

Proyecto - A multi-robot system for unconfined video-conference

Luis Enrique Ruiz-Fernandez

Diciembre 2021

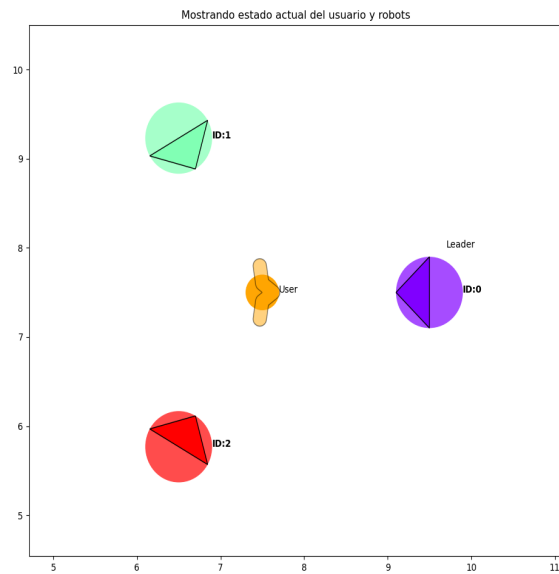
Para este proyecto, se realizara la implementación del artículo [1], exceptuando la parte de procesamiento de imagen, que de acuerdo al artículo se utiliza para estimar la posición y velocidad del usuario, con lo que para este trabajo se supondrá ya dada. Para no extender el reporte, solo se mencionara a grandes rasgos los pasos que se siguieron para llevar a cabo la implementación.

Primero tenemos un número n de robots, estos robots son *differential drive* (DDR), que van a ser controlados por dos controles, velocidad angular ω y velocidad lineal v , para trasladarse en el mapa, estos DDR son capaces de rotar en sitio, los DDR se distribuyen de manera uniforme al rededor del usuario como se muestra en la Figura (1a), en la cual se puede observar que el robot con mejor vista hacia el usuario, sera considerado *leader*, este *leader*, es el encargado de estimar posición y velocidad del usuario, con esa información, se generara un árbol de estados (configuraciones) que se genera a base de 3 controles, hacia atrás (*BC*), hacia atrás e izquierda (*BLC*) y hacia atrás y derecha (*BRC*).

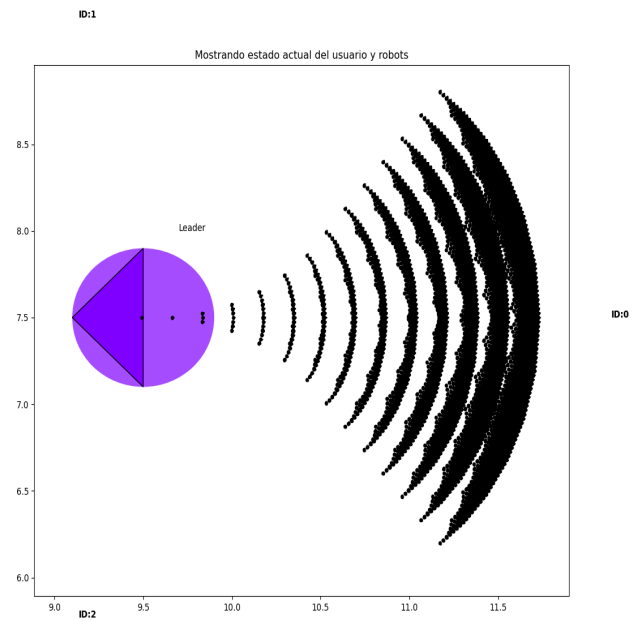
Recalcando que el usuario solo hace dos tipos de movimiento, rotar en sitio y movimiento en linea recta, es por eso que solo se consideran estos dos controles, ya que si el usuario va de frente al robot este se mueve hacia atrás para mantenerlo en vista el mayor tiempo posible. En la Figura (1b), se puede observar el árbol que se genera. A la hora de generar el árbol cada nodo o configuración tiene un costo que maximiza el tiempo que el robot líder mantiene en *mejor* vista al usuario, la manera en la que se calcula cada costo se ve en la siguiente ecuación,

$$C(q(t)) \equiv \langle r(t), \theta(t), \alpha(t) \rangle = C(\text{parent}(q(t))) + \text{isDesirable}(q(t)) * \Delta t - K|\theta(t) - \theta_{\text{sector}}|, \quad (1)$$

donde, $r(t)$ es la distancia a la que esta el robot del usuario, $\theta(t)$ es la orientación del robot, $\alpha(t)$ es el ángulo entre la orientación del usuario y la del robot, $\text{isDesirable}(q(t))$ es una función que nos retorna 1 o 0, retorna 1 si el robot esta dentro del margen aceptado de distancia $r(t)$ y dentro del rango de vista con respecto al usuario θ_{sector} , y K es una ganancia que convierte la diferencia angular en unidades de tiempo, por ejemplo dividiendo la diferencia angular entre la velocidad angular máxima del robot.



(a) Condiciones iniciales para arrancar el algoritmo.



(b) Robot líder, con su árbol de estados.

Figure 1: Primera parte del algoritmo.

Una vez que el árbol se hace una búsqueda para maximizar el tiempo, en la Figura (2) se puede ver en la simulación como el robot recorre una trayectoria (puntos rojos) dentro del árbol, que maximiza el tiempo que mantiene en vista al usuario, y en la Figura (3) se muestra que lo hace iterativamente si el usuario sigue en línea recta.

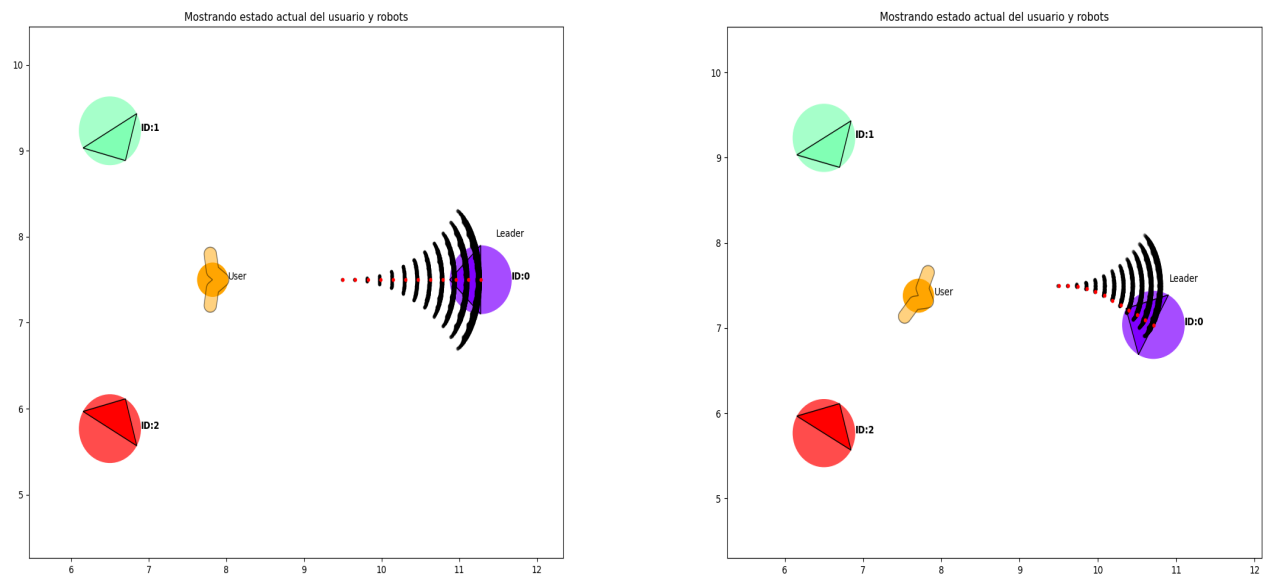


Figure 2: Obteniendo trayectorias dentro del árbol.

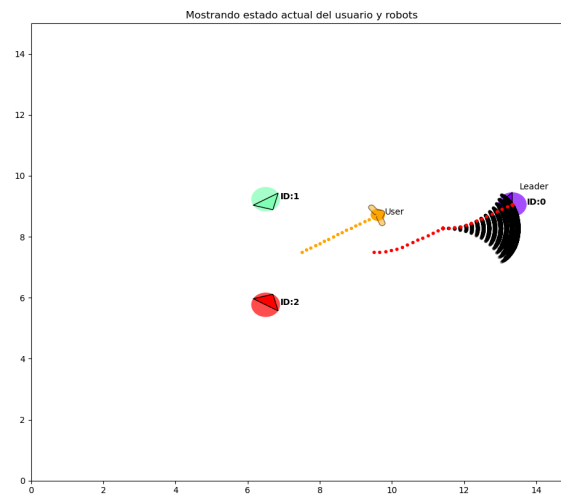


Figure 3: Resultado de simulación iteraciones, línea recta.

Para los otros robots simplemente tenemos que generar controles que mantengan la formación al rededor del usuario, en la Figura (4) se puede mostrar como los robots que no son el líder, mantienen la formación, por si cuando el usuario pare y rota en sitio, otro robot tiene que tomar el papel de líder.

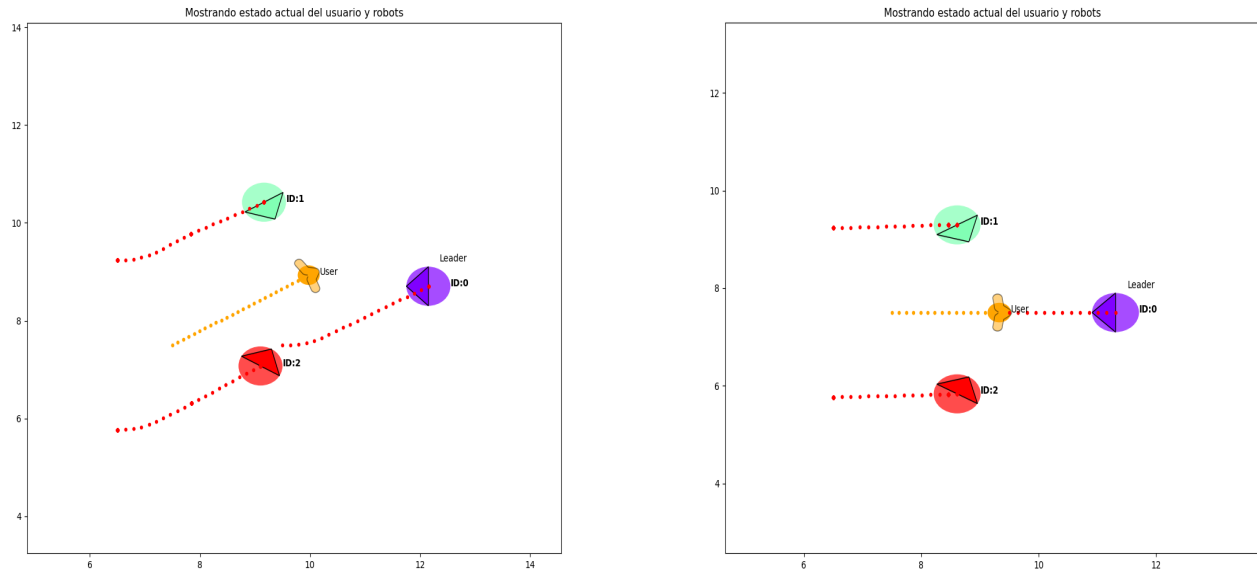


Figure 4: Prueba de controles en simulación para robots no lideres.

Tomando otro robot como el líder, se ve en la Figura (5) que el algoritmo se ejecuta correctamente sin importar el líder.

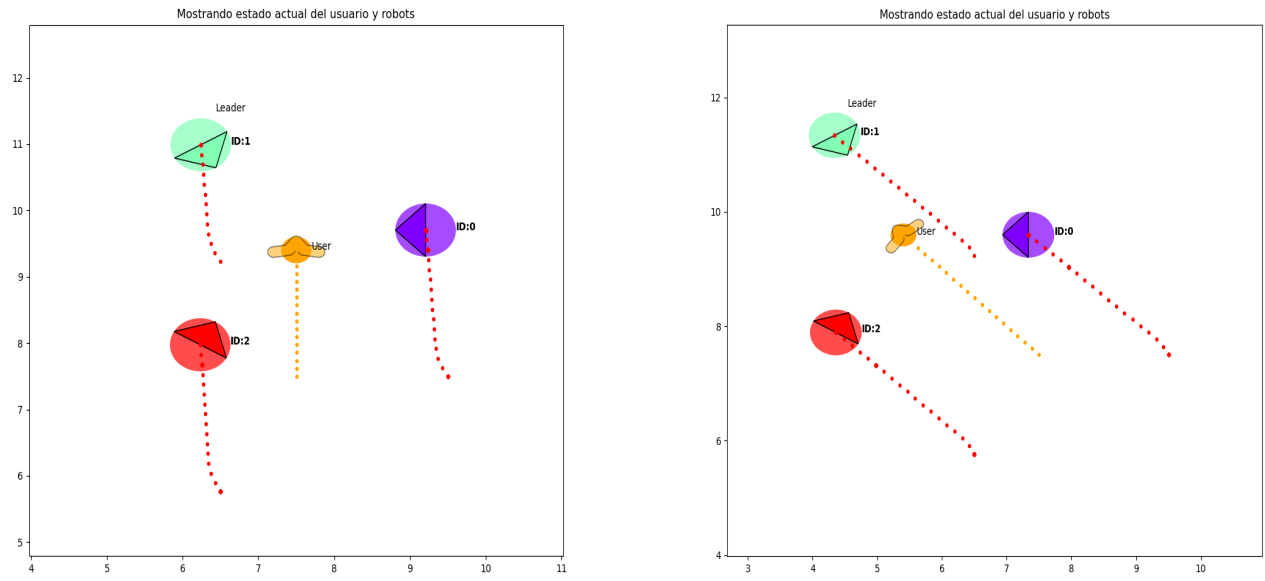


Figure 5: Prueba de lideres en simulación.

En la Figura (6) se puede observar la simulación corriendo, con rotaciones y movimientos aleatorios del usuario, como los robots tratan de mantener en vista al usuario.

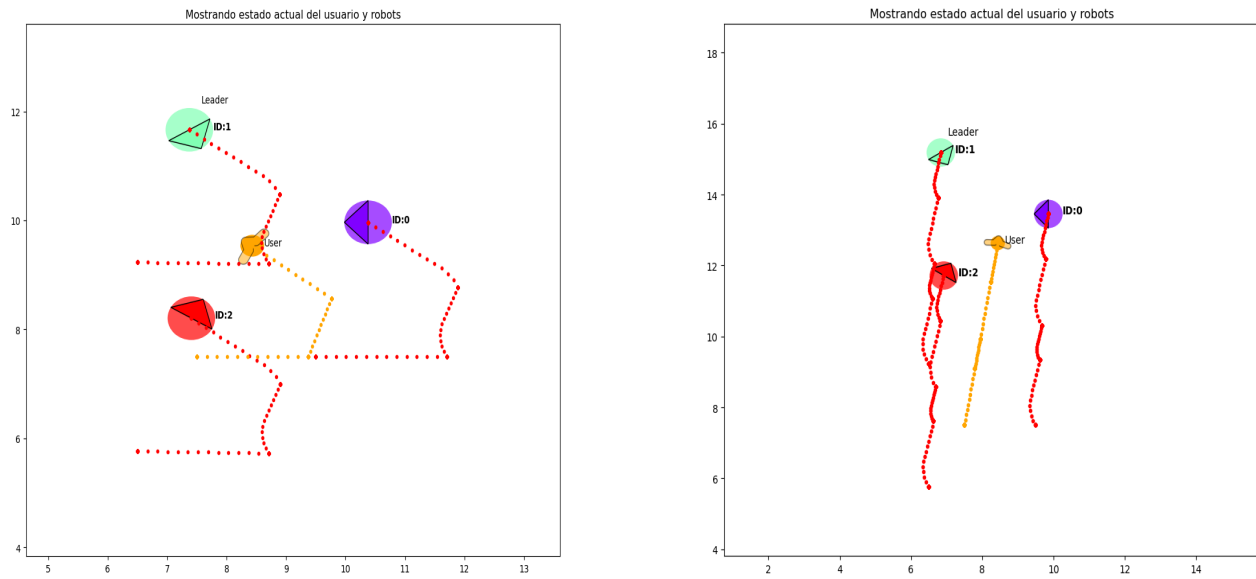
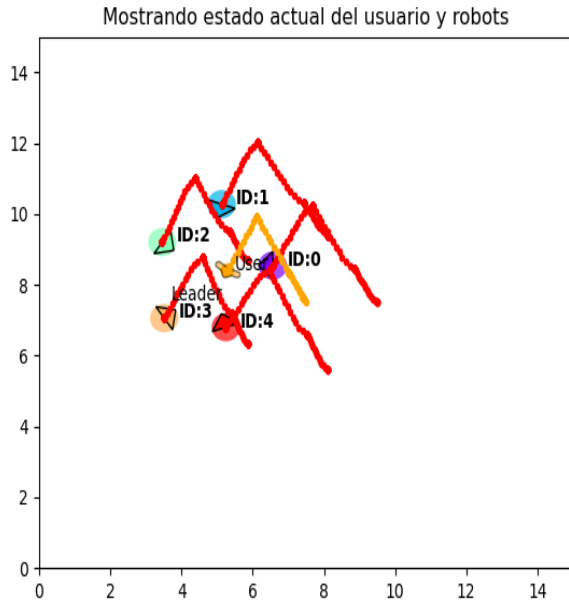
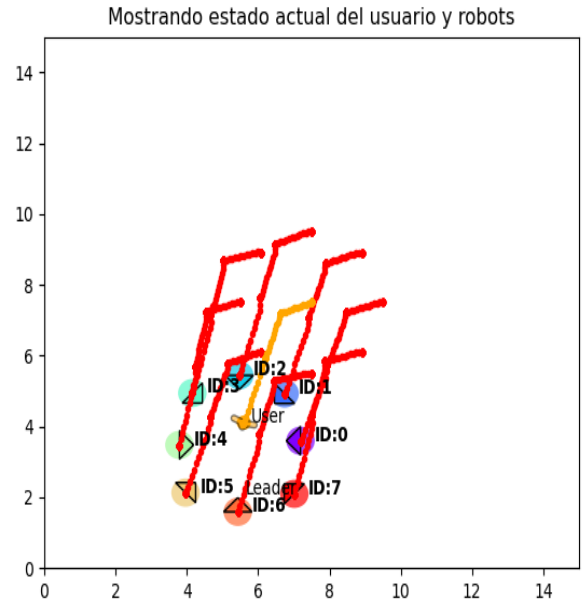


Figure 6: Simulación final, con 3 robots.

Y por ultimo en la Figura (7) se muestra en el caso con mayor numero de robots, muestra que el algoritmo se ajusta de manera correcta.



(a) Numero de robots 5.



(b) Numero de robots 8.

Figure 7: Resultado simulación, mayor numero de robots.

Observaciones: unas observación que se realizaron a lo largo del proyecto y que se pueden mejorar; 1) existen ángulos muerto, esto quiere decir que hay ocasiones en las que la orientación del usuario queda exactamente entre la vista de dos robots, lo que hace que no se fije un líder y se muevan sin seguir al usuario; 2) no existe la evasión si existen obstáculos o incluso entre ellos, se supone que se mueven en formación pero no hay nada que asegure que no chocaran entre ellos; 3) generar trayectorias, basado en árbol de estados (configuraciones) hace para ciertas orientaciones complicado hacer el seguimiento del usuario, considerando que se tiene un numero limitado de robots.

Referencias

- [1] Nikhil Karnad and Volkan Isler. “A multi-robot system for unconfined video-conferencing”. In: *2010 IEEE International Conference on Robotics and Automation*. 2010, pp. 356–361. DOI: 10.1109/ROBOT.2010.5509600.