

Informe de Práctica 3

Programación de Robots

David Ávila Jiménez || Pedro Antonio Aguilar Lima

Contenido

Introducción.....2

 Planteamiento 2

 Colocación del robot 2

Contenido3

 Calibración de luz blanca y negra 3

 Definición de constantes..... 4

 Declaración de variables 4

 Main 4

 Controlador PID..... 4

 Actuación con Wind-Up 5

 Ajuste de los motores con saturación 5

Introducción

En este informe se recoge los datos sobre la práctica 3, que trata sobre el manejo de manejo de microbots, en este caso con un controlador PID, complementando al controlador de bucle cerrado de la práctica anterior. El objetivo de esta práctica es profundizar en el uso del programa V-REP PRO además de MATLAB mediante el robot de Lego EV3. Este robot deberá realizar un seguimiento de líneas al mapa de la provincia de Granada, tratando siempre de ir por el borde exterior o borde interior del mapa.

Planteamiento

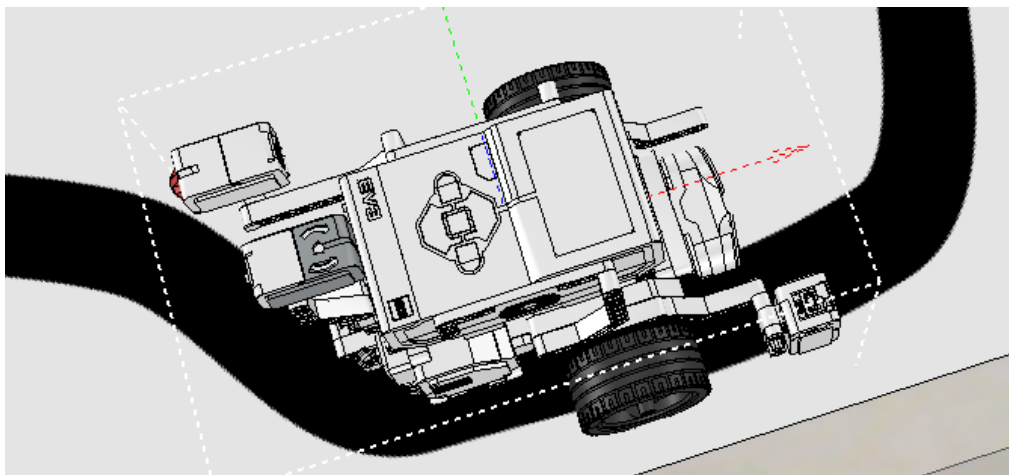
El seguidor de línea que vamos a desarrollar va a seguir en realidad, el borde la línea negra, ya que así será más fácil detectar cuando abandona la línea y por donde se ha salido. El algoritmo debe seguir un esquema que se especifica más abajo y que se hablará de él más en profundidad en el contenido.

El algoritmo general del programa es el siguiente:

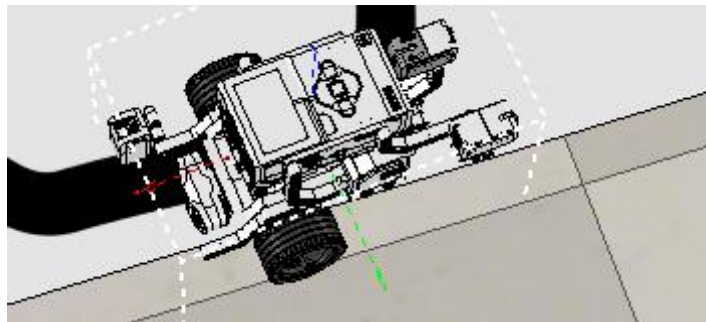
1. Definición de constantes.
2. Main.
 - 2.1. Declaración de variables.
 - 2.2. Calibración de la luz del sensor.
 - 2.3. Algoritmo de control en un bucle
 - 2.3.1. Lectura del sensor
 - 2.3.2. Ajuste de la lectura a los niveles de luz
 - 2.3.3. Actualización de la actuación del controlador
 - 2.3.4. Ajuste de la actuación con Wind-Up
 - 2.3.5. Ajuste de la potencia de los motores con Saturación

Colocación del robot

En este apartado hablaremos brevemente de la colocación inicial en la que debe estar el robot, ya que es muy importante debido a que si no comenzará a tener errores acumulados mal que harán que no llegue en ningún momento a seguir la línea y se limite a girar sobre si mismo.



El sentido que debe tener el robot al comenzar es este, debido a que nosotros hemos decidido que siga la línea exterior del mapa, estableciendo este sentido como el inicial llevará a que él solo ya sepa analizar bien el camino a seguir y lo haga de forma correcta. Si ponemos el robot en sentido contrario, el robot dará una vuelta sobre si mismo para volver a colocarse como en la posición de arriba y comenzará a seguir la línea correctamente, pero ese giro inicial es algo que no está correcto, por lo que es mejor dejar esa posición como la inicial para el robot. También ha ocurrido que estableciendo el sentido contrario como el inicial el robot se ha recolocado en el borde interior, siguiendo después de forma perfecta la línea, pero no siendo el sentido que hemos establecido. Si queremos que recorra el borde interior este debe ser el sentido que debemos poner.



Contenido

Calibración de luz blanca y negra

En este apartado explicaremos cada paso seguido según el planteamiento general del programa. Comenzaremos primero de todo creando otro mini código fuera del programa principal en donde detectaremos con el sensor la intensidad del negro y del blanco.

- **Código de lectura de color blanco**

```
while (~ButtonPressed(BTNCENTER))  
    % Esperamos a que se pulse el boton central  
end  
ClearScreen();  
  
SetSensorLight(IN_1);      % Inicia el sensor de luz  
  
while (ButtonPressed(BTNCENTER))  
    l = Sensor(IN_1);  
    TextOut(1,LCD_LINE2,strcat('Light: ',num2str(l)));  
  
end
```

Viendo el código de arriba, se espera a que pulsemos a que pulsemos el botón central una vez colocado el robot en una zona del papel que sea del color blanco. Mientras se deje pulsado

mostrará el valor, oscilará entre 79 u 80, por lo que nosotros hemos decidido poner 80. Después el programa esperará a que se coloque encima del color negro.

- **Código de lectura de color negro**

El código es exactamente igual que en el caso de arriba, el programa espera a que se pulse el botón y una vez que se deje pulsado mostrará un valor que apuntaremos. El valor oscila entre 7 u 8, por lo que hemos decidido que sea 8.

Definición de constantes

El siguiente paso es establecer en el programa las constantes con sus valores. En este caso tenemos las siguientes constantes, BLANCO y NEGRO, establecidas en 35 y 70 respectivamente. También definimos Kd, Kp, Ki, constantes que se han ido variando hasta encontrar el mejor resultado con el robot y que se establecen en 0.7, 0.6, 0.3 respectivamente. Por último, tras haber realizado algunas pruebas se han definido WIND-UP con un valor de 20 y Saturación con un valor de 40.

Declaración de variables

Ahora se declaran las siguientes variables que van a ser usadas para realizar los cálculos en el programa principal y que serán modificadas dentro de él. Tenemos error_acumulado, error_anterior, tAnterior, tActual, pDer, plzq y dt, establecidas a 0. Luego tenemos la variable pBase en donde se le dará una potencia inicial al motor de las ruedas del robot y que es igual a 5. Por último, se encuentran las variables negro y blanco que ya se han establecido sus valores anteriormente.

Main

Ahora pasaremos a hablar del programa principal, que consta de 2 bucles while, el primero que espera a que el botón sea pulsado, antes de comenzar a seguir la línea. Una vez pulsado entra en el segundo while y comienza a repetir los siguientes pasos que se van a describir.

Controlador PID

```
% Lee el sensor de luz
lectura = Sensor(IN_1);

%AJUSTE DE LECTURA
lectura_recta = (lectura*(BLANCO-NEGRO) / (blanco-negro)) + ((-
negro*(BLANCO-NEGRO)) + (NEGRO*(blanco-negro)))/(blanco-negro);

%ACTUALIZACION DEL CONTROLADOR
%Calculamos el error
error = (lectura_recta - (BLANCO+NEGRO)/2);
%Calculamos la derivada del error
tActual = CurrentTick()-t_ini;
dt = double(tActual - tAnterior);
derivada_error = (error - error_anterior)/dt;
%Calculamos la integral del error
integral_error = 1 / (error + error_acumulado)*dt;

Gc_pid = Kp*error + Kd*derivada_error+Ki*integral_error;
pDer = pBase - Gc_pid;
pIzq = pBase + Gc_pid;

error_anterior = error;
```

```
error_acumulado = error_acumulado + error;
tAnterior = tActual;
```

En el apartado del controlador PID se comienza guardando la lectura de luz del robot y se escala y normaliza la lectura con la fórmula de la recta. Después, se calcula el error que es la media de la diferencia entre el valor normalizado y la suma de las constantes negro y blanco.

El siguiente paso es calcular derivada del error, en donde siguiendo la fórmula de la derivada, $\frac{derror}{dt}$, calculamos la diferencia del error, es decir, el error actual menos el anterior, y todo esto se divide entre el diferencial de tiempo, que es la diferencia entre el tiempo actual y el tiempo anterior.

Después, se calcula la integral del error que es la inversa del error acumulado por el diferencial de tiempo, $\frac{1}{\sum error} dt$.

Tras estos cálculos, aplicamos la fórmula $G_c(s) = K_p + K_D s + K_I \frac{1}{s}$, en donde el resultado lo guardaremos en la variable PID y que se aplican junto con la potencia base a la rueda izquierda y derecha.

Actuación con Wind-Up

%AJUSTE DE LA ACTUACION CON WIND-UP

```
if(Gc_pid > WINDUP || Gc_pid < -WINDUP)
    Gc_pid = Gc_pid - (integral_error*Ki);
    error_acumulado = error_acumulado-error;
end
```

El siguiente paso es evitar que el error calculado no haga que el robot se salga de su recorrido, lo que implicaría que acumulara los errores y no siguiera en ningún momento el recorrido. Para ello se controla con el WIND-UP y se dejaría de tener en cuenta el error acumulado para las siguientes iteraciones.

Ajuste de los motores con saturación

%AJUSTE DE POTENCIA DE LOS MOTORES CON SATURACION

```
if(pIzq > Saturacion)
    pIzq = Saturacion;
else
    if (pIzq < -Saturacion)
        pIzq = - Saturacion;
    end
end

if(pDer > Saturacion)
    pDer = Saturacion;
else
    if (pDer < -Saturacion)
        pDer = - Saturacion;
    end
end
```

El último paso, ajustaremos los motores en función a una saturación, ya que estableceremos que los motores no tengan una potencia más allá de 40, para evitar que se salga de las curvas más cerradas. Con esta última cota terminamos el ajuste de los motores, con lo que ya podemos aplicar la potencia a los motores, y probar el robot, comprobando instantáneamente su buen funcionamiento.