Universidad Católica del Uruguay

Proyecto final de Robótica

Diseño e implementación de un sistema multiagente

Diego Toledo, Juan Pablo Becoña, Soffía Pereira y Diego Medina

Profesores: Enrique Ferreira y Jose Job

Universidad Católica del Uruguay
Facultad de Ingeniería y Tecnologías
Montevideo, Uruguay

Julio 2017

INTRODUCCIÓN

El proyecto final del curso consiste en controlar dos robot (agentes uniciclo) para que formen determinada posición entre ellos. A su vez, uno será de líder siguiendo una trayectoria. Esto se realizará mediante la obtención de las posiciones con una cámara web, el control y comunicación con los agentes.

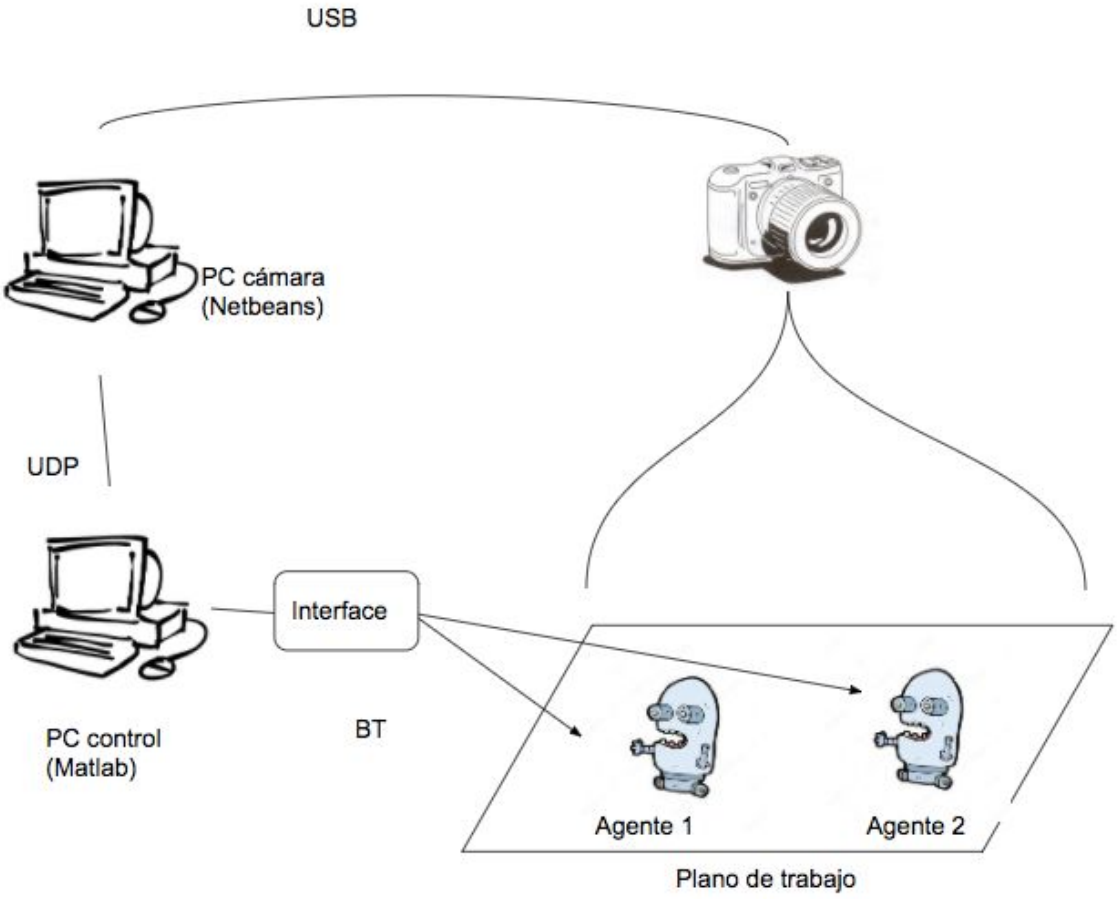

Figura 1. *Esquema inicial de la solución*

Composición de cada agente:

- Dos motores de continua
- Batería de li-on
- Sistema embebido (Beaglebone black y arduino nano V3)
- Modulo BT HC-05
- Protoboard para conexiones
- Puentes H, L298n (2 puentes H independientes)
- Impresión 3D para la carcasa del robot.

Agente uno

El agente uno está armado con el BeagleBone Black conectando el módulo bluetooth a uno de los puertos seriales del mismo y pines PWM a las entradas del driver del motor de tipo puente H.

Su programa está desarrollado en python y su funcionamiento principal consta en la lectura de la información en el puerto serial mandada a través del módulo bluetooth, el parseo de la misma para la obtención de los PWM correspondientes y finalmente la ejecución de una función encargada de asignar la señal correspondiente a cada motor durante el tiempo especificado.

Al utilizar los agentes fue notoria la necesidad de definir velocidades mínima y máxima de giro de las ruedas ya que el robot presenta una zona muerta de velocidades en un entorno cercano a cero y su velocidad debe ser menor a un máximo porque de lo contrario es difícil para la cámara seguirle la posición.

Cabe destacar además que para este agente fue necesario establecer una diferencia predefinida entre los PWM de ambas ruedas, ya que si le asignamos la misma señal a las dos ruedas el robot no avanzara un línea recta sino con una pequeña desviación.

En la siguiente figura se puede ver un diagrama de conexionado de este agente.

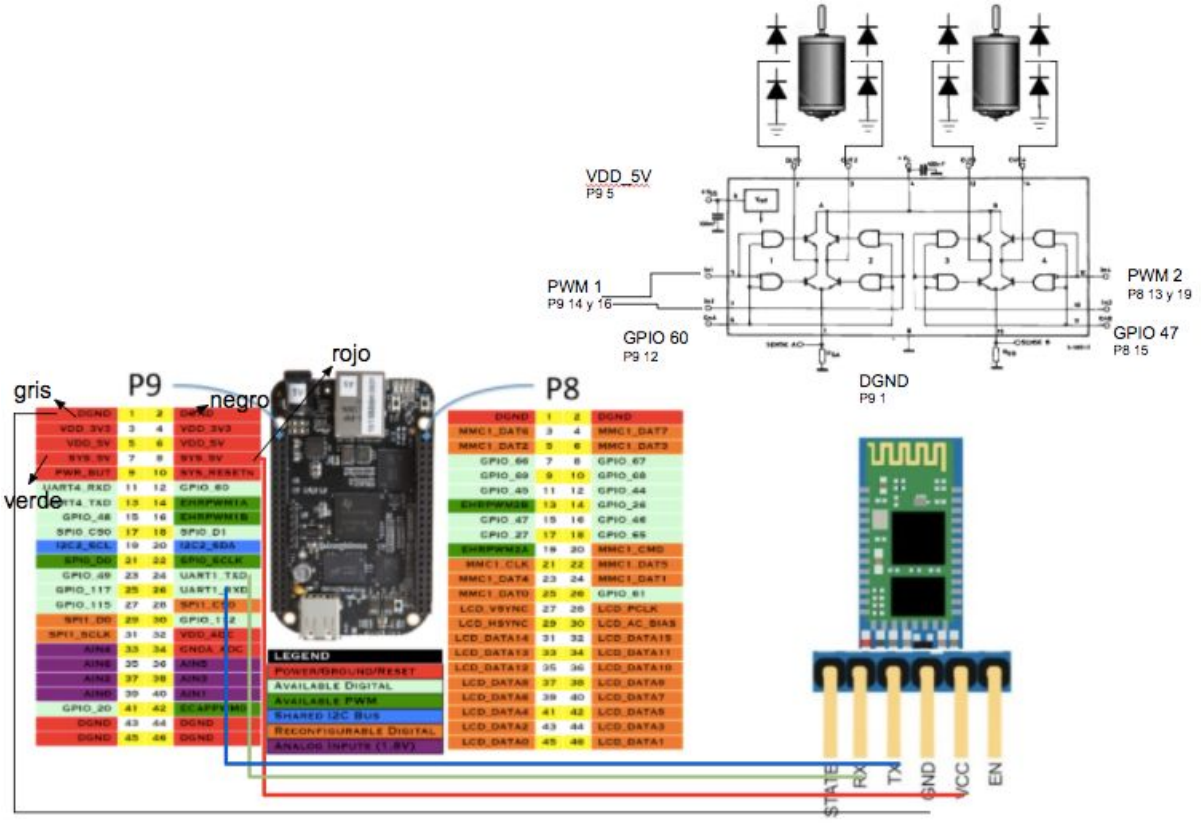

Figura 2. Esquema de conexiones del agente 1

Agente dos

El agente dos fue armado haciendo uso de una placa Arduino nano v3, utiliza el módulo Bluetooth HC-05 y el driver del motor es el L298N que cuenta con dos puente H, necesarios para mover ambos motores.

Se realiza un programa en C en el IDE proporcionado por Arduino. Se tienen pines designados para la conexión serial con el módulo Bluetooth mediante los cuales es posible la comunicación. La información que se obtiene es parseada y enviada a los motores, donde se controla el sentido del giro y la velocidad mediante una señal PWM, que varía en un rango de frecuencias de 0 a 255.

Experimentalmente se llega a que las velocidades para que el robot tenga un funcionamiento correcto y no se pierda de las cámaras son aquellas para las cuales las frecuencias del PWM varían entre 55 y 145.

En la siguiente figura se puede ver un diagrama de conexionado de este agente.

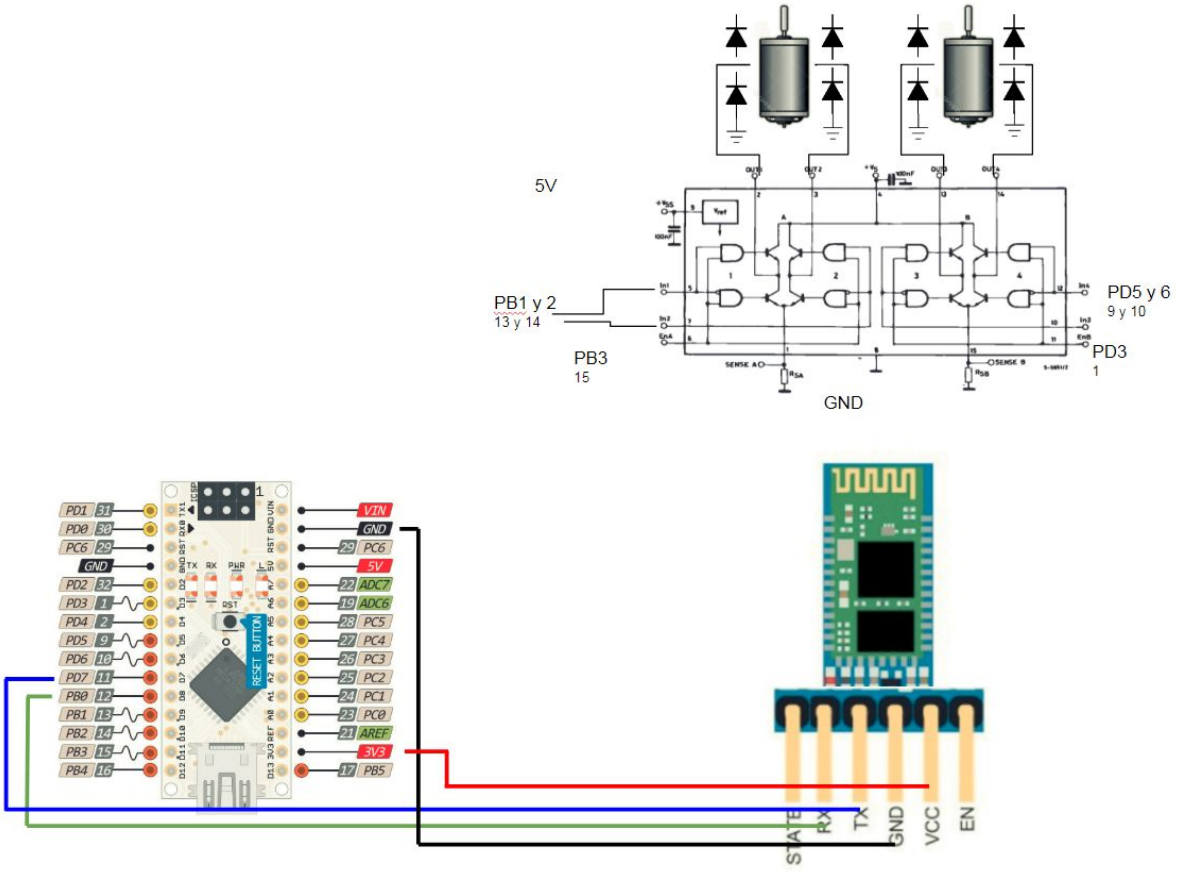

Figura 3. Esquema de conexiones del agente 2.

INSTALACIÓN

En la PC de la cámara se necesita instalado el entorno de desarrollo NetBeans, con librerías instaladas para el procesamiento de imagen y librerías de java. (Ver procedimiento en proyecto final de Robótica de Ricardo Illa adjunto con este informe). Luego en la PC del control se instaló Matlab en el cual no se necesito ninguna librería extra para obtener el resultado esperado. Por último, se modificaron las configuraciones de Windows para la comunicaciones tanto de entre PC como para los Agentes.

Para la comunicación entre los agentes se necesitaron crear dos interfaces Bluetooth, con los nombres “Robot1” y “Robot2” en la PC del control, para poder referirse a cada uno.
Para la comunicación entre las dos pc se utiliza una comunicación udp entre ambas mediante una red punto a punto, se debió de quitar los firewall de la pc, porque la seguridad de Windows impedía realizar la comunicación correctamente.

DESARROLLO

La realización del control de dos agentes uniciclo se realiza mediante:

- Una cámara que obtiene las posiciones del punto frontal y central de cada agente esta PC la nombraremos PC A.
- En otra PC se realizará el control de formación de los robot mediante consenso, esta pc la nombraremos PC B
- Los resultados, velocidades de las ruedas, se envían al programa de los uniciclos (BeagleBone Black y Arduino)

En la PC A mediante el entorno NetBeans se utiliza el programa en JAVA descrito anteriormente al cual se adaptó para dos agentes simultáneos, un programa que obtiene la posiciones frontal y central de los dos agentes y los envía mediante un puerto en un formato string.

En la PC B tendremos las siguientes funciones y programas:

Main.m

Maneja la comunicación y cálculo del control. Comienza inicializando la comunicación bluetooth con los robots, luego lee de la cámara las posiciones de los robots, con ellas calcula el control y parsea la velocidad de las ruedas para luego enviarlas a los robots. Este procedimiento se puede ver en el esquema de la figura 4.

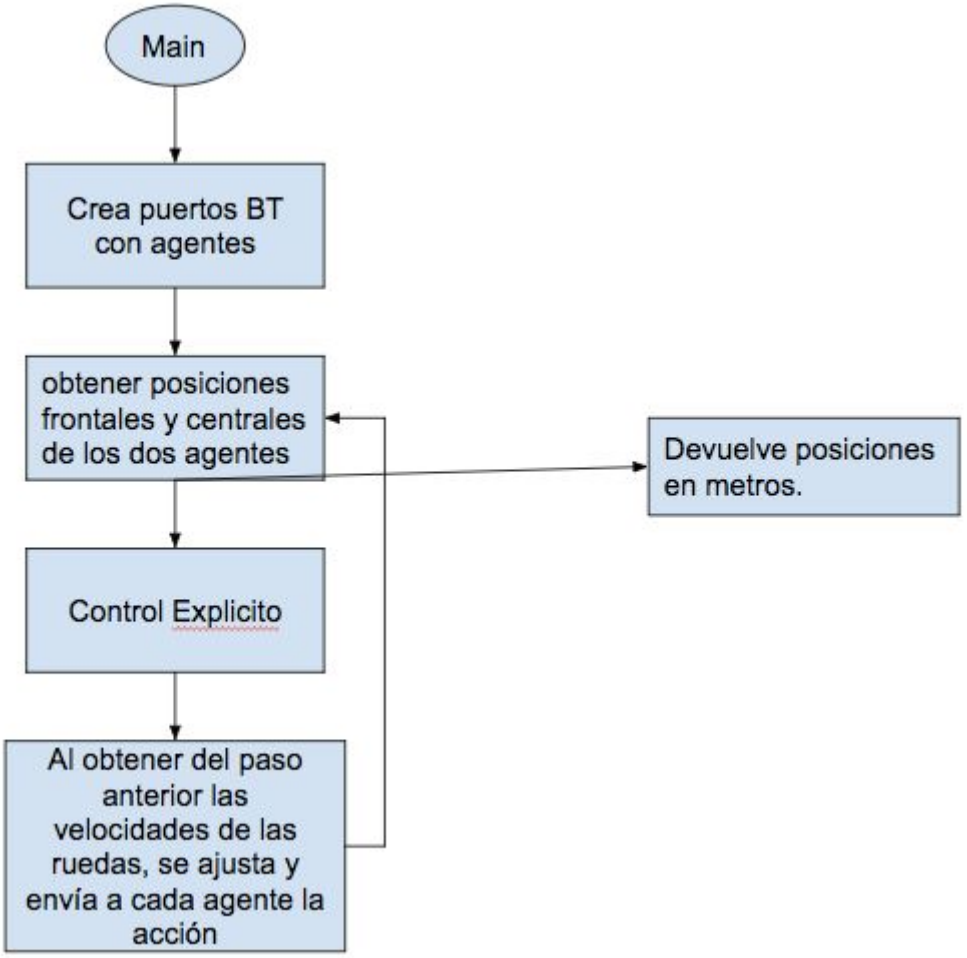

Figura 4. Esquema global del flujo entre programas.

UDPparaCamara.m (parseMessage)

Este programa tiene la comunicación con la PC que contiene el programa de la cámara. Obteniendo las posiciones de los dos robots. El mismo realiza un parseo en el buffer del puerto UDP que recibe un string con las dos posiciones de cada robot.

controlFormacionExp.m

Programa basado en el control de agentes uniciclo mediante formación, estudiado en el curso, para dos agentes. En la figura 5 se puede observar el

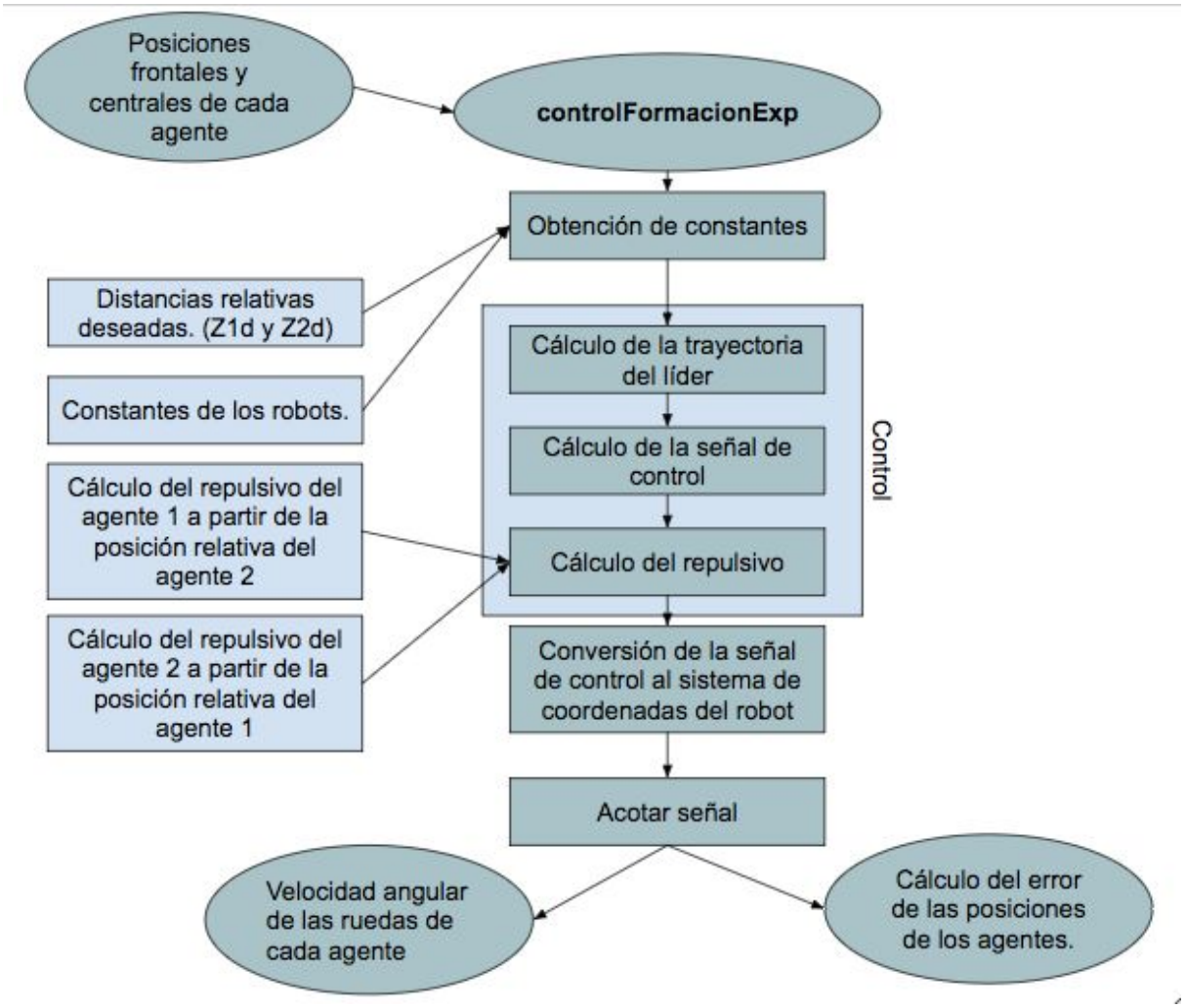

Figura 5. Diagrama de flujo de la función ControlFormacionExp.

CONTROL DE FORMACION

Se consideró el sistema de coordenadas de la figura siguiente. El sistema de coordenadas de la cámara, que inicialmente utilizamos y nos generó gran cantidad de inconvenientes al no ser directo, originalmente tiene los ejes x e y invertidos.

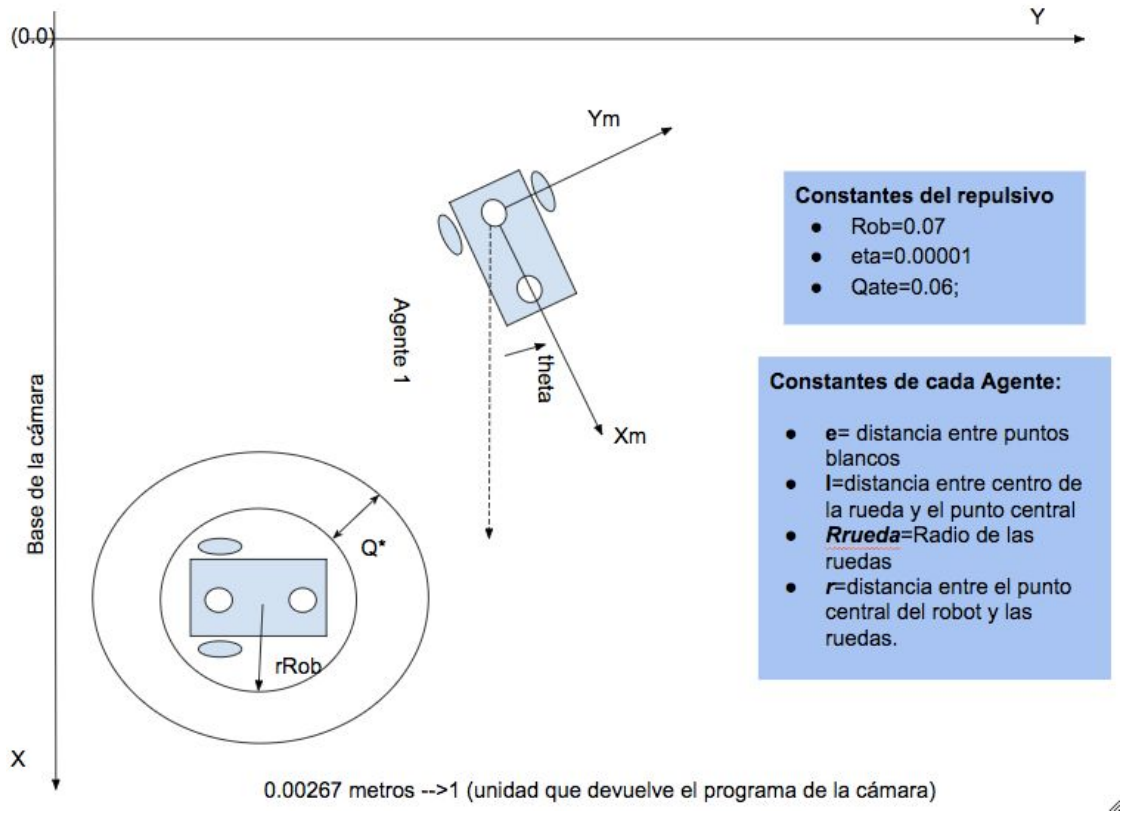

Figura 6. Sistema de coordenadas del sistema Cámara-Agentes.

El control implementado fue por consenso y para formación. Se plantea la siguiente topología de grafos donde ambos agentes conocen la posición del otro.

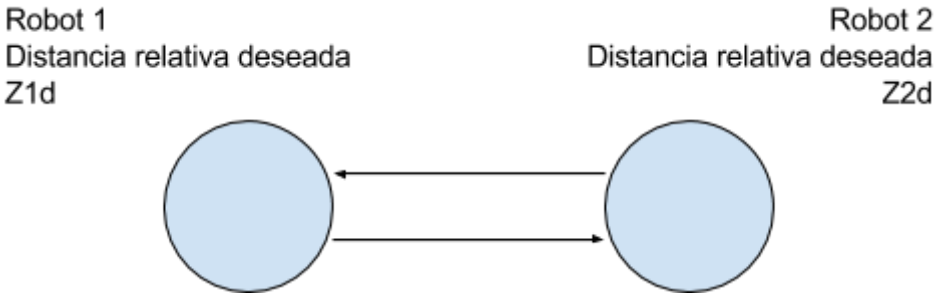

Figura 7. Sistema de coordenadas del sistema Cámara-Agentes.

Utilizando la topología anterior se desarrolla la matriz laplaciana

$$A_d = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\Delta = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$z = \Delta - A_d$$

$$z = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

Con ella se calculan las velocidades que deben tener los robots en el sistema absoluto

$$un = -k * (z \otimes I_m) z_i - C_{ij}$$

Im es la matriz identidad de 2x2 al tener dos dimensiones el sistema y Cij son las distancias que deben mantener los robots en la formación final. La matriz un contiene las velocidades en x e y que deben tener los agentes en el siguiente ciclo de programa para acercarse al destino. Como se las debe implementar en el robot primero se la traslada al sistema relativo de los agentes con la inversa de la matriz de posición del uniciclo, que con los ejes mostrados al comienzo es de la siguiente manera.

$$A_i = \begin{pmatrix} \cos\theta & -e\,sin\theta \\ sin\theta & e\,cos\theta \end{pmatrix}$$

Luego de obtener las velocidades lineales y angulares relativas al robot se pasa a las velocidades angulares de las ruedas mediante las ecuaciones siguientes.

$$\omega_{rueda\ derecha} = \frac{u1 + L * u2}{r}$$

$$\omega_{rueda\ izquierda} = \frac{u1 - L * u2}{r}$$

Siendo r el radio de la rueda, u1 y u2 las velocidades lineales y angulares de los robots y L la distancia del eje de las ruedas al centro del robot. Estas velocidades angulares son procesadas para cada robot calculando el duty cycle de los PWM, considerando que sistema embebido tienen y las particularidades tanto de los motores como de la construcción física. Para por último enviar esta información a los robots para ejecutar el control.

COMUNICACIÓN DEL CONTROL CON LOS AGENTES.

Al momento de comunicar lo calculado por el control a los dos agentes, se encontró con la problemática de que los dos agentes tienen sus diferencias, tanto en la programación, ya que las librerías de Arduino manejan de forma ligeramente distinta el PWM que Pyton, como en sus características, los motores y sus drivers tienen diferencias. Por lo tanto, se

implementa una interfaz de comunicación, que toma de entrada lo enviado por el control, velocidades angulares de las ruedas, y lo envía a cada agente según su protocolo de comunicación.