TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

IMPLEMENTACIÓN DE UN CONTROLADOR EMBEBIDO PARA CONTROLAR UNA MESA XY

Tesis para optar el Título de Ingeniero Electrónico, que presenta el bachiller:

Diego Bustamante Varillas

ASESOR: Eddie Ángel Sobrado Malpartida

Lima, Julio del 2013

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

RESUMEN

La presente tesis tiene como objetivo principal desarrollar un controlador embebido, basado en un microcontrolador ATmega128, que permita controlar la posición de una mesa XY.

Los objetivos específicos son:

- Diseñar la tarjeta de desarrollo del ATmega128 donde se va a implementar el algoritmo de control y el perfil de velocidad.
- Analizar la respuesta en lazo abierto del sistema de control.
- Diseñar e implementar en un microcontrolador ATmega128 el algoritmo de control PID y el perfil de velocidad trapezoidal para controlar la posición de una mesa XY, usando el lenguaje de programación C.
- Diseñar una interfaz en una PC, usando el software Labview, que le permita al usuario ingresar la posición a la cual quiere que se dirija el portaobjetos montado sobre el eje Y de la mesa XY.

Se logró cumplir con todos los objetivos propuestos y luego de realizar las pruebas respectivas se llegaron a las siguientes conclusiones, que se explican más detalladamente en el documento.

- En el ATmega128 se puede implementar desde un controlador de movimiento simple hasta uno complejo porque se pueden manejar hasta 6 ejes de movimiento.
- El factor de conversión (mm/revolución), hallado experimentalmente, es diferente para cada eje porque uno soporta mayor carga que el otro.
- Para implementar un algoritmo de control PID junto con un perfil de velocidad trapezoidal en un ATmega128 es más recomendable programarlo en lenguaje C que en lenguaje ensamblador.
- No se alcanza error cero en estado estable porque la resolución de la señal PWM no es la óptima, pero la precisión final es aceptable.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

ÍNDICE

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

INTRODUCCIÓN

Problemática

Debido a los elevados estándares de calidad que deben tener los productos que son elaborados por los sistemas de producción actuales, los elementos que los componen tienen que disponer de una mayor precisión, versatilidad y fiabilidad.

La capacidad de un trabajador hábil para realizar operaciones precisas en un plano bidimensional o tridimensional sólo es superada por la tecnología ofrecida por los fabricantes de sistemas de movimiento lineal. Por esta razón, los procesos de manufactura manual son procesos ineficientes que dan como resultado un nivel de producción bajo o moderado dependiendo del número y tamaño de los componentes con los que se trabaja.

Entonces, se busca un sistema de posicionamiento basado en un controlador que permita lograr la mayor precisión posible en el movimiento. Por eso, en este trabajo de tesis se plantea realizar la implementación de un controlador PID embebido para poder controlar la posición de una mesa XY, la cual tiene sobre el eje Y un portaobjetos y es usada en la estación de ensamble y control de calidad del Centro de Tecnologías Avanzadas de Manufactura (CETAM).

Justificación

La razón principal por la cual se implementa un algoritmo de control PID y no se aplican técnicas de control avanzado como el control adaptativo, lógica difusa o espacio estado multivariable es porque la mesa XY que se desea controlar presenta una dinámica simple de primer orden, cuyo modelo matemático se presentará más adelante. Entonces, por más que no sea un control óptimo será suficiente para cumplir con los requerimientos de diseño.

Además, este controlador ofrece la ventaja de ser más intuitivo al momento de hallar sus parámetros de sintonización, los cuales se resumen a tres ganancias: proporcional, integrativa y derivativa. Es decir, es más fácil deducir el comportamiento esperado del proceso cuando se cambia alguno de los tres parámetros, que analizar el comportamiento de los polos y ceros.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

Por ejemplo, en la industria química la mayoría de controles de lazo son no lineales, pero presentan una región lineal cerca del punto de operación para la cual un control PID se desempeña bien para la aplicación [8].

En muchos casos se dice que para un sistema de primer orden es suficiente un control proporcional integrativo, pero para esta aplicación donde se requiere que cada uno de los dos ejes de la mesa XY cumpla las siguientes especificaciones: sobreimpulso menor al 5%, tiempo de establecimiento menor a 200ms y error en estado estable igual a cero, también es necesaria la ganancia derivativa porque permite reducir el sobreimpulso y el tiempo de establecimiento. Es importante mencionar que la parte integrativa es la encargada de lograr que el error en estado estable llegue a cero.

De esta manera se puede mejorar el desempeño en la manufactura, logrando un movimiento preciso de los componentes, alta capacidad de producción, flexibilidad para sesiones cortas y manufactura integrada. Es lo que se exige hoy en día debido a la gran competencia que existe en la industria y donde un detalle puede marcar la diferencia.

Estado del arte

Actualmente se puede encontrar en el mercado una gran variedad de controladores de movimiento debido a la gran demanda generada por la industria. Los fabricantes dedicados a este rubro son: Moog, Newmark, Phytron, Siemens, Robox, Advantech, Motrona, Contec, Omrom, sólo por nombrar algunos.

Por ejemplo, Newmark Systems ofrece el controlador NSC-1S que está exclusivamente diseñado para trabajar con sus sistemas de un solo eje gobernados por motores a pasos.

Este mismo fabricante, también vende sistemas que controlan varios ejes simultáneamente como el controlador de la serie NSC-G que puede manejar hasta 4 ejes de manera coordinada o independiente, según los requerimientos del usuario.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

Figura 1: Controlador NSC-1S (izquierda) y controlador NSC-G (derecha) [11].

Los motores más usados para control de movimiento son los motores a pasos y los servomotores sin escobillas. Dependiendo del fabricante, estos motores pueden ser compatibles con diferentes controladores de movimiento. Este es el caso del motor a pasos y servomotor de la marca Newmark Systems que son compatibles con los diversos controladores de la serie NSC-G, NSC-1S y NSC-1M.

Figura 2: Motor a pasos (izquierda) y servomotor con escobillas (derecha) de la marca Newmark Systems [11].

Los algoritmos de control que se implementan en los controladores de posición modernos son generalmente PID, lógica difusa y control adaptativo, aunque también se ha implementado una combinación de estos dos últimos logrando un control robusto y muy preciso.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Objetivos

Objetivo general

- Desarrollar un controlador embebido, basado en un microcontrolador ATmega 128, que permita controlar la posición de una mesa XY.

Objetivos específicos

- Diseñar una tarjeta de desarrollo, basada en un microcontrolador ATmega 128, donde se va a implementar el algoritmo de control y el perfil de velocidad.
- Analizar la respuesta en lazo abierto del sistema de control.
- Diseñar e implementar en un microcontrolador ATmega 128 el algoritmo de control PID y el perfil de velocidad trapezoidal para controlar la posición de una mesa XY, usando el lenguaje de programación C.
- Diseñar una interfaz en una PC, usando el software Labview, que le permita al usuario ingresar la posición a la cual quiere que se dirija el portaobjetos montado sobre el eje Y de la mesa XY.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Capítulo 1: MARCO TEÓRICO

1.1 Sistema de Control

Un sistema de control está compuesto por un conjunto de dispositivos de distinta naturaleza, los cuales pueden ser: mecánicos, eléctricos, electrónicos, hidráulicos o neumáticos. En todo sistema de control se puede considerar una señal de entrada que actúa sobre el mismo y una señal de salida proporcionada por el sistema, según el esquema de la Figura 1.1.1.

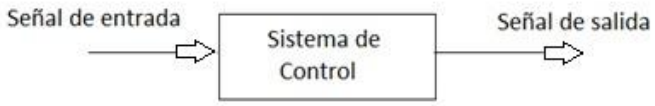


Figura 1.1.1: Esquema de control [12]

El objetivo de un sistema de control es controlar la salida en alguna forma prescrita o llevarla a un valor deseado de acuerdo a la entrada a través de los elementos del sistema de control [12]. Existen dos configuraciones para un sistema de control: lazo abierto y lazo cerrado.

1.1.1 Sistema de Control de Lazo Abierto

Los elementos de un sistema de control en lazo abierto se pueden dividir en dos: el controlador y el sistema o proceso controlado, como se muestra en la Figura 1.1.2.

Una señal de entrada, x(t), se aplica al controlador y como resultado se obtiene una señal actuante, m(t), la cual se encarga de controlar el sistema o proceso de tal forma que la variable de salida o variable controlada, y(t), se desempeñe de acuerdo a los requerimientos preestablecidos como lo son el porcentaje de sobreimpulso, tiempo de establecimiento, tiempo de subida y error en estado estable.

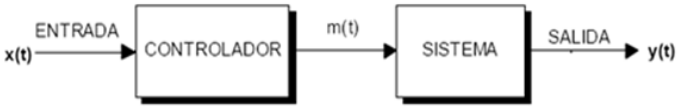


Figura 1.1.2: Sistema de control en lazo abierto [12]

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

1.1.2 Sistema de Control de Lazo Cerrado

En la configuración mostrada en la Figura 1.1.3, el controlador recibe la señal de error de actuación e(t), que es la diferencia entre la entrada x(t) y la señal de realimentación, con el objetivo de reducir el error y llevar la salida del sistema a un valor deseado. El término control en lazo cerrado siempre implica el uso de una acción de control realimentado para reducir el error del sistema [15].

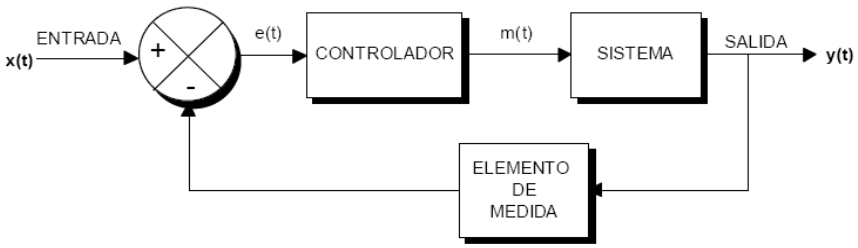


Figura 1.1.3: Sistema de control en lazo cerrado [12]

A continuación, en la Tabla 1.1 y Tabla 1.2 se procede a explicar las ventajas y desventajas de la estrategia de control a lazo abierto y a lazo cerrado respectivamente.

Tabla 1.1: Ventajas y desventajas de un sistema de lazo abierto

LAZO ABIERTO	
VENTAJAS	**DESVENTAJAS**
Simple	No permite corregir disturbios o variaciones en la planta
Bajo costo	

Tabla 1.2: Ventajas y desventajas de un sistema de lazo cerrado

LAZO CERRADO	
VENTAJAS	**DESVENTAJAS**
Mejor respuesta en estado estable y transitorio	Complejo
Menos sensible al ruido, disturbios o variaciones en la planta	Costoso
Más exactitud en comparación con el lazo abierto	

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

1.2 Sistema de Control de Movimiento

Un sistema de control de movimiento está compuesto por los siguientes elementos:

Software de aplicación

Permite diseñar la interfaz con el usuario para que este pueda ingresar la posición final y en algunos casos, elegir el perfil de velocidad. Debe ser diseñada lo más amigable posible porque es el único medio por el cual la persona se va a comunicar con el sistema de control. Además, los fabricantes de estos sistemas entregan sus controladores con su respectivo software de aplicación que permite configurar, generar prototipos, visualizar la respuesta del sistema en tiempo real, etc.

Controlador de movimiento

El controlador de movimiento cumple la función de ser el cerebro del sistema. Es el encargado de cerrar el lazo de control ya que esta tarea requiere un alto nivel de determinismo y es vital para una operación consistente. Además de las tareas mencionadas, este dispositivo también ejecuta un control de supervisión al monitorear los límites y las paradas de emergencia. Por ejemplo, cuando se produce una sobre corriente en los servomotores se detiene el sistema para garantizar una operación segura [7].

Amplificador

El amplificador, también llamado driver, recibe la señal del controlador y genera la corriente necesaria para hacer girar el o los motores dependiendo de la cantidad de ejes que maneje el sistema [7].

Su diseño tiene que ser adecuado porque si proporciona muy poca corriente, el motor no alcanzará la capacidad de torque completa. Además, si el voltaje que proporciona es demasiado bajo, el motor no podrá girar a su máxima velocidad cuando es requerido.

Motor

El motor convierte la energía eléctrica en energía mecánica y produce el torque requerido para mover la estructura acoplada mecánicamente a él hacia la posición deseada.

Tesis publicada con autorización del autor
No olvide citar esta tesis

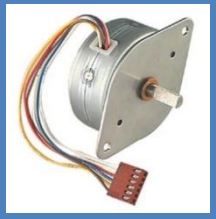
TESIS PUCP

Los motores más utilizados en los sistemas de control de posicionamiento son:

- Motores DC
- Servomotores
- Motores paso a paso

En la Tabla 1.3 se presentan las ventajas y desventajas de cada uno de los motores mencionados.

Tabla 1.3: Cuadro comparativo de motores

	Ventajas	Desventajas
Motor DC  Fuente: sgmada.com	▪ Elevado par de retención sin corriente. ▪ Es de fácil manejo y económico.	▪ No puede ser enclavado en una posición específica sin encoder. ▪ Gira a la máxima velocidad si es que no se usa la modulación por ancho de pulso.
Servomotor  Fuente: directindustry.es	▪ Excelentes características de aceleración y desaceleración. ▪ Torque constante desde su velocidad de reposo hasta su velocidad nominal.	▪ Requiere mantenimiento. ▪ Peligroso en ambientes explosivos. ▪ Es caro.
Motor a pasos  Fuente:micropik.com	▪ Alta resolución de posición. ▪ Buenas propiedades de sincronismo. ▪ Es económico.	▪ Solo sirve para control en lazo abierto. ▪ Bajo torque a alta velocidad. ▪ No es apto para cargas variables.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Dispositivo de retroalimentación o sensor de posición

El dispositivo de retroalimentación, generalmente un "encoder", detecta la posición del motor y envía la información al controlador. Es decir, con estas señales provenientes del encoder, el controlador realiza el lazo de corriente, velocidad y posición.

Existen dos tipos de "encoder" y sus diferencias se presentan en la Tabla 1.4.

Tabla 1.4: Tipos de encoder

TIPOS DE ENCODER	
INCREMENTAL	**ABSOLUTO**
No guarda la posición cuando se le quita la alimentación	Guarda la posición a pesar de desenergizarlo

Es importante mencionar que en algunas aplicaciones de control de movimiento, por ejemplo cuando se usan motores de pasos, no se requiere ningún dispositivo de realimentación para controlar la posición. Pero, en el caso de los servomotores sí es vital y es por esa razón que en la mayoría de casos lo tienen incorporado.

1.3 Algoritmo de Control PID

Los controladores PID son los más usados en aplicaciones que involucran lazos de control a nivel industrial. Más del 90% de los controladores empleados usan el algoritmo PID debido a su simplicidad, funcionalidad y gran aplicabilidad [3].

No solo proporciona flexibilidad en el algoritmo de control, sino también en el tratamiento de la señal de referencia. Más adelante se muestran las diferentes variantes y arquitecturas que se pueden encontrar en el mercado actual.

En la Figura 1.3.1 se muestra un controlador analógico. Este tipo de controladores no se ha visto desplazado por los modernos algoritmos de control frutos del desarrollo de las áreas de electrónica e informática. El motivo radica en las dos grandes ventajas que presenta: robustez y las intuitivas relaciones entre sus parámetros y la respuesta del sistema. Esto no significa que haya quedado al margen de los avances en las áreas tecnológicas de control, sino que debido a su flexibilidad se ha podido beneficiar de ello.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

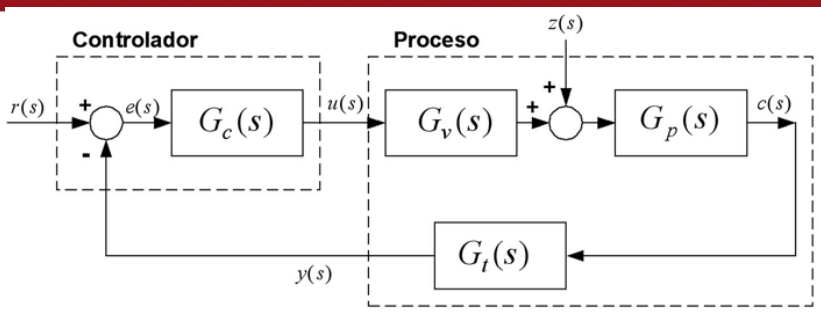


Figura 1.3.1: Diagrama de bloques de un sistema de control analógico [2]

Las siglas PID hacen referencia a los términos proporcional, integral y derivativo, que son las acciones de control aplicadas usualmente sobre el error como en la Fórmula 1.3.1.

$$u(t) = K_p\left(e(t) + \frac{1}{T_i}\int e(t)dt + T_D\frac{de(t)}{dt}\right) \quad (1.3.1)$$

Donde:

Kp: ganancia proporcional
Ti: tiempo integral
Td: tiempo derivativo.

A continuación se explica el efecto que produce cada una de las acciones de control por separado.

Acción de control proporcional

Esta acción genera a la salida una señal de control que es proporcional a la señal de error. De este modo, tenemos:

$$u(t) = K*e(t) \quad (1.3.2)$$

Donde:
u(t): Señal de control
K: Sensibilidad proporcional o ganancia proporcional
e(t): Señal de error

En el dominio de Laplace se representa según la Fórmula 1.3.3.

$$U(s) = K* E(s) \quad (1.3.3)$$

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

Cuanto mayor es la ganancia del control proporcional, mayor es la señal de control generada para un mismo valor de señal de error. Por lo que un aumento de la ganancia del control proporcional permite reducir el error en estado estacionario hasta cierto límite como se puede observar en la Figura 1.3.2.

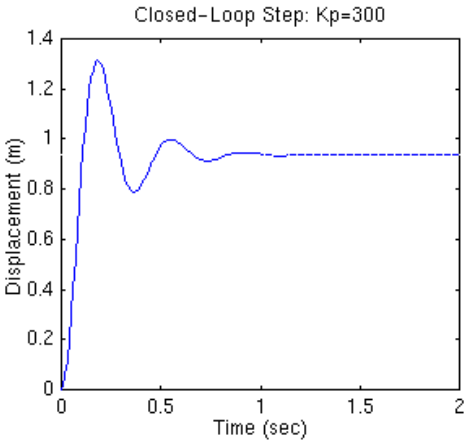


Figura 1.3.2: Respuesta en el tiempo de un sistema con controlador proporcional frente a un escalón unitario de entrada

Se caracteriza por el aumento del sobreimpulso y disminución del tiempo de subida.

Accion de control integral

Genera una señal de control proporcional a la integral de la señal de error:

$$u(t) = \frac{K_p}{T_i} \int_o^t e\, dt \tag{1.3.4}$$

La acción correctiva se efectúa mediante la integral del error. Esto permite obtener error estacionario nulo en un sistema de control, como se presenta en la Figura 1.3.3, pero la disminución exagerada de la constante de tiempo integral, Ti, puede provocar un efecto desestabilizador alargando el estado transitorio y dando como resultado un mayor sobreimpulso.

Tesis publicada con autorización del autor
No olvide citar esta tesis

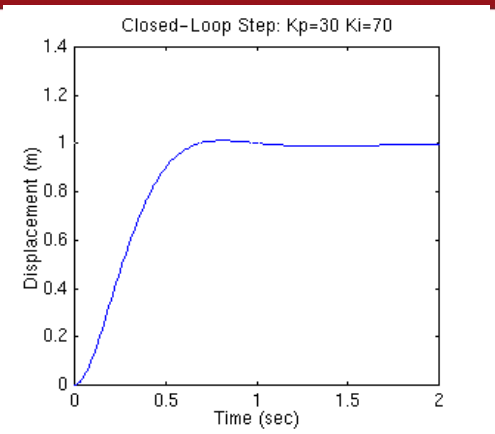


Figura 1.3.3: Respuesta en el tiempo de un sistema con controlador proporcional integral frente a un escalón unitario de entrada

Acción de control derivativa

La acción de control derivativa añade sensibilidad al sistema y tiene un efecto de aumento de estabilidad relativa como se muestra en la Figura 1.3.4.

$$u(t) = K_c T_d \frac{de}{dt} \tag{1.3.5}$$

Sin embargo, el control derivativo no puede utilizarse en solitario ya que es incapaz de responder a una señal de error constante y se comprueba en la Fórmula 1.3.6.

$$e(t) = cte \rightarrow u(t) = 0 \tag{1.3.6}$$

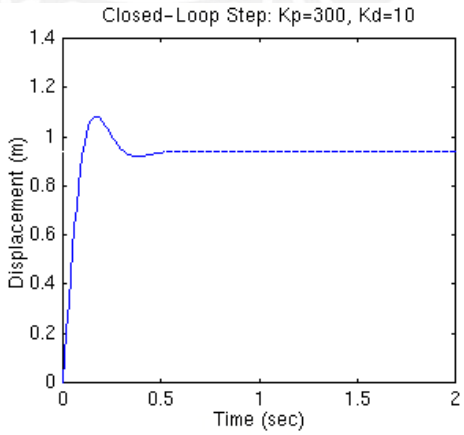


Figura 1.3.4: Respuesta en el tiempo de un sistema con controlador proporcional derivativo frente a un escalón unitario de entrada

Tesis publicada con autorización del autor
No olvide citar esta tesis

En conclusión, un sistema con un control derivativo puro no alcanzará nunca el estado estacionario. Por lo tanto, se recomienda usarlo junto con otros controles por su influencia estabilizadora mediante la acción anticipativa.

Filtro en la acción derivativa

El modo derivativo tiene la desventaja de que amplifica las señales de alta frecuencia, lo que da como resultado que el controlador sea sumamente sensible y actúe de manera errónea. Como solución, se introduce un filtro pasa-bajos en el modo derivativo según la Fórmula 1.3.7.

$$\frac{U_d(s)}{E\ (s)} = \frac{K_p T_D s}{\alpha\ T_D s + 1} \tag{1.3.7}$$

En los controladores PID comerciales, el valor de α óptimo debe ser elegido para estar entre el rango de [0.05, 0.2] [2]. Esto se entiende como un filtro de primer orden, cuya constante de tiempo es α*Td. La nueva acción derivativa actuará como derivada solo a baja frecuencia, mientras que a alta frecuencia su ganancia tendrá como máximo un valor de Kp/α.

De esa forma el ruido a altas frecuencias será amplificado por el valor mencionado y no por uno elevado como se daría en el caso ideal. Esta acción también es llamada red de adelanto.

1.3.1 Modificaciones del Algoritmo PID

En la actualidad, el mercado ofrece distintas versiones del algoritmo de control PID donde cualquiera de ellas se puede asociar a uno de los tres grupos de controladores PID: No interactivo o ideal, interactivo o serie y paralelo.

Algoritmo No interactivo

Es el algoritmo de control por excelencia, de carácter general y reconocido por la ISA (Instrumentation Society of America). Con esta denominación se quiere hacer énfasis en que las acciones de control derivativo e integral no dependen de los parámetros de los otros controladores pues cada uno de ellos actúa sobre una misma señal de error, como se puede ver en la Figura

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

1.3.5. Sin embargo, el control proporcional es el último en actuar ya que afecta a la señal de error, a la derivada y a la integral del error [10].

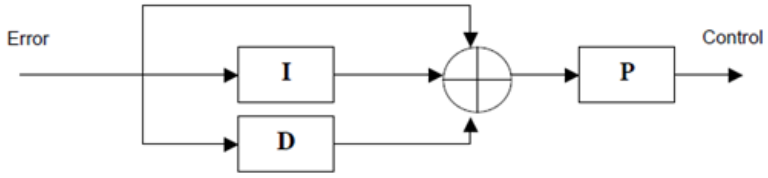


Figura 1.3.5: Diagrama de bloques del algoritmo PID no interactivo [10].

En la Fórmula 1.3.8 se expresa su función de transferencia.

$$G_{cPIDIdeal}(s) = K_c\left(1+\frac{1}{sT_i}+sT_d\right) \tag{1.3.8}$$

Algoritmo interactivo

Se denomina algoritmo interactivo, serie o clásico al algoritmo en el que cualquier modificación de alguno de sus parámetros repercute en las entradas de las demás constantes de tiempo: integral, derivativa o proporcional, según la Figura 1.3.6. Aparentemente esto complica la sintonización manual, pero en realidad no sucede así.

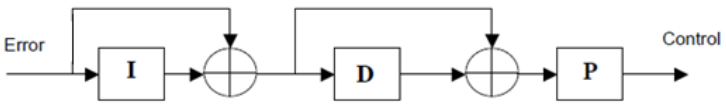


Figura 1.3.6: Diagrama de bloques del algoritmo PID interactivo [10].

La expresión que representa la relación entre su entrada y salida está dada por la Fórmula 1.3.9.

$$G_{cPIDSerie}(s) = K_c'\left(1+\frac{1}{T_i's}\right)\left(1+T_d's\right) \tag{1.3.9}$$

Algoritmo paralelo

Es la forma más general de realizar el algoritmo PID. Las tres acciones de control actúan directamente sobre la misma señal de error como se presenta

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

en la Figura 1.3.7. Por un lado, es la fórmula más flexible, pero sus parámetros obtenidos no tienen una interpretación física muy directa [10]. En este caso, cada acción de control puede ser modificada por separado sin que ello pueda influir en las demás acciones de control.

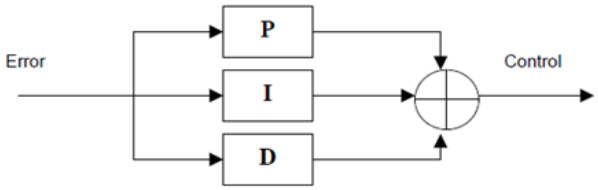


Figura 1.3.7: Diagrama de bloques del algoritmo PID paralelo [10]

Este algoritmo se representa en el dominio de Laplace con la Fórmula 1.3.10.

$$G_{cPIDParalelo}(s) = \left(K_p + \frac{K_i}{s} + K_d s \right) \tag{1.3.10}$$

1.3.2 Arquitecturas de Control PID

Tomando como referencia la estructura PID no interactiva, se pueden efectuar variaciones para intentar mejorar su funcionamiento. Cuando aparecen perturbaciones a la entrada de una planta, el controlador actuará en función de una señal no actual. Es decir, estaría trabajando con cierto desfase y para evitar este efecto se recurre a estructuras alternativas como las siguientes:

Estructura PI-D:

La acción derivativa actúa únicamente sobre la salida del proceso, Y(s), y no sobre la señal de error, E(s), como se ve en la Figura 1.3.8. De esa forma, se impide que aparezcan señales de control, u(s), muy bruscas o excesivamente elevadas cuando la entrada varía bruscamente [10].

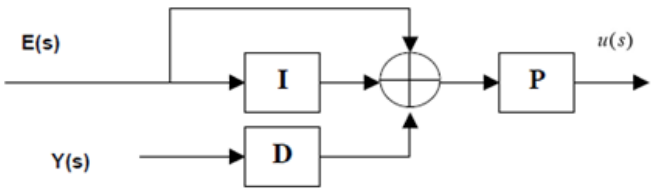


Figura 1.3.8: Estructura PI-D [10]

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Estructura I-PD:

En esta estructura, mostrada en la Figura 1.3.9, tanto la acción derivativa como la acción proporcional actúan sobre la salida, Y(s), y no sobre el error. Además, atenúa las perturbaciones y mejora el estado transitorio porque la señal de control, u(s), es mucho más suave [10].

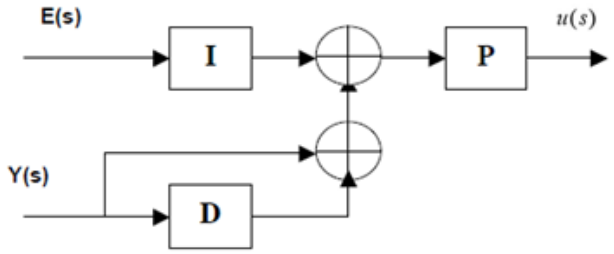


Figura 1.3.9: Estructura I-PD [10]

1.4 Generación de Trayectoria

1.4.1 Tipos de Movimiento

Movimiento punto a punto

El movimiento de un solo eje y punto a punto es uno de los más utilizados con más frecuencia debido a su sencillez. Este requiere la posición a la cual se debe mover el eje y en muchos casos también requiere la velocidad y la aceleración, que por lo general son valores predeterminados por el usuario.

Este movimiento se puede extender a dos o tres ejes siguiendo el mismo principio, es decir sólo moviendo un eje a la vez [6].

Movimiento coordinado de múltiples ejes

En este tipo de trayectoria el movimiento de cada eje empieza y finaliza al mismo tiempo. Se debe calcular cuál es el eje que invertirá más tiempo en su movimiento. Entonces, se ralentiza el movimiento de los otros ejes para que inviertan el mismo tiempo, acabando todos ellos simultáneamente. Normalmente el tiempo invertido es el menor posible y no se exigen aceleraciones y velocidades elevadas a los actuadores [7].

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Movimiento combinado

Este movimiento implica dos movimientos fusionados por una combinación que causa que los movimientos actúen como uno. La combinación es útil para las aplicaciones que requieren continuidad entre dos movimientos diferentes. Sin embargo, en este caso particular el sistema no pasa por todos los puntos de su trayectoria original. Si lo más importante es la posición específica a lo largo del camino, se debe considerar el movimiento de contorno. En la Figura 1.4.1 se representan gráficamente los tres movimientos mencionados anteriormente.

Figura 1.4.1: Trayectorias de movimiento eje a eje, coordinado y combinado

Movimiento de contorno

Con un movimiento de contorno se puede proporcionar un historial de posiciones y crear un camino suave o trazado a través de ellas. Este movimiento de la Figura 1.4.2 tiene una ventaja sobre el combinado porque garantiza que el sistema pasará por cada posición elegida.

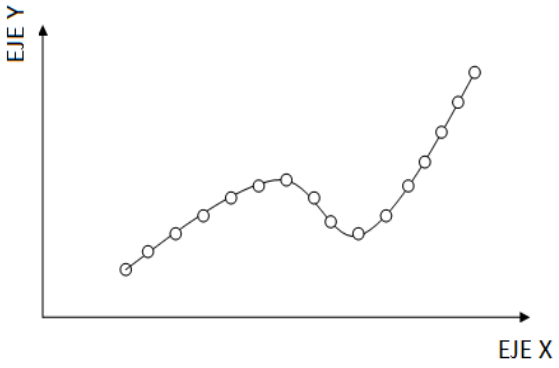


Figura 1.4.2: Movimiento de contorno [6]

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

1.4.2 Perfiles de velocidad

En un sistema de control de movimiento, la manera en que la carga llega a su destino se le conoce como perfil de velocidad.

Existen diferentes perfiles de velocidad y dentro de los más usados se encuentran el perfil de velocidad constante, el perfil de velocidad trapezoidal y el de curva S o sinusoidal. La distancia total recorrida es igual al área encerrada por la curva obtenida, donde el eje horizontal es el tiempo y el eje vertical representa la velocidad.

Velocidad constante

Este perfil mantiene una velocidad constante entre dos puntos y es el más simple porque tiene a la velocidad como único parámetro. Los sistemas de posicionamiento que requieren precisión no usan este perfil porque ninguna máquina real puede cambiar de velocidad instantáneamente. Habrá un tiempo de retardo que dependerá del sistema y de los cambios en la carga.

En la Figura 1.4.3, la línea punteada representa la velocidad real que se alcanzará, donde Ta y Td representan el tiempo necesario para acelerar y desacelerar, pero estos tiempos pueden variar de acuerdo al proceso que se desea controlar. Puede ser usado en fajas transportadoras o ventiladores [16].

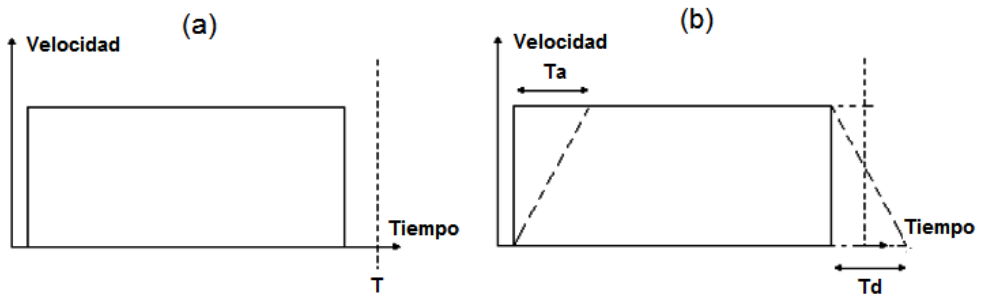


Figura 1.4.3: Perfil de velocidad constante: (a) Trayectoria ideal
(b) Trayectoria real debido a los retardos [16]

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Trapezoidal

En este perfil, el motor parte de una posición de detenimiento y acelera constantemente hasta alcanzar la velocidad establecida. El movimiento continúa a la velocidad deseada por un determinado periodo hasta que el controlador decide que es tiempo de comenzar el segmento de desaceleración, y disminuye la velocidad del movimiento para detenerse exactamente en la posición deseada como se puede observar claramente en la Figura 1.4.4.

Si el movimiento es demasiado corto de manera que el punto de inicio de la desaceleración ocurre antes de que se haya completado el tramo de aceleración, el perfil tomará una forma triangular en vez de trapezoidal y la velocidad alcanzada será menor a la velocidad que se deseaba alcanzar. Por eso, se tiene que cumplir una condición de aceleración mínima que se presenta más adelante.

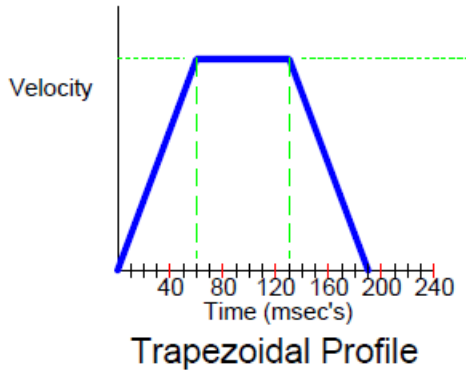


Figura 1.4.4: Perfil de velocidad trapezoidal [14]

Sinusoidal o curva S

El perfil de la Figura 1.4.5 permite un cambio gradual en la aceleración. Esto ayuda a reducir y en algunos casos a eliminar los problemas causados por el sobreimpulso, siendo el resultado una mejora en la vibración mecánica vista desde el sistema.

Los puntos de mínima aceleración se producen al comienzo y al final del periodo de aceleración, mientras que la máxima aceleración se alcanza entre estos dos puntos. Esto permite un movimiento más rápido, preciso y suave.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

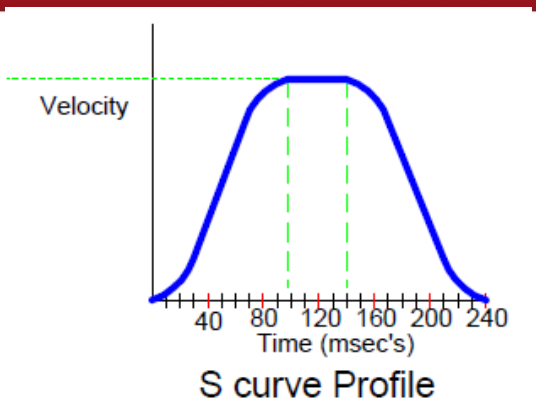


Figura 1.4.5: Perfil de velocidad sinusoidal [14]

Para una misma distancia, este perfil demorará un tiempo mayor en llegar al punto de destino comparándolo con el perfil de velocidad trapezoidal. Pero, la degradación mecánica de los componentes será menor porque su arranque es más suave.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Capítulo 2: DISEÑO

En esta etapa se va a explicar cómo se realizó la sintonización de parámetros de los dos controladores PID que van a implementarse para poder controlar la posición de la mesa XY. Para lograr esto es necesario definir los requerimientos de respuesta en el tiempo que se quieren alcanzar con el diseño a realizar. Estos son los siguientes: sobreimpulso menor al 5%, tiempo de establecimiento menor a 200ms y error en estado estable igual a cero.

Además, se brindan detalles del diseño del hardware del controlador embebido y los pasos a seguir para generar el perfil de velocidad trapezoidal.

2.1 Descripción del Sistema de Control

En la Figura 2.1.1 se presenta el Diagrama de Bloques del Sistema de control de posición implementado.

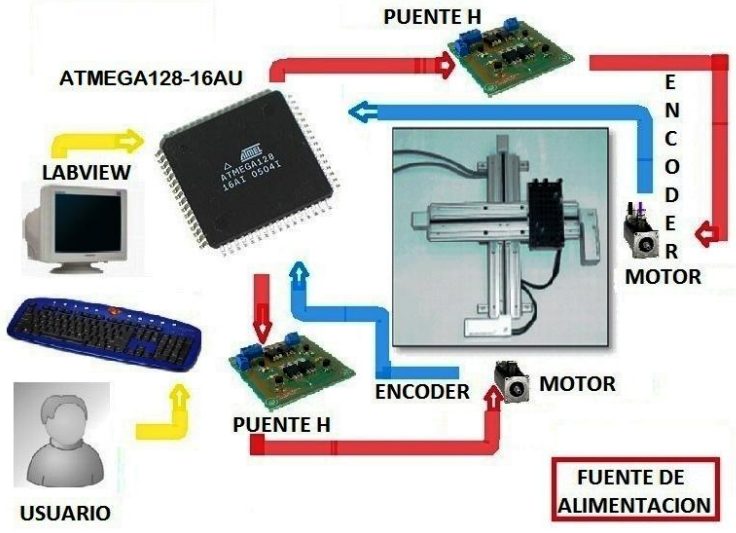


Figura 2.1.1: Diagrama de bloques del sistema de control de posición

El usuario ingresará desde una PC, a través de una interfaz diseñada en el software Labview 8.5 del fabricante National Instruments, las coordenadas de posición X e Y a las cuales quiere que se dirija el portaobjetos de la mesa XY.

El microcontrolador usado para esta aplicación es el ATmega128-16AU. Este recibe las coordenadas enviadas por el usuario desde la PC a través de la interfaz de comunicación serial USART y las toma como el valor de referencia.

Las señales de error se calculan restando la posición de referencia menos la posición actual, la cual es medida por el encoder incremental que posee cada

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

uno de los dos ejes de movimiento. Esta señal de error es la entrada a su respectivo algoritmo de control PID, el cual generará una señal de control que pasará por una etapa previa de amplificación, basada en un puente H, y que permitirá entregar la corriente necesaria a cada motor para poder controlarlo.

2.1.1 Características de la Mesa XY

En la Figura 2.1.2 se puede apreciar a la mesa XY. Este equipo es de la marca estadounidense Intelitek [13] y fue importado en el año 2000 para que formara parte de la Estación de Ensamble y Control de Calidad por Visión del Centro de Tecnologías Avanzadas de Manufactura (CETAM) de la PUCP. Está compuesta por el eje X, eje Y y un portaobjetos montado sobre el eje Y, el cual tiene acoplado un pistón neumático.

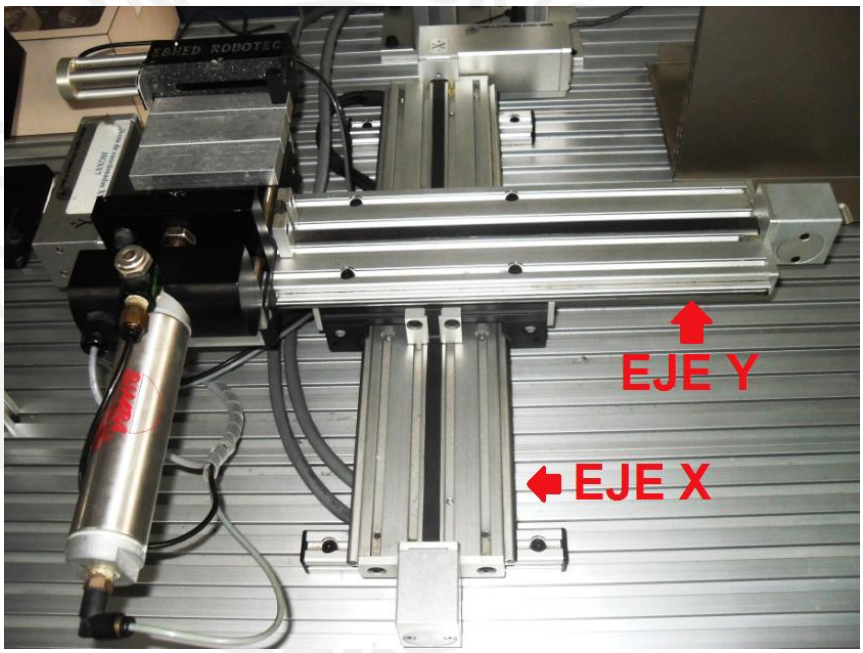


Figura 2.1.2: Fotografía de la mesa XY

Está construida en base a vigas de aluminio extruido con ranuras que permiten un fácil montaje y fijación colocando un tornillo o cualquier otro componente con las dimensiones adecuadas [13]. En la Tabla 2.1 se muestran sus dimensiones:

Tabla 2.1: Dimensiones de la mesa XY

Largo	Ancho	Altura
494mm (19.5")	494mm (19.5")	115mm (4.5")

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

Además, presenta las siguientes características:

- Recorrido de cada eje: 297 mm.
- Capacidad de carga: 5kg.
- 02 Motores DC de 24 Vdc.
- 02 Encoders incrementales de 20 cuentas por revolución.
- Modo de transmisión por correa.
- Agujeros pre taladrados para el montaje sobre la mesa de trabajo.
- Se puede separar en dos ejes individuales para ser usados como dispositivos de posicionamiento lineal.

2.1.2 Características del microcontrolador ATmega128

El ATmega128-16AU (Figura 2.1.3) es un microcontrolador de 8 bits que pertenece a la familia AVR de la compañía Atmel. Una de las ventajas que presentan los dispositivos de la familia AVR es que aparte de poder ser programados en lenguaje ensamblador, son totalmente compatibles con el lenguaje C y por lo tanto, también pueden ser programados a alto nivel.

En el Anexo 1 se mencionan los pasos a seguir para instalar el compilador AVR-GCC y cómo usarlo desde el simulador VMLAB (Anexo 3) o en el AVR Studio (Anexo 2).

Tampoco es necesario extraerlos del circuito impreso o tarjeta de desarrollo donde se encuentran para ser programados, ya que permiten la programación ISP (In System Programming). En el Anexo 4 se explica cómo programar un ATmega128 vía ISP usando el programa AVR Studio 4.

Figura 2.1.3: Microcontrolador ATmega128-16AU (www.futurlec.com)

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

Este microcontrolador presenta una arquitectura RISC, la cual combina una gran cantidad de instrucciones con 32 registros de propósito general que se encuentran directamente conectados a la Unidad Lógica Aritmética (ALU).
De esa manera, se permite el acceso a dos registros por medio de una sola instrucción que es ejecutada durante un ciclo de reloj. La arquitectura que presenta es más eficiente en cuanto a la ejecución de código porque puede alcanzar un rendimiento hasta 10 veces más rápido que un microcontrolador convencional con una arquitectura CISC [4].

En la Figura 2.1.4 se muestra la arquitectura que presentan los microcontroladores de la familia AVR. A nivel de memoria presentan un arquitectura Harvard, esto implica que la memoria de datos y la memoria de programa se encuentran separadas y cada una con su respectivo bus.

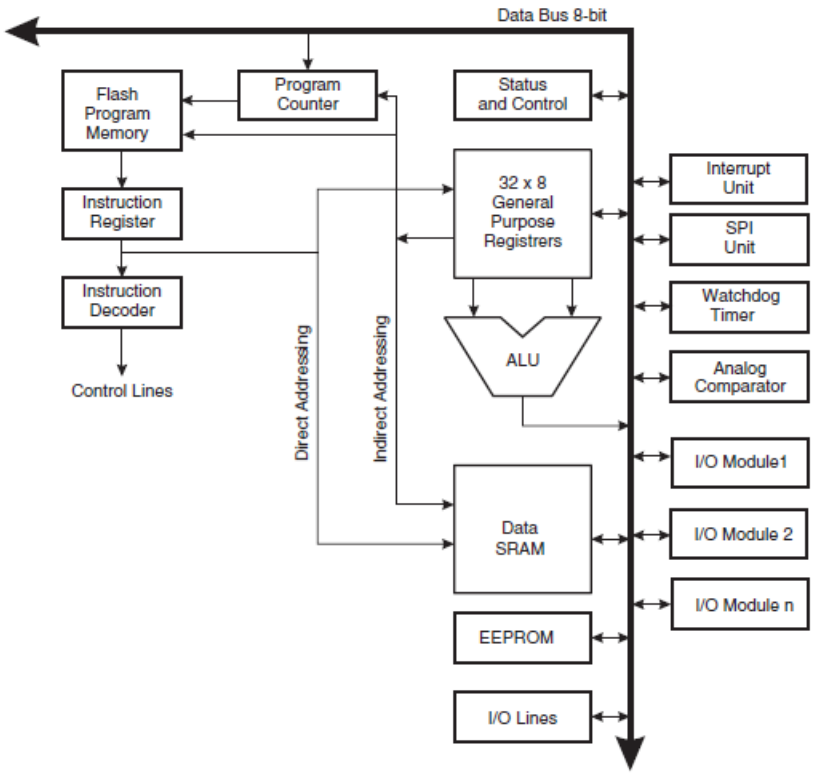


Figura 2.1.4: Diagrama de bloques de la arquitectura de la familia AVR.

A continuación, se mencionan las características principales de este microcontrolador de alto rendimiento y bajo consumo de potencia, las cuales prevalecieron para su elección.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

- 128Kb de memoria de programa (Flash)
- 4Kb de memoria de datos (SRAM)
- 6 canales PWM con resolución de 2 a 16 bits.
- 2 canales PWM con resolución de 8 bits
- 53 puertos de entrada/salida
- 8 interrupciones externas
- 2 interfaces USART

Otras especificaciones importantes son:

- Frecuencia de reloj: 0 - 16Mhz
- Interfaz para memoria de datos externa de hasta 64Kb.
- 35 vectores de interrupción
- 8 canales ADC de 10 bits de resolución
- Interfaz SPI (full duplex)

2.2 Diseño del Hardware del Controlador

En esta etapa se mencionan las características principales de la tarjeta de desarrollo que se diseñó en el programa Eagle Layout Editor 5.6.0 y que luego se mandó a fabricar para poder implementar el controlador embebido.

El hardware del controlador de la mesa XY es alimentado por un adaptador AC-DC regulable modelo AM-C10 de la marca Miray (Figura 2.2.1) con las siguientes características:

- Voltaje de entrada: 220VAC / 60 Hz
- Voltajes de salida: 3 – 4.5 – 6 – 7.5 – 9 - 12 Voltios DC
- Corriente: 500 mA
- Potencia máxima: 6 VA

Figura 2.2.1: Adaptador AC-DC usado para alimentar la tarjeta de control

Tesis publicada con autorización del autor
No olvide citar esta tesis

La tarjeta tiene incorporada una fuente regulada de 5 voltios (Figura 2.2.2). La razón principal por la cual se usa esta fuente es por seguridad, ya que de este modo el ATmega128 no se dañará por un sobrevoltaje porque el regulador se encargará de entregar siempre 5 voltios, con la condición de que el adaptador le entregue un voltaje mayor a 7 Vdc. Además, esta tarjeta posee un led indicador de color verde que se enciende cuando la tarjeta es alimentada correctamente.

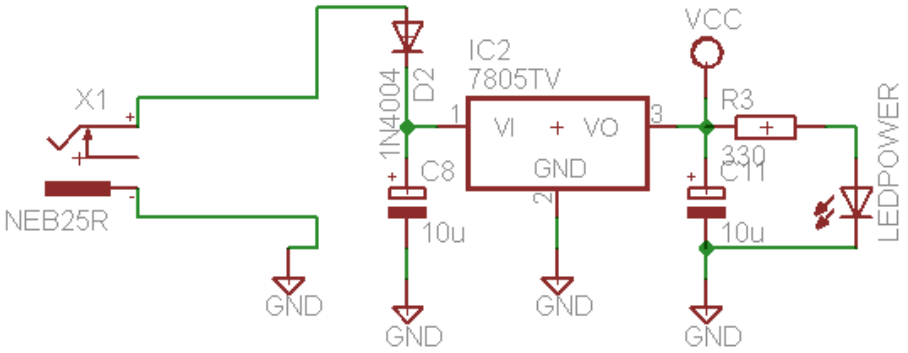


Figura 2.2.2: Diagrama esquemático de la fuente regulada (LM7805)

Además, el kit de desarrollo tiene incorporado un circuito que permite establecer la comunicación serial bajo el estándar RS-232. Este circuito está conformado por el circuito integrado MAX232 con sus respectivos componentes, los cuales se encargan de convertir los niveles de voltaje con que trabaja el puerto serial de la computadora a los niveles de voltaje TTL que maneja el ATmega128.

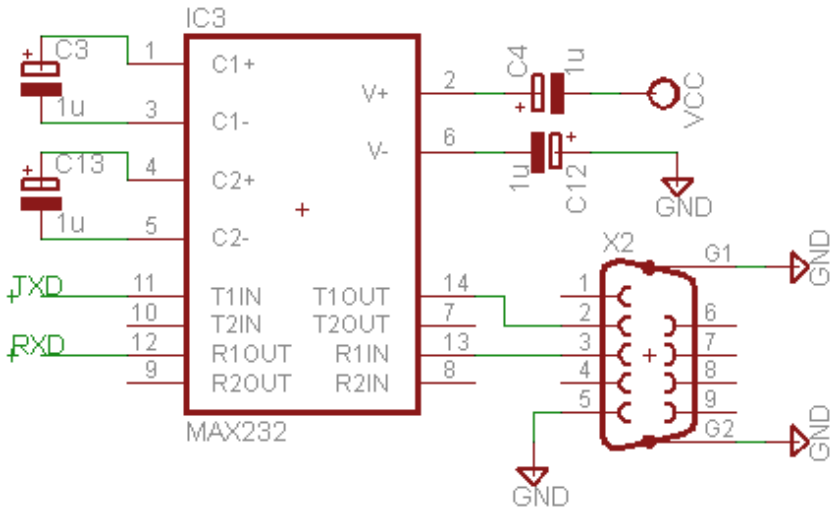


Figura 2.2.3: Diagrama esquemático del circuito de comunicación serial

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

La tarjeta de desarrollo cuenta con un conector de 6 pines que sirve para la descarga del programa en el microcontrolador. Este componente es imprescindible ya que al ser un componente de montaje superficial, está soldado de manera permanente y no puede ser extraído de la tarjeta para ser programado en otro módulo.

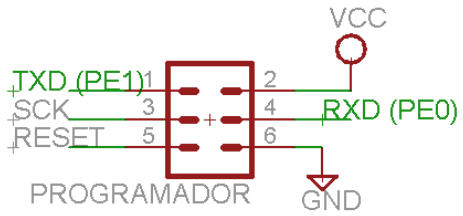


Figura 2.2.4: Diagrama esquemático del programador ISP

El ATmega 128 presenta una particularidad frente a los otros microcontroladores de la serie ATmega con respecto a los pines usados para la programación. En la hoja de datos se menciona que los pines generalmente usados para programar, MISO y MOSI, son reemplazados por el pin PE1 y PE0 respectivamente, mientras que el pin SCK y el RESET se mantienen como en los demás microcontroladores de la familia. Para más detalle ver la Tabla 2.2.

Tabla 2.2: Disposición de los pines del programador ISP [4]

Símbolo	Pines	I/O	Descripción
MOSI (PDI)	PE0	I	Datos de recepción serial
MISO (PDO)	PE1	O	Datos de envío serial
SCK	PB1	I	Reloj serial

2.3 Diseño de la Interfaz de Usuario

El objetivo de la mesa de posicionamiento XY es que el portaobjetos que se encuentra montado sobre el eje Y, se ubique en la posición indicada por el usuario con el mínimo error posible.

Es por eso, que tiene que existir un medio de comunicación entre la persona y el controlador para que este último interprete los datos recibidos y genere las señales de control correspondientes para accionar los motores respectivos.

Labview es un entorno de programación gráfico desarrollado por la compañía estadounidense National Instruments donde se puede crear una interfaz de

Tesis publicada con autorización del autor
No olvide citar esta tesis

manera simple y que permite enviar datos a través del puerto serial de la PC desde la cual se está ejecutando el programa.

Los parámetros establecidos para la recepción y transmisión serial de datos usando el protocolo RS-232 asíncrono son:

- Velocidad de transmisión: 9600 bps
- Bits de datos: 8
- Bits de paridad: Ninguno
- Bits de parada: 1

Los pasos que tiene que seguir la persona que desea manipular la mesa se encuentran en el Anexo 5.

La interfaz cuenta con lo siguiente:

- Los parámetros para establecer la comunicación serial

Figura 2.3.1: Parámetros de comunicación serial

- El set point para el eje X y el set point para el eje Y

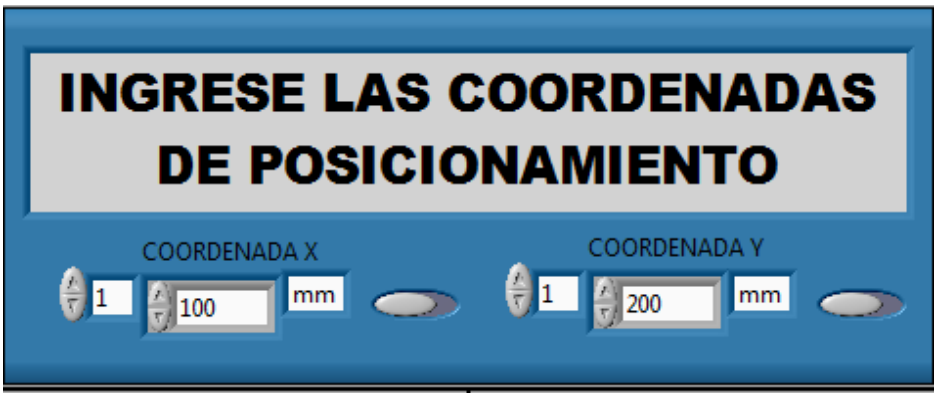


Figura 2.3.2: Coordenadas de posición para ambos ejes

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

- Los parámetros de los dos controladores PID

Figura 2.3.3: Parámetros de los algoritmos PID

- Una gráfica que simula la trayectoria que debe seguir la mesa XY de acuerdo a las posiciones de referencia ingresadas

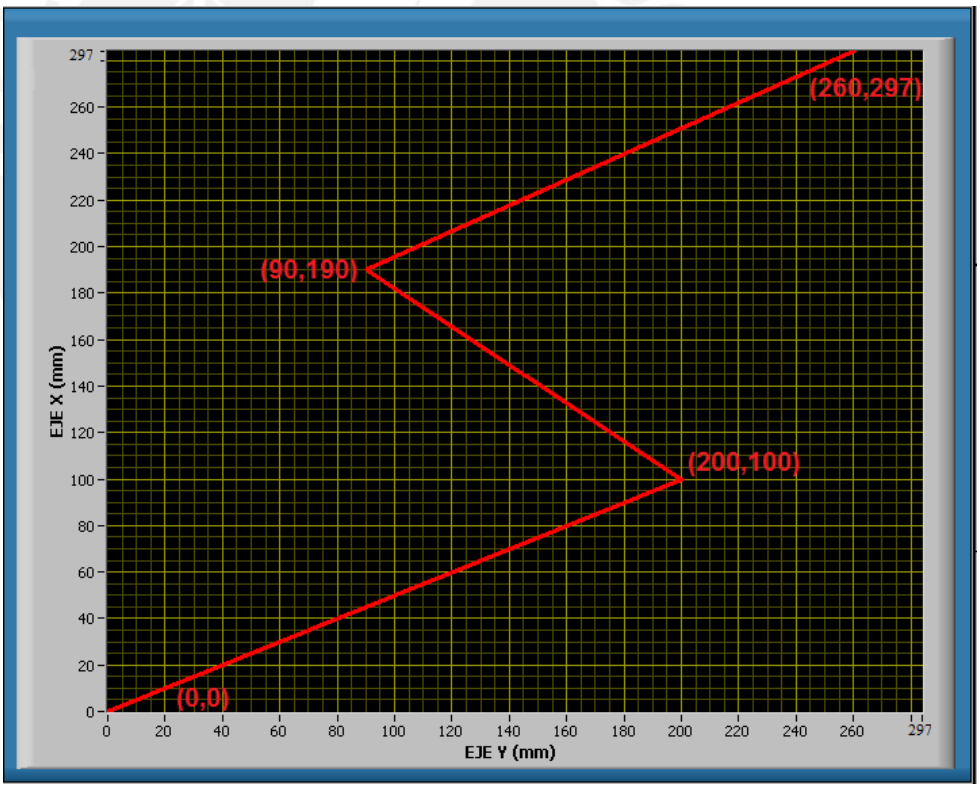


Figura 2.3.4: Gráfica de trayectoria.

Tesis publicada con autorización del autor
No olvide citar esta tesis

A continuación se presenta la interfaz gráfica completa:

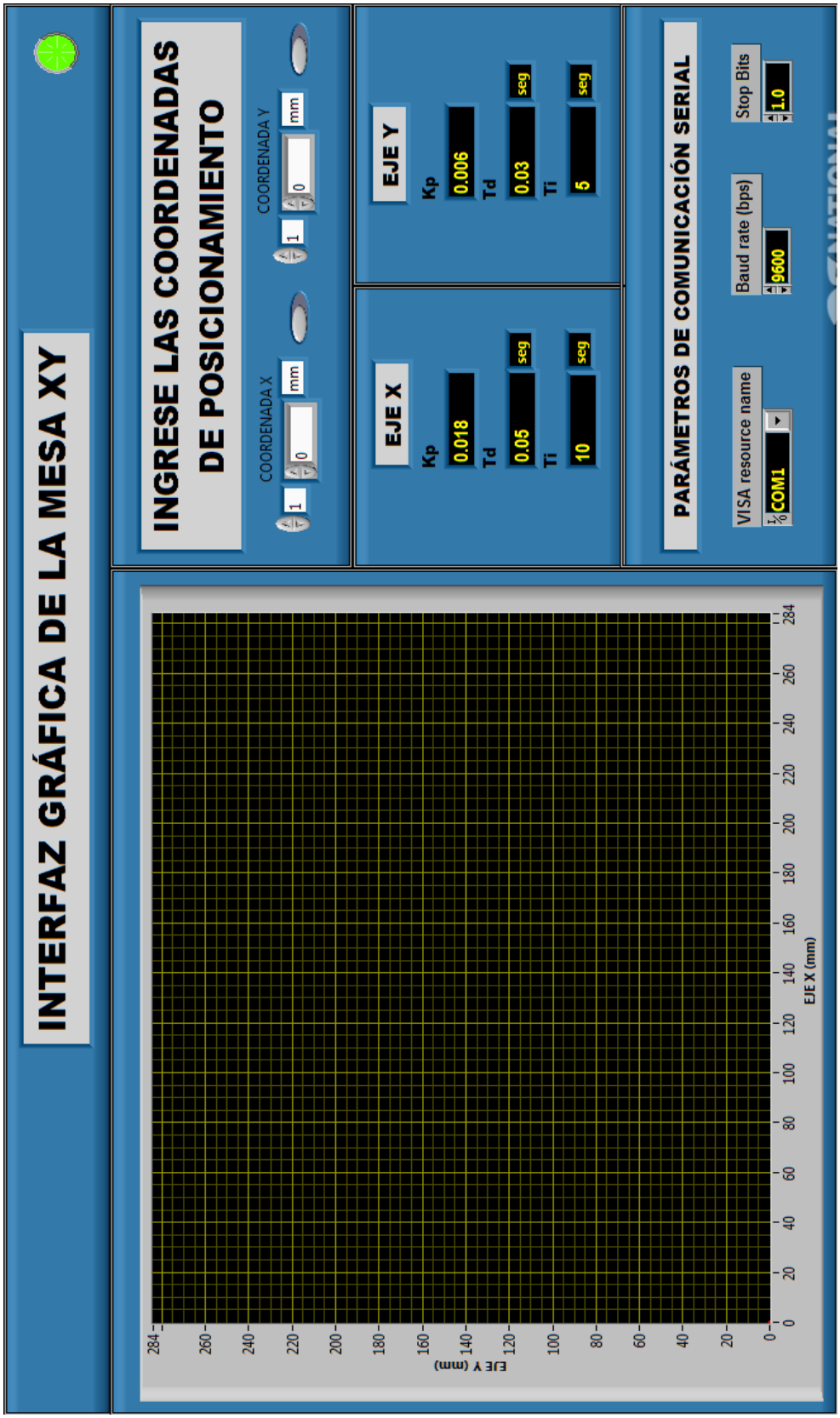


Figura 2.3.5: Interfaz gráfica de la mesa XY

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

2.4 Diseño del Controlador PID

En esta etapa se aplicó el algoritmo PID no interactivo presentado en el Capítulo 1. El primer paso fue obtener los modelos matemáticos de ambas plantas a controlar, tanto del motor del eje X y como del motor del eje Y.

Los modelos matemáticos para ambos motores e interfaces de potencia fueron hallados anteriormente de manera experimental [7]. Los resultados fueron:

Motor del eje X:

$$\frac{\omega(rad/s)}{V_M(V)} = \frac{28.75}{0.04s+1}e^{-0.005s} \tag{2.4.1}$$

Motor del eje Y:

$$\frac{\omega(rad/s)}{V_M(V)} = \frac{32}{0.032s+1}e^{-0.008s} \tag{2.4.2}$$

Amplificador del eje X:

$$\frac{Vm}{DutyCycle} = 52 \tag{2.4.3}$$

Amplificador del eje Y:

$$\frac{Vm}{DutyCycle} = 144.4 \tag{2.4.4}$$

Donde:
w: Velocidad del motor en rad/s.
Vm: Voltaje medio aplicado al motor en voltios.
Duty Cycle: Ciclo de trabajo de la señal generada por el controlador.

Con estos modelos matemáticos se puede hallar la relación entre la velocidad del motor y el duty cycle o ciclo de trabajo de entrada. Pero, como el objetivo es lograr un control de posición se debe agregar un integrador a la función de transferencia para poder simular la respuesta del sistema frente a una entrada determinada.

Entonces, la función de transferencia de la planta queda representada de la siguiente manera:

Tesis publicada con autorización del autor
No olvide citar esta tesis

Eje X:

$$\frac{1495}{0.04s^2+s}e^{-0.001s} \tag{2.4.5}$$

Eje Y:

$$\frac{4620.8}{0.032s^2+s}e^{-0.008s} \tag{2.4.6}$$

Luego de obtener el modelo matemático del proceso, se debe elegir el controlador que permita cumplir con los requerimientos establecidos. En este caso, para el correcto funcionamiento del sistema de posicionamiento XY cada eje de movimiento deberá cumplir las siguientes especificaciones:

Un sobreimpulso menor al 5%, tiempo de establecimiento menor a 200ms y error en estado estable igual a cero [7].

El algoritmo PID no interactivo o forma estándar expresado en la Fórmula 1.3.8 es una solución adecuada para este sistema porque presenta una dinámica de primer orden, exige error nulo en estado estable y un mínimo sobreimpulso. Para implementar un algoritmo de control en un microcontrolador o en un procesador es necesario discretizarlo. Por lo tanto, se usó el método de diferencias en retroceso que consiste en aplicar la siguiente igualdad:

$$s = \frac{z-1}{zT} \tag{2.4.7}$$

Como resultado, se obtuvo el algoritmo PID discreto:

$$PID = k[1+\frac{T}{T_i}\frac{1}{1-z^{-1}}+\frac{T_d}{T}(1-z^{-1})] \tag{2.4.8}$$

El tiempo de muestreo (T), mostrado en la Fórmula 2.4.8, tiene gran influencia sobre la dinámica del sistema porque mientras mayor sea, afectará en mayor magnitud la estabilidad de este. Una forma práctica de hallar el tiempo de muestreo es usando la Fórmula 2.4.9:

Tesis publicada con autorización del autor
No olvide citar esta tesis

$$T_{muestreo} \leq \frac{T_{minimo}}{5} \quad (2.4.9)$$

, donde $T_{mínimo}$ representa la menor constante de tiempo presente en el sistema. Entonces, tomando en cuenta la regla teórica y luego de hacer las pruebas respectivas se determinó el tiempo de muestreo igual a 5ms.

El siguiente paso es la sintonización de los parámetros de ambos algoritmos PID. Esta etapa consiste en hallar los valores adecuados para las constantes Kp, Ti y Td. Existen varios métodos para sintonizar un controlador PID, entre ellos: método de Ziegler-Nichols en lazo abierto, método de Cohen-Conn en lazo abierto, método de Chien-Hrones-Reswick en lazo abierto, método de Ziegler-Nichols en lazo cerrado, método manual, etc.

Los parámetros de sintonización de ambos ejes se hallaron en Simulink aplicando el método manual. En la Tabla 2.3 y la Tabla 2.4 se muestran los valores elegidos para el controlador del Eje X y el del Eje Y, mientras que la respuesta de cada eje de movimiento en el tiempo se puede ver en la Figura 3.2.2 y Figura 3.2.4 respectivamente.

Tabla 2.3: Parámetros de sintonización del controlador PID del eje X

EJE X		
Kp	**Ti (seg)**	**Td (seg)**
0.018	10	0.05

Tabla 2.4: Parámetros de sintonización del controlador PID del eje Y

EJE Y		
Kp	**Ti (seg)**	**Td (seg)**
0.006	5	0.03

Reemplazando las constantes de las Tablas 2.3 y 2.4 en la Fórmula 2.4.8 y luego de calcular la transformada zeta inversa, se obtuvo la ecuación de diferencias del eje X y del eje Y respectivamente.

$$u_x[k] = 0.198*e_x[k] - 0.179982*e_x[k-1] \quad (2.4.10)$$

$$u_y[k] = 0.042*e_y[k] - 0.035994*e_y[k-1] \quad (2.4.11)$$

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

Las Ecuaciones 2.4.10 y 2.4.11 son las que finalmente se van a implementar en el microcontrolador. Para que se tenga una idea más clara, en las siguientes líneas se muestra el código en lenguaje C correspondiente al algoritmo PID del eje X y la única diferencia con el algoritmo del eje Y radica en la ecuación de diferencias.

```
// ALGORITMO DE CONTROL PID (EJE X)
posx_rad=0.314*pulsos_x; // Se hace la conversión de pulsos de encoder a radianes
error_act_x=arr_sp1[contador] - posx_rad; // Se calcula del error actual
error_act_x=fabs(error_act_x); // Valor absoluto del error del eje X
salida_x=(0.198*error_act_x) - (0.179982*error_ant_x); // Se ejecuta la ecuación de diferencias
salida_x=(salida_x)*100; // Se expresa la salida del algoritmo de control en porcentaje
  if(salida_x<min) // Se establece los límites mínimo y máximo de saturación de la señal PWM según las características del motor
      salida_x=min;
  if(salida_x>max)
      salida_x=max;
OCR1A=floor(salida_x/0.405);  // Se redondea el valor de senal PWM a la escala del registro OCR1A
error_ant_x=error_act_x; // El error actual se convierte en el error anterior para la siguiente muestra
 _delay_loop_2(10000); // Se aplica el tiempo de muestreo equivalente a 5ms
```

2.5 Diseño del Perfil de Velocidad

El perfil de velocidad trapezoidal es elegido para esta aplicación porque minimiza el desgaste mecánico de los componentes y el movimiento tarda menos tiempo comparado con el perfil sinusoidal.

En este caso, el controlador de movimiento calcula los puntos de toda la trayectoria del movimiento en base a los parámetros programados. Estos parámetros son: el punto de partida y el punto de llegada, la velocidad máxima que se desea alcanzar y la aceleración definida por el usuario. Con estos

Tesis publicada con autorización del autor
No olvide citar esta tesis

parámetros se puede determinar cuánto tiempo empleará en cada uno de los tres segmentos: aceleración, velocidad constante y desaceleración.

Los pasos a seguir son los siguientes:

- Calcular la trayectoria para el eje que demora más tiempo en llegar a su posición final. Entonces, se calcula el tiempo final (tf) de cada eje, según la Fórmula 2.5.1.
- Calcular el tiempo crítico (tc) del eje de mayor tiempo final, según la Fórmula 2.5.2.

$$t_f = \frac{V_{max}^{\ 2} - a(x_0 - x_f)}{aV_{max}} \tag{2.5.1}$$

$$t_c = \frac{x_0 - x_f + V_{max}t_f}{V_{max}} \tag{2.5.2}$$

Donde:

tf: Tiempo final

tc: Tiempo crítico

Xo: Posición inicial

Xf: Posición final

Vmax: Velocidad máxima

a: Aceleración definida por el usuario

Es importante mencionar que el tiempo crítico es el tiempo que dura el tramo de aceleración y es el mismo periodo que demora el segmento de desaceleración hasta que el motor llega a detenerse.

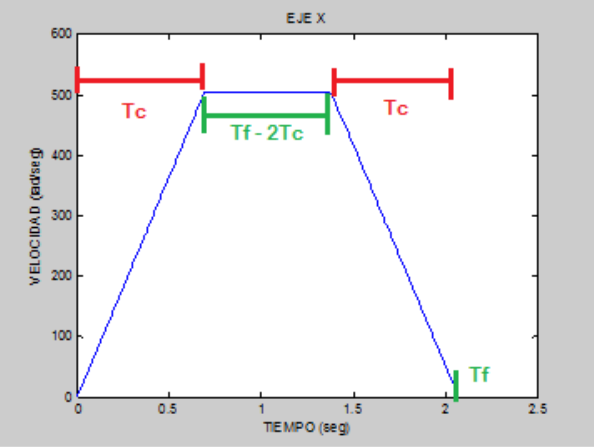


Figura 2.5.1: Tiempo crítico (Tc) y tiempo final (Tf)

Tesis publicada con autorización del autor
No olvide citar esta tesis

- Cumplir la condición de aceleración mínima, que permita cumplir la siguiente desigualdad:

$$tc < tf/2 \quad (2.5.3)$$

En el caso de no cumplirse la condición determinada por la expresión 2.5.3 se debe aumentar el valor de aceleración, siempre y cuando esté dentro del rango permitido por el motor o sino modificar la velocidad máxima

- Determinar el número total de muestras. Esto se obtiene dividiendo el tiempo final entre el periodo de muestreo usado para el algoritmo de control.
- Dividir el número total de muestras entre tres para tener igual número de puntos *(N)* en los tres segmentos del perfil de velocidad: aceleración, velocidad constante y desaceleración.
- Con los datos hallados hasta este momento, se pueden aplicar las ecuaciones cinemáticas para calcular cada tramo del perfil de velocidad trapezoidal. El código escrito en Lenguaje C sería el siguiente:

```
// Tramo de aceleración

 for ( i=0;  i<tc;   i=i+(tc/(N-1)) )
 {
     pos_1[contador]=(sp1_ini) + (0.5*acel*i*i);
     vel_1[contador]= a * i ;
     contador++;
 }

// Tramo de velocidad constante

 for ( j=tc;   j<(tf-tc);  j=j+((tf-2*tc)/(N-1)) )
 {
     pos_1[contador]= sp1_ini + ( acel*tc*(j-(tc/2)) );
     vel_1[contador]= a * tc ;
     contador++;
 }
```


Tesis publicada con autorización del autor
No olvide citar esta tesis

```
// Tramo de desaceleración

for ( k=(tf-tc);  k<tf;  k=k+(tc/(N-1)) )
{
    pos_1[contador]=sp1_fin - ( 0.5*acel*(tf-k)*(tf-k) );
    vel_1[contador]= a * (tf - k) ;
    contador++;
}
```

Luego de contar con todos los puntos intermedios del eje con mayor tiempo final, se genera el vector con los set points del eje restante.

- El siguiente paso es hallar la pendiente de la recta formada por los puntos inicial (Xo, Yo) y final (Xf, Yf) usando la siguiente fórmula:

$$pdte = \frac{(y_f - y_0)}{(x_f - x_0)} \tag{2.5.4}$$

- Aplicar la ecuación de una recta cuando se conoce la pendiente y un punto de paso.

$$y = y_0 + pdte(x - x_0) \tag{2.5.5}$$

- Finalmente, aplicar la Fórmula 2.5.6 para hallar la velocidad del eje restante usando las muestras del vector tiempo y del vector posición calculado en el paso anterior.

$$v_i = \frac{x_i - x_{i-1}}{t_i - t_{i-1}} \qquad \forall i = 1 \ldots N, v_0 = 0 \tag{2.5.6}$$

Hasta el momento los vectores de posición que contienen la trayectoria para cada eje se encuentran en milímetros, pero para poder ejecutar el algoritmo de control se tiene que realizar la conversión de milímetros a radianes. Esta conversión se tuvo que calcular experimentalmente y la explicación detallada de cómo se hallaron los factores de conversión para cada eje se explica más adelante en el Capítulo 4.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Capítulo 3: SIMULACIÓN E IMPLEMENTACIÓN

Las simulaciones que se presentan a continuación corresponden al algoritmo de generación de trayectoria y al algoritmo de control PID para el eje X e Y. Este último fue simulado con dos tipos de referencias: escalón unitario y el generador de trayectoria. El objetivo es cumplir a nivel de software de simulación (Matlab 7.0 y Simulink 6.0) con lo planteado en la etapa de diseño para luego pasar a la etapa de implementación y poder comparar ambos resultados.

3.1 Simulación del Generador de Trayectoria

Luego de ejecutar el programa "Generador de trayectoria" presentado en el Anexo 7, se obtuvieron las gráficas de las Figuras 3.1.1 y 3.1.2 respectivamente.

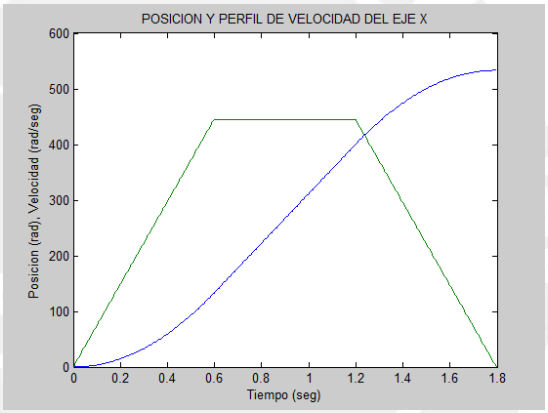


Figura 3.1.1: Posición y velocidad del eje X en función del tiempo

En la Figura 3.1.1 se observa el perfil de velocidad (color verde) y la posición del eje X (color azul) en la misma gráfica. Se puede ver que el generador de trayectoria tarda 1.8 segundos en enviar el último punto del recorrido.

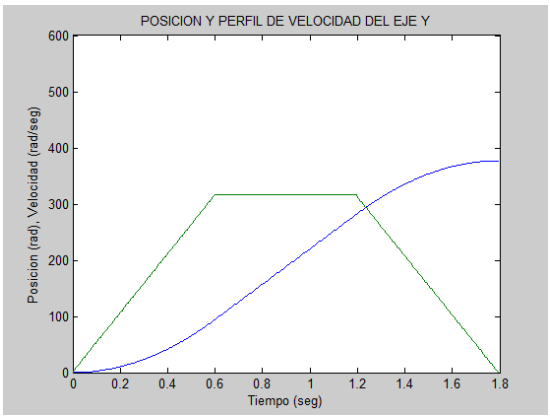


Figura 3.1.2: Posición y velocidad del eje Y en función del tiempo

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

Asimismo, en la Figura 3.1.2 se comprueba que el tiempo que demora el generador de trayectoria en enviar el último set point del eje Y es el mismo que tardó para el eje X, a pesar de que este último tiene que cubrir el doble de la distancia. Con esto se demuestra que es un movimiento coordinado de dos ejes porque ambos comienzan y terminan de hacer su recorrido en el mismo instante de tiempo.

3.2 Simulación del Algoritmo de Control PID

3.2.1 Respuesta a una Referencia Escalón Unitario

Simulink es una herramienta de Matlab que permite simular sistemas de control. En la Figura 3.2.1 se muestra el diagrama de simulación correspondiente al sistema del eje X, que incluye el controlador PID sintonizado con un tiempo de muestreo de 5ms.

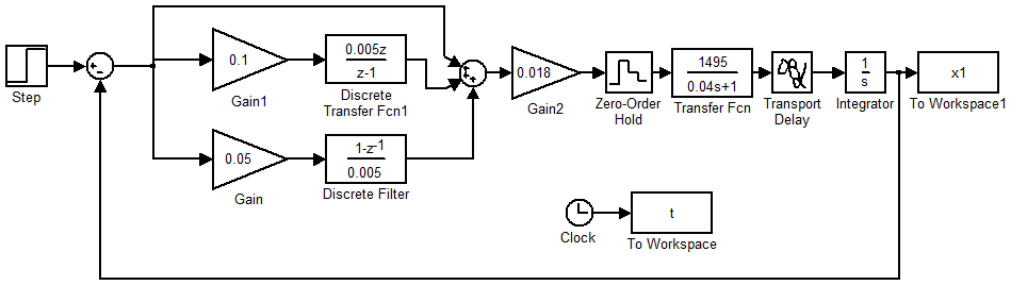


Figura 3.2.1: Diagrama de simulación del controlador PID del eje X

En la Figura 3.2.2 se observa que la respuesta del Eje X es sobreamortiguada con un tiempo de establecimiento menor a 200ms y con un error nulo en estado estable. Por lo tanto, se puede afirmar que en la simulación se cumple con los requerimientos del sistema.

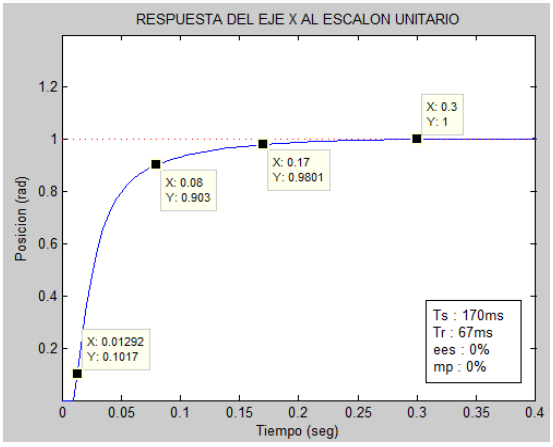


Figura 3.2.2: Respuesta del eje X al escalón unitario

Tesis publicada con autorización del autor
No olvide citar esta tesis

En la Figura 3.2.3 se presenta el diagrama de simulación usado para conocer la respuesta del sistema del eje Y junto al controlador PID frente a una referencia escalón unitario. El tiempo de muestreo usado es de 5ms.

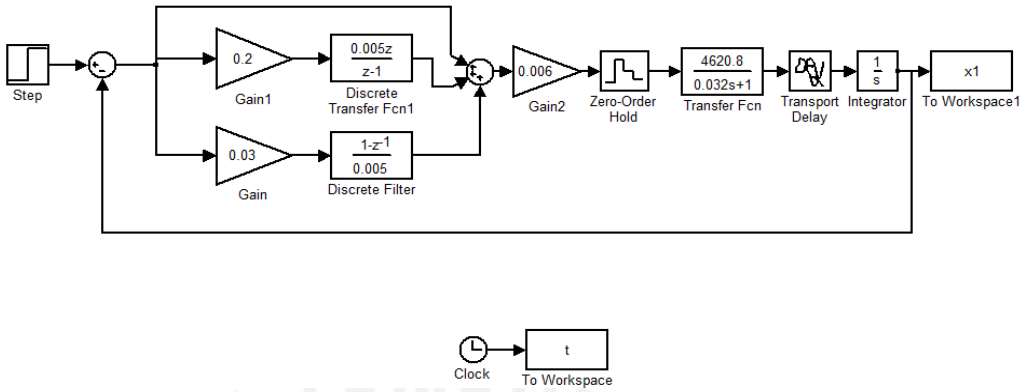


Figura 3.2.3: Diagrama de simulación del controlador PID del eje Y

Según la Figura 3.2.4, la respuesta del eje Y también cumple con las especificaciones de diseño establecidas e incluso su tiempo de establecimiento es menor al del eje X.

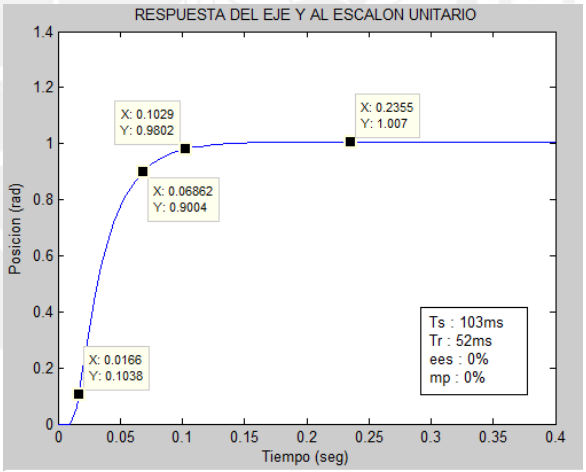


Figura 3.2.4: Respuesta del eje Y al escalón unitario

3.2.2 Respuesta Frente al Generador de Trayectoria

El siguiente paso es simular la respuesta de ambos sistemas cuando la referencia es una trayectoria generada que depende del punto de partida y del punto de llegada. En las Figuras 3.2.5 y 3.2.6 se presentan los diagramas de simulación implementados en Simulink para analizar la respuesta en el tiempo de las plantas en estudio.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

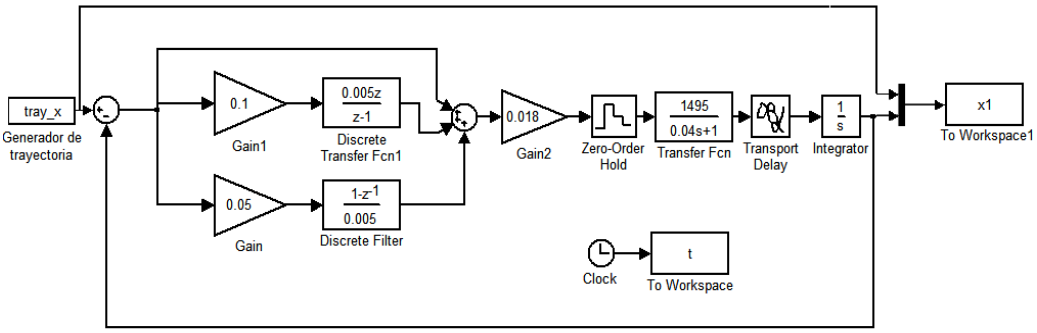


Figura 3.2.5: Diagrama de simulación del controlador PID del eje X con el generador de trayectoria como referencia del sistema

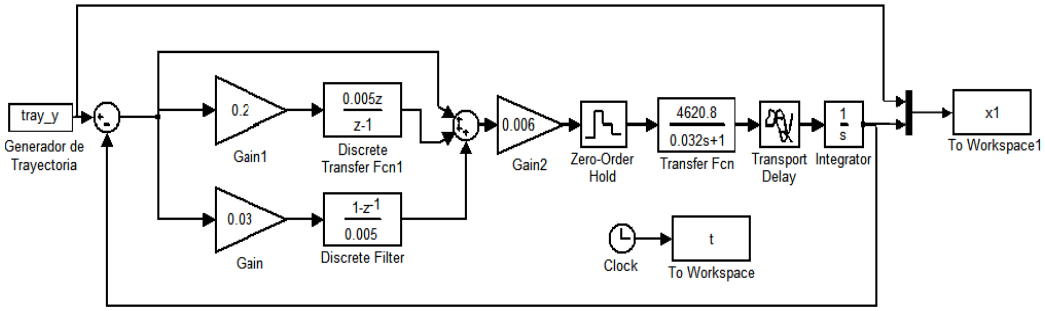


Figura 3.2.6: Diagrama de simulación del controlador PID del eje Y con el generador de trayectoria como referencia del sistema

En la Figura 3.2.7 está graficada en color verde la trayectoria ideal y en color azul la trayectoria real o controlada, ambas en función del tiempo. Se puede notar que existe un error mínimo de 13.4 radianes que equivale a 4mm y se debe principalmente a que la sintonización manual se realizó teniendo en cuenta una referencia tipo escalón unitario.

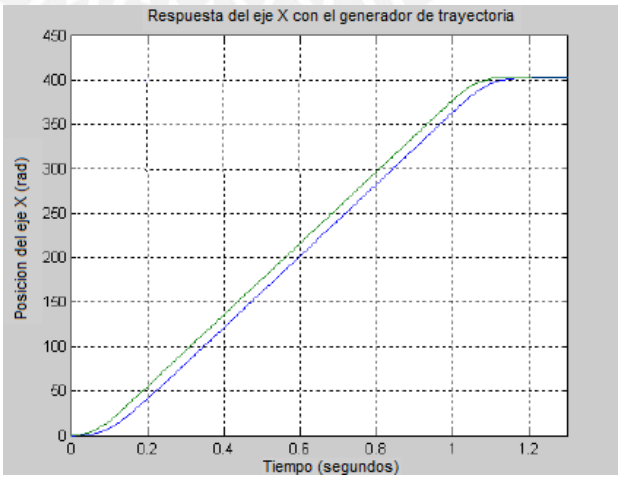


Figura 3.2.7: Trayectoria generada y respuesta del eje X frente a esta trayectoria como referencia.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

3.3 Implementación del Algoritmo de Control

Después de haber concluido con las etapas de diseño y simulación, se pasa a la etapa de implementación. En esta última, se va a comprobar el funcionamiento real que, en teoría, debería tener una respuesta similar a la obtenida en la simulación. Los programas empleados para el desarrollo en el microcontrolador ATmega128 fueron:

- VMLAB
- AVRStudio 4.18
- Compilador AVR-GCC

Entonces, se deben realizar correctamente todas las conexiones del sistema y luego se ejecutará el programa desarrollado para controlar la mesa XY.

3.3.1 Conexión del Hardware Experimental

El hardware del sistema de control de posición de la mesa XY está conformado por:

- 2 fuentes de alimentación: Cada una proporciona 24Vdc y 5Vdc para la etapa de potencia y 5Vdc para la etapa de control. Ambas con respectivas tierras, como se puede ver en la Figura 3.3.1.
- 2 interfaces de potencia: Una para cada motor como se muestra en la Figura 3.3.2. Cada una está compuesta por mosfets de potencia IRF 540 y un circuito que sirve para detener el sistema en caso se produzca una sobrecorriente en alguno de los motores.
- 2 circuitos de acondicionamiento de la señales de encoder: Una por cada encoder, como se muestra en la Figura 3.3.2, para adaptar las impedancias y evitar las caídas de voltaje cuando se conecten a las entradas del microcontrolador.
- 1 tarjeta de desarrollo: Basada en el ATmega128, mostrado en la Figura 3.3.3, donde se va a implementar el algoritmo de control y desde la cual se va a enviar y recibir señales de control.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Figura 3.3.1: Fotografía de las fuentes de alimentación.

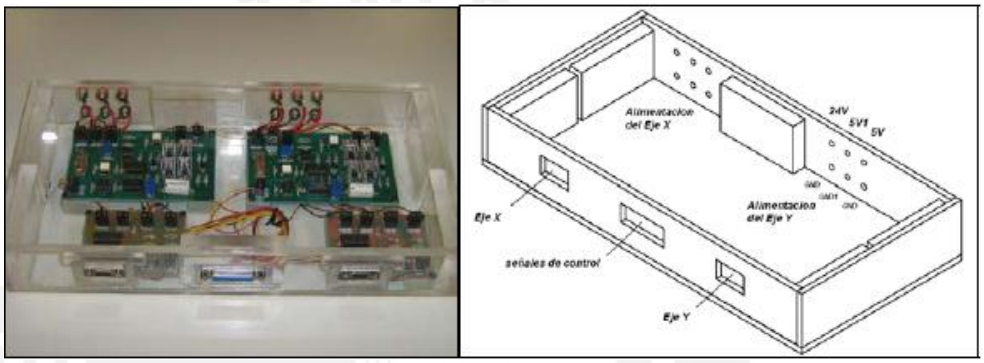


Figura 3.3.2: Fotografía y diagrama del módulo de potencia y acondicionamiento de las señales de encoder del sistema de posicionamiento XY [7]

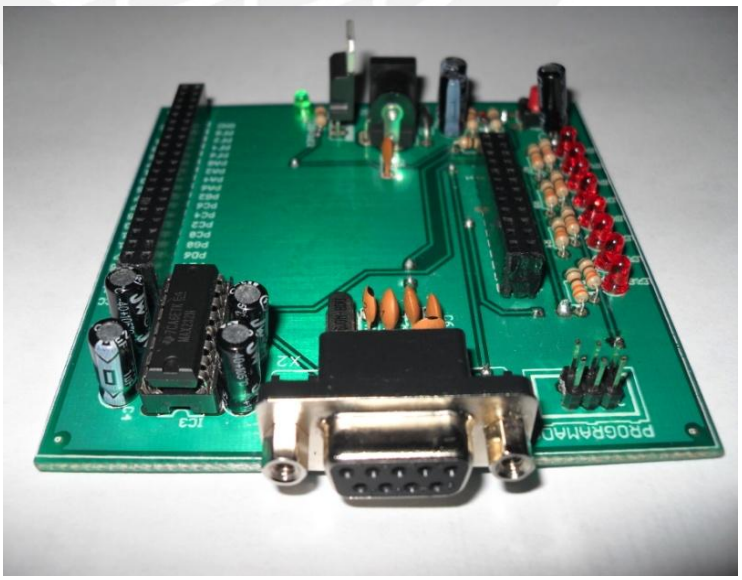


Figura 3.3.3: Fotografía de la tarjeta de desarrollo del ATmega 128

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Como inicialmente se disponía de las fuentes de alimentación, las interfaces de potencia y los circuitos de acondicionamiento de señal para los encoders, las conexiones se reducen a 3 cables por los cuales viajan todas las señales necesarias para implementar el sistema de control.

Estos son:

- 2 cables con conectores DB9 machos. Ambos provienen de la mesa XY, uno por cada eje, y van conectados a los conectores DB9 hembras que se encuentran en la caja de la etapa de potencia.
- 1 cable con conector DB25 macho. A través de este se envían 9 señales que son entradas o salidas del ATmega128. Este conector está sobredimensionado porque la caja con los conectores fue fabricada para usarse con una tarjeta de adquisición de datos que necesitaba señales adicionales a las que se están usando en la presente tesis.

En las Tablas 3.1, 3.2 y 3.3 se indican los pines con las señales de cada uno de los conectores utilizados y los puertos del microcontrolador ATmega 128 que se definieron para la implementación del sistema de control de posición.

Tabla 3.1: Señales del conector DB9 del eje X y eje Y

CONECTOR DB9	
Pin	**Señal**
1	Bornera "B" del motor
2	Sin conexión
3	Alimentación del encoder
4	Sin conexión
5	Tierra digital
6	Canal B del encoder
7	Sin conexión
8	Canal A del encoder
9	Bornera "A" del motor

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

Tabla 3.2: Señales del conector DB25

CONECTOR DB25	
Pin	**Señales**
3	Sobrecorriente Eje Y
4	Sobrecorriente Eje X
5	Bit de giro Eje Y
6	Bit de giro Eje X
7	Tierra digital
8	Canal A del encoder X
11	Canal A del encoder Y
17	PWM del Eje Y
18	PWM del Eje X

Tabla 3.3: Puertos del ATmega 128 dedicados para el sistema de control

CONEXIONES DEL ATMEGA128	
Puerto	**Señales**
PE4	Canal A (encoder del eje X)
PE5	Sobrecorriente eje X
PE6	Canal A (encoder del eje Y)
PE7	Sobrecorriente eje Y
PB5	PWM eje X
PB6	PWM eje Y
PC0	Bit de giro eje X
PC1	Bit de giro eje Y

3.3.2 Programación del Algoritmo de Control

El programa escrito en lenguaje C consta de dos partes: La generación de los set points para el perfil de velocidad trapezoidal y el algoritmo de control PID que se encargará de enviar las señales de control en base a las señales de referencia proporcionadas por el generador de trayectoria.

En la Figura 3.3.4 se muestra el diagrama de flujo del programa principal.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

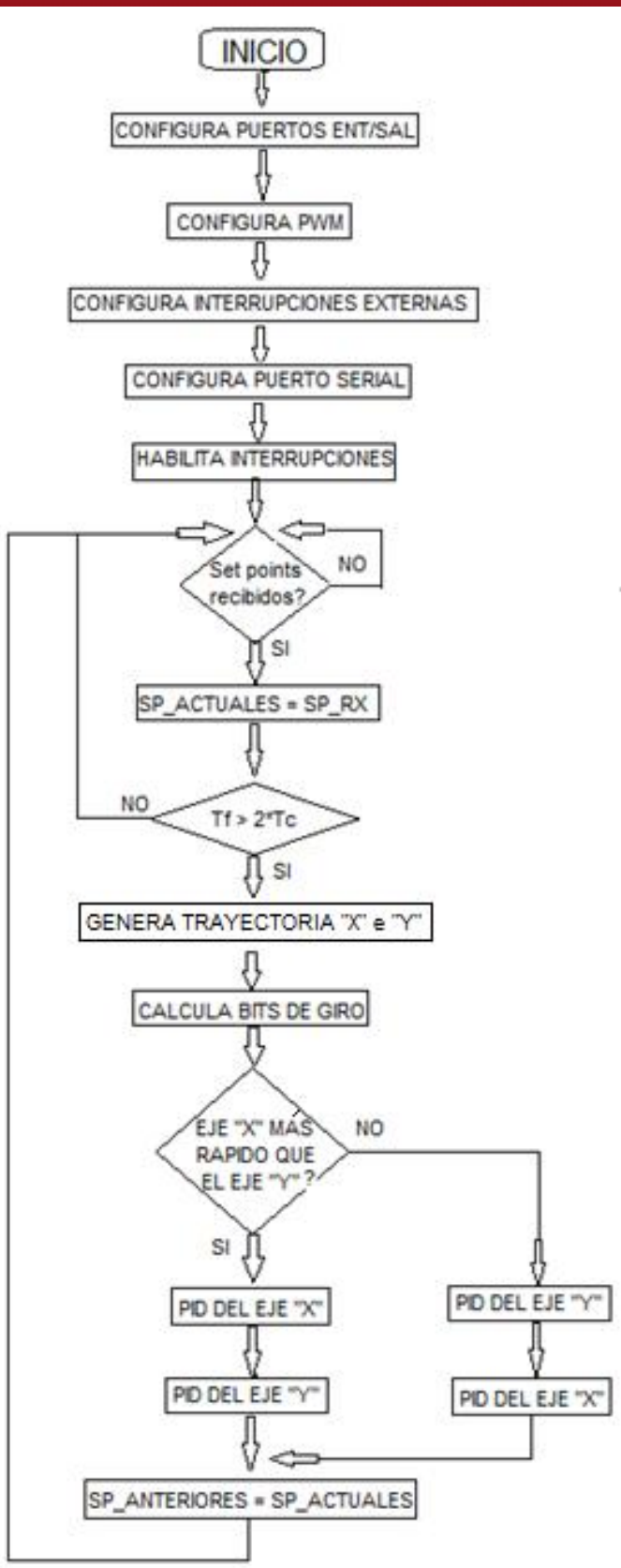


Figura 3.3.4: Diagrama de flujo del programa principal

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

En el “Programa principal” del Anexo 7, primero se declaran las librerías listadas en la Tabla 3.4 que contienen las funciones necesarias para que el programa ejecute correctamente el algoritmo de control.

Tabla 3.4: Librerías declaradas en el programa principal

LIBRERÍAS
#include <avr\io.h>
#include <avr\interrupt.h>
#include <avr\signal.h>
#include <stdint.h>
#include <math.h>
#include <stdlib.h>

Por otro lado, en la Tabla 3.5 se mencionan las rutinas de servicio de interrupción empleadas en el “Programa principal” del Anexo 7.

Tabla 3.5: Rutinas de servicio de interrupción del programa principal

INTERRUPCIÓN	SEÑAL	TIPO DE FLANCO
SIGNAL(SIG_INTERRUPT4)	Encoder eje X	Flanco de subida
SIGNAL(SIG_INTERRUPT5)	Sobrecorriente eje X	Flanco de subida
SIGNAL(SIG_INTERRUPT6)	Encoder eje Y	Flanco de subida
SIGNAL(SIG_INTERRUPT7)	Sobrecorriente eje Y	Flanco de subida

Mientras que en la Tabla 3.6 se presentan las funciones utilizadas.

Tabla 3.6: Funciones declaradas en el programa principal

FUNCIONES
void config_puertos(void)
void config_serial(void)
void config_pwm(void)
void config_int_ext(void)
void bits_de_giro(float spx_fin,float spx_ini,float spy_fin,float spy_ini)

Por último, en la Tabla 3.7 se detalla la cantidad y tipo de variables utilizadas en el programa principal.

Tabla 3.7: Lista de variables utilizadas en el programa principal

VARIABLES	
CANTIDAD	TIPO
4	uint8_t
11	uint16_t
28	float

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Capítulo 4: PRUEBAS Y RESULTADOS

4.1 Pruebas a Lazo Abierto

El objetivo de estas pruebas es hallar el factor de conversión de milímetros a radianes para cada eje de movimiento por separado. Debido a que los modelos matemáticos de cada motor están expresados en radianes y se tiene que mantener una uniformidad en las unidades empleadas. Para obtener esta conversión se elaboró el "Programa para pruebas a lazo abierto" del Anexo 7, el cual incrementa el ciclo de trabajo de una señal PWM en 0.4% cada 65.28 ms con la condición que cuando el motor comience a girar, es decir cuando el microcontrolador reciba el primer pulso de encoder, se detenga el incremento del ancho de pulso. Entonces, con el ciclo de trabajo constante se envían los pulsos de encoder serialmente desde el microcontrolador al Labview hasta que se cumpla la condición de parada. Así se podrá establecer una relación entre la distancia lineal recorrida en milímetros y los pulsos de encoder recibidos.

4.1.1 Pruebas a Lazo Abierto del Eje X

En la Tabla 4.1 se muestran los resultados que se obtuvieron luego de realizar varias pruebas con condiciones de parada distintas.

Tabla 4.1: Resultados de las pruebas en lazo abierto del eje X

EJE X	
PULSOS	**DISTANCIA RECORRIDA (mm)**
200	26.5
514	54.9
853	90.2
1000	110.9
1028	109.1
1553	160.8
2100	229.1
2600	283.5
2900	296.8

Estas pruebas permiten hallar que un pulso de encoder equivale a 0.10325 milímetros de desplazamiento lineal y además, se tiene el dato que el encoder es de 20 pulsos por revolución. Entonces, haciendo una equivalencia se concluye que el factor de conversión es igual a 2.12milímetros/revolución.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

Por lo tanto, usando la fórmula 4.1.1 se puede conocer cuál es la posición en coordenadas articulares:

$$posx\_rad = posx\_mm * (2\pi / 2.12) \quad (4.1.1)$$

En las Figuras 4.1.1, 4.1.2 y 4.1.3 se presentan las gráficas con los resultados obtenidos en el programa implementado en Labview para conocer el número exacto de pulsos que el microcontrolador recibió provenientes del encoder del motor del eje X.

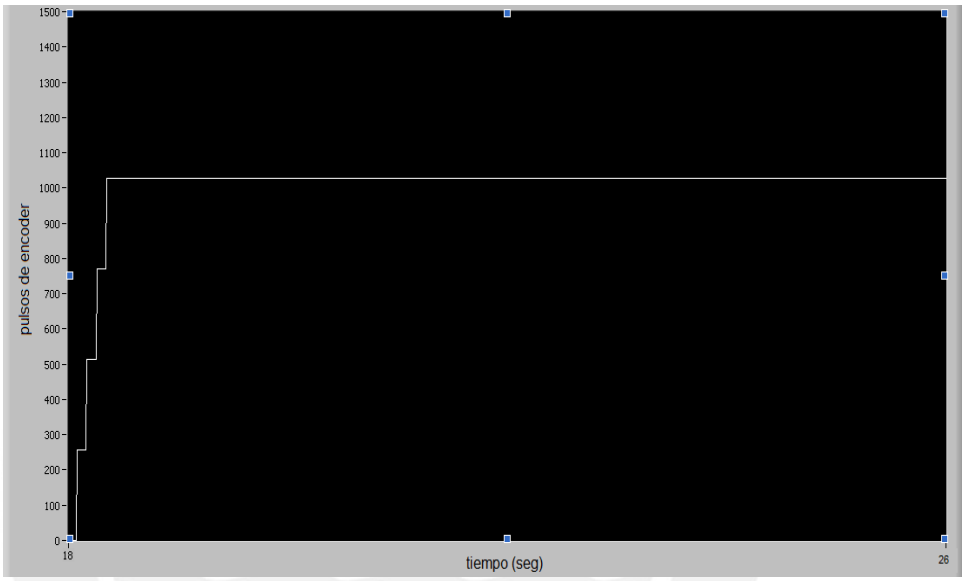


Figura 4.1.1: Respuesta del eje X en lazo abierto (Condición de parada: 1028 pulsos del encoder)

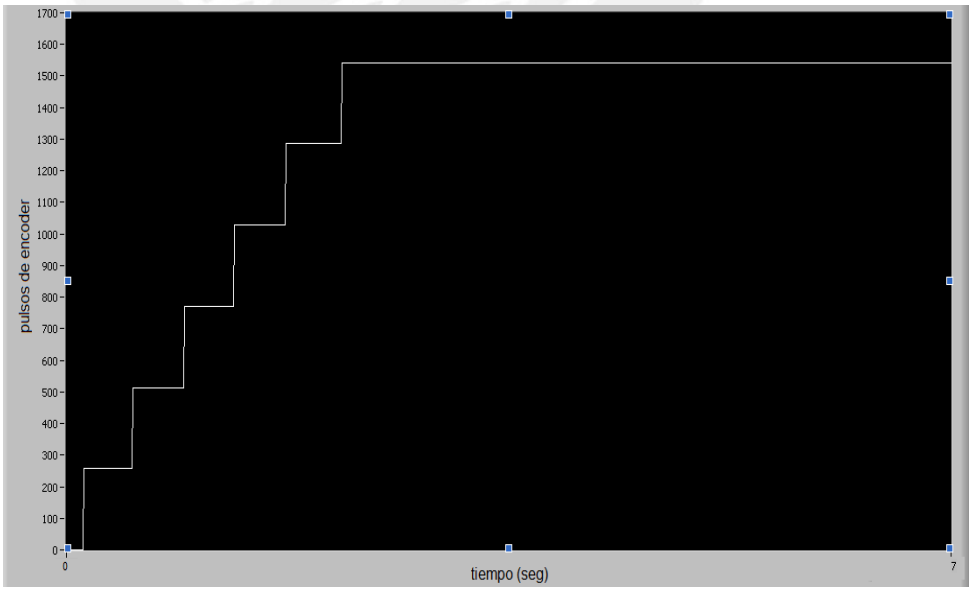


Figura 4.1.2: Respuesta del eje X en lazo abierto (Condición de parada: 1542 pulsos del encoder)

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

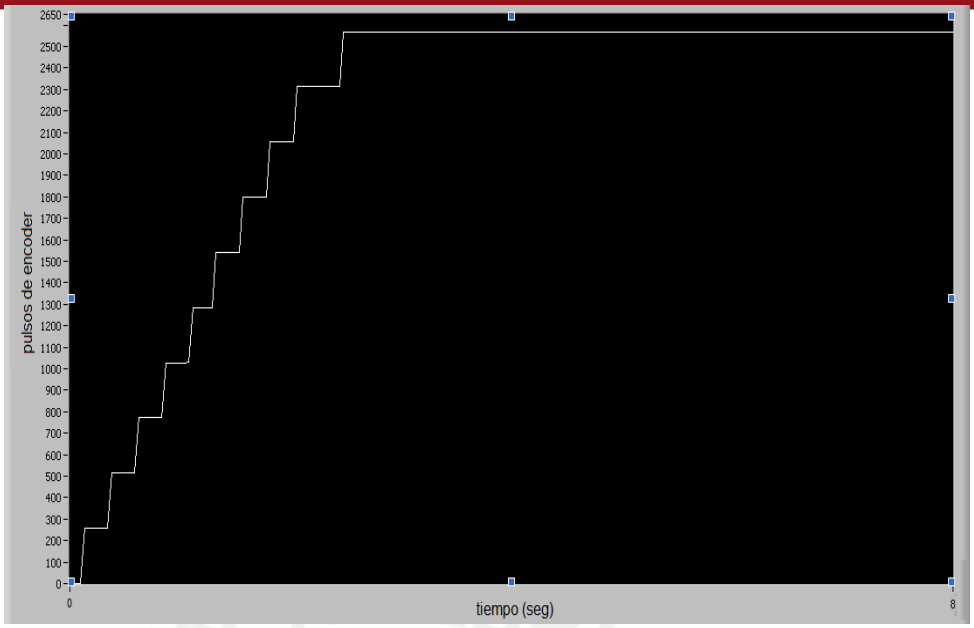


Figura 4.1.3: Respuesta del eje X en lazo abierto (Condición de parada: 2570 pulsos del encoder)

4.1.2 Pruebas a Lazo Abierto del Eje Y

Para las pruebas a lazo abierto del Eje Y se realizó el mismo procedimiento que con el Eje X, pero con diferentes condiciones de parada. En la Tabla 4.2 se detallan las distancias recorridas en milímetros por cada condición de parada expresada en pulsos de encoder.

Tabla 4.2: Resultados de las pruebas en lazo abierto del eje Y

EJE Y	
PULSOS	**DISTANCIA RECORRIDA (mm)**
150	15.2
450	45.2
900	89.8
1028	103.1
1500	152.9
1660	166.5
2900	289.9
2950	295
2965	296.85

Luego de realizar las pruebas correspondientes se comprobó que el factor de conversión para el eje Y no es el mismo que para el eje X. Según los resultados de la Tabla 4.2, un pulso de encoder equivale a 0.1mm de

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

desplazamiento lineal y por dato del fabricante de la mesa XY se conoce que el encoder es de 20 pulsos por revolución. Con esta información se calcula el nuevo factor de conversión igual a 2.00 milímetros/revolución. Entonces, la posición en coordenadas articulares se puede calcular partiendo de las coordenadas cartesianas aplicando la siguiente fórmula:

$$posy\_rad = posy\_mm * (2\pi / 2.00) \quad (4.1.2)$$

En las Figuras 4.1.4, 4.1.5 y 4.1.6 se pueden ver las respuestas obtenidas en el software Labview:

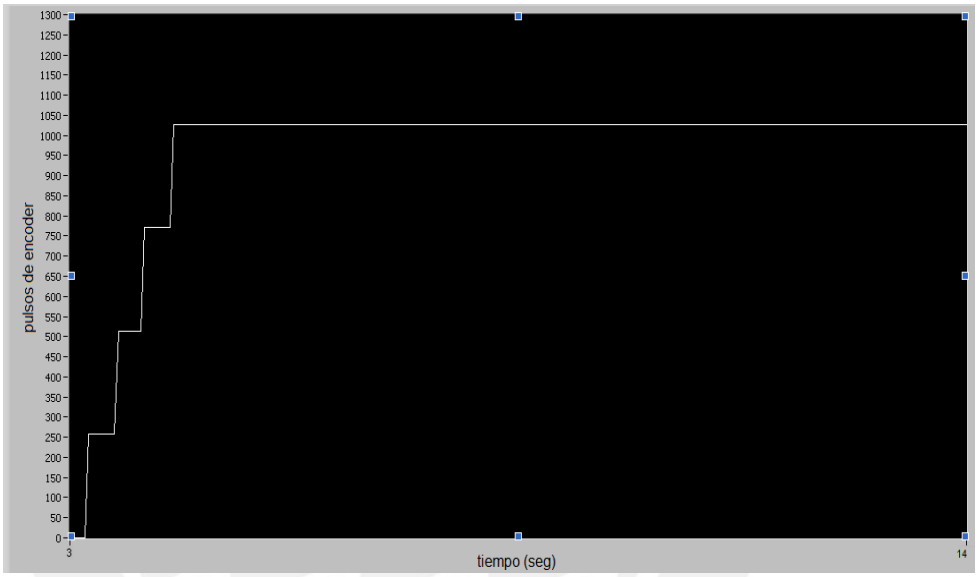


Figura 4.1.4: Respuesta del eje Y en lazo abierto (Condición de parada: 1028 pulsos del encoder)

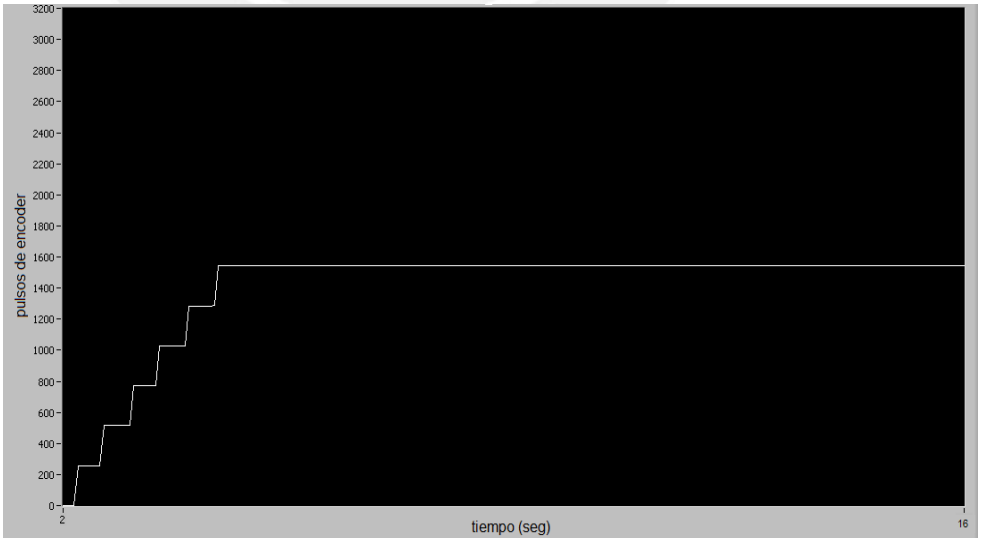


Figura 4.1.5: Respuesta del eje Y en lazo abierto (Condición de parada: 1600 pulsos del encoder)

Tesis publicada con autorización del autor
No olvide citar esta tesis

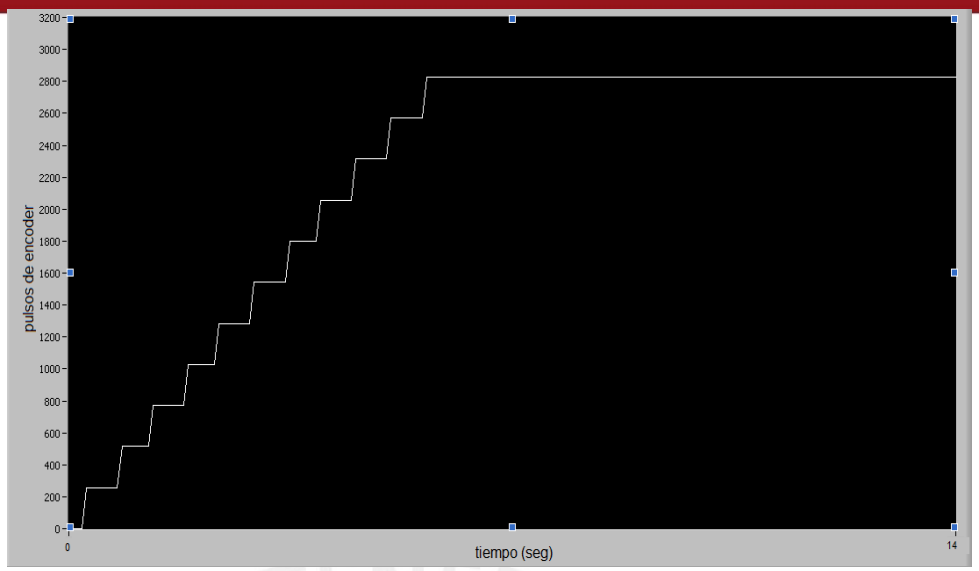


Figura 4.1.6: Respuesta del eje Y en lazo abierto (Condición de parada: 2900 pulsos del encoder)

4.2 Pruebas a Lazo Cerrado

En esta etapa se realizaron las pruebas de ambos algoritmos de control PID ejecutando el "Programa principal" del Anexo 7 para poder verificar que los parámetros de sintonización del Eje X y el Eje Y de la Tabla 2.3 y la Tabla 2.4 respectivamente eran los adecuados para cumplir con los requerimientos establecidos.

En la Tabla 4.3 se presentan los resultados del Eje X.

Tabla 4.3: Resultados de las pruebas a lazo cerrado del eje X

EJE X	
Set Point (mm)	**Distancia recorrida (mm)**
200	206.1
100	103
150	154.3
250	259

Como resultado de las pruebas realizadas se obtuvo un error aproximado del 3% en estado estable para el eje X. Asimismo, en la Tabla 4.4 se muestran los resultados de las pruebas del Eje Y.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

Tabla 4.4: Resultados de las pruebas a lazo cerrado del eje Y

EJE Y	
Set Point (mm)	**Distancia recorrida (mm)**
200	209.9
100	104.7
150	156.6
250	262.1

En cambio, en el eje Y el error promedio en estado estable fue de 4.7%. Por lo tanto, ambos ejes necesitaban un reajuste de sus constantes para lograr la mayor precisión posible que es lo que se busca. Luego de realizar varias pruebas modificando solo la acción proporcional en el diagrama de simulación del Simulink, se logró obtener una mejor respuesta en el tiempo con los valores que se muestran en la Tabla 4.5 y la Tabla 4.6.

Tabla 4.5: Nuevos parámetros del controlador PID del eje X

EJE X		
Kp	**Ti (seg)**	**Td (seg)**
0.072	10	0.05

Tabla 4.6: Nuevos parámetros del controlador PID del eje Y

EJE Y		
Kp	**Ti (seg)**	**Td (seg)**
0.055	5	0.03

Con estos nuevos valores se logró minimizar el error hasta un 1.12% para el eje X, mientras que en el eje Y se obtuvo un 1.35% de error en estado estable.

4.3 Análisis del Controlador Embebido Frente a Trayectorias Consecutivas

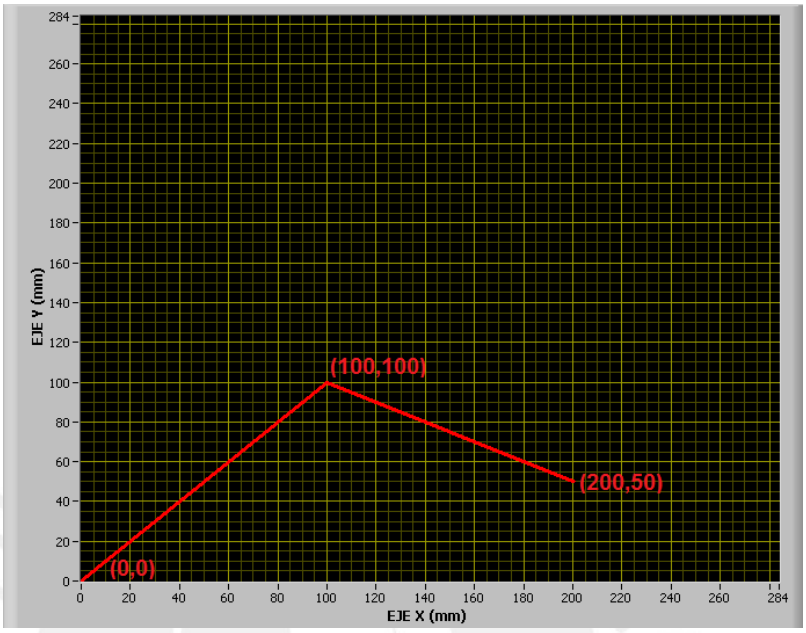
Se realizaron dos trayectorias consecutivas donde ambas iniciaron su recorrido desde el origen de coordenadas (0,0).

La primera, mostrada en la Figura 4.3.1, consistió en ir desde el punto de origen (0,0) hasta la coordenada (100,100) y luego al punto (200,50), de manera que en el primer movimiento ambos motores giraran en el sentido de avance, pero para llegar a la segunda coordenada, el motor del eje X siguió con el sentido giro anterior mientras que el motor del eje Y cambió el sentido de giro para retroceder hasta el punto 50.

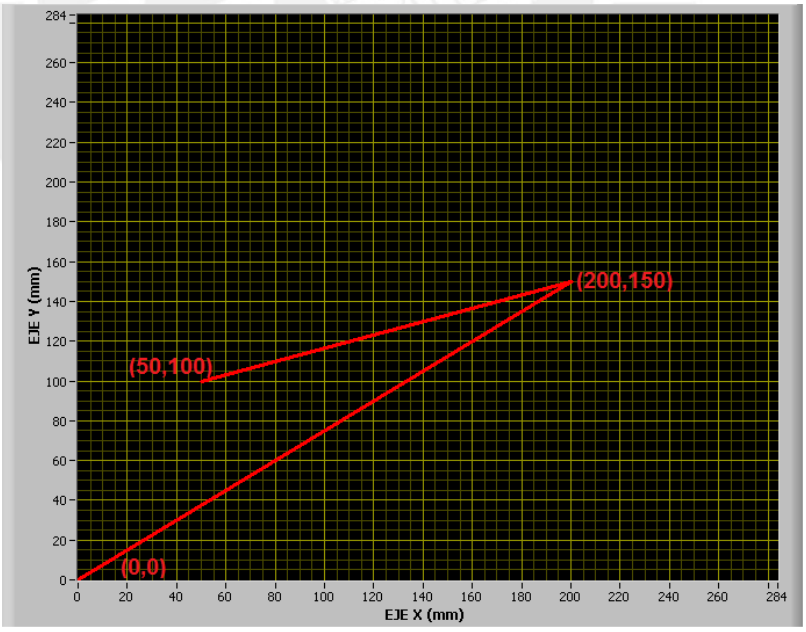
Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

En la Figura 4.3.2 se puede ver que la segunda trayectoria parte del origen (0,0) para dirigirse al punto (200,150) y finalmente llega al punto (50,100). Con estas coordenadas se logró que primero avancen ambos motores y luego cambien el sentido de giro inicial por el de retroceso.



Figurar 4.3.1: Trayectoria del punto (100,100) al punto (200,50)



Figurar 4.3.2: Trayectoria del punto (200,150) al punto (50,100)

A pesar de contar con encoders incrementales que se caracterizan por no guardar la posición absoluta con respecto al punto de origen, no hubo problema

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

porque se guardó la cantidad de pulsos recibidos de cada encoder en una variable tipo entera de 16 bits.

4.4 Análisis de los Costos de Implementación del Sistema de Control

En la Tabla 4.7 se presenta el presupuesto con el costo detallado de cada uno de los componentes adquiridos para la fabricación de la tarjeta de control y de los accesorios adicionales usados en la etapa de implementación del sistema embebido para el control de posición de la mesa XY.

Tabla 4.7: Presupuesto final para la implementación del Sistema de Control

Presupuesto final			
Cantidad	**Componente**	**C. Unit. (S/.)**	**C. Total (S/.)**
1	PCB en fibra de vidrio	50	50
1	Programador AVRISPmkII	150	150
1	Microcontrolador ATmega 128	55	55
1	Fuente de alimentación DC variable	20	20
1	Circuito integrado MAX 232	2	2
1	Regulador de voltaje LM7805	1	1
2	Condensador de 22pf	0.2	0.4
4	Condensador de 0.1 uf	0.2	0.8
5	Condensador de 1uf	0.2	1
2	Condensador de 10uf	0.2	0.4
9	Resistencia de 330 Ohms - 1/4 W	0.05	0.45
2	Resistencia de 10kOhms - 1/4 W	0.05	0.1
1	Resistencia de 100 Ohms - 1/4 W	0.05	0.05
8	Led rojo 3mm	0.1	0.8
1	Led verde 3mm	0.1	0.1
1	Cristal de 16 Mhz	2.5	2.5
1	Conector DB9 hembra para PCB	3	3
1	Conector hembra de fuente de alimentación	1	1
1	Pulsador	1	1
1	Espadín programador	0.5	0.5
1	Espadín hembra (40 pines por fila)	2.5	2.5
1	Conector DB25 macho	4.5	4.5
1	Cable de comunicación serial	6	6
		Subtotal (S/.)	303.1

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

Es importante mencionar que además de los componentes mencionados, se utilizaron:

- 2 fuentes de 24 voltios.
- 4 fuentes de 5 voltios (2 para cada eje).
- 2 circuitos de potencia para ambos ejes.
- 2 circuitos de adaptación para las señales del encoder.

Este hardware fue utilizado junto con la tarjeta de control. Sin embargo, no es considerado dentro de los costos porque fueron desarrollados años atrás por un alumno que también trabajó con la mesa XY, pero con un enfoque diferente al realizado en la presente tesis.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

CONCLUSIONES

- En el microcontrolador ATmega128 se puede implementar desde un controlador de movimiento simple hasta uno complejo porque se pueden manejar hasta 6 ejes de movimiento ya que cuenta con 6 canales PWM de buena resolución y 8 interrupciones externas, de las cuales 6 pueden usarse para las interfaces de encoder y las dos interrupciones restantes puede ser utilizadas para señales de sobrecorriente.

- El factor de conversión (mm/revolución), hallado experimentalmente, es diferente para cada eje porque uno soporta mayor carga que el otro y por lo tanto, no van a recorrer la misma distancia lineal para un determinado número de pulsos de encoder. En este caso, el eje X tiene como carga al eje Y con el portaobjetos, mientras que el eje Y sólo tiene como carga al portaobjetos.

- Se llega a la conclusión de que para implementar un algoritmo de control PID junto con un algoritmo para generar un perfil de velocidad trapezoidal en un microcontrolador ATmega128 es más recomendable programarlo en lenguaje C que en lenguaje ensamblador. La razón principal es porque el programador puede trabajar con números en punto flotante de manera sencilla y además, le permite realizar todas las operaciones matemáticas solo con declarar las librerías correspondientes.

- El porcentaje de error obtenido en el eje X y el eje Y se debe a que la resolución de la señal PWM sólo es de 0.4% y lo recomendable sería contar con una resolución de 0.1% porque los motores se saturan para valores bajos de ciclo de trabajo debido a que trabajan a una frecuencia de 33 Khz.

- Se comprobó que el software Labview es una muy buena alternativa para diseñar una interfaz de comunicación con el usuario porque su lenguaje de programación basado en diagrama de bloques es muy intuitivo para el programador y le permite diseñar con una apariencia amigable.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

RECOMENDACIONES

- Antes de descargar un programa en el microcontrolador ATmega128 se debe desactivar el fusible de compatibilidad con el microcontrolador ATmega103 (M103C) que se encuentra activado por defecto e impide al programador disponer de todos los recursos del ATmega128 limitándolo solo a los que dispone su antecesor, el ATmega103.

- Tener en cuenta que en ambos amplificadores se debe prender el interruptor de la parte digital antes que el interruptor de la parte analógica porque en caso contrario, le llegarán 24 voltios al motor y este comenzará a girar en cualquier sentido.

- Debido a que las computadoras modernas no cuentan con el puerto serial embebido, se puede utilizar un conversor USB – Serial de tipo industrial manteniendo la misma tarjeta de desarrollo para poder establecer la comunicación con la interface de usuario desarrollada en el software Labview 8.5.

- Los set points que determinan la trayectoria se deberían generar en cada periodo de muestreo para almacenarlos en una sola variable y no en dos arreglos porque para recorridos mayores a 297mm se sobrepasaría la capacidad de la memoria de datos del microcontrolador ATmega128.

- Lo más conveniente sería colocar fines de carrera mecánicos en ambos ejes para que envíen la señal al controlador y este se encargue de detener al motor correspondiente cuando haya llegado al final del recorrido porque la única forma de hacerlo durante las pruebas a lazo abierto fue observando el movimiento del eje para apagar la fuente respectiva.

- Durante las pruebas no se debe maniobrar la mesa XY manualmente para regresarla a su posición inicial ya que las fajas de cada eje se destensan y se pierde precisión en el movimiento.

- La mesa XY puede sincronizarse con un brazo robótico para que manipule la pieza colocada sobre el portaobjetos, ya que el microcontrolador ATmega128 cuenta con un periférico USART adicional a través del cual se puede establecer la comunicación entre el robot y el controlador.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

BIBLIOGRAFÍA

[1] V. Alfaro, "¿Son todos los controladores PID iguales?", *Ingeniería: Revista de la Universidad de Costa Rica*, Vol. 1, N°. 1, pp. 11-19, Enero 1993.

[2] V. Alfaro, "Ecuaciones para controladores PID universales", *Ingeniería*, *Revista de la Universidad de Costa Rica*, Vol. 12, N°. 1-2, pp. 11-20, Enero 2002.

[3] K. H. Ang, G. Chong y Y. Li., "PID Control System Analysis, Design and Technology", *IEEE Transactions on Control Systems Analysis Technology*, Vol. 13, N° 4, pp. 559-576, Julio 2005.

[4] *ATmega128 Datasheet*, Atmel Corporation, San Jose, CA, 2009.

[5] S. Bennett, "El pasado de los Controladores PID", en *Conferencia de Control Digital de la IFAC: Pasado, presente y futuro del Control PID*, 2000.

[6] G. S. Bohl (2009, Enero). Fundamentos de control de movimiento. National Instruments Developer Zone, Massachusetts. [En línea]. Disponible en: http://zone.ni.com

[7] J. Curay, "Control de movimiento en tiempo real de un posicionador XY", Tesis de Bachiller, Pontificia Universidad Católica del Perú, Lima, 2007.

[8] D. Greenfield, "Cuándo el Control PID no es la solución?", *Ingeniería de Control*, Vol. 57, N° 3, pp. 54-57, Marzo 2010.

[9] *Guía aplicativa del robot cartesiano*, Schneider Electric España, Barcelona, España, 2008.

[10] *Guía de Tecnologías de sistemas de control del Departamento de Ingeniería de sistemas, automatización e informática industria*, Universidad Politécnica de Catalunya, Barcelona, 1999.

[11] G. Keneth (2010, Febrero). Motion Controllers. Newmark Systems Inc., California. [En línea]. Disponible en: http://www.newmarksystems.com

[12] B. C. Kuo, *Sistemas de control automático.* 7ma edición. México DF, México: Prentice Hall Hispanoamérica, 1996.

[13] L. S. Monfils (2009, Julio). Robotic Accessories. Intelitek, New Hampshire. [En línea]. Disponible en: http://www.intelitek.com

[14] G. M. Morrison, "Velocity profiles – Trapezoidal vs. S-Curve", Precision Microcontrol Corporation, Carlsbad, CA, TN1031, 2000.

[15] K. Ogata, *Ingeniería de control moderna*. 3ra edición. México DF, México: Pearson Prentice Hall, 1998.

Tesis publicada con autorización del autor
No olvide citar esta tesis

TESIS PUCP

[16] M. R. Taylor (2009, Abril). Motion Control Overview. Advanced Motion Control, California. [En línea]. Disponible en: http://www.a-m-c.com/university/mco.html

[17] V. J. Van Doren (2003, Octubre). PID: Still the one. Control Engineering Digital Edition, Illinois. [En línea]. Disponible en: http://www.controleng.com

Tesis publicada con autorización del autor
No olvide citar esta tesis