



EXPERIMENTO 8

CONTROL DE UN ROBOT MÓVIL CON PID

PROFESOR: ATOANY FIERRO

UNIDAD DE FORMACIÓN: MR3001B – DISEÑO Y DESARROLLO DE ROBOTS

EXPERIMENTO 8

CONTROL DE UN ROBOT MÓVIL CON PID

OBJETIVO

Implementar y ajustar un controlador PID en un robot móvil mediante simulación en Simulink, con el fin de controlar su posición a 1 metro del punto inicial. Se utilizará un encoder para sensar la posición actual y retroalimentar el sistema, comparándola con la posición deseada para calcular el error. A través de la configuración y ajuste del controlador PID, se analizará el desempeño del sistema y se optimizarán los parámetros para mejorar la estabilidad y precisión del control..

INTRODUCCIÓN

El control de posición en robots móviles es un aspecto fundamental en la robótica y la automatización. Uno de los métodos más utilizados para lograr un control preciso es el uso de un controlador PID (Proporcional-Integral-Derivativo), el cual permite minimizar el error entre la posición deseada y la posición real del robot.

En esta actividad, se diseñará e implementará un controlador PID en un entorno de simulación utilizando Simulink. A través de la retroalimentación proporcionada por un encoder, el sistema ajustará la señal de control para llevar al robot a una posición objetivo de 1 metro. Inicialmente, se configurará un controlador proporcional (P), para posteriormente optimizar los parámetros del controlador PID utilizando la herramienta de ajuste automático (Tune). Finalmente, se aplicará el controlador en la simulación del robot móvil y se analizará su desempeño.

En el ejemplo anterior, utilizamos un controlador de tipo on/off para hacer que el robot se detenga después de haber recorrido 1 metro de distancia. Este sistema de control es deficiente ya que la salida del sistema es diferente a la entrada de referencia. El controlador PID (Proporcional, Integral, Derivativo) nos permitirá hacer que la velocidad del vehículo sea directamente proporcional a la distancia del punto objetivo. Además, la salida del sistema tendrá un menor error comparado con la salida de un sistema on/off.

Un controlador PID se puede visualizar como un sistema cuya entrada es una señal de error y su salida es una señal de control que determinará el funcionamiento de un actuador.



El controlador PID está conformado por tres tipos de controladores, el Proporcional (P), el Integral (I) y el Derivativo (D). A continuación, se explicará la función que tiene cada uno de estos controladores.

Controlador Proporcional (P)

Este controlador se obtiene como $P = k_p \times \text{error}$, donde k_p es la constante de proporcionalidad y **error = señal de referencia – señal medida**. Luego entonces, entre más grande es el error, más grande es la señal de control, y viceversa; lo que ocasiona una convergencia más rápida a la señal de referencia. Sin embargo, se debe tener cuidado que la constante de proporcionalidad no sea lo suficientemente grande como para que el sistema comience a oscilar. Más formalmente, el controlador P se puede calcular como sigue:

$$P = k_p e(t) \quad (1)$$

En donde k_p es la constante de proporcionalidad y $e(t)$ es la señal de error en función del tiempo.

Controlador Integral (I)

El controlador P reduce la señal de error de sistema, sin embargo, tiene dificultades para reducirla a un valor más cercano a cero en el estado estable, ya que es en este estado donde regularmente el error es muy pequeño matemáticamente, por lo que la acción de control P es casi nula. Para ello, se utiliza el controlador I, el cual suma todos los errores que ha tenido el sistema a través del tiempo. En este caso, no importa qué tan pequeño sea el error del sistema, ya que, al irse sumando en cada instante de tiempo, el error llega a ser lo suficientemente grande como para hacer que se pueda ajustar la salida del controlador. El controlador I se puede obtener como sigue:

$$I = k_i \int e(t) dt \quad (2)$$

En donde k_i es la constante del controlador integral y $\int e(t)dt$ es el área bajo de la curva del error, o, mejor dicho, la sumatoria de los errores a través del tiempo.

Controlador Derivativo (D)

El controlador Derivativo permite ajustar la señal de control de acuerdo a la velocidad con que varía el error, es decir, la tasa de cambio del error. Si la tasa de cambio es grande, la señal de control D es grande, y viceversa. El controlador D se puede obtener como sigue:

$$D = k_d \frac{d}{dt} e(t) \quad (3)$$

En donde k_d es la constante de proporcionalidad derivativa y $\frac{d}{dt} e(t)$ la velocidad con que cambia el error. Si se utiliza un valor grande para k_d hará que el sistema de control reaccione con más fuerza a los cambios del error y aumentará la velocidad de respuesta del sistema de control en general. Por lo regular se utiliza un valor pequeño de k_d ya que la respuesta de la derivada es muy sensible al ruido.

Para tener un controlador PID, se tienen que sumar las señales de control P, I y D como sigue:

$$PID = k_p e(t) + k_i \int e(t)dt + k_d \frac{d}{dt} e(t) \quad (4)$$

Sistemas de Control en Lazo Abierto y Lazo Cerrado

En un sistema de control de lazo abierto, el sistema no utiliza ninguna señal de retroalimentación que le indique como mover el vehículo. En un sistema de control retroalimentado en lazo cerrado, la salida del sistema se compara con una señal de referencia para poder desplazar el vehículo de acuerdo con la señal de error que se obtenga. Por ejemplo, para un sistema de vehículo autónomo, un control en lazo abierto es cuando la computadora acciona los motores de acuerdo con instrucciones ya establecidas. Como se ve en la Figura 14, la señal de referencia es de 50 km/h, la computadora ejecuta las instrucciones que previamente ya fueron establecidas para accionar el pedal y así alcanzar los 50 km/h. El sistema no tiene ningún sensor que le permite ver en el tacómetro la velocidad a la que está viajando el vehículo.

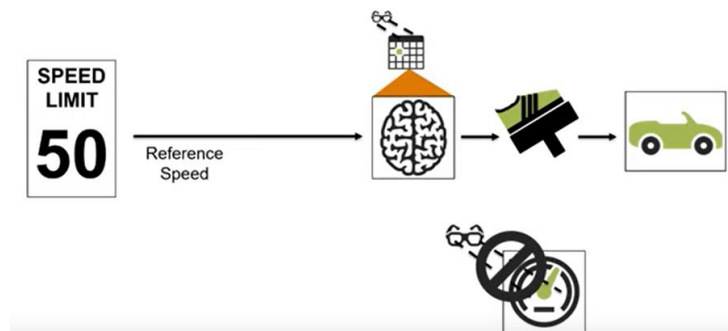


Fig. 1. En un sistema de control de lazo abierto, no hay sensores que le permitan al sistema comparar su salida con la señal de referencia. La señal de referencia es de 50 km/h; la computadora ejecuta instrucciones previamente cargadas para tratar de mantener el vehículo a una velocidad constante.

Los sistemas de control en lazo abierto son simples y generalmente estables, sin embargo, no son precisos y no son robustos frente a perturbaciones.

Por otro lado, un sistema de control en lazo cerrado cuenta con sensores los cuales retro alimentan la salida del sistema para poder compararla con la señal de referencia, calcular el error y con base a ello, el controlador trata de mantener una señal del sistema muy similar a la señal de referencia. En la Figura 15 podemos observar un ejemplo de navegación autónoma utilizando un sistema de control en lazo cerrado. La señal de referencia está definida como 50 km/h a la cual se le resta la salida real del sistema, es decir, la velocidad a la

que se está desplazando, obteniéndose así la señal de error. Esta señal de error es procesada por el sistema de control, que, con base a ella, se toman decisiones de accionar el pedal de acelerador o no para mantener una velocidad constante de 50 km/h.

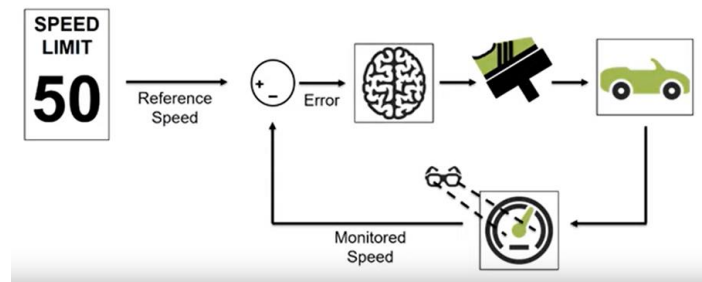


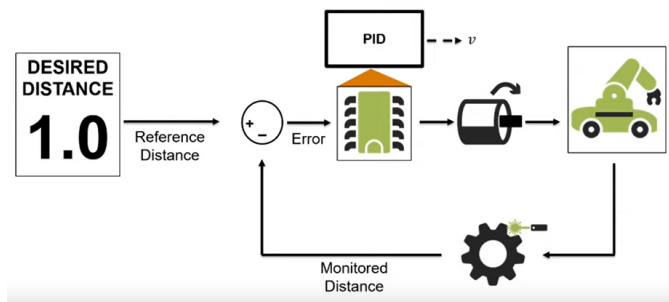
Fig. 2. En un sistema de control de lazo cerrado, un sensor retro alimenta la salida real del sistema con la señal de referencia. La señal de error es la resta de la señal de referencia menos la señal real. El controlador procesa la señal de error para que, de acuerdo a su magnitud, se accione o no el pedal del acelerador.

MATERIAL Y EQUIPO

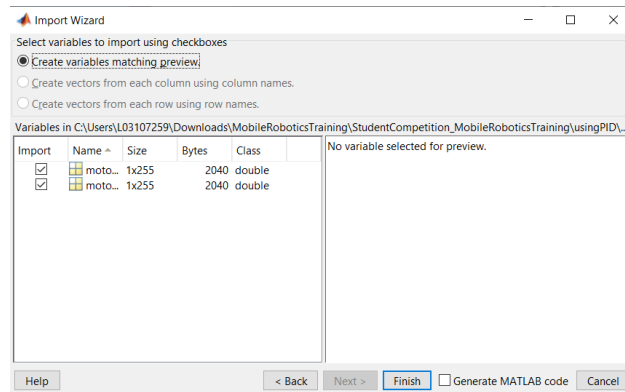
- Computadora con capacidad suficiente para ejecutar MATLAB/Simulink sin problemas
- MATLAB y Simulink instalados
- Mobile Robotics Training Library para Simulink.
- Acceso a internet para consulta de documentación o resolución de duda

INSTRUCCIONES

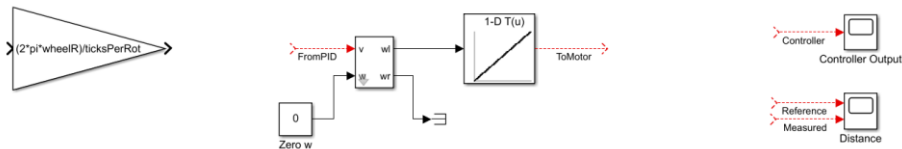
En nuestro caso realizaremos una simulación de un pequeño robot móvil en donde el controlador PID controlará su posición por medio de un motor. Se utilizará un encoder para sensar la posición actual, se retroalimentará al sistema y se comparará con la posición deseada (1 metro del punto inicial) para obtener el error del sistema, como se muestra en la Figura 3.



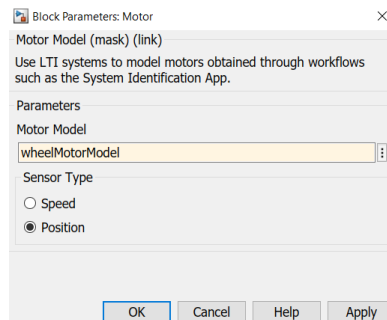
Descargar y abre el archivo llamado [wheelMotorLUT.mat](#) y cargar las variables `motorIp` y `motorSp`, como se muestra a continuación.



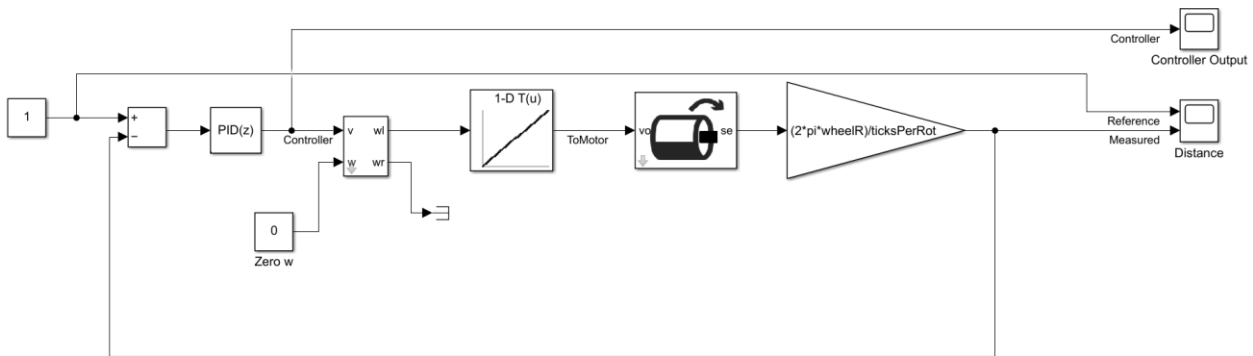
Descarga y abre el código con nombre [motorDistCtrl_start.slx](#). Aparecerá un archivo simulink como se muestra a continuación:



Guardar el archivo como `motorDistCtrl.slx`. Agregar un bloque *constante* el cual se encuentra en la categoría *sources*. Este bloque constante será nuestra señal de referencia (1 metro). Posteriormente, agregar el bloque *motor* que se encuentra en la categoría *Mobile Robotics Training Library*. Este bloque representará las propiedades físicas que la planta que queremos controlar. Dichas propiedades se encuentran en un archivo llamado [wheelMotorModel.mat](#) el cual tendrás que descargar y cargar la variable `wheelMotorModel`. Ahora dar bloque clic al bloque *motor* de la librería *Mobile Robotics Training Library*. Configura sus parámetros como sigue:

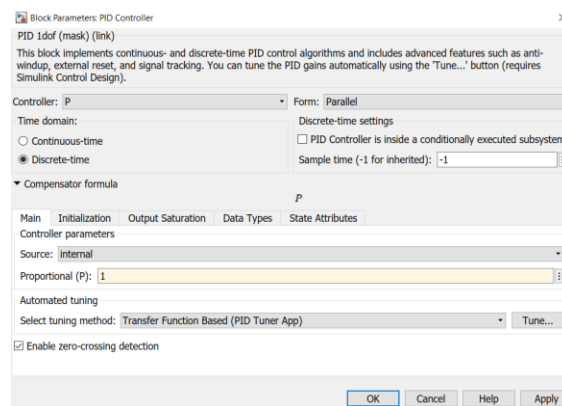


Agregar un bloque *Subtract* que se encuentra en la categoría *Math Operations*. A continuación, agregar un bloque *PID* que se encuentra en la librería *Mobile Robotics Training Library*. Conectar los bloques de la siguiente manera:



Copyright 2017 The MathWorks, Inc.

Ahora, configuraremos el bloque PID dándolo doble clic al bloque *PID*. En primer lugar, seleccionaremos solamente un controlador P con una ganancia de 1.

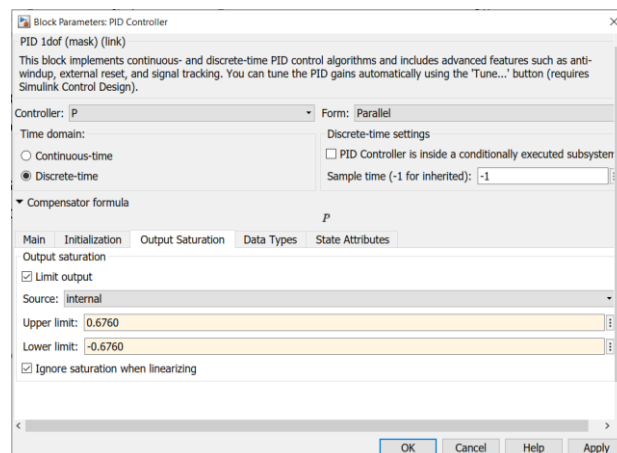


Posteriormente, damos clic en *Output Saturation* para configurar las propiedades físicas del motor. Supongamos que se utilizará un motor cuya velocidad angular máxima es 13 radianes por segundo:

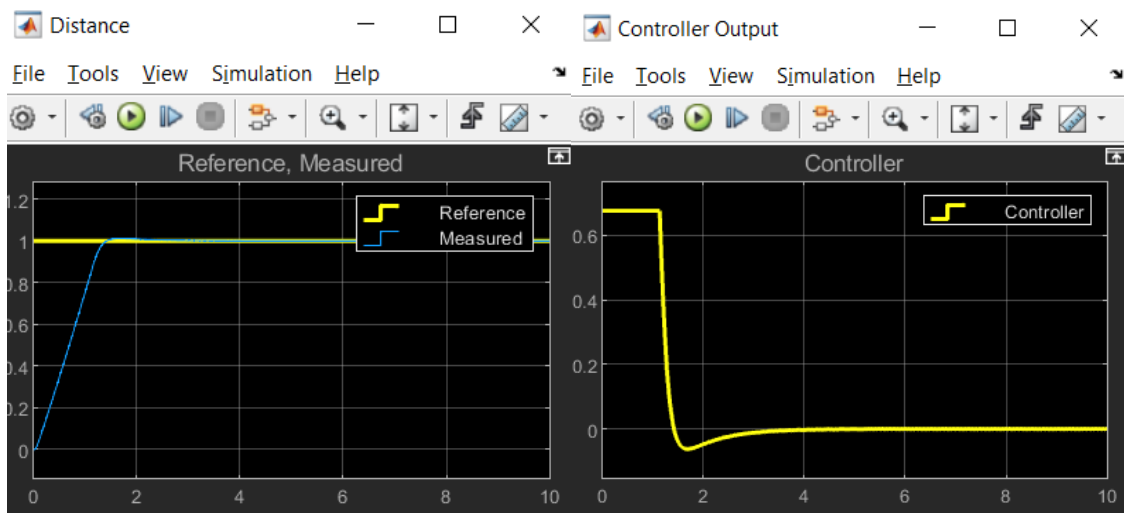
$$v_{max} = \text{wheel radius} \times 13$$

$$v_{max} = 0.052 \times 13 = 0.6760$$

Este valor de velocidad máxima se configurará como sigue:



Simular el sistema y visualizar los scopes *Controller Output* y *Distance*. La simulación debería entregar el siguiente resultado:



En donde se observa de la salida *Distance* la señal de control ocasiona que la señal real alcance de manera rápida la señal de referencia.

Ejercicio 1

1. ¿Qué parámetro tienes que modificar del controlador para que la respuesta del sistema alcance el nivel de referencia en un menor tiempo?
2. Configura la ganancia P en 50
3. ¿Qué es lo que le pasa al sistema? Argumenta tu respuesta

Ahora copia el bloque de *subtract* y *PID Controller* y pégalo en el área de trabajo del archivo *deadReckoningPlant*. Guarda el archivo como (*deadReckoningDistanceControll*)

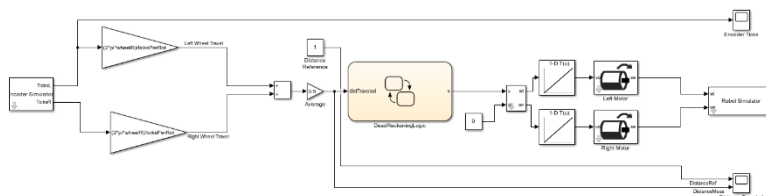


Fig. 3. Área de trabajo del archivo *deadReckoningPlant*

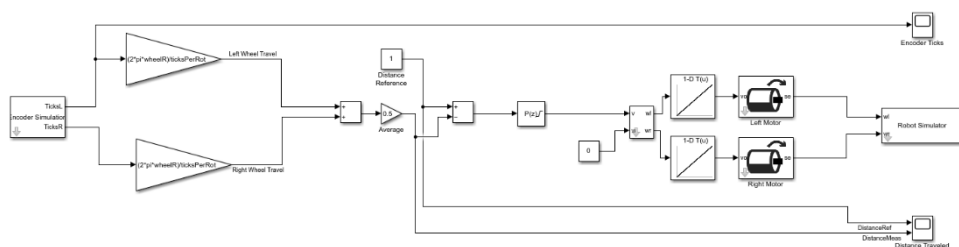
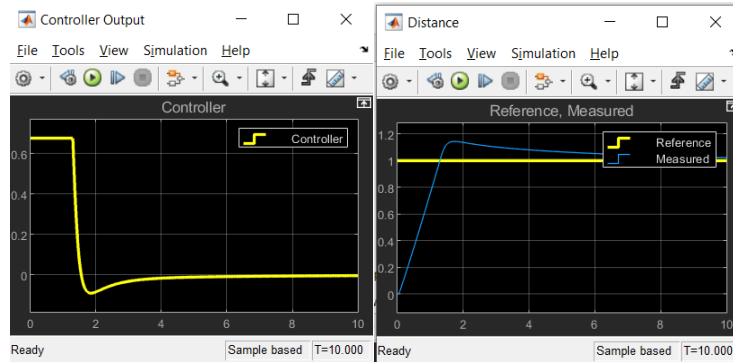


Fig. 4. Área de trabajo del archivo *deadReckoningDistanceControll* modificado (bloque *subtract* y *PID* implementados)

Ejercicio 2

Vuelve a tu simulador (archivo *motorDistCtrl*) e implementa un controlador PI con una ganancia $P = 5$ y una ganancia $I = 1$. Deberás de obtener las siguientes gráficas del controlador y de distancia.



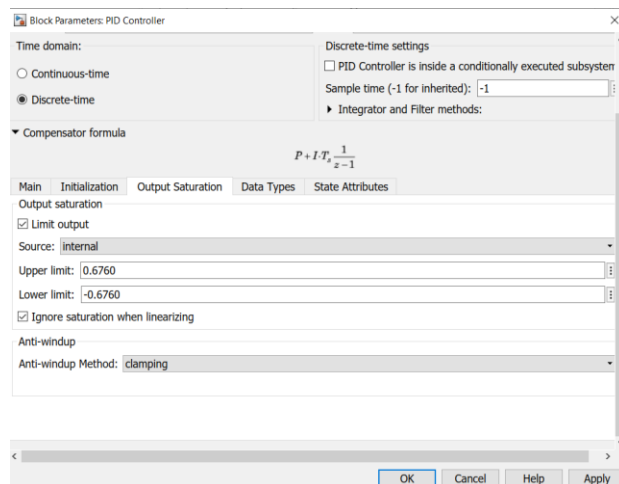
1. De acuerdo con las gráficas obtenidas, ¿qué ventajas presenta el sistema?
2. De acuerdo con las gráficas obtenidas, ¿qué desventajas presenta el sistema?

Ejercicio 3

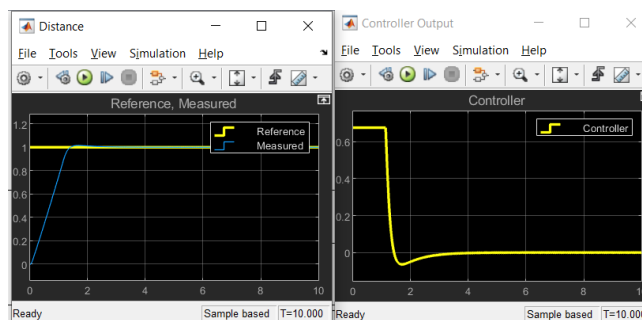
Implementa el controlador PI simulado en el archivo deadReckoningPlant, observa el comportamiento del robot y escribe tus observaciones.

Ejercicio 4

Agrega el método anti-acumulación “clamping” para evitar el sobre impulso de la respuesta de la simulación. Observa el comportamiento de la simulación y escribe tus observaciones

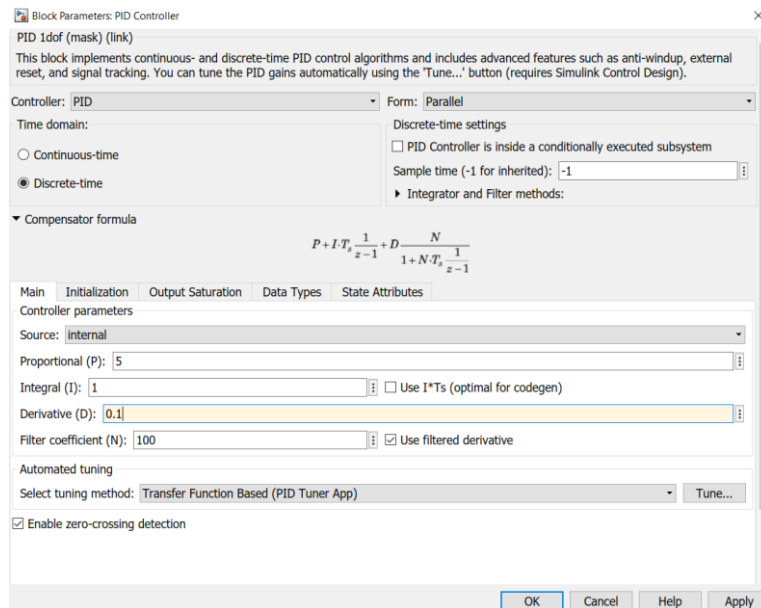


Obtendrás las siguientes gráficas:



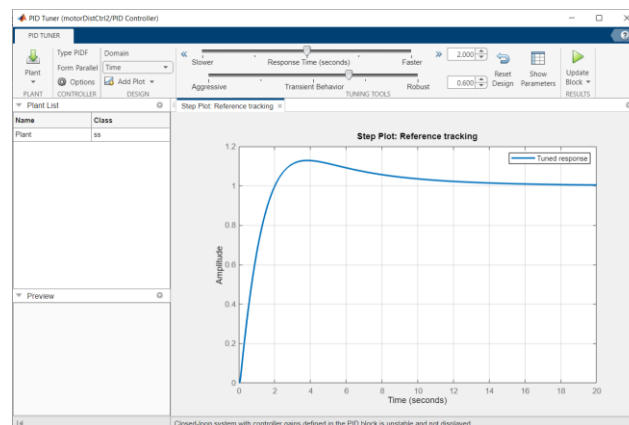
Ejercicio 5

Implementa un controlador PID como se muestra a continuación y obtén su respuesta en la simulación. Observa los resultados y escribe tus observaciones.

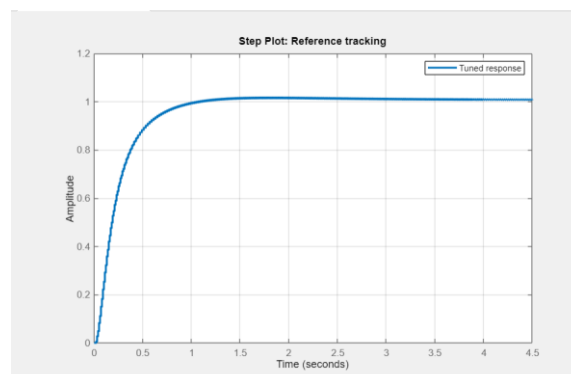


Ejercicio 6

Realiza una afinación a tu controlador PID seleccionando en la sección *Automated tuning*, desplegar la lista *Select tuning method* y seleccionar *Transfer Function Based (PID Tuner App)*, dar clic en *Apply* y en *Tune*. Aparecerá la siguiente ventana:



Modifica los controles *Response Time* y *Transient Behavior* hasta que obtengas una señal sin sobre impulso, sin oscilaciones y que tenga un tiempo de respuesta corto, como la siguiente figura:



Cuando hayas encontrado los parámetros correctos, da clic en *Update Block*. Regresa a tu ventana de parámetros de PID y observarás que se han cargado las nuevas constantes del controlador. Obtén las gráficas de *Distancia* y del *Controlador*.

Ejercicio 7

Implementa el PID afinado en el archivo *deadReckningPlant*. Observa el comportamiento y escribe tus observaciones.

ENTREGABLES

- Archivo de simulación: El archivo .slx de Simulink con la implementación del controlador PID (25%)
- Video de la simulación (individual): Incluir una explicación breve del proceso y resultados obtenidos (50%)
- Reporte en PDF: Presentar los resultados de los 7 ejercicios realizados en la simulación, incluyendo análisis y conclusiones (25%)

RÚBRICA DE EVALUACIÓN

La simulación se evaluará con la siguiente rúbrica de evaluación

Criterio	Destacado 95-100	Sólido 85-94	Básico 75-84	Incipiente 0-74
Presentación de gráficas	Se presentan ambas gráficas (distancia recorrida y señal del controlador) con datos claros y bien organizados. La gráfica de distancia recorrida no muestra sobreimpulsos ni oscilaciones en el estado estable.	Se presentan ambas gráficas, pero la gráfica de distancia recorrida tiene ligeros sobreimpulsos u oscilaciones mínimas.	Se presentan las gráficas, pero la de distancia recorrida muestra sobreimpulsos o oscilaciones significativas, afectando la estabilidad del sistema.	No se presentan las gráficas o la información es incorrecta e insuficiente para evaluar el desempeño del sistema
Implementación del controlador PID	Las constantes del controlador PID están correctamente implementadas en la simulación y el robot se desplaza exactamente 1 metro.	Las constantes del controlador PID están implementadas, pero el robot presenta un desplazamiento cercano a 1 metro con pequeñas desviaciones.	Las constantes del PID están implementadas, pero el robot se desplaza incorrectamente, mostrando errores notables en la distancia recorrida.	No se implementan correctamente las constantes del PID o el robot no se desplaza de manera adecuada.
Entrega de archivos (.slx)	Se entregan correctamente los archivos <i>deadReckoningPlant.slx</i> y <i>motorDistCtrl.slx</i> con	Se entregan los archivos, pero con detalles menores en la implementación que no afectan	Se entregan los archivos, pero contienen errores significativos que afectan la simulación.	No se entregan los archivos o estos están incompletos y no permiten evaluar la simulación.

	Destacado	Sólido	Básico	Incipiente
Criterio	95-100	85-94	75-84	0-74
	la implementación completa y funcional.	gravemente la simulación.		

El video individual se evaluará con la siguiente rúbrica de evaluación

	Destacado	Sólido	Básico	Incipiente
Criterio	95-100	85-94	75-84	0-74
Claridad en la explicación	La explicación es clara, bien organizada y demuestra un dominio profundo del tema.	Se explica correctamente el procedimiento con una estructura clara.	Se explica el procedimiento, pero con omisiones o falta de claridad en algunos puntos.	La explicación es confusa o incompleta, sin describir correctamente el procedimiento.
Demostración de la simulación	La simulación se presenta sin errores y con una excelente explicación de cada paso y resultado.	Se muestra la simulación correctamente y con una explicación adecuada de los resultados.	Se muestra la simulación, pero con errores o falta de detalles clave.	No se muestra la ejecución de la simulación o el video no es funcional.
Análisis de resultados	Se presenta un análisis detallado, reflexionando sobre el comportamiento del robot y posibles mejoras.	Se analizan los resultados correctamente con observaciones adecuadas.	Se menciona el comportamiento del robot, pero sin análisis profundo o con errores.	No se analiza el comportamiento del robot o la información es incorrecta.
Presentación y lenguaje	La presentación es profesional, con excelente comunicación y uso adecuado del lenguaje técnico.	La presentación es clara y con buen uso de lenguaje técnico.	La presentación es aceptable, pero con deficiencias en fluidez o lenguaje técnico.	El video es poco profesional, con pausas excesivas, lenguaje inadecuado o desordenado.
Originalidad y autenticidad	La explicación es completamente individual, con un enfoque personal y evidencia de comprensión única.	La explicación es original, con algunas similitudes esperadas en términos técnicos.	La explicación es similar a la de otros estudiantes, con diferencias mínimas.	La explicación es idéntica a la de otros estudiantes o hay evidencia de plagio.

El reporte se evaluará con la siguiente rúbrica de evaluación

Criterio	Destacado 95-100	Sólido 85-94	Básico 75-84	Incipiente 0-74
Presentación de los resultados de los 7 ejercicios	Se presentan correctamente los resultados de los 7 ejercicios con tablas, gráficas y explicaciones detalladas. Cada resultado está bien fundamentado.	Se presentan los resultados de los 7 ejercicios, pero con explicaciones o análisis que podrían ser más detallados.	Se presentan algunos resultados incompletos o con explicaciones superficiales que afectan la comprensión.	No se presentan todos los ejercicios o la información es insuficiente para evaluar el desempeño.
Calidad del análisis y conclusiones	El análisis es profundo y bien argumentado. Se identifican patrones, errores y mejoras. Las conclusiones reflejan un buen entendimiento de la simulación.	El análisis es adecuado, aunque podría ser más detallado. Las conclusiones son correctas, pero carecen de profundidad.	El análisis es superficial o tiene errores de interpretación. Las conclusiones no reflejan completamente los resultados obtenidos.	No se presenta un análisis claro y las conclusiones son vagas o incorrectas.
Claridad y calidad de redacción	La redacción es clara, sin errores ortográficos o gramaticales. Se utilizan términos técnicos correctamente.	La redacción es comprensible, aunque con algunos errores menores. El uso de términos técnicos es adecuado.	Hay varios errores ortográficos o gramaticales. La redacción no es clara en algunas partes.	La redacción es deficiente, con numerosos errores ortográficos o gramaticales, dificultando la comprensión.
Uso adecuado de figuras, gráficas y referencias	Se incluyen figuras y gráficas bien etiquetadas y explicadas. Se utilizan referencias adecuadas en formato correcto.	Se incluyen figuras y gráficas, pero algunas no están bien explicadas o etiquetadas. Las referencias están en su mayoría correctas.	Faltan algunas figuras o gráficas importantes. Las referencias tienen errores de formato.	No se incluyen figuras o gráficas relevantes. No hay referencias o están incorrectamente citadas.