

Reto Final - Manchester Robotics

DANIELA BERENICE HERNÁNDEZ DE VICENTE, ALEJANDRO ARMENTA ARELLANO,
DANA MARIAN RIVERA OROPEZA

¹Instituto Tecnológico de Estudios Superiores de Monterrey, Campus Puebla, Puebla, México

*Autores correspondientes: a01735346@itesm.mx, A00830027@itesm.mx, a01734879@tec.mx

Publicado el 14/05/2023

El siguiente reporte contiene una investigación acerca del significado y funcionamiento de los siguientes conceptos: Control en lazo cerrado para un robot móvil, PID aplicado a un robot móvil diferencial y los temas derivados al cálculo del error, robustez de un controlador, procesamiento de imágenes en una tarjeta embebida, interconexión entre Jetson y la cámara, detección de contornos o formas en una imagen, detección de colores, robustez en sistemas de procesamiento de imágenes. De igual forma, se realizarán diferentes códigos los cuales buscarán controlar desde la tarjeta embebida al robot móvil (**Puzzlebot**) incluyendo su funcionalidad de procesamiento de imágenes, detección de colores y bordes.

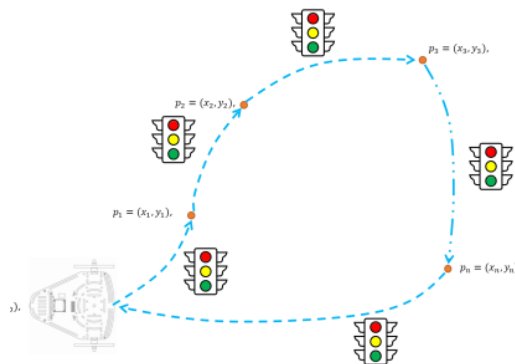


Diagrama 1.

El diagrama 1 hace referencia al objetivo final del reto presentado por MCR2, el cual se trabajó durante las últimas dos semanas de este segundo periodo académico.

1. Introducción.

A. Introducción .

A lo largo de la historia la evolución de la tecnología ha sido inevitable, y con ella han surgido muchos retos relacionados con la misma.

Uno de ellos es el manejo de robots junto con el procesamiento de imágenes, ya que la robótica tiene una naturaleza multidisciplinaria, esto presenta retos para las personas que se dedican a la rama de la robótica, ya que deben poder desenvolverse en diferentes sectores, ya sea con características individuales o con múltiples de ellas, como por ejemplo Móvil (Individual) o Móvil con procesamiento de imágenes (Múltiple). Una vez superados estos retos vienen las siguientes partes, como por ejemplo el modelado del robot, la implementación de sistemas de control para reducir los errores y en un futuro la generación de robots completamente autónomos o de líneas de producción más eficientes para la optimización de procesos.

Ahora bien, durante este proyecto uno de los factores fundamentales para la resolución del reto semanal es la implementación de controladores para el ajuste del voltaje enviado al motor, para que el robot móvil sea capaz de avanzar y/o moverse en la dirección deseada con una cantidad muy pequeña de error.

Mientras que por otro lado, otro factor fundamental fue el procesamiento de imágenes mediante la cámara integrada en el robot móvil proporcionado, así como el manejo del puzzlebot y la plataforma de simulación desde un nivel de dominio muy bajo.

2. Objetivos.

A. Objetivos Generales.

A continuación se presentan los objetivos, donde se describe la función general con la que debe cumplir cada uno de los componentes de nuestro sistema.

1. Se deberá agregar una capa de toma de decisiones al algoritmo de navegación punto a punto previamente desarrollado, esto con la finalidad de la detección del color de un "semáforo".
2. El comportamiento esperado ante la primera instrucción es el siguiente:
 - a. Luz roja: Detenerse hasta que la luz verde sea vista.

- b. Luz amarilla: Avanzar lentamente hasta que la luz roja sea vista para posteriormente detenerse.
 - c. Luz Verde: Continuar con el camino previsto.
3. Como punto adicional, el robot móvil debe permanecer detenido hasta que la luz verde sea vista, aun incluso cuando la luz roja desaparezca o este no sea capaz de detectarla.
4. El algoritmo de visión y el controlador de lazo cerrado debe ser robusto, por lo que el estudiante debe definir el significado de robusto e implementar estrategias para que esté de un buen resultado.
5. El algoritmo de visión y el controlador deben estar sintonizados de manera apropiada.
6. El controlador debe tomar en consideración la perturbación, las no linealidades y el ruido.
7. Se alienta, pero no se requiere, al estudiante a utilizar un archivo de configuración o un parámetro en el archivo de lanzamiento para establecer los objetivos de modo que se pueden cambiar fuera del código (no codificado).

3. MARCO TEÓRICO

A. Control en lazo cerrado para un robot móvil.

Un robot móvil se presenta como un control de lazo cerrado, el cual puede realizar un cambio de posición de sus elementos, esto en función de la información captada de su entorno mediante sus sensores. [1]

Es importante reconocer que es casi imposible sintonizar el controlador para que este genere una trayectoria perfecta hacia la posición deseada en el robot.

Ante esta situación también es importante reconocer que al no ser exactamente perfecto, se busca generar una tasa de error muy baja. [2]

De igual manera es importante destacar que este es un sistema que utiliza fundamentalmente la retroalimentación para mantener una variable controlada, al contar con la medición de dicha variable, esta se compara con el valor de referencia dado anteriormente, y cualquier diferencia entre estos dos valores es utilizada para ajustar el controlador empleado.

B. PID aplicado a un robot móvil diferencial y los temas derivados al cálculo del error.

Como anteriormente se mencionó la programación en ROS no fue lo más importante durante este proyecto, aunque sí es la parte que predomina en este proyecto, ya que la gran parte del proyecto es la creación de los nodos y la comunicación entre ellos, pero el sistema implementado no podría funcionar correctamente sin un controlador que satisfaga las necesidades del mismo.

Por lo que para poder implementar el controlador adecuado para el sistema, fue necesario comprender los tipos de controladores existentes y en qué momento es necesario utilizarlos, al igual que la función de cada una de las variables de los mismos, de igual manera es importante conocer que el diseño del sistema de control consistirá en elegir adecuadamente la localización de los ceros, polos y la ganancia

K, por otro lado gracias al controlador podemos llegar a tener cierto tiempo muerto, por lo que es importante conocer el significado de dicho concepto [3].

Ganancia de Proceso(K). Es la relación de cambio de salida al cambio de la variable de entrada. Esta define más específicamente la sensibilidad que tiene la variable de salida a un cambio dado en la variable de entrada. Dicho comportamiento se muestra matemáticamente en la ecuación 1 [3].

$$K = \frac{\Delta \text{Output}}{\Delta \text{input}}$$

Ecuación 1.

Esta solo puede describirse como un parámetro de estado estacionario y es independiente de las variables de diseño y operación. Cuenta con solo tres componentes, el signo, el valor y las unidades.

El primero indica cómo responderá la salida a la entrada del proceso. El segundo depende del proceso que se esté considerando, por lo que dependen de las variables que se mencionan.

Tiempo Muerto (t₀). Sucede cuando se da el cambio en una variable de entrada y cuando inicia la variable de salida. Cabe mencionar que es muy importante, porque afecta a la controlabilidad de nuestro sistema de control, por lo que un cambio en el punto de ajuste no suele ser inmediato gracias a este parámetro. De igual manera, este tiempo muerto debe ser siempre considerado para los procesos de afinación y modelado [3].

Control Proporcional. Es una forma de control de retroalimentación, siendo la forma más fácil de control continuo que se puede utilizar en un sistema de lazo cerrado. También minimiza la fluctuación en la variable de proceso, el lado negativo es que no siempre lleva al sistema al punto de ajuste que se desea [3].

Por otro lado proporciona una respuesta más rápida que la mayoría de los otros controladores, lo que permite inicialmente que el controlador solo P llegue a responder unos segundos más rápido. Aunque el controlador P ofrece como ventaja un tiempo de respuesta más rápido, llega a producir una desviación del punto de ajuste, la cual se conoce como desplazamiento, y es algo que no se desea durante un proceso.

La existencia de ella implica que el sistema no podría mantenerse en el punto de ajuste deseado en estado estacionario. Este se puede minimizar combinando el control P con algún otro controlador (I o D)

Por lo que el controlador P correlaciona la salida del controlador con el error. Dicho comportamiento se muestra matemáticamente en la ecuación 2.

$$c(t) = K_e e(t) + b$$

Ecuación 2.

Control Integral. Es una segunda forma de control de retroalimentación. Usualmente se utiliza porque es capaz de eliminar cualquier desviación que pueda existir, de esta manera

el sistema vuelve tanto al estado estacionario como a su configuración original [3].

Por lo tanto, un error negativo hará que la señal al sistema disminuya, mientras que un error positivo hará que la señal aumente. Sin embargo este tipo de controladores suelen ser más lentos en su tiempo de respuesta que los controladores P solo porque dependen de más parámetros. Este tiempo de respuesta se puede aumentar si es combinado con algún otro controlador (P o D). De igual manera suelen usarse por separado cuando las variables medidas deben permanecer dentro de un rango muy estrecho y requieren un ajuste más fino.

Estos suelen afectar al sistema al responder a errores pasados acumulados, por lo que su filosofía es que las desviaciones se verán afectadas en proporción a la suma acumulada de su magnitud. Ahora bien, una gran ventaja de estos es que se eliminara el desplazamiento, mientras que sus grandes desventajas son que puede llegar a desestabilizar el controlador, y que hay una windup del integrador, lo cual aumenta el tiempo que tarda el controlador en realizar ciertos cambios.

El control I correlaciona la salida del controlador con la integral del error, por lo que la integral del error se toma con respecto al tiempo. Es el error total asociado a lo largo de una cantidad de tiempo especificada. Dicho comportamiento se muestra matemáticamente en la ecuación 3.

$$c(t) = \frac{1}{T_i} \int e(t)dt + c(t_0)$$

Ecuación 3.

Control Derivativo. Es una forma de control de avance, ya que este anticipa las condiciones del proceso analizando el cambio en el error. Este funciona para minimizar el cambio de error, de esta forma mantiene el sistema en una configuración consistente. Uno de sus principales beneficios es que resiste el cambio en el sistema, siendo el más importante de estas oscilaciones. Ahora bien, la salida del control se calcula en base a la tasa de cambio del error con el tiempo, por lo que mientras más grande sea la tasa de cambio de error, más pronunciada será la respuesta del controlador [3].

A diferencia de los otros controladores, el derivativo no guía al sistema a un estado estacionario, ahora bien, debido a esto se recomienda que deben estar combinados con los controladores P, I o PI para poder controlar de manera correcta el sistema.

El control D correlaciona la salida del controlador con la derivada del error. La derivada del error se toma con respecto al tiempo. Dicho comportamiento se muestra matemáticamente en la ecuación 5.

$$c(t) = T_d \frac{de}{dt}$$

Ecuación 5.

Control PI. El control PI es una forma de control de retroalimentación. Este proporciona un tiempo de respuesta más rápido que el control I debido a la suma de la acción proporcional, aunque sigue siendo hasta un 50% más lento que el control P. Por lo que con el fin de aumentar el tiempo de

respuesta, el control PI suele combinarse con el control D. De igual manera este impide que el sistema fluctúe, y es capaz de devolver el sistema a su punto de ajuste [2].

Como se ha mencionado el control PI correlaciona la salida del controlador con el error y la integral del error. Dicho comportamiento se muestra matemáticamente en la ecuación 4.

$$c(t) = K_c \left(e(t) + \frac{1}{T_i} \int e(t)dt + T_d \frac{de}{dt} \right) + C$$

Ecuación 4.

Control PD. El control PD es una combinación entre control de avance y retroalimentación, debido a que opera tanto en las condiciones actuales del proceso como en las condiciones del proceso predichas. En este controlador, su salida es una combinación lineal de la señal de error y su derivada. De igual manera contiene la amortiguación del control proporcional de la fluctuación y la predicción del error del control derivado [2].

Como se ha mencionado el control PD correlaciona la salida del controlador con el error y la derivada del error. Dicho comportamiento se muestra matemáticamente en la ecuación 5.

$$c(t) = K_c \left(e(t) + T_d \frac{de}{dt} \right) + C$$

Ecuación 5.

Control PID. Este es el más utilizado, ya que combina las ventajas de cada tipo de control. Por lo tanto tenemos un tiempo de respuesta más rápido debido al control P, junto con el desplazamiento decrecido/cero de los controladores D e I. Es importante destacar que el desplazamiento anteriormente mencionado se elimina mediante el uso adicional del control I [2].

La adición del controlador D aumenta en gran medida la respuesta del controlador cuando se usa en combinación con los demás, ya que predice la respuesta del sistema midiendo el cambio en el error. Sin embargo, aunque el controlador PID parece ser el controlador más adecuado, también es el más caro. Por lo tanto, realmente no se utiliza a menos que el sistema necesite esa precisión y estabilidad [3].

Por lo que el controlador PID correlaciona la salida del controlador con el error, siendo estos el integral del error y derivada del error. Dicho comportamiento se muestra matemáticamente en la ecuación 6.

$$c(t) = K_c \left(e(t) + \frac{1}{T_i} \int e(t)dt + T_d \frac{de}{dt} \right) + C$$

Ecuación 6.

C. Robustez de un controlador.

Esta es la capacidad de un controlador para mantener un buen rendimiento y función cuando se presentan ciertas alteraciones y/o incertidumbres ante el sistema controlado. [4]

De igual manera es capaz de manejar ciertas perturbaciones, no linealidades, variaciones, entre otros factores en las condiciones de operación, los cuales podrían afectar el comportamiento del sistema controlado y que en algunas ocasiones puedan pasar desapercibidos.

D. Procesamiento de imágenes en una tarjeta embebida

Por otro lado también durante este reto se maneja el uso de tarjetas embebidas como lo es la tarjeta Jetson, esto con la finalidad de lograr un buen procesamiento de imágenes, siendo esta una parte fundamental del proyecto.

Aunado a esto se busca realizar un procesamiento de imágenes en tiempo real, esto es posible gracias a los algoritmos empleados los cuales han sido optimizados, con la finalidad de obtener el mejor resultado posible.

Cabe destacar que al ser un proyecto de bajo recurso este contiene técnicas de programación de bajo nivel, aunque esto no define el producto final entregado.

E. Interconexión entre Jetson y la cámara

Siendo esta la conexión entre la tarjeta NVIDIA y la cámara que el puzzlebot posee, esta es utilizada para capturar imágenes mediante una aplicación de visión por computadora.

Lo cual se logra a través del uso de diferentes puertos de entrada y salida, los cuales son de alta velocidad junto con la implementación de ciertos protocolos de comunicación, haciendo de este sistema algo óptimo para la resolución del reto. [4]

F. Detección de contornos o formas en una imagen

Ahora bien la detección de contornos sirve para identificar el contorno del círculo correspondiente a cualquier color del semáforo utilizado durante este challenge.

La detección de contorno es parte de un proceso de segmentación, el cual consiste en la identificación de objetos dentro de una imagen o vídeo.

Como es de esperarse, existen varias definiciones de contorno, estas dependen de las distintas circunstancias. [5]

Lo cual se logra a través del uso de diferentes puertos de entrada y salida, los cuales son de alta velocidad junto con la implementación de ciertos protocolos de comunicación, haciendo de este sistema algo óptimo para la resolución del reto.

Uno de los más empleados y que en este caso fue el utilizado durante la resolución del reto, es el análisis de la intensidad de píxeles y la detección de bordes.

G. Detección de colores.

Ahora bien, es cierto que se han detectado los bordes, pero para que se pueda lograr esto primero hay que procesar la imagen para poder así detectar los colores Rojo, Amarillo y Verde, con la finalidad de que el robot móvil pueda observar correctamente y pueda ejecutar los movimientos correspondientes.

Ahora bien, este se realiza mediante HSV (Hue, Saturation, Value / Matiz, Saturación, Brillo), ya que posee 3 componentes, es similar al RGB.

El espacio de color HSV se utiliza ya que es más sencillo determinar los rangos de colores que se desean detectar. Ahora bien, para poder determinar exactamente un color hay que centrarse principalmente en el componente H el cual corresponde al matiz.

Es importante destacar que cada componente tiene un rango de valor específico, por lo que son mencionados a continuación:

- H: 0 a 179.
- S: 0 a 255.
- V: 0 a 255.

Aunado a esto es importante conocer que en el desarrollo del proyecto y principalmente en el código se emplea la librería OpenCV, siendo esta la encargada de todo el tema de procesamiento de imágenes en el lenguaje de programación Python.

Al usar esta librería existen pasos que se deben seguir para poder obtener una detección de colores con HSV.

El primer paso es procesar la imagen o video en tiempo real, con esta obtenida se obtienen los fotogramas con los que se va a trabajar. Posteriormente se transforma de BGR a HSV una vez obtenida dicha transformación se determinan los rangos en donde se encuentra el color a detectar, por último al tener una imagen binaria, en donde el color blanco representa el color detectado, esto según los rangos establecidos, es hora de visualizar el resultado. [6]

H. Robustez en sistemas de procesamiento de imágenes

Esto es posible gracias al diseño de ciertos algoritmos avanzados de procesamiento de imágenes y de algunas técnicas de aprendizaje automático, las cuales pueden adaptarse y ajustarse dependiendo de la situación ya que estas pueden verse alteradas dependiendo de las condiciones del ambiente.

Lo anteriormente mencionado permite que el sistema sea capaz de compensar las variaciones en temas de iluminación, presencia de ruido y ciertas variabilidades en la apariencia del objeto de interés.

4. Marco Metodológico.

Cada uno de los conceptos anteriormente mencionados se han utilizado para la resolución de manera práctica de los diferentes problemas que han surgido durante las 5 semanas del bloque.

A. Funcionamiento del sistema general ROS.

Se generan dos nodos principales, Path generator que se encarga de leer las posiciones que enviamos desde los parámetros, las envía al controlador por medio de un mensaje en string. El segundo nodo Image generator, enciende la cámara y envía el video al controlador, ambos nodos no se conectan entre sí, pero envían la información al nodo Controlador.

El nodo controlador recibe la información enviada en forma de string, lo separa en una cadena de caracteres separada por comas, después lo convierte en enteros y se lee como posiciones del robot, se toma la posición actual y se le resta el error.

B. Funcionamiento del controlador.

Después de convertir el string en cadena de caracteres, esta se publica como la posición ideal, haciendo uso de encoders para

calcular la distancia que avanza considerando el tamaño de las ruedas del robot, se hace una estimación de la distancia a partir de los encoders para la que sería nuestra posición actual que se resta a la posición deseada para obtener el error para el controlador; para adaptar la velocidad según el error por medio de un controlador integral, para el controlador de la distancia se usó un controlador potencial, seleccionándolo a partir de diferentes pruebas, para el controlador angular, por medio de cálculos obtenemos la distancia y el ángulo para llegar al punto deseado, cada que se da un giro de la llanta genera un error en el ángulo que se corrige a partir de una resta entre el valor deseado y el actual, por la necesidad de precisión en el cálculo se implementó un controlador PID, para así obtener valores precisos y evitar la desviación del robot.

Con respecto a las imágenes, al ser recibidas se lee y convierte a HSV volviéndola más ligera, se aplican filtros a la imagen y generamos rangos de color para detectar el estado del semáforo, se aplica una detección de bordes en la que consideramos una figura mayor a once bordes como circular, detectamos el color y se envía una señal de salida indicando el tipo de movimiento que deberá tener el robot a partir de la información obtenida.

C. Procesamiento de imágenes mediante HSV.

Para esto utilizamos la siguiente imagen como referencia.

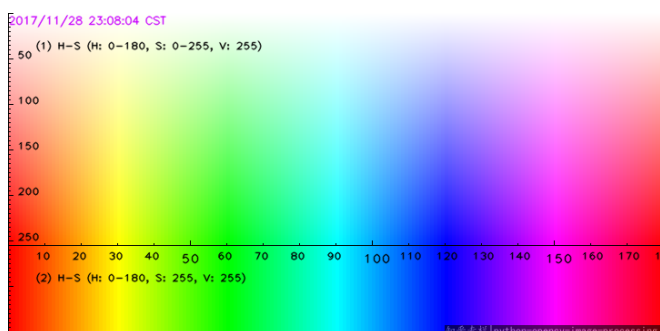


Imagen 1.

En esta se puede observar el rango de colores que se manejan para HSV, donde nosotros aplicamos los siguientes rangos para el color Rojo (179 a 160), Verde (55 a 100) y Amarillo (15 a 45), si bien estos rangos abarcan más del color esperado, se han puesto de esa forma ya que nuestra cámara presentaba modificaciones en su colorimetría.

5. Resultados

Una vez realizada la explicación de cada uno de los análisis y cálculos para obtener los valores específicos del controlador y de la ejecución de cada uno de los códigos para los nodos, se obtuvieron los siguientes resultados.

Como resultado del challenge se identifica de manera correcta el estado del semáforo enviando la señal correcta para el movimiento considerando rangos para el color y manipulando la imagen por medio de filtros y funciones para identificar el círculo del semáforo para reconocer que esa es la señal correcta ya que si detecta el color por sí mismo puede tener un movimiento erróneo.

Por otra parte se genera el control para el movimiento donde se consideran las señales enviadas por los nodos calculando los errores de la posición y ángulos para ajustar la velocidad y la posición de las ruedas.

A. Controlador Final.

Se implementaron dos controladores, el primero para la velocidad requerida para ajustarla según la distancia de la posición actual a la deseada, el segundo para ajustar la posición de las llantas según el ángulo óptimo para llegar al punto deseado y la posición actual de las llantas, se implementaron controladores integrales y PID respectivamente, esto considerando las necesidades de respuesta y precisión.

En donde se tienen los siguientes valores para los controladores.

- $a=0$
- $a2=0$
- $k=0$
- $r=1$
- $z=0$
- $h=0$
- $q=0$
- $n=0$
- $o=0$
- $w2=0$

6. Conclusión.

A pesar de los retos presentados por el challenge, pudimos completarlo de manera satisfactoria, sin embargo se nos presentaron diversas dificultades como el tiempo en el que se envían las imágenes de la cámara del robot ya que si se retrasa el procesamiento de la imagen puede causar movimientos desfasados, de igual manera realizar los controladores y volverlos precisos para tener movimientos exactos, sin embargo logramos sobrellevar estos retos con el apoyo de los profesores.

7. Referencias.

1. Control y Robótica. (s. f.). robotica1. Recuperado 25 de abril de 2023, de [CONTROL Y ROBÓTICA](#)
2. Manchester Robotics [MCR2]. (s. f.). Closed Loop Control_v2. Github. Recuperado 25 de abril de 2023, de https://github.com/ManchesterRoboticsLtd/TE3002B_Intelligent_Robotics_Implementation/blob/main/Week%203/Challenges/MCR2_Mini_Challenge2.pdf
3. Libretexts. (2022, 2 noviembre). 9.2: Control de P, I, D, PI, PD y PID. LibreTexts Español. [https://espanol.libretexts.org/Ingenieria/Ingenier%C3%ADa Industrial y de Sistemas/Libro:_Din%C3%A1mica_y_Contr%C3%B3les_de_Procesos_Qu%C3%ADmicos_\(Woolf\)/09:_Control_proporcional-integral-derivado_\(PID\)/9.02:_Control_de_P,_I,_D,_PI,_PD_y_PID](https://espanol.libretexts.org/Ingenieria/Ingenier%C3%ADa Industrial y de Sistemas/Libro:_Din%C3%A1mica_y_Contr%C3%B3les_de_Procesos_Qu%C3%ADmicos_(Woolf)/09:_Control_proporcional-integral-derivado_(PID)/9.02:_Control_de_P,_I,_D,_PI,_PD_y_PID)
4. Perez, M. A., Perez Hidalgo, A., & Perez Berenguer, E. (2007). INTRODUCCION A LOS SISTEMAS DE CONTROL Y MODELO MATEMÁTICO PARA SISTEMAS LINEALES INVARIANTES EN EL TIEMPO. Universidad Nacional de San Juan. Recuperado 25

de abril de 2023, de [introduccion a los sistemas de control y modelo matemático para sistemas lineales invariantes en el](#)

5. 4. (s. f.). [4. Detección de contorno](#).
6. Administrador. (2020). Detección de colores en OpenCV – Python (En 4 pasos). OMES. [Detección de colores en OpenCV - Python \(En 4 pasos\) » omes-va.com](#)

8. Anexos.

1. Repositorio de GitHub:
https://github.com/Bere901/-Retos_Manchester_Robotics_IRI/tree/main/Week%204