

CONTROL EN TIEMPO CONTINUO
Informe sobre procesos de Control de Posición PID



Presentado por:
Jesús Humberto Dorado De La Cruz
Daniel Alejandro Rodríguez López
Juan Camilo Vásquez
Luis Fernando Miranda

Presentado a:
Ing. Hermes Fabián Vargas

Ing. en Automática Industrial

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones

2016

Objetivo de la Practica:

Proporcionar a una planta de posición un Controlador PID por medio de una tarjeta Arduino (Mega 2560 en este caso), y utilizando las herramientas de Matlab, Simulink para hacer la identificación de la planta. El control del sistema una vez identificado y sintonizado el controlador, será a través de un set point que modificará el valor de la temperatura a la cual queremos llevar la planta, y así suministrar los valores respectivos de PWM para el motor que moverá el riel a la posición indicada.

Conceptos a tener en cuenta:

- **Arduino:**

Arduino es una plataforma de creación de prototipos de código abierto basado en fácil de usar hardware y software. Placas Arduino son capaces de leer las entradas- la luz en un sensor, un dedo sobre un botón o un mensaje de Twitter - y lo convierten en una salida - la activación de un motor, encender un LED, publicar algo en línea. Se puede decir que su tablero qué hacer mediante el envío de un conjunto de instrucciones al microcontrolador en el tablero. Para ello se utiliza el lenguaje de programación de Arduino (basado en el cableado), y el software de Arduino (IDE), sobre la base de procesamiento.

(Arduino, 2016, www.arduino.cc)

- **Arduino Mega 2560:**

El Mega 2560 es una placa electrónica basada en el Atmega2560. Cuenta con 54 pines digitales de entrada / salida (de los cuales 15 se pueden utilizar como salidas PWM), 16 entradas analógicas, 4 UARTs (puertos serie de hardware), un oscilador de 16MHz, una conexión USB, un conector de alimentación, una cabecera ICSP, y un botón de reinicio. Contiene todo lo necesario para apoyar el microcontrolador; basta con conectarlo a un ordenador con un cable USB o la corriente con un adaptador de CA a CC o una batería para empezar. El tablero de 2560 mega es compatible con la mayoría de los shield para el Uno y las anteriores juntas de Duemilanove o Diecimila. El Mega 2560 es una actualización de la Arduino Mega, al que sustituye.

(Arduino, 2016, www.arduino.cc)

- **Matlab**

Es una plataforma que está optimizado para la solución de problemas de ingeniería y científicas. El lenguaje MATLAB basado en la matriz es la forma natural en la mayoría del mundo para expresar matemática computacional. Los gráficos integrados hacen que sea fácil de visualizar y obtener información a partir de datos. Cuenta con una vasta biblioteca de cajas de herramientas prediseñados que le permite comenzar de inmediato con algoritmos esenciales a su dominio. El entorno de escritorio invita a la experimentación, la exploración y el descubrimiento. Estas herramientas y capacidades de MATLAB están rigurosamente probados y diseñados para trabajar juntos.

(MathWorks, 2016, www.mathworks.com)

- **Simulink**

Simulink es un entorno de diagrama de bloques para la simulación multidominio y diseño basado en modelos. Es compatible con la simulación, generación automática de código, y la prueba continua y verificación de sistemas integrados.

Simulink ofrece un editor gráfico, bibliotecas de bloques personalizables y solucionadores para el modelado y simulación de sistemas dinámicos. Se integra con MATLAB, lo que le permite incorporar algoritmos de MATLAB en los modelos de simulación y los resultados de exportación a MATLAB para su posterior análisis.

(MathWorks, 2016, www.mathworks.com)

- **Sensor Ultrasonido**

Los **sensores de ultrasonido** o **sensores ultrasónicos** son detectores de proximidad que trabajan libres de roces mecánicos y que detectan objetos a distancias que van desde pocos centímetros hasta varios metros. calificado para operar en un rango de temperatura de -55 °C a 150 °C, mientras que el dispositivo LM35C está pensado para un rango de -40 °C a 110 °C (-10 °C con una precisión mejorada).

(Wikipedia, 2016)

- **PWM en Arduino**

Modulación de ancho de pulso o PWM, es una técnica para obtener resultados

análogos con medios digitales. Control digital se utiliza para crear una onda cuadrada, una señal de conmutación entre encendido y apagado. Este patrón

de encendido y apagado puede simular tensiones en el medio completo en (5 voltios) y desactivación (0 voltios) cambiando la parte de las veces la señal pasa en comparación con el tiempo que la señal pasa fuera. La duración del "tiempo" se llama el ancho de pulso. Para conseguir variando los valores analógicos, se cambia, o modular, que el ancho de pulso. Si repite este patrón de encendido y apagado suficientemente rápido con un LED por ejemplo, el resultado es como si la señal es una tensión constante entre 0 y 5V controlar el brillo del LED".

(Arduino, 2016, www.arduino.cc)

Desarrollo de la practica:

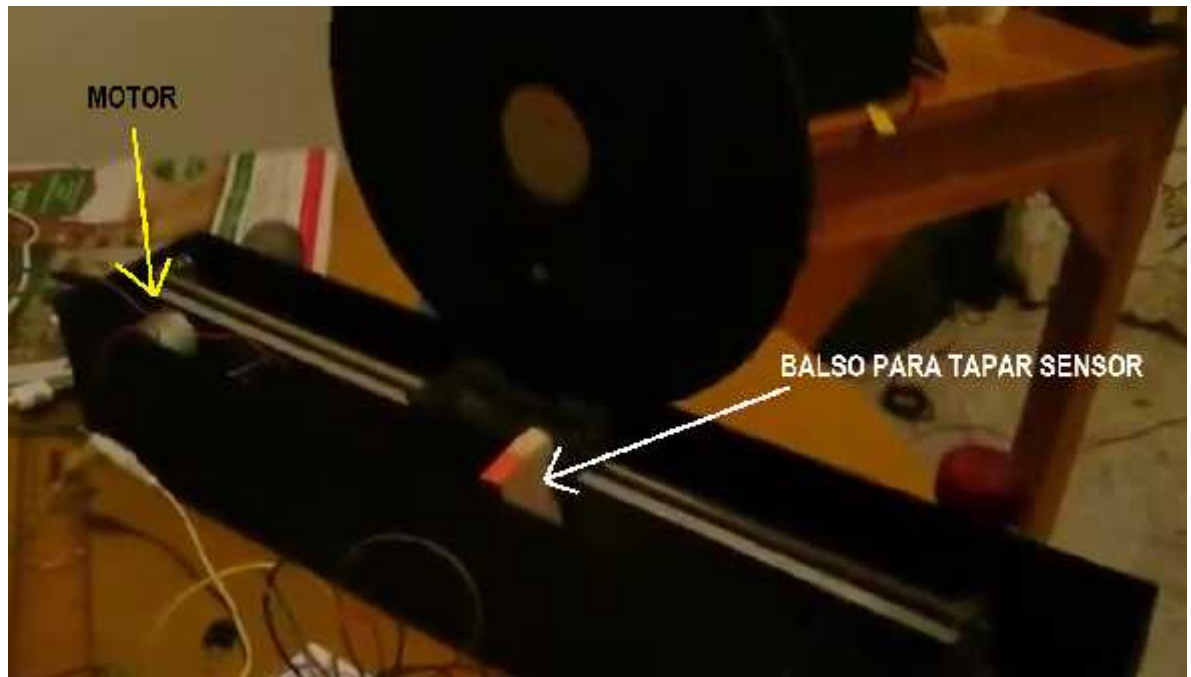
Elementos y equipos utilizados:

- ❖ Caja para montaje en metálica (previamente diseñada)
- ❖ 1 Arduino Mega 2560.
- ❖ 1 sensor de ultrasonido
- ❖ 1 Fuente de Voltaje de 9V DC
- ❖ Cables jumper's.
- ❖ 1 Potenciómetro de 10 k .
- ❖ 1 Protoboard
- ❖ 1 Motor DC 12V
- ❖ 1 Extensión para fuente alterna domiciliaria de 120V AC
- ❖ MATLAB 2014a y librerías de Simulink para Arduino.
- ❖ Software de para lenguaje de programación Arduino.
- ❖ Riel de impresora
- ❖ Balso

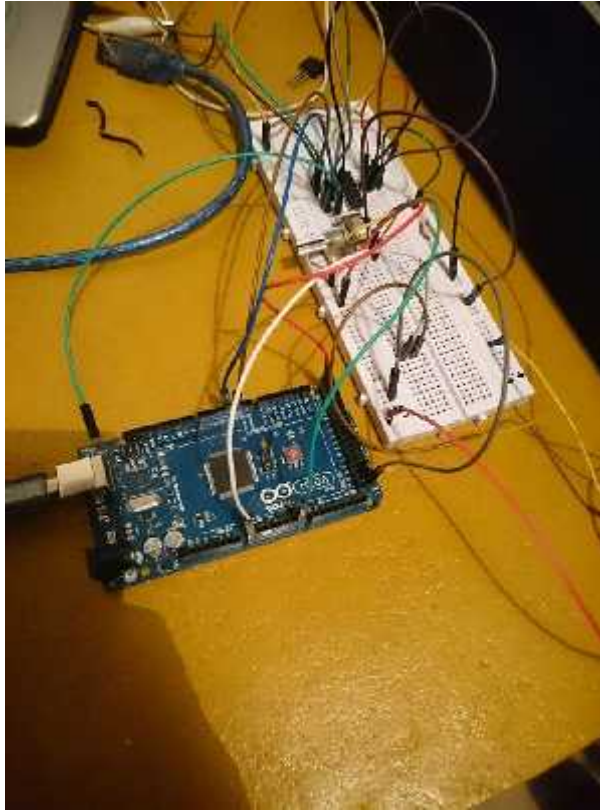
Procedimiento:

1. Montaje.

Se procede a adecuar el espacio necesario para el correcto funcionamiento de la planta esto supuso adecuar a caja metálica con balsa para proporcionar estabilidad a la hora del desplazamiento del riel de la impresora y también se adecuo una pared de balsa como se muestra en imágenes posteriores para la correcta lectura del sensor de ultrasonido.



Posteriormente se procede a ensamblar todos los componentes mencionados anteriormente en la protoboard, para generar el set point el cual se le aplicara al riel de la impresora.

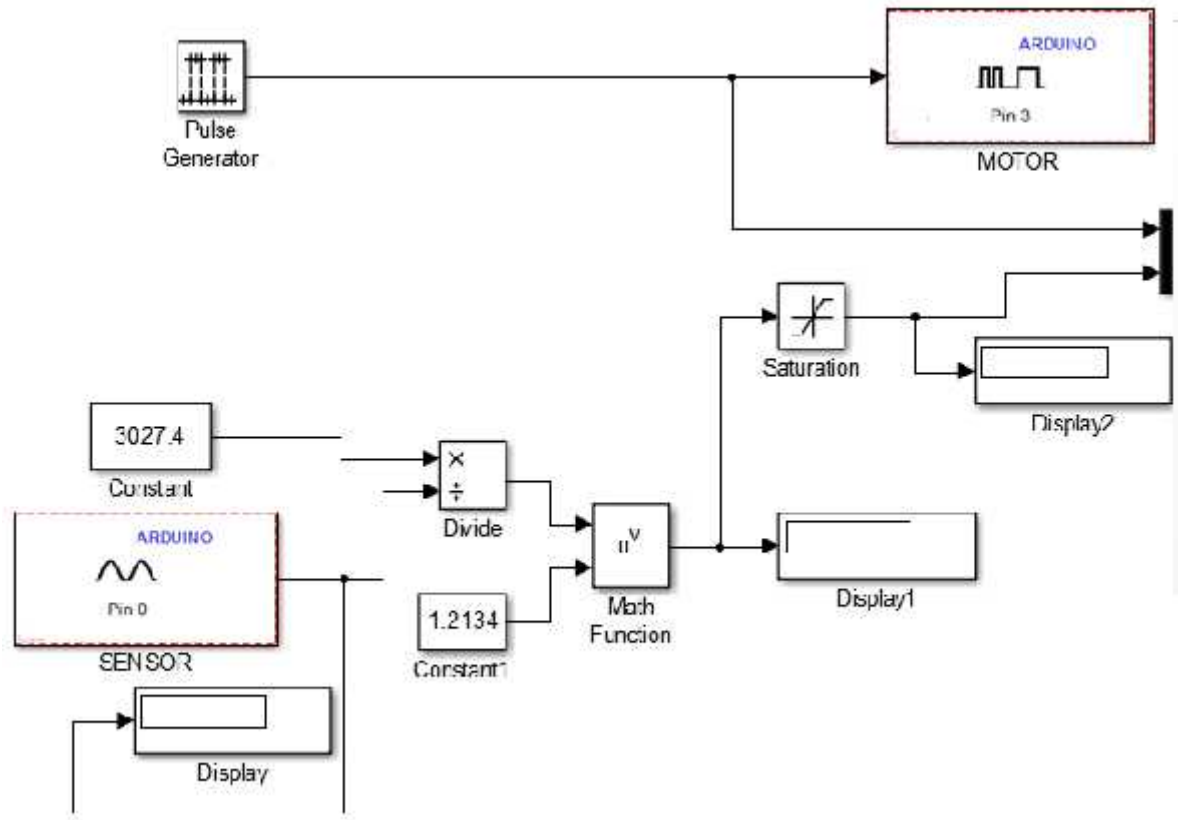


Como se puede apreciar en la anterior imagen se adecua en la protoboard el potenciómetro, y el driver para la correcta función del motor.

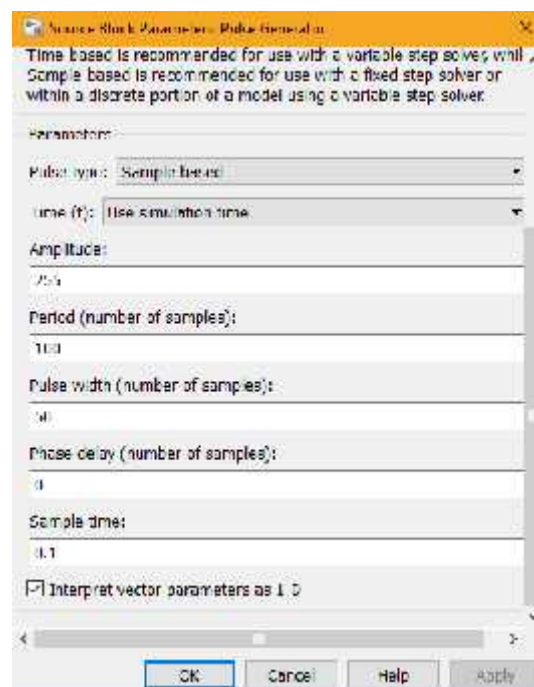
2. Procedimiento para la Identificación de la Planta.

Como primera medida se generó un impulso al actuador, es decir el que va movimiento para el riel de la planta que en este caso es el motor.

A continuación, se proporcionan las imágenes donde se puede apreciar el código en Simulink que hace posible la lectura de las señales para determinar el modelo de la planta.

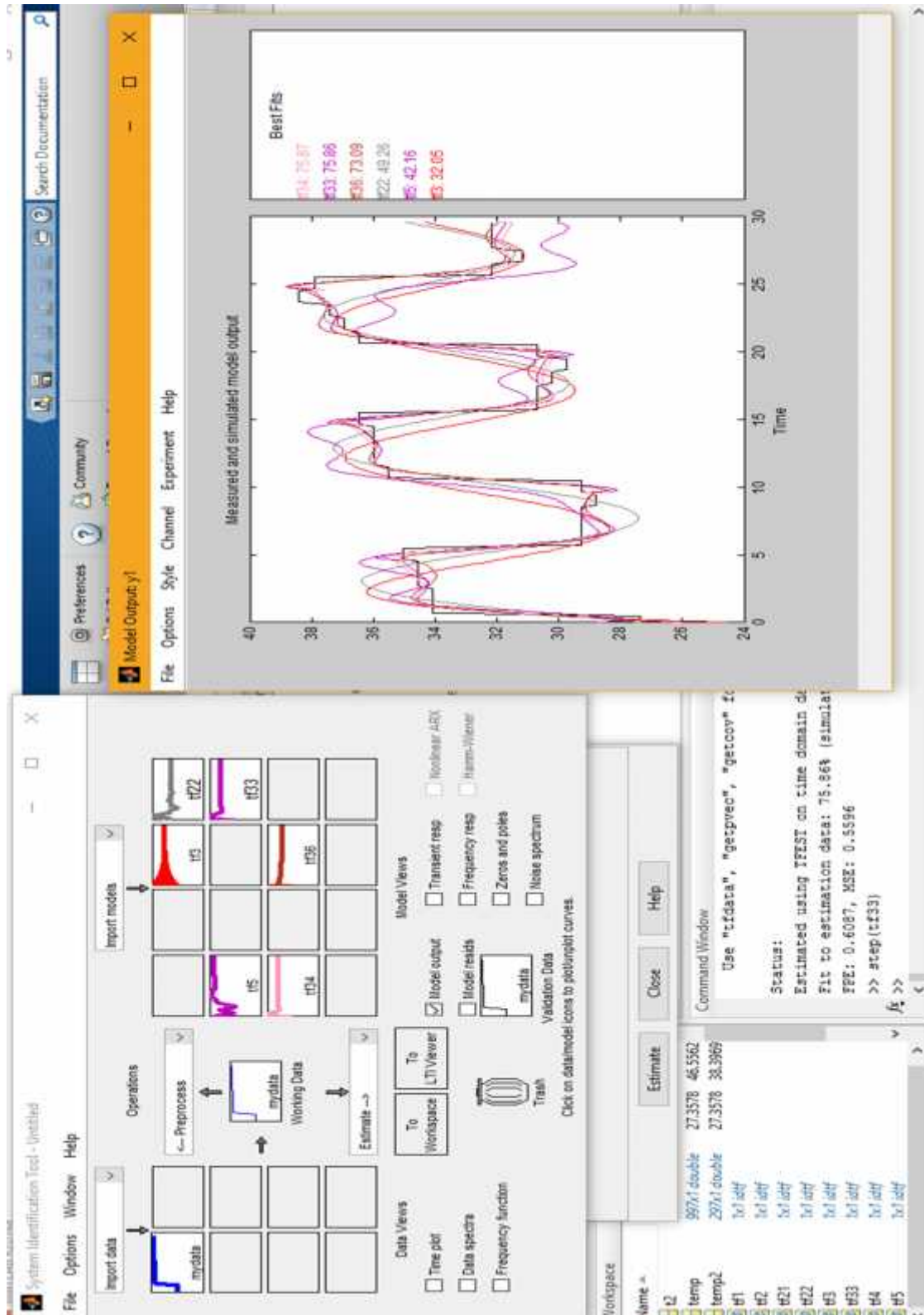


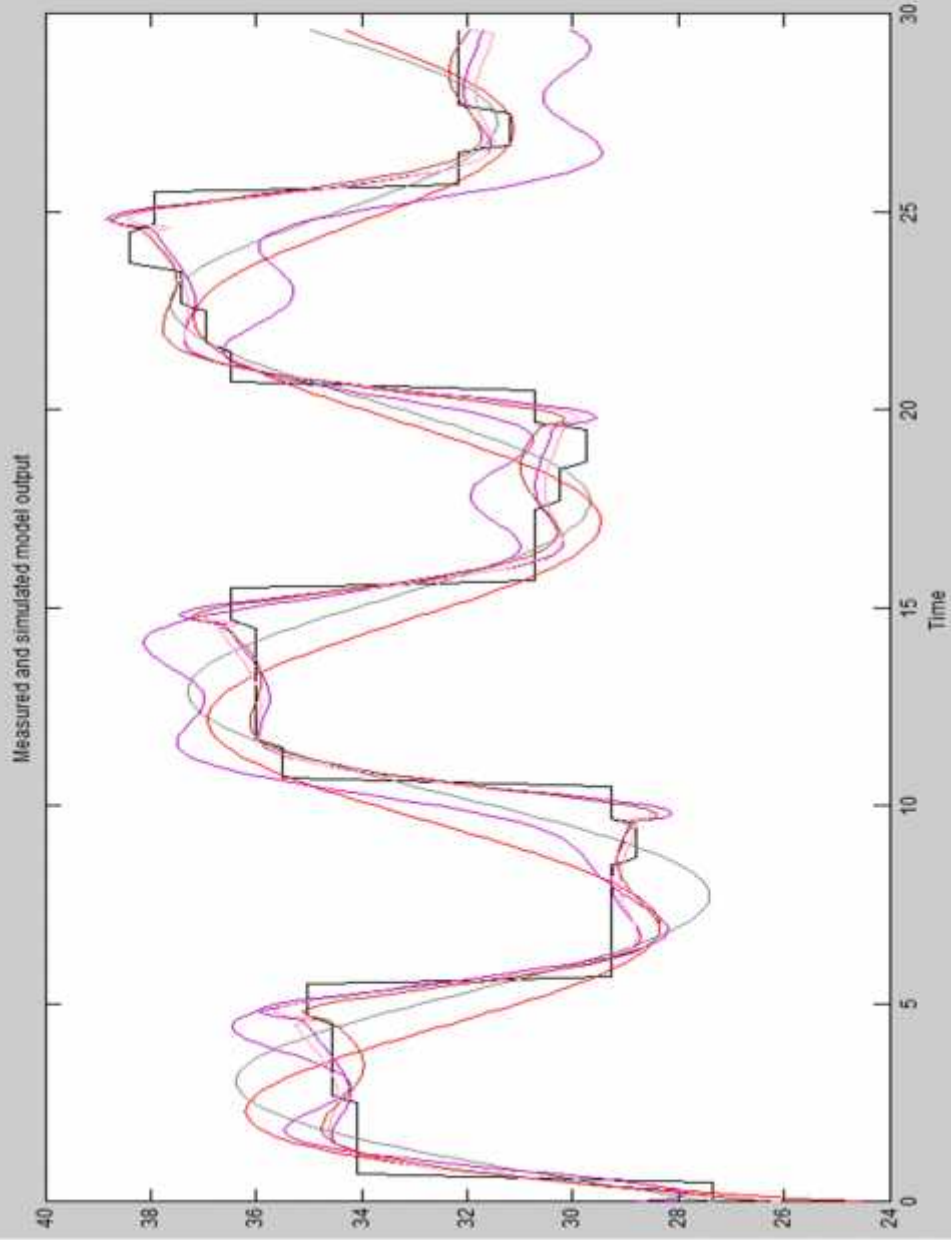
En el procedimiento se hicieron de 0 a 500 muestras en intervalos de 0.1, para obtener un modelo más exacto, con lo cual se escoge las mejores muestras para obtener los valores de I/O que se van a utilizar para hacer la identificación



Después se procede a crear dos vectores (para cada señal), con el rango de tiempo y los datos en ese rango para proporcionarlos a Matlab y crear una gráfica de I vs O. Este procedimiento se hace con la función **ident**, la cual estima una función de transferencia.

El paso siguiente es empezar a variar los polos y ceros de la función obtenida para obtener una exactitud como mínimo de 80%.





Best Fits

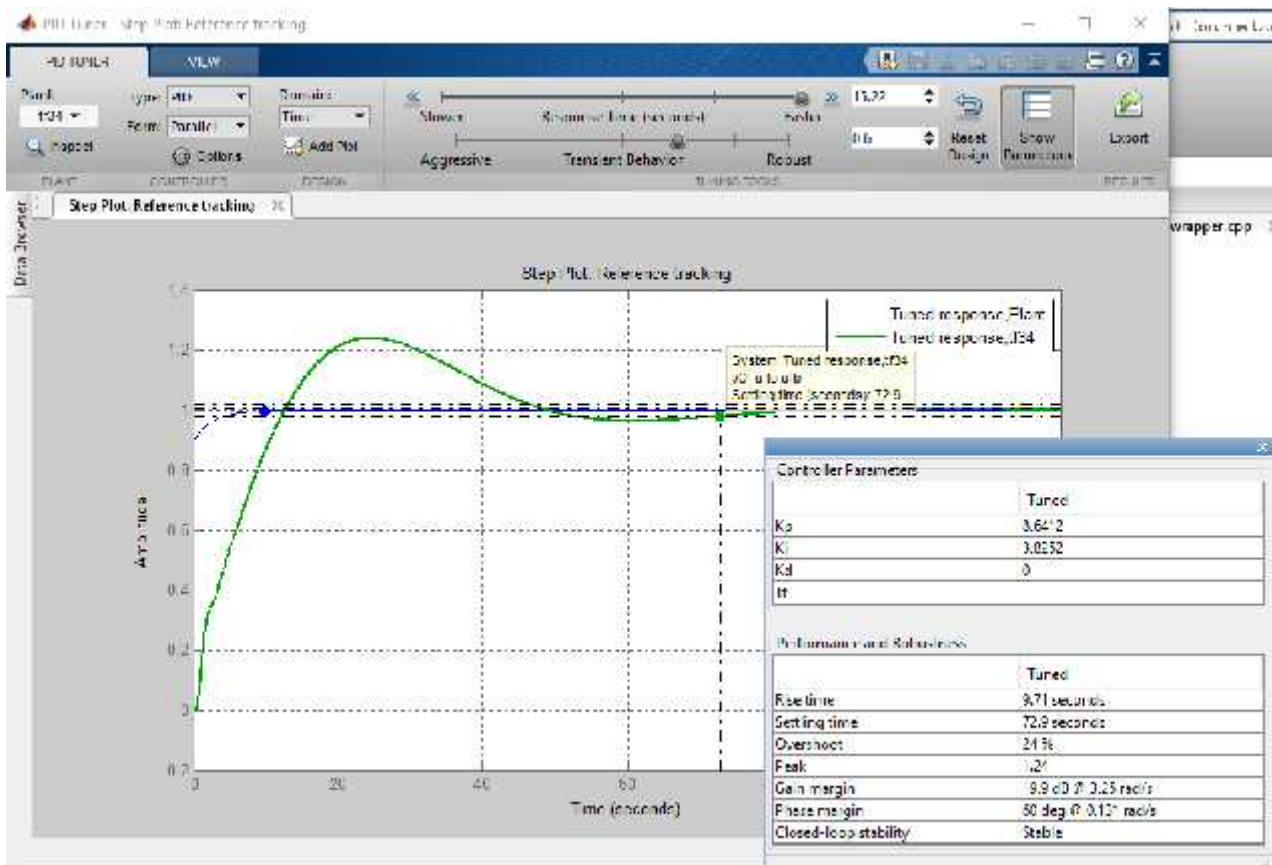
334: 75.67
333: 75.86
336: 73.09
322: 49.26
335: 42.16
333: 32.05

3. Estimación y sintonización del Controlador

Para este procedimiento se hace uso de la herramienta **pidtool**, para hacer la sintonización de la planta, tomando la función de transferencia estimada para modificar la respuesta del sistema siendo más rápido o más robusto, lo cual modifica las constantes del controlador que estima el **pidtool**.

Una vez obtenido un modelo que cumpla unas condiciones de rapidez, pero estable, se exporta la función de transferencia del controlador, con las variables del mismo.

Como se puede observar en la siguiente imagen, la sintonización se hace para que el sistema cumpla unos requerimientos de rapidez y estabilidad, lo que se hace es hacer una compensación para que el sistema sea eficiente para estas dos características.

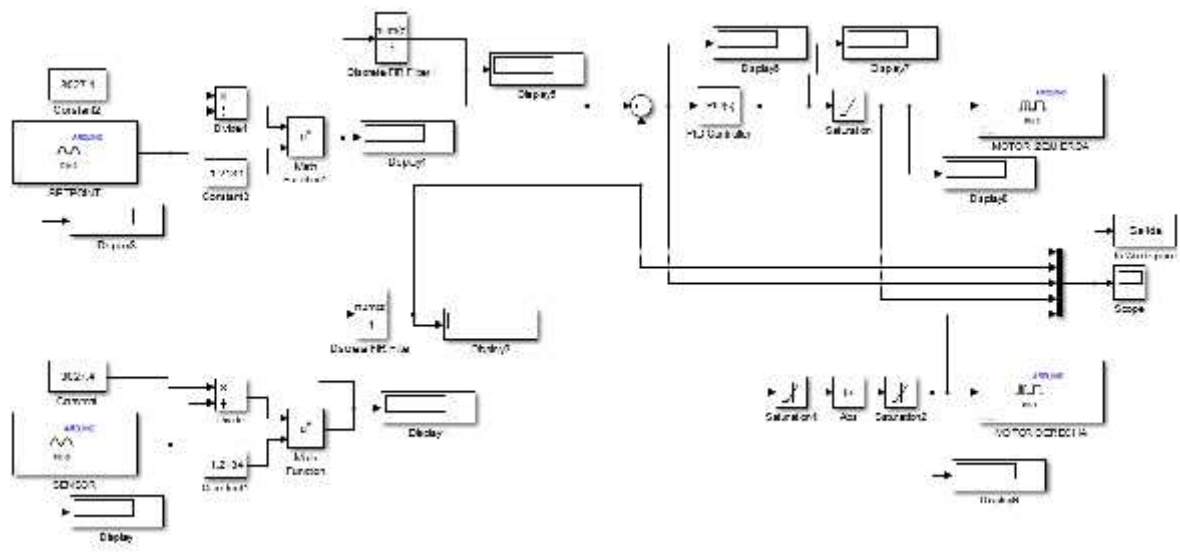


Una vez obtenidas las constantes del controlador y la función de transferencia de la planta, se procede a hacer las pruebas tanto en el código de Simulink como en la interfaz de Arduino.

4. Pruebas con Simulink y Arduino

En el caso de Simulink se proporcionan las constantes obtenidas a la función/bloque PID y se procede a activar el sistema. En nuestro caso la planta se comporta de manera muy eficiente y para comprobar que sea el mismo comportamiento de manera teórica, se simula la planta con la función de transferencia estimada y las constantes del controlador para observar gráficamente la respuesta del sistema.

Se muestra la conformación de la planta en Simulink y se realizan las pruebas en Arduino dándonos como resultado las siguientes graficas



Simulando la planta con Simulink y Arduino obtuvimos los siguientes resultados:



Se pudo concluir las clasificaciones de la anterior grafica estudiando lo que pasaba en la gráfica del osciloscopio en la simulación cuando se movía el potenciómetro que determinaba el set point así pudimos identificar cual era el esfuerzo de control determinado para cada dirección del motor así mismo el error y lo que identificaba el sensor.

Conclusiones.

Después de identificar y sintonizar la planta se pudo obtener unos resultados muy congruentes mediante la simulación Simulink/Arduino, permitiéndonos identificar fácilmente las componentes a estudiar en el proceso, tales como el esfuerzo de control cuando queríamos variar la posición de izquierda a derecha, permitiéndonos identificar dos esfuerzos diferentes de control mostrados en una gráfica anterior, así mismo también se logró asemejar el error y la señal detectada por el sensor.

Como el modelo no presento el 100% de exactitud, se pudo observar que el sistema ya compensado funciona correctamente y con una eficiencia muy buena, respondiendo de manera rápida ante una variación del set point y manteniendo la temperatura, activando la bombilla como el motor para regular de manera eficiente la lectura del sensor.

Para la identificación del modelo de la planta se necesitó tomar valores con intervalos más cortos ya que un sistema de control de temperatura trabaja con tiempos prolongados y por esto es sistema no se estabiliza rápidamente, por lo tanto, al tomar valores en intervalos más cortos podemos generar una función más precisa para que la planta reaccione de manera más rápida.

Las gráficas que se generan al momento de estimar el modelo de la planta, presentan muchas oscilaciones debido a que los instrumentos no son de precisión, como por ejemplo la variable set point que se modifica a través de un potenciómetro, ante cualquier movimiento genera una señal que se ve como un disturbio en la planta.

El sensor genera una buena lectura, es decir con un error bajo, pero ya que es un elemento muy básico también presenta interferencias ante alguna corriente parasita o algún movimiento.