29-9-2023

Prácticas de Laboratorio

Práctica 1 - Robótica

Christian Delgado Cruz y Juan

Ingenieria informática

índice

[Introducción 2](#_Toc146908601)

[Día 1 – 27/09/2023 3](#_Toc146908602)

# Introducción

# Día 1 – 28/09/2023

El día de hoy hemos aprendido a encender el robot que nos acompañará en la primera práctica, su funcionamiento es sencillo, tiene varios puertos y se puede conectar por bluetooth, wifi o USB.

Nosotros usamos un cable USB, una vez conectado y encendido (Pulsando el botón central y dejando que la luz se ponga a verde y pite) abrimos *MatLab* para controlarlo.

Primero había que esquematizar el robot, para poder entender como estaban conectados los puertos y que partes tenía el mismo. La parte principal es la base del robot y es donde tiene todos los puertos y por donde se enciende y apaga, mientras que además el robot tiene dos motores para hacer funcionar dos ruedas, un sensor de luz, dos pulsadores, un motor para mover la cabeza, y la cabeza propiamente tiene un sensor sonar.

Nuestra distribución es la siguiente:

Texto

Descripción generada automáticamente

Imagen - 1.0

El comando para vincularlo es: “*mi \_Robot=legoev3('USB')”* mostrando lo siguiente:

Texto

Descripción generada automáticamente con confianza media

Imagen - 1.1

Para vincular el motor: *“motor\_Cabeza = motor(mi\_Robot,'A');”* Para iniciarlo: *“start(motor\_Cabeza)”* Para asignarle una velocidad:*“motor\_Cabeza.Speed=10”* Para pararlo: *“stop(motor\_Cabeza)”* o :*“motor\_Cabeza.Speed=10”*

Con esto ya se mueve la cabeza del robot, pero nosotros queremos que nos de información sobre su movimiento, para ello usaremos una modificación de la función que representa el seno que dimos en prácticas, para ello definimos lo siguiente:

*“giro\_Cabeza(1) = double(readRotation(motor\_Cabeza));”*

Con este comando podemos saber en el instante uno, cuanto vale la rotación total de la cabeza, hay que ponerlo double para que no de errores, luego simplemente, haciendo un while de 10s con la función tic y toc recogemos los datos en una matriz que vamos representando con el drawnow y el plot.

Todo esto lo podemos ver en la siguiente foto, donde en resumen se ve todo el script.

Texto

Descripción generada automáticamente Imagen - 2.0

Para la segunda parte, simplemente utilizamos el comando siguiente para definir el Sonar y para leer la distancia el otro:

*“Sonar = sonicSensor(mi\_Robot,4);” “distancia(1) = double( readDistance(Sonar));”*

Hacemos lo mismo con la gráfica y poniendo la mano en el sensor cambian sus valores

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente Imagen - 3.0

# Día 2 – 05/10/2023

El día de hoy hemos aprendido a configurar el robot para que solo inicie, o termine de hacer la función que nosotros queramos cuando pulsemos un botón, en nuestro caso, el superior.

Para ello hemos utilizado el ya definido “RobotCabeza.m” que es un script que ya definimos en la última clase, el cual movía la cabeza durante un tiempo definido de 10 segundos, ahora, empezará cuando nosotros queramos pulsar el botón, y de forma análoga cuando queramos pararlo.

Es decir, utilizando el esquema de la “**Imagen - 2.0**” solo cambiamos la parte del “*while*” y le añadimos la definición del botón “*Pulsador=touchSensor(mi\_Robot,1);*” quedándose de la siguiente forma:

Texto

Descripción generada automáticamenteImagen - 4.0

Para la siguiente parte, hemos definido una referencia de un ángulo de 90, para que el robot gire 90 grados, y se pare, para ello tenemos que calcular el error que se genera, y multiplicarlo por una constante que hemos definido como “*kp*”, y nos dará el “*power*”, que es la velocidad en la que se deberá mover para colocarse en el ángulo, tras ello en la siguiente imagen, podemos ver que dibujamos el error, la referencia y el giro que realiza, haciéndose referencia y giro el mismo valor al corregir el error.

Texto

Descripción generada automáticamente

Imagen - 4.1

Texto

Descripción generada automáticamente

Imagen - 4.2

Para la última parte hemos recogido el esquema anterior, y realizado los cambios de las siguientes imágenes. Con esto conseguimos que la cabeza no para de moverse, porque la referencia sigue haciéndose mayor cada segundo, y nuestro script intenta corregirlo constantemente, también, el error será superior en los primeros segundos, hasta que la velocidad se mantenga porque al empezar en 200, la cabeza tendrá que hacer un desplazamiento más rápido para corregir el error.

Imagen que contiene Texto

Descripción generada automáticamente

Imagen - 5.0

Texto, Carta

Descripción generada automáticamente

Imagen - 5.1