

DESAFIO DE PROGRAMACION

III ASTI ROBOTICS 2018/2019

1. Explicación del desafío

El objetivo del desafío es ayudar a un robot a llegar a su destino controlando las velocidades de las ruedas. El robot es de tipo diferencial de dimensiones 10cm x 10cm.

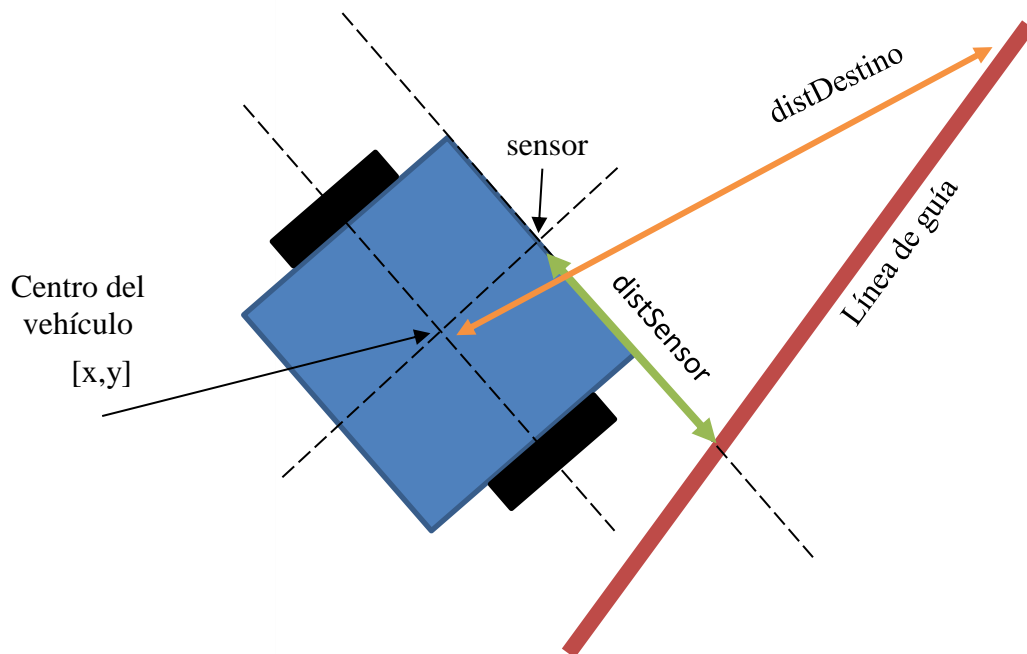


Fig.1 Vista en planta desde arriba

En un robot diferencial si las dos ruedas tienen la misma velocidad el robot se desplaza en línea recta. Si la velocidad de la rueda derecha es mayor que la rueda izquierda, el robot tenderá a girar hacia la izquierda. Si la velocidad de la rueda izquierda es mayor que la de la rueda derecha, el robot tenderá a girar hacia la derecha.

Para ayudar al robot a llegar a su destino se dispone de una línea de guía y 2 sensores, esta línea es sólo una ayuda no es obligatorio seguirla. El punto destino siempre pertenecerá a la línea de guía. El sensor *distDestino* envía la distancia en línea recta entre el centro del robot y el punto destino. El sensor *distSensor* indica la distancia entre el centro del frontal del vehículo (ubicación del sensor) y el punto de corte de la prolongación de la línea del frontal del vehículo con la línea de guía (Ver figura 1). Cuando el sensor está a la izquierda de la línea de guía el resultado es positivo, cuando el sensor está colocado a la derecha de la línea el resultado es negativo. Cuando el sensor esta justo encima de la línea la distancia *distSensor* es 0. Si el robot se desplaza perpendicularmente a la línea de guía, la prolongación del frontal nunca corta la línea de guía y el valor devuelto por el sensor es Nan (Not a number).

La figura 2 muestra un ejemplo de desafío: la línea azul es la línea de guía, el robot parte del círculo verde, el punto destino es el punto rojo, y la trayectoria descrita por el robot se representa en color negro.

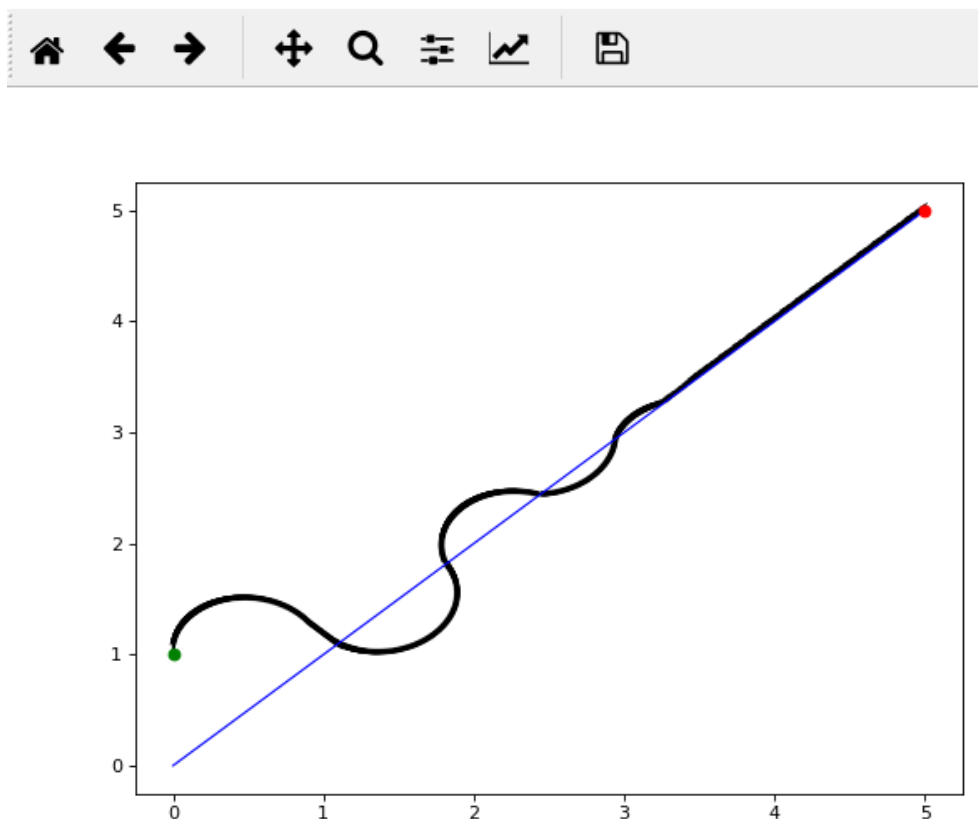


Fig.2 Vista en planta desde arriba

El lenguaje de programación empleado para la implementación del algoritmo será Python. El lenguaje se ha escogido por su sencillez de programación y ejecución, capacidad de abstracción en interoperabilidad. Estas características hacen que sea uno de los lenguajes más utilizados en la comunidad científica y académica para el desarrollo y validación de algoritmos.

Se le proporciona al usuario el código que modela el comportamiento del robot, línea y sensores, así como las funciones necesarias para visualizar la trayectoria del robot. **Este código no se podrá modificar.** El usuario solo podrá escribir en las zonas del programa habilitadas para ello (Zona A y Zona B ver figura 3) . La zona A sirve para que el usuario inicialice todas las variables que necesite antes de que se ejecute cada fase. El código en la zona B se ejecuta cada 100ms y esta pensado para incluir las instrucciones para controlar el robot.

El participante sólo debe enviar el código de la zona A y de la zona B. El código será validado automáticamente por una herramienta. El código que no cumpla este requisito no será tenido en cuenta.

La modificación de la velocidad de las ruedas se realiza con la función:

```
def fijarVel(self,vIzq,vDer):  
    if vIzq > 2:  
        self.__vizq=2  
    elif vIzq < -2:  
        self.__vizq=-2  
    else:  
        self.__vizq=vIzq  
    if vDer > 2:  
        self.__vder=2  
    elif vDer < -2:  
        self.__vder=-2  
    else:  
        self.__vder=vDer
```

Esta función recibe como primer parámetro de entrada la velocidad de la rueda izquierda y como segundo parámetro de entrada la velocidad de la rueda derecha. La máxima velocidad permitida es 2 m/s en ambas direcciones.

```

#PROGRAMA PRINCIPAL
#-----
Tfin=200
r=Robot(Tfin)

while r.leerT() < Tfin and not r.leerFinFases():
    r.actTiempo()
    Tfin=200

#-----
# Tocar a partir de aqui {
#-----

#-----
# } No tocar desde aqui
#-----

while r.leerT() < Tfin and not r.leerFinal():
    r.actTiempo()

#-----
# Tocar a partir de aqui {
#-----

#-----
# } No tocar desde aqui
#-----

#-----

if r.leerFasesSuperadas()==2:
    print "-----"
    print " :) Ha llegado al final"
    print "-----"
    print "Fases superadas:"
    print r.leerFasesSuperadas()
    print "Tiempo total: "
    print r.leerTiempoTotal()
else:
    print "-----"
    print " :( Intentelo de nuevo"
    print "-----"
    print "Fases superadas:"
    print r.leerFasesSuperadas()
    print "Tiempo total: "
    print r.leerTiempoTotal()

#DIBUJO
r.pintar()

```

ZONA A
Zona para escribir tu código

ZONA B
Zona para escribir tu código

Fig.3 Programa principal

2. Reglas adicionales

- El plagio supondrá la descalificación del equipo participante y la imposibilidad de participar en el resto de las pruebas del torneo.
- El usuario podrá implementar funciones y hacer llamadas a sus funciones.
- En caso de necesitar incluir alguna librería de Python adicional a las que aparecen actualmente en el código, se consultará con la organización.

3. Ranking y puntuación

El desafío tendrá varias fases. El ranking se establecerá mediante el número de fases superadas. En caso de empate en el número de fases se tendrá en cuenta el tiempo empleado, un menor tiempo supondrá una posición superior en el ranking.

Al final del desafío, en caso de ser necesario, si hay empate en número de fases y tiempo se resolverá revisando la calidad del código entregado.

4. Procedimiento de envío de respuestas

El algoritmo se enviará por correo electrónico a desafio@astichallenge.com. Se podrá enviar como máximo un correo diario, los restantes correos no se tendrán en cuenta.

5. Procedimiento de publicación de resultados

Los resultados se irán publicando regularmente en un documento público Google Sheets. Se informará por correo electrónico del enlace, una vez que se hayan publicado los primeros resultados.

6. Recomendaciones

- Se recomienda la instalación y el uso de *Anaconda* que incluye *Spyder* para el desarrollo del algoritmo y todas las librerías necesarias.

<https://www.anaconda.com/download/>

- Se recomienda utilizar Python 2.7
- Si se usa *Spyder*. Se recomienda visualizar las gráficas empleando Qt en vez de la representación gráfica en línea. Para ello:
 - Herramientas --> preferencias--> Terminal de IPython --> Gráficas --> Salida gráfica: Automático

