cualquier otra máquina que necesite movimientos coordinados en dos o más ejes.*

## **Reseña histórica**

Durante varios años el grupo de investigación responsable de este proyecto tuvo como objetivo final el desarrollo de dispositivos y sistemas de medición de bajo costo, con el objeto de automatizar y modernizar máquinas herramientas manuales para transformarlas en sistemas de control numérico computarizado (CNC) [1]. El presente trabajo describe la culminación de esta investigación, los problemas encontrados, las soluciones propuestas a tal fin y los resultados obtenidos.

## **Motivación**

En la Argentina existen numerosos talleres de mecanizado que poseen diversos tipos de máquinas herramientas. Estas máquinas, en muchas ocasiones, significan una inversión a largo plazo que con el pasar del tiempo, debido al deterioro o a la

obsolescencia, no ofrecen una ventaja competitiva. Sumado a esto se añade el factor de los tiempos invertidos en la creación de las piezas a producir, lo que retrasa la producción de nuevos pedidos. Si bien lo planteado hace foco en una cuestión meramente económica, es necesario destacar que el deterioro natural de estas máquinas herramientas produce inconvenientes en el resultado final de la pieza mecanizada. Éstos problemas derivan en inexactitudes en el mecanizado, las que en un principio se encontrarán dentro de las tolerancias permitidas, aunque luego se transformarán en errores difíciles de corregir o engorrosos de controlar. Esta fue en parte la motivación inicial para desarrollar esta investigación, cuyo objetivo final fue automatizar y compensar los errores de este tipo de máquinas herramientas, por medio del uso de un sistema de control a lazo cerrado.

## **Componentes del sistema**

En un sistema de control numérico existen distintos elementos a ser controlados o monitoreados, como motores, distintos tipos de actuadores, elementos para realizar mediciones (como reglas ópticas o codificadores rotativos), entre otros. Junto con todos estos elementos existe un dispositivo electrónico que debe coordinar las acciones que se deben realizar con estos componentes.

En función del tipo de máquina herramienta a controlar [2], los componentes pueden ser muy variados. Si se toma como ejemplo un torno, la máquina herramienta incluirá entre sus partes un actuador rotatorio (que rota la pieza a maquinar) el que debe ser controlado en términos de velocidad (R.P.M.). Además, existen también uno o más actuadores lineales que posicionan la herramienta de desbaste a lo largo de uno o varios ejes. En este caso la coordinación debe darse de manera precisa y en conjunto entre la velocidad de rotación de la pieza, y la de los distintos ejes que posicionan a la herramienta.

Del ejemplo anterior, se puede deducir y hacer extensible que surgen dos tipos de elementos de control: aquellos elementos que se controlan en velocidad y aquellos que se controlan en base a su

posición. Si los elementos deben ser monitoreados, el sensor que realiza la medición, debe adaptarse al tipo de elemento de control. En general se utilizan codificadores rotativos para elementos controlados en velocidad y reglas ópticas para elementos de control lineal o discreto.

Si el elemento a controlar es del tipo de control por velocidad, debe utilizarse una interfaz de control acorde. En estos casos el elemento suele ser un motor universal en alguna de sus variantes (corriente continua o alterna) elegido para cumplir con los requisitos de velocidad y torque acordes a la aplicación. El sistema de control numérico controla el motor utilizando una salida de modulación por pulsos (PWM) donde la variación de ancho de pulso se traduce en distintas velocidades del motor.

Si el elemento a controlar es del tipo de control por posición, suele utilizarse un motor paso a paso en donde la relación de paso a distancia recorrida es fija y conocida. Dado que este tipo de motores solamente se desplaza en incrementos discretos que producen un avance fijo, es posible llevar una referencia interna de la posición y actualizarla luego de cada movimiento del motor. En este caso en particular el control se realiza a “lazo abierto”, ya que no existe una retroalimentación de la posición real, sino una posición calculada. En el caso donde se encuentre un elemento auxiliar que informa la posición real y esta información es tenida en cuenta por el sistema de control, éste trabaja a “lazo cerrado”.

Por otra parte el dispositivo electrónico que lleva el control del sistema puede acarrear distintos grados de complejidad [3]. Su objetivo principal es llevar el registro de la posición o velocidad donde se encuentra la máquina herramienta y realizar acciones para moverla hacia una nueva posición, velocidad o combinación de ambas. Las distintas posiciones suelen estar programadas en una secuencia acompañada de tipos de movimientos predeterminados. Estos tipos de movimientos definen el camino intermedio que recorre la maquina desde que parte de un punto a otro. Los más comunes son movimientos en línea recta y movimientos en arco.

El proceso de generar todos los puntos intermedios desde un punto de origen a un punto de destino se denomina interpolación[4]. Esta interpolación puede ser resuelta en un sistema secundario o auxiliar y luego ser utilizada por el sistema de control numérico, aunque el caso más común es que el sistema de control numérico realice la interpolación por sí mismo. El control de los motores requiere menos capacidad de procesamiento del sistema electrónico, cuando se lo compara con la capacidad de procesamiento requerida para realizar la interpolación. Aquí reside el grado de complejidad mencionado anteriormente respecto del dispositivo electrónico de control.

Si bien el grado de complejidad es menor para el control de los motores, es en el manejo de los tiempos, para la generación de señales de control,

donde radica el principal problema, mientras que no tiene demasiada importancia durante la interpolación.

### Problemas comunes

En un sistema de mecanizado donde cualquiera de sus ejes se desplaza mediante un tornillo sin fin, hacia ambas direcciones, y dicho eje deba realizar un esfuerzo al mover una carga (como una mesa) su operador debe lidiar con ciertos problemas que se detallan a continuación entre otros.

- **Necesidades de torque:** En función de cómo interactúen los distintos ejes y de las imperfecciones inherentes de la fabricación del tornillo, la fuerza de torsión (torque) que debe aplicar el motor al sistema puede variar dependiendo de la sección del tornillo que se utilice, con lo cual puede ocurrir que el motor no sea capaz de generar el torque necesario y por ende desplace al sistema menos de lo esperado.

- **Juego (Backlash):** Este fenómeno ocurre cuando se invierte el sentido de rotación del tornillo y en el momento de dicha transición, el acople de la mesa al tornillo gira en falso, no produciéndose ningún tipo de movimiento. Esto ocurrirá hasta que el sistema mesa-tornillo se sincronice nuevamente. Dicho fenómeno es inherente a una unión de una tuerca con un tornillo, ya que este último posee juego para no desgastar tan rápidamente el tornillo, aunque en máquinas que están desgastadas por su uso, este fenómeno se acentúa. El backlash implica que los motores giren y hagan girar al tornillo sin lograr ningún movimiento por unos momentos. En la figura 1 se grafica un ejemplo de sistema tuerca tornillo donde se muestra de dónde proviene el backlash.

- **Movimientos impredecibles en el sistema:** En máquinas antiguas en donde existen más de un eje, el movimiento de uno de ellos provoca movimientos sobre otros, creando desplazamientos impredecibles. El desgaste de la máquina también puede generar oscilaciones que no pueden advertirse con anterioridad.

Para lidiar con los problemas relacionados con las distintas necesidades de torque, se debe configurar el sistema de mecanizado con las tolerancias correspondientes del motor, y variar el torque dependiendo de cada tarea. Para lograr este fin, se deben calibrar algunos parámetros como la velocidad y la aceleración máxima de los motores, con el objeto de evitar los problemas resultantes de un torque deficiente.

Para lidiar con el backlash, en este tipo de sistemas, las acciones a tomar se tornan un poco más

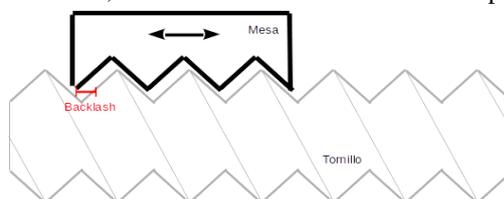


Figura 1: Ejemplificación de backlash

complejas. El backlash puede ser atacado desde varios flancos, siendo el más común utilizar tornillos con bolas recirculantes o tornillos comunes de alta calidad. Esto implica sucumbir en una solución mecánica que no satisface el objetivo principal de este proyecto. Además, esto involucra adaptar las máquinas de una forma mucho más profunda, lo que necesariamente implica un costo mucho mayor. Otro punto negativo que poseen los tornillos con bolas recirculantes es que los residuos del maquinado pueden producir que el sistema falle o lamentablemente deje de funcionar. Aun cuando se tomen todas las precauciones para evitar problemas en esta clase de tornillos los mismos poseen imperfecciones en su fabricación que se trasladan al mecanizado aunque estas imperfecciones son casi despreciables.

Por otra parte, una de las formas de evitar el backlash es calcularlo. Para esto es imprescindible configurar el sistema (a lazo abierto) con el movimiento necesario que debe realizar el motor para compensar el backlash cada vez que ocurra un cambio de sentido. Este método puede funcionar correctamente siempre y cuando el backlash sea constante, pero si el tornillo posee defectos o el sistema en su conjunto posee defectos haciendo que el backlash no sea constante, este método no cumplirá satisfactoriamente con su cometido.

Existe además otra forma en la que se puede atacar este fenómeno, de manera mucho más drástica. Es posible evitar el backlash realizando el mecanizado de tal forma que los ejes del sistema siempre se muevan en un solo sentido. Esto es viable, ya sea mediante la realización de cambios físicos en la máquina herramienta, o retirando la herramienta en la máquina cuando se configura el cambio de sentido en algún eje. Esta técnica (muy utilizada cuando el maquinado es manual y no mediante un sistema de control numérico) presenta algunas complicaciones a la hora de ejecutarse, debido que se requiere volver a sincronizar la posición de la pieza en el momento del cambio de sentido. Cabe destacar que en la mayoría de los casos el backlash no se elimina, sino que se lo trata de minimizar lo máximo posible, excepto en este último caso en el que si bien se elimina, volver a sincronizar la pieza puede provocar errores.

Por último, existe una última manera para compensar o minimizar el backlash, sin la necesidad de modificar la máquina herramienta a gran escala o evitando cambiar de sentido el mecanizado.

La solución consiste en el uso de un sistema a lazo cerrado. Cuando se utiliza un sistema a lazo cerrado, el backlash se detecta cuando se exceda cierto nivel de error. Este error será la diferencia entre donde se encuentra efectivamente la mesa y en donde efectivamente debería estar. Detectado el error, el sistema a lazo cerrado compensará esta inconsistencia, mediante el movimiento del motor para compensar el backlash o cualquier otro tipo de

error mediante una nueva sincronización del o de los motores.

Utilizar un sistema a lazo cerrado por otra parte soluciona el problema de los movimientos o fallas impredecibles en el sistema, o al menos lo minimiza.

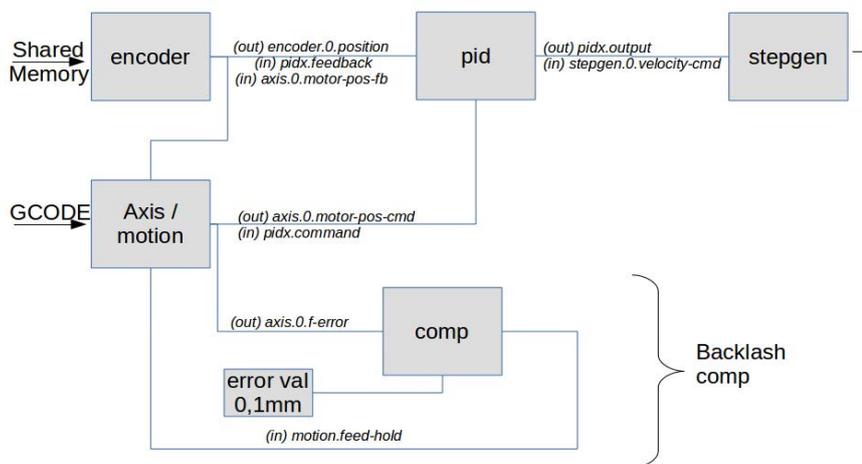
### **Implementación**

Para ofrecer una solución a los problemas mencionados en la sección anterior, se decidió tomar una mesa de mecanizado modelo, la cual sufría varios de los inconvenientes descriptos. A esta mesa se le desarrolló e implementó un sistema el cual se detallará a continuación:

El sistema desarrollado se implementó sobre una mesa de mecanizado del tipo XY. Este tipo de mesa posee dos ejes compuestos de un tornillo y un carro que se desplaza a lo largo de una recta. Los ejes se montan de forma perpendicular de forma tal que el carro puede moverse sobre un plano.

Cada uno de los tornillos es girado por un motor paso a paso que se encuentra unido al tornillo mediante una serie de acoples que compensan desfases en la alineación. Estos motores están energizados por excitadores (drivers) que regulan el torque del motor en función de la corriente máxima de salida prefijada. Estos drivers brindan entradas digitales para ser interconectadas a un sistema de control de forma opto-aislada, separando así las perturbaciones eléctricas que puedan ocurrir en el manejo de la potencia del motor y de la lógica de control de los mismos.

La posición de los carros de cada eje en el plano XY se obtiene mediante reglas ópticas. Estas reglas son de tipo incrementales, con lo cual se requiere un circuito capaz de almacenar la posición y modificarla en función de los incrementos y decrementos de las reglas. En una primera etapa del desarrollo del proyecto, se elaboró este circuito utilizando dispositivos de lógica programable, aunque por una razón de costos y practicidad se utilizó la interfaz de temporización de un microcontrolador ARM (STM32F103) [5] en modo lectura de codificador. Este dispositivo brinda un registro de 16 bits con la posibilidad de poner 16 bits previos de escala. Las reglas utilizadas tienen una resolución de 5 micrones, lo que, dadas las condiciones de la mesa XY utilizada crea un efecto de oscilación (jitter) sobre la lectura de las mismas durante el movimiento y mecanizado. Para evitar esto se configuró el registro de escala para dividir el valor de la posición en dos. Esto reduce la resolución a 10 micrones (0,01 milímetros) pero elimina el jitter de forma eficiente; sumado a esto también reduce la capacidad del registro para medir distancias, ya que con, los 16 bits con resolución de 0,01mm se pueden medir como máximo 655,35 milímetros. En el caso de esta implementación fue suficiente ya que el largo de cada eje no superaba los 200 milímetros aunque en el caso de ser necesario puede utilizarse un registro adicional de mayor tamaño y llevar la cuenta de la



**Figura 2: Implementación de los módulos en Machinekit**

posición en dos etapas. El valor de la posición para los ejes X e Y se reporta mediante una interfaz serie sincrónica (SPI) que opera en modo esclavo. Por ende, el dispositivo que desee leer la posición debe enviarle un pulso de selección (para permitir almacenar el valor de la posición en el buffer de salida) y 32 pulsos de reloj en los cuales se obtiene bit a bit la posición de la mesa. Esta posición se transmite como dos números de 16 bits en complemento a la base, los que indican la cantidad de incrementos de 0,01mm que se encuentra desplazado cada eje con relación a un punto de origen arbitrario. El valor del punto de origen

puede definirse mediante pulsadores en cualquier momento. La posición de la mesa se puede visualizar mediante un display alfanumérico de 2 líneas donde se representa la posición en milímetros (con dos decimales de precisión en punto fijo).

El sistema de control se implementa en una placa de desarrollo embebida Raspberry Pi 3B[6]. Esta placa de propósitos generales (orientada al desarrollo de aplicaciones embebidas) posee un procesador ARM, memoria de escritura/lectura e interfaces varias para video, GPIO y SPI entre otros.

El núcleo del sistema operativo utilizado debe brindar soporte a aplicaciones de tiempo real. Es por ello que se utilizó el núcleo GNU/Linux, el que fue modificado para soportar distintas prioridades en el planificador de tareas.

Estas extensiones al planificador de tareas se conocen como *Preempt-RT*. El resto del sistema operativo está basado en Tynycore Linux[7], con paquetes armados a medida de la aplicación. El software de mecanizado se ejecuta dentro de la placa Raspberry Pi. El software es una versión modificada del proyecto LinuxCNC llamado MachineKit que puede ser ejecutada en arquitecturas ARM. Este software no existe en forma completa para sistemas embebidos, sino que se brinda como un conjunto de bibliotecas y componentes que deben ser compilados y configurados según la aplicación.

Luego de compilarse MachineKit se procedió a configurarlo para adaptarlo a la configuración de

motores y sensores utilizados en la mesa XY. Se desarrolló un programa secundario que, corriendo en segundo plano, se dedica a leer el estado de las reglas ópticas utilizando el canal SPI además de comunicar la posición mediante un mecanismo de memoria compartida al que accede MachineKit desde su módulo Encoder. Dado que esta información es la representación momento a momento de la posición de la mesa, el programa secundario se encarga de registrar en un archivo de bitácora todas las posiciones reportadas por las reglas. De esta forma pueden reconstruirse todos los movimientos ejecutados pudiendo ser representado de forma gráfica. Esto resulta muy útil durante la configuración y puesta en marcha de la mesa, ya que permite hacer simulaciones de maquinados moviendo la mesa sin realizar el maquinado propiamente dicho. Las imágenes presentadas en este trabajo fueron capturadas y generadas de esta forma.

Como se mencionó anteriormente existen dos configuraciones que pueden seleccionarse para la operación de la máquina herramienta. La primera de ellas es a lazo abierto donde la posición de la mesa se mantiene en una referencia interna y el movimiento depende exclusivamente de esta referencia. Con esta configuración, la mesa XY no compensa errores (principalmente mecánicos) al no existir ningún tipo de realimentación. Esta configuración es trivial y no se recomienda su uso a menos que la máquina herramienta tenga otros mecanismos de compensación de fallas mecánicas.

La segunda opción de configuración es a lazo cerrado. En esta configuración existe un dispositivo que informa la posición de los actuadores y de la máquina al software y al sistema de control. En nuestro caso las reglas son el dispositivo que le informa a Machinekit la posición de la mesa XY. De esta forma es posible tomar acciones cuando la máquina esta cometiendo un error de mecanizado, es decir, cuando el resultado de la interpolación no sea exactamente el que se este ejecutando Por esta razón (junto con otras de mayor peso) es que es necesario utilizar un sistema operativo de tiempo real, ya que la

corrección del mecanizado debe ocurrir en el menor tiempo posible (o respetando ciertas tolerancias) y el error debe ser detectado rápidamente.

Para poder implementar el lazo cerrado en Machinekit aunando todos los componentes descriptos y solucionar los problemas mencionados en la sección “Problemas comunes” se hizo uso de los módulos que ofrece Machinekit. Si bien no es objetivo de este trabajo describir en detalle la interconexión modular realizada se puede observar la misma en la figura 2.

El diagrama muestra el módulo *Encoder* el cual se encarga de leer la posición de las reglas. Dichas posiciones son leídas por memoria compartida, mecanismo de IPC (comunicación entre procesos) que resultó ser el más apropiado y efectivo para este fin. Luego se encuentra el módulo *Axis/Motion* que se encarga de interpretar GCODE, el código que describe el mecanizado. Por otra parte, el módulo *Stepgen* se dedica a generar los pulsos para los motores.

Entre todos los módulos, se encuentra un módulo *PID*. Este módulo se encargará de compensar ciertos errores repetitivos en la máquina herramienta. En nuestro caso modelo solo fue necesario ajustar el valor proporcional.

Por último, se muestra una sección representada como *Backlash comp*. Esta sección se encarga de mantener la interpolación pausada en el caso que el error en algún eje sea mayor a 0,1mm. De esta forma el *PID* corrige el error hasta que vuelve a valores tolerables sincronizando los motores. Esto es imprescindible para solucionar el problema de backlash.

Cabe destacar que el módulo *Encoder* tuvo que ser adaptado para poder leer las reglas a través de la memoria compartida.

### Resultados obtenidos

La figura 3 muestra la relación entre el aumento de la precisión y el costo relativo[8]. Como se puede observar en dicha figura, una máquina con precisión “áspera” tiene una tolerancia de  $\pm 0,762$ mm. Esta es la tolerancia esperada en un maquinado manual, comparable con la máquina herramienta que fuera utilizada como modelo en el proyecto. Una máquina estándar preparada para control mecanizado tiene una tolerancia de  $\pm 0,127$ mm. Mediante el agregado de las reglas se logra reducir el error a 0,1mm, particularmente en el caso del backlash (uno de los principales problemas de dicha máquina), con una precisión de 0,01mm.

Por lo tanto, se puede concluir que los resultados obtenidos satisfacen a la resolución del problema original: convertir una máquina herramienta manual a una máquina de control numérico. Esta conclusión es validada a partir de la conversión exitosa de la máquina modelo en un sistema de control numérico, teniendo en cuenta las tolerancias aceptables a las que operan. Un ejemplo de la diferencia entre el

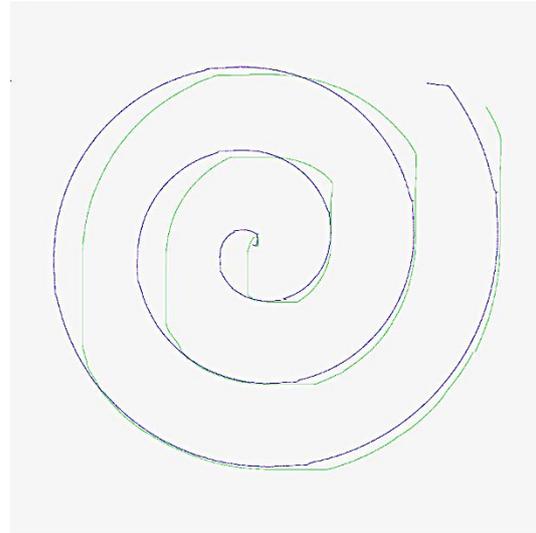


Figura 4: En verde el mecanizado a lazo abierto. En azul el mecanizado a lazo cerrado

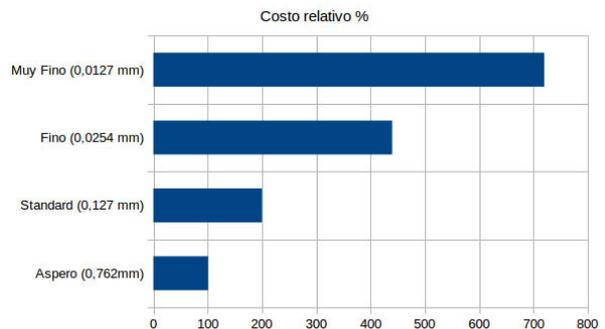
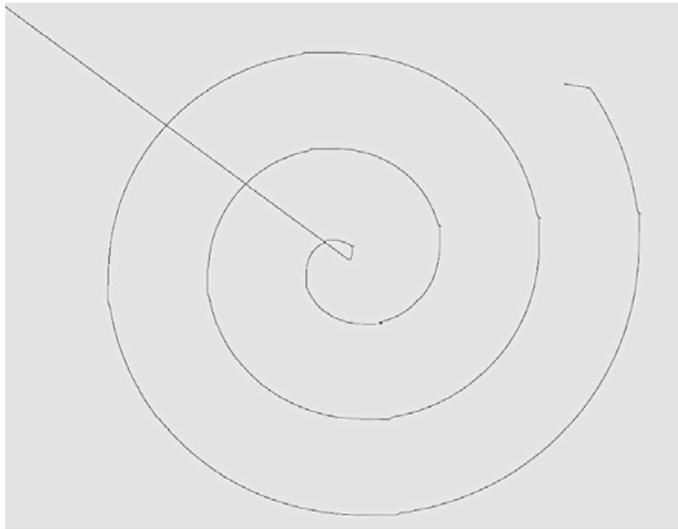


Figura 3: Relación calidad de maquinado-costo

maquinado a lazo abierto y a lazo cerrado se puede observar en la figura 4. Si bien el sistema a lazo cerrado permite solucionar inconvenientes en el mecanizado, no puede solucionar todos los problemas mecánicos. Suponemos que, si se modificara la máquina utilizada con piezas que reduzcan el backlash, la misma podría alcanzar la categoría de maquinado fino (“fine machining”) definido como aquella que ofrece precisiones de 0,0254 mm. Cabe destacar el incremento en el costo relativo de dicha precisión. Este incremento se debe a los costos de los materiales necesarios para incrementar la precisión, como las reglas ópticas que deben tener una resolución acorde. Sin embargo, los elementos de control (que son el alcance de este trabajo) pueden acomodarse sin problema a mayores precisiones. Todos los parámetros de precisión son modificables y las dimensiones configurables con lo cual el sistema desarrollado es escalable.

### Caso de Prueba: Espiral

Para poder estudiar el comportamiento de la máquina herramienta utilizada en el desarrollo del proyecto, se utilizó como caso de prueba un mecanizado en espiral. La forma del mecanizado no fue elegida al azar, sino que se eligió una espiral



**Figura 5: Maquinado a lazo cerrado con compensación de backlash**

debido a los distintos cambios de sentido que posee esta figura. Además, la espiral posee distintos arcos, una de las figuras más complejas de realizar para nuestra máquina herramienta.

Debido a que los cambios de sentido en la espiral fuerzan a que ocurra backlash, sería posible comprobar si el mismo resulta compensado dentro de las tolerancias especificadas por el sistema de control.

Por otra parte, los arcos verificarían si los movimientos impredecibles en el sistema pueden ser corregidos. Éste sería entonces un caso complejo de resolver para la máquina herramienta de prueba.

Dado que el mecanizado depende en gran medida de la calibración, la puesta a punto de la mesa y del material a ser mecanizado, se determinó realizar la captura de la posición de las reglas en forma paralela a la ejecución del maquinado. Esto permite obtener todas las posiciones por las que se desplazaron los ejes del plano XY, por lo que el resultado puede ser analizado en forma gráfica de una manera sencilla y con el mayor grado de precisión posible como fue explicado en el apartado “Implementación”

En las imágenes que se analizarán a continuación, cada píxel representa un punto en el plano XY con una resolución de 0,01mm. El eje Z no fue tenido en cuenta para las capturas, por lo tanto, todos los

movimientos de posicionamiento previos al mecanizado se encuentran también reflejados en la imagen. La figura 5 muestra una espiral contenida en un espacio de 30x30 mm maquinada en la máquina herramienta utilizada. La línea diagonal representa el posicionamiento de la máquina en el centro de la espiral. Puede apreciarse que dicha línea posee una pendiente constante y que el error en este tipo de

Si se divide la figura 5 en cuatro cuadrantes, se puede notar fácilmente errores de continuidad en el momento que algún eje cambia de sentido. Estos errores se repiten de manera consistente en los cuadrantes diametralmente opuestos lo cual coincide con la idea de que se producen por backlash.

En los fragmentos capturados en la figura 6 a se puede observar fácilmente un cambio de sentido en el eje horizontal (mientras que el eje vertical continúa con su movimiento ascendente) y el error asociado en las tres instancias.

Si se amplía la primera imagen (figura 6 b), se puede observar con mayor detalle el cambio de sentido en el eje horizontal y cómo este en un principio falló, moviéndose el eje horizontal hacia la derecha.



**Figura 6: a) Detalle de error debido a backlash y compensación. b) Primer figura donde se puede apreciar el cambio de sentido**

Utilizando la grilla como referencia (con una unidad de 0,01mm por cuadrado) se observa que el eje horizontal se corrige y el error queda contenido dentro de los 0,1mm de tolerancia que fueron especificados para el sistema. Esto significa que cuando se detectó el error en el módulo “Backlash comp”, pausó el módulo axis/motion y el módulo PID compensó el error.

En el caso opuesto (figura 7), el eje vertical se desplaza de arriba hacia abajo de forma constante y el eje horizontal cambia de sentido (de manera inversa a la imagen previa). En este caso no se aprecia el error corregido de forma tan drástica en el eje horizontal, sino que el mismo se distribuye a lo largo del movimiento del eje vertical. Esto ocurre ya que el cambio de sentido y la inercia del eje quedaron contenidos dentro de la tolerancia de 0,1mm, por lo tanto, no se necesitó pausar el eje vertical para compensar el movimiento. Es por este motivo que se generó una recta.

En la figura 8 se observan puntos faltantes en el trazo durante un cambio de sentido. El movimiento se realizó de izquierda a derecha y de abajo hacia arriba. En este punto el eje vertical es el que cambia de sentido y el eje horizontal se mantiene constante. Los puntos faltantes se deben a problemas de comunicación entre las reglas y el sistema de control.

Si bien hubiese sido deseable no tener dichos inconvenientes, es interesante observar cómo se comporta el sistema ante su aparición. Cuando se retomó la comunicación, el sistema calculó el error en los ejes y procedió a detener el eje horizontal por completo. Luego de avanzar con el eje vertical hasta disminuir el error a menos de 0,1mm donde se volvió a detener la marcha en el mismo eje para luego compensar el error en el eje horizontal. Luego la máquina avanzó con el eje vertical hasta que el error en ambos ejes fue menor que la tolerancia. Por último ambos ejes se movieron en conjunto (otra vez compensando backlash en el eje horizontal).

El sistema compensó dos veces problemas de backlash en el eje horizontal y el error total estuvo contenido dentro de 0,16mm. Esto es normal ya que el sistema actúa luego de detectar errores mayores a 0,1mm. Se puede decir que el sistema se comportó de manera adecuada ante el problema y resolvió de manera aceptable la situación.

### Conclusiones

Como se describió durante todo el trabajo, se logró cumplir con el objetivo original del proyecto de transformar una máquina herramienta de acción manual en una máquina herramienta automatizada mediante un sistema de control numérico. El sistema desarrollado es escalable a máquinas que requiera mayor precisión. El sistema además permite reutilizar máquinas herramientas desgastadas con problemas mecánicos definiendo tolerancias que son estándares en el mercado actual.

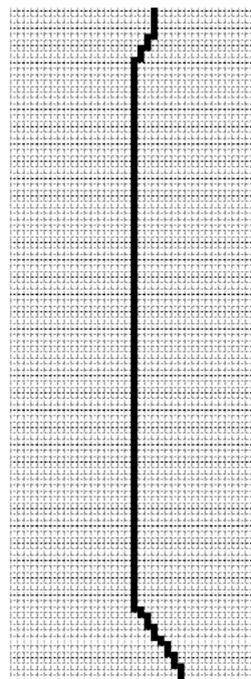


Figura 7: Cambio de sentido y backlash dentro de la tolerancia

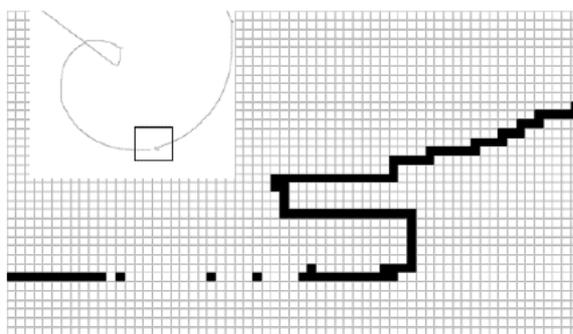


Figura 8: Problemas de comunicación en un cambio de sentido y corrección del error.

Si bien, por distintos requerimientos del proyecto (ajenos al objetivo de este documento), tanto de índole tecnológica como económica, el proyecto comenzó su desarrollo mediante la utilización de dispositivos lógicos programables de alto costo, el mismo fue migrando a circuitos y componentes muy económicos y con gran disponibilidad en el mercado actual. Las bibliotecas de software utilizadas son de uso libre con lo cual su uso no implica el pago de licencia o regalía alguna a su uso. Además los cambios realizados en los mismos están debidamente identificados en el caso de ser requeridos para cumplir con licencias de software libre al momento de su publicación.

La máquina herramienta utilizada para este desarrollo fue sufriendo a lo largo del tiempo desgastes y diversos problemas mecánicos. El sistema se adaptó a estos problemas y logro hacerla funcionar dentro de las tolerancias configuradas y

esperadas. Este trabajo tuvo como finalidad modernizar una máquina de herramienta antigua, pero a lo largo del desarrollo del mismo se hicieron pruebas con componentes más modernos, los cuales pueden utilizarse para armar una máquina más moderna (como puede ser una impresora 3D, un láser para realizar grabados, un sistema CNC de pequeño porte). Durante el desarrollo del proyecto aparecieron en el mercado componentes de gran calidad a muy bajo costo, los cuales son utilizados en sistemas de mecanizado de pequeño porte (hogareños o hobbistas). Estos sistemas suelen tener sistemas de control con prestaciones básicas. Una alternativa para los mismos es adaptarlos a sistemas de control industriales, pero los costos de los mismos hacen que esto sea prohibitivo. El sistema de control desarrollado puede fácilmente adaptarse a estos sistemas de pequeño porte brindando una alternativa muy completa para extender las capacidades de estos equipos.

Por último, durante el desarrollo del proyecto se realizaron pruebas utilizando la API de Python. Esta permitiría en una etapa posterior de este proyecto realizar automatizaciones industriales para líneas de producción en serie que requieran mecanizado de piezas en una o varias de sus etapas.

### **Agradecimientos**

Los autores desean agradecer el apoyo continuo recibido desde el Departamento de Ingeniería e Investigaciones Tecnológicas desde el comienzo del proyecto desarrollado. Asimismo desean hacer especial mención de la participación del Ing. Elio A. A. De María, durante toda su pertenencia al grupo de investigación. También desean hacer constar y valorar la participación permanente del Lic. Carlos E. Maidana y del Ing. Carlos A. Rodríguez. No puede dejar de mencionarse además la colaboración,

desde su incorporación a la Universidad, del Ing. Alejandro G. Fourcade. Todos los mencionados pusieron a disposición del grupo de investigación sus mejores esfuerzos para la concreción exitosa del proyecto al que se refiere el presente trabajo.

### **Referencias**

- [1] Ing. Fernando I. Szklanny, Ing. Elio A. A. De María, Lic. Carlos E. Maidana, Ing. Carlos A. Rodríguez, Ing. Roberto Di Lorenzo, Ing. Hugo R. Tantignone, Sr. Edgardo Gho. "Utilización de dispositivos y sistemas de lógica programable en Sistemas de control numérico para aplicaciones industriales.", Congreso de Microelectronica Aplicada 2010 Libro de memorias, Grupo editor Tercer Milenio S.A , ISBN 978-987-9374-65-8
- [2] Suk Hwan Suh et al, "Theory and Design of CNC Systems", Springer, 2008, ISBN 978-1848003354
- [3] Peter Smid, "CNC Control Setup for Milling and Turning: Mastering CNC Control Systems", Industrial Press, Enero 2010, ISBN 978-0831133504
- [4] Ken Evans, "Programming of CNC Machines", Industrial Press, Abril 2007, ISBN 978-0831133160
- [5] Donald Norris, "Programming with STM32: Getting started with the Nucleo board and C/C++", McGraw Hill, 2018, ISBN 978-1260031317
- [6] Simon Monk, "Raspberry Pi Cookbook", O'Reilly 2014, ISBN 978-1-449-36522-6
- [7] Lauri Kasanen et al, "Into the Core. A look at Tiny Core Linux", Creative Commons, 2013, ISBN 978-952-93-3391-2
- [8] N.E.Worldman, "Machinability and Machining of Metals" McGraw Hills, 1951 , ASIN B000B7CIW0