

Informe de Práctica 2

Programación de Robots

David Ávila Jiménez || Pedro Antonio Aguilar Lima

Índice

Resumen.....	2
Introducción	2
Ejercicio 1.A.....	2
Objetivo.....	2
Restricciones	2
Contenido del código	2
Cuestiones.....	3
Ejercicio 1.B.....	3
Objetivo.....	3
Restricciones	3
Contenido del código	3
Cuestiones.....	4
Ejercicio 2	4
Objetivo.....	4
Restricciones	5
Contenido del código	5
Cuestiones.....	6

Resumen

En este informe se recoge los datos sobre Programación de microbots, práctica 2 (“Control en bucle abierto/cerrado. Seguimiento de líneas”), el objetivo de esta es aprender el uso del programa V-REP PRO además de MATLAB mediante el robot de Lego EV3 en modo simulación, software, comandos y librerías proporcionadas en el Campus Virtual.

Introducción

Tanto en el ejercicio uno como en el dos vamos a trabajar con un robot Lego EV3. El primer ejercicio se divide en dos apartados, A y B, en donde debemos realizar tanto el control de un bucle abierto como el de un bucle cerrado respectivamente. Además, en el ejercicio dos realizaremos un seguimiento de líneas mediante un control de bucle cerrado. Para realizar esta práctica es necesario descargar la escena (“escenaSeguimiento.ttt”) dada en el campus virtual. Además, se ha creado un script por cada ejercicio en el software de MATLAB (“Ejercicio1BucleAbierto.m”, “Ejercicio1BucleCerrado.m”, “Ejercicio2.m”). Se usa el siguiente robot educativo, EV3 de Lego:



Ejercicio 1.A

Objetivo

El robot debe andar en línea recta mediante una orden de potencia “P” igual y constante en ambos motores (OnFwd) y la mantenga durante 2 segundos, tratando de aproximar la rotación de ángulos de los motores a 860. Todo esto se debe realizar programando un control en bucle abierto.

Restricciones

1. El valor del sensor se muestra después de pasar el tiempo máximo establecido.
2. El robot debe pararse en 860 grados con un error de ± 10 grados (aunque se permite mayor error).
3. Tiempo máximo de 2 segundos.

Contenido del código

Nos centraremos en una parte del código, la esencial, como se hace el control de bucle abierto. Lo primero que hacemos es dar potencia a los motores y después entra en el bucle en donde lo único que hace es leer los valores de los motores y salirse cuando haya pasado 2 segundos. Después apagamos los motores, quedando así programado el control de bucle abierto.

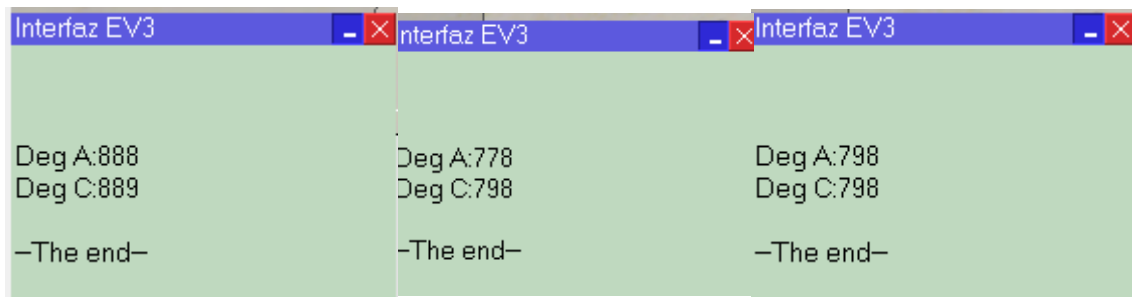
```

OnFwd(OUT_AC,40); % Arranca ambos motores con potencia = 40
while( (CurrentTick()-t_ini) <= tiempo)
    ra = MotorRotationCount(OUT_A); % Lee el encoder del motor
    izquierdo
    rc = MotorRotationCount(OUT_C); % Lee el encoder del motor derecho
end
Off(OUT_AC);

```

Cuestiones

Tras realizar tres pruebas con el robot, se han obtenido los resultados mostrados abajo. En el ejercicio se nos pide que los motores hubieran girado 860 grados, en este caso no hemos conseguido acercarnos al valor objetivo en ningún momento, habiéndose probado todo el rato valores entre 40 y 45 de potencia, pero siendo aleatorios todos ellos, ya que en algunas ocasiones salían o muy altos o bajos, por lo que se ha optado por dejar 43, un término medio y arrojando estos valores, con un error muy grande, pero siendo difícil de controlar al ser bucle abierto.



Ejercicio 1.B

Objetivo

En este caso se debe realizar lo mismo que en el apartado A, pero con un control de bucle cerrado donde en cada iteración vamos a ir calculando el resultado de restar la salida deseada (860) al valor leído en cada momento para cada rueda.

Restricciones

1. El valor del sensor se muestra después de pasar el tiempo máximo establecido.
2. El robot debe pararse en 860 grados con un error de ± 10 grados (aunque se permite mayor error).
3. Tiempo máximo de 2 segundos.
4. El motor solo admite valores entre -100 y 100, si la diferencia es mayor a 100, se acota a 100, si es menor de -100, se acota a -100.

Contenido del código

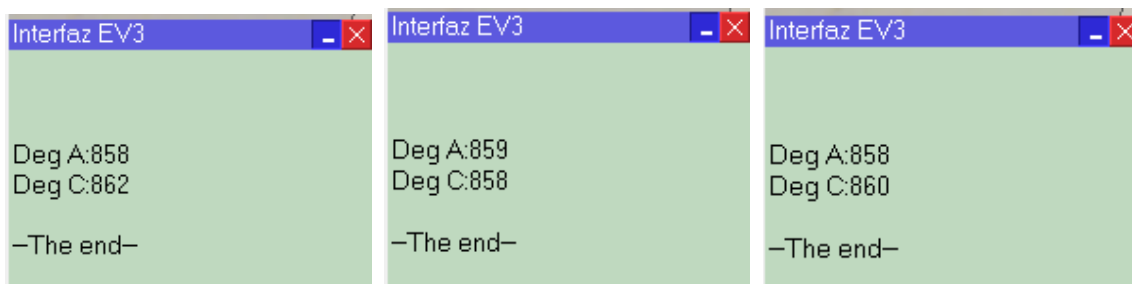
Igual que en el apartado anterior, aquí hablaremos de lo esencial del código, la implementación del control de bucle cerrado. El while controla el tiempo para que no se pase de 5 segundos. Al principio del bucle se apagan ambos motores debido a que, de no hacerlo, el robot calcula mal cuanto debe girar cada rueda y entonces el proceso se vuelve aleatorio, dando algún caso bueno y la mayoría muy malos. Apagando los motores sale prácticamente

perfecto. Después se calcula los nuevos valores de potencia de las ruedas y se acotan a los mínimos y máximos.

```
while( (CurrentTick()-t_ini) <= tiempo)
    Off(OUT_AC); %Apagamos motores para que no tropiece una ordena de potencia anterior con una nueva y sea más exacto
    ra = MotorRotationCount(OUT_A); %lee el encoder del motor izq
    rc = MotorRotationCount(OUT_C); %idem para der
    %calcula la diferencia entre los grados a conseguir y los que lleva en ambos motores
    ruedaIzq = 860 - ra;
    ruedaDer = 860 - rc;
    %Ajusta la potencia de las ruedas entres -100 y 100
    if (ruedaIzq > 100 && ruedaDer > 100)
        OnFwd(OUT_AC,100);
    elseif (ruedaIzq < -100 && ruedaDer < -100)
        OnFwd(OUT_AC,-100);
    else
        OnFwd(OUT_A, ruedaIzq);
        OnFwd(OUT_C, ruedaDer);
    end
end
Off(OUT_AC); %Apaga los motores
```

Cuestiones

Tras realizar tres pruebas con el robot, se han obtenido los resultados mostrados abajo. En el ejercicio se nos pide que los motores hubieran girado 860 grados, en este caso no es del todo exacto habiendo pequeños errores de cálculos, pero que a su vez son menores los errores y los datos son más precisos que los valores obtenidos en el control de bucle abierto que se basa en prueba y error, dando aun así resultados más dispares del valor objetivo. En este punto decir que es más eficiente para este caso el control de bucle cerrado, ya que al retroalimentarse a si mismo ayuda a que sea más fácil la corrección de los errores.



Ejercicio 2

Objetivo

Para este ejercicio se nos propone que el robot debe moverse siguiendo una franja de cierto nivel de gris. Esto se va a realizar con un control de bucle cerrado. El robot deberá ser colocado en el papel con la elipse de la escala de grises. Se realiza así para evitar

perturbaciones incesantes en la lectura del sensor de luz que se producirían si el robot siguiese una línea recta. Mientras el robot realiza esto se guardará los valores del sensor de luz en un fichero junto con cada instante de tiempo. Por último, deberemos cambiar la potencia de los motores y sacar conclusiones acerca de que ocurre con valores bajos de P y con los valores altos.

Restricciones

1. La banda gris debe de referencia es 40.
2. La banda de histéresis es 5.
3. El tiempo máximo será de 60.
4. La potencia de los motores vendrá determinada por P o -P, cuyos valores pueden ser 5, 10, 30 o 50.
5. Los valores del sensor de luz se guardarán en el fichero y no se mostrarán en la ejecución del programa.

Contenido del código

Lo esencial en este ejercicio es ver que se crea un fichero en el código para guardar los valores obtenidos por el sensor de luz en el bucle. Dentro del bucle mediante control de bucle cerrado se calcula el error del sensor con la referencia dada y se compara con la banda de histéresis. Tras esto, se le da potencia a los motores para que sigan una determinada escala de gris.

```
%Creamos manejador y cantida y el fichero
manejador=[];
cantidad=[];
CreateFile('Ejercicio2',50000,manejador);
R=40; %Referencia sobre la banda gris
U=5; %Banda de Histéresis
P=50; %Potencia motores (5, 10, 30, 50)
while( (CurrentTick()-t_ini) <= tiempo)
    Off(OUT_AC); %Apagamos motores para que no tropiece una ordena de potencia anterior con una nueva y sea más exacto
    t = CurrentTick()-t_ini; %Calculamos el tiempo que ha pasado
    l = Sensor(IN_1); % Lee el sensor de luz
    E=R-l; %Calculo del error
    %Escribimos en fichero
    WriteLnString(manejador,sprintf('%u\t%u',t,l),cantidad);
    %Damos potencia a los motores en función del error
    if(E<-U)
        OnFwd(OUT_A,-P);
        OnFwd(OUT_C,P);
    elseif(E>U)
        OnFwd(OUT_C,-P);
        OnFwd(OUT_A,P);
    else
        OnFwd(OUT_A,P);
        OnFwd(OUT_C,P);
    end
end
Off(OUT_AC); %Apagamos motores
CloseFile(manejador); %Cerramos el fichero
```

Cuestiones

Realizando varios experimentos en donde hemos variado los valores de P (usando 5, 10, 30 y 50), se ha sacado las siguientes conclusiones. Para valores bajos de P , el seguimiento es más adecuado debido a que el sensor de luz no se mueve de forma tan brusca a causa de la alta potencia de los motores. Como también se trata de un espacio pequeño para el seguimiento, a menor velocidad, menor número de errores. Cuanto más alto es el valor de P , más rápido va el robot por tanto hace una lectura menos fiable del sensor de luz, obteniendo valores muy dispares a la referencia establecida y teniendo que corregir errores muy altos a diferencia de los valores bajos, en donde los errores a corregir son menores.