

NAVEGACIÓN AUTÓNOMA DE UN DRON PARA LOCALIZAR UN TRANSMISOR DE RADIO FRECUENCIA BASADO EN APRENDIZAJE POR REFUERZO

GRADO EN INGENIERÍA DE ROBÓTICA SOFTWARE



Cristian Sánchez Rodríguez

▶ DRONES

Dispositivos aéreos no tripulados, útiles en labores de:

- Inspección.
- Inventariado.
- SAR.



► NAVEGACIÓN AUTÓNOMA

O conjunto de algoritmos enfocados a planificar y mover un dispositivo sin intervención externa. Existen diversos enfoques:

- Imitation Learning.
- Deep Learning.

Requieren de datasets, por ello surge la necesidad de probar nuevas soluciones.



► SEÑALES RF

Entre ellas se opta por estudiar sistemas capaces de navegar en base a señales emitidas:

- Detección de fuentes maliciosas (jamming).
- Rescates con poca visibilidad, donde se pueda navegar hacia un transmisor situado en una salida.



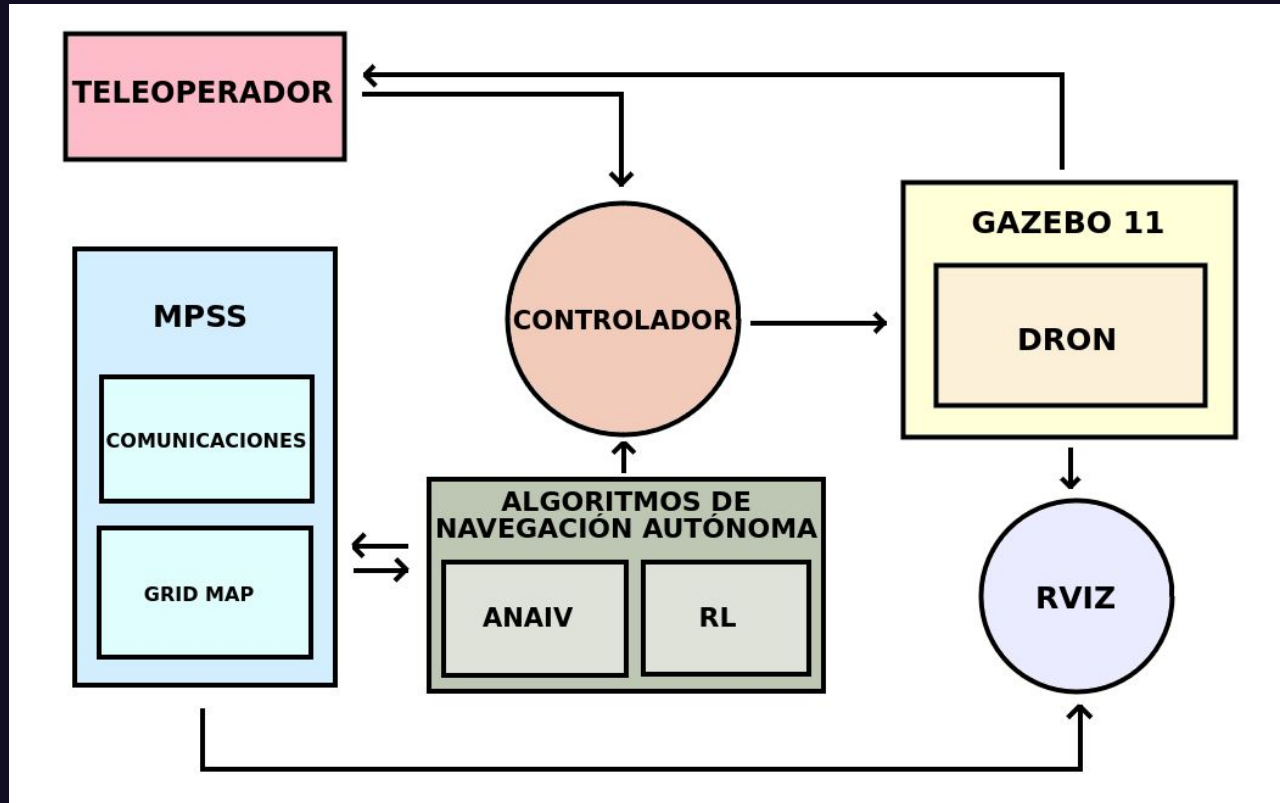
► DESCRIPCIÓN DEL PROBLEMA

Navegar autónomamente hacia un emisor de señal radio frecuencia, empleando técnicas de aprendizaje por refuerzo, de modo se plantean los siguientes **objetivos**:

- Desarrollar de una aplicación de teleoperación.
- Obtener un modelo de propagación de señales.
- Desarrollar un comportamiento autónomo para navegar hacia el transmisor (sin y con obstáculos).
- Comparar las soluciones planteadas.



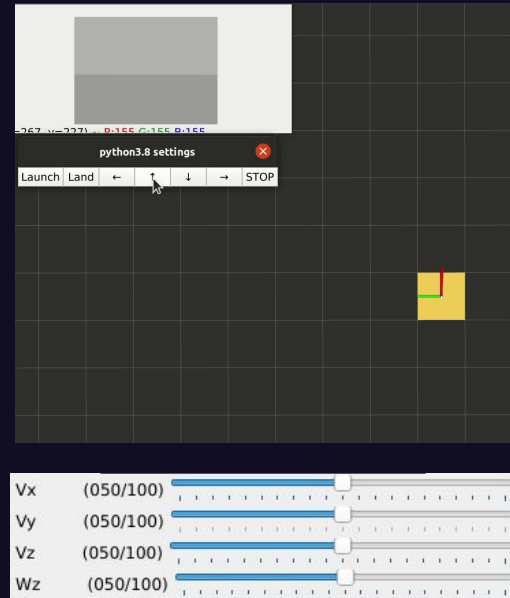
► ARQUITECTURA PROPUESTA



► TELEOPERADOR

Interfaz gráfica capaz de controlar al dron:

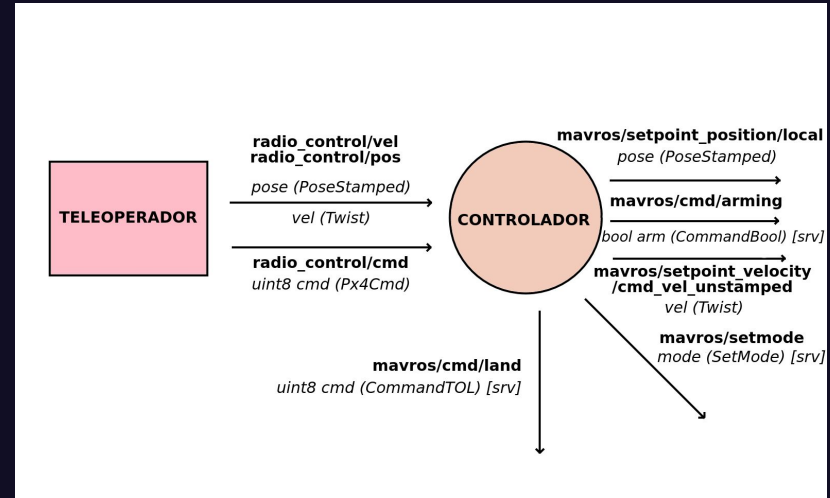
- Cámara.
- Comandos, posiciones y velocidades.
- Órdenes predefinidas.
- RVIZ



► CONTROLADOR

Gestiona la comunicación del resto de bloques con PX4 (OS que permite controlar el vuelo en el dron):

- Listo para recibir comandos (pose/vel/cmd).
- Traduce órdenes a MAVROS (protocolo de comunicaciones MAVLINK sobre ROS)
- Se conecta con el teleoperador y el bloque de algoritmos de navegación autónoma.



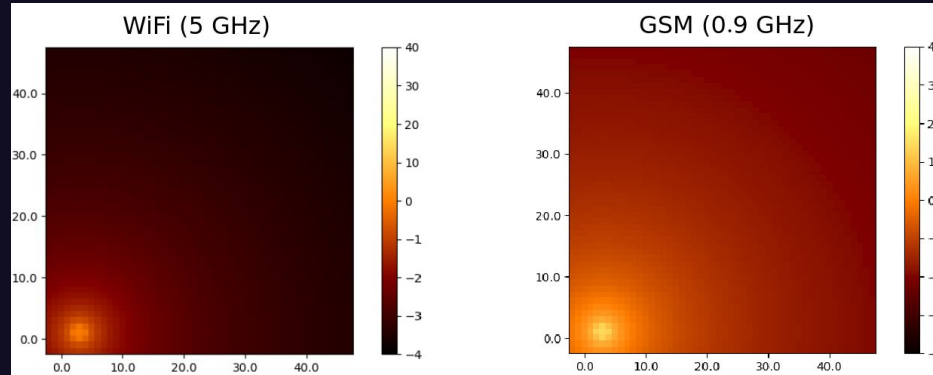
► MODELO DE PROPAGACIÓN DE SEÑAL

Se desarrolla un modelo de propagación de señales, basado en la aproximación de Friis, en python, para generar poder generar mapas de propagación (calor) sobre los que programar soluciones:

- Interfaz gráfica.
- Integración en nodo ROS para la comunidad.

$$P_r = P_t \cdot \left(\frac{G_t G_r \lambda^2}{(4\pi)^2 d^n L} \right)$$

► MODELO DE PROPAGACIÓN DE SEÑAL



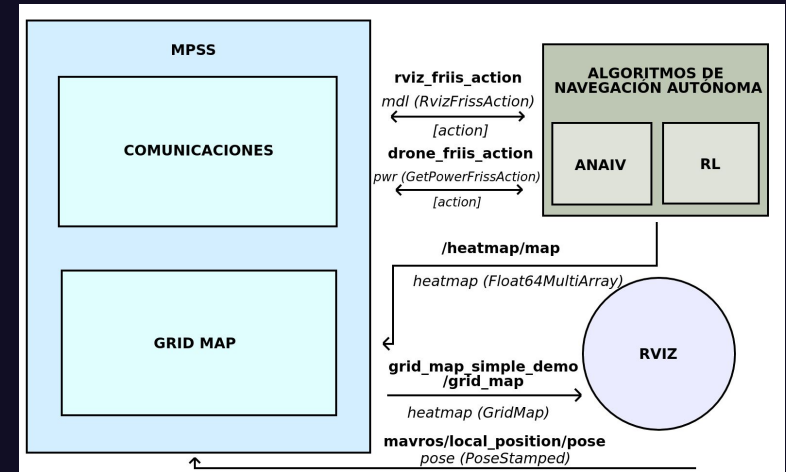
PowerTransmitter (W)	<input type="range"/>	10
GainTransmitter (W)	<input type="range"/>	1.5
GainReceiver (W)	<input type="range"/>	2
OtherLosses [L]	<input type="range"/>	1
PathLossExponent [n]	<input type="range"/>	2.0
MapResolution MapSize (AXA)	<input type="range"/>	10 50
		<input type="button" value="Set"/>

Frequency (GHz)	<input type="range"/>	5
Frequency (GHz)	<input type="range"/>	0.9

► MPSS - MÓDULO DE PROPAGACIÓN DE SEÑALES SIMULADO

Integra el modelo de Friis en un nodo ROS (a través de acciones y topics) para suplir una doble función:

- Proporcionar la potencia de la señal, dada una posición del mapa de calor.
- Representar gráficamente el mapa mediante un plugin de RVIZ (grid map).

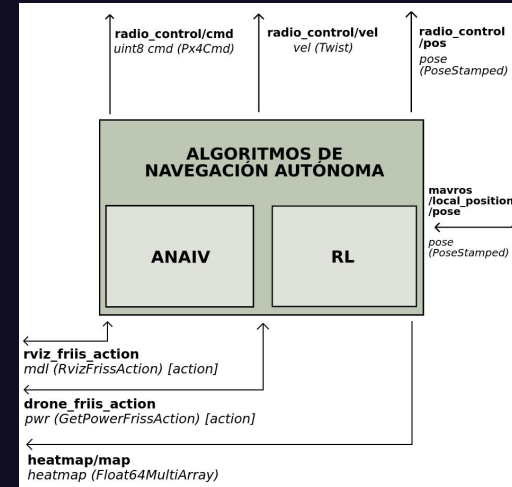


► ALGORITMOS DE NAVEGACIÓN AUTÓNOMA

Conforma el conjunto de soluciones propuestas, que dotan al dron de la capacidad para planificar y moverse, sin la intervención directa de un operador:

- ANAIV, acrónimo de “Algoritmos de Navegación Autónoma basados en la Información de la Vecindad”, enfoque tradicional.
- RL, o *Reinforcement Learning*.

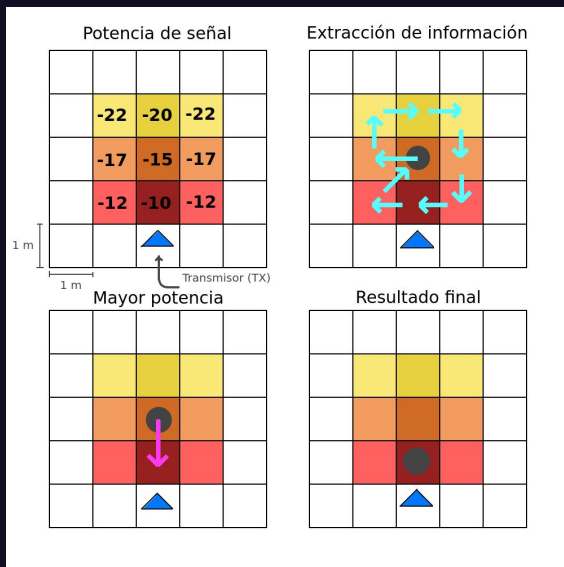
Se comunican con el bloque MPSS para navegar hacia el transmisor, que modela la propagación de una señal en celdillas de 1x1 m (configurable).



▶ ALGORITMOS DE NAVEGACIÓN AUTÓNOMA BASADOS EN LA INFORMACIÓN DE LA VECINDAD

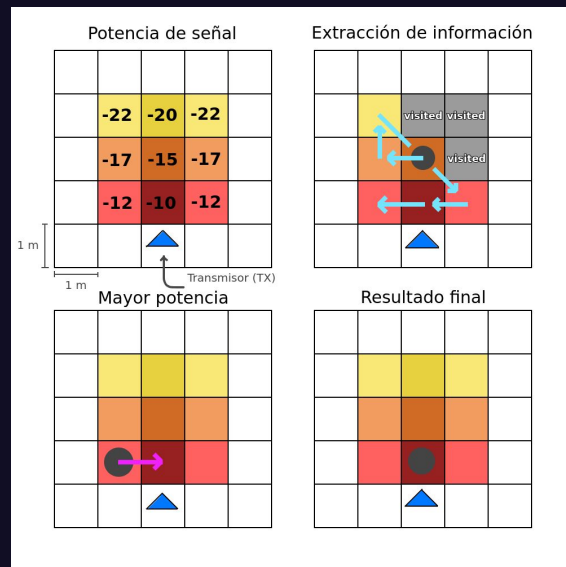
ANAIV

- Visita todos los vecinos.
- Condición de parada basada en objetivo anterior y actual.



ANAIV +

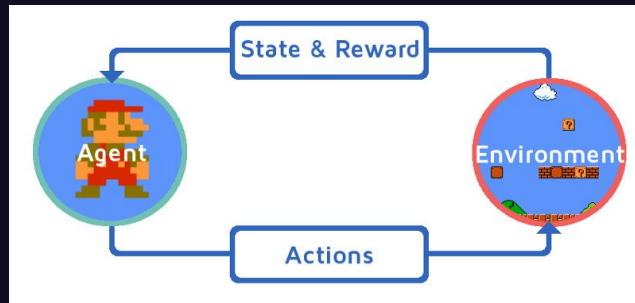
- No revisita, guarda información de 18 poses.
- Misma condición de parada.



► ALGORITMO RL

Está basado en Q-Learning, que es un algoritmo iterativo de aprendizaje por refuerzo basado en estados y acciones, donde a través de una función de recompensa, se converge hacia qué acción es la mejor para según qué estado se encuentre el agente. Para ello tenemos:

- Fase de entrenamiento, donde se distinguen episodios (o intervalo de eventos dentro del escenario), en los que el agente aprende a través de la toma de acciones.
- Fase de inferencia, donde el agente actúa en base a lo que aprendió durante el entrenamiento.



► ALGORITMO RL

En el caso planteado, para llegar al transmisor, describimos los estados como coordenadas del mapa de calor, y las acciones como los movimientos que el dron puede realizar.

La función de recompensa sigue esta fórmula:

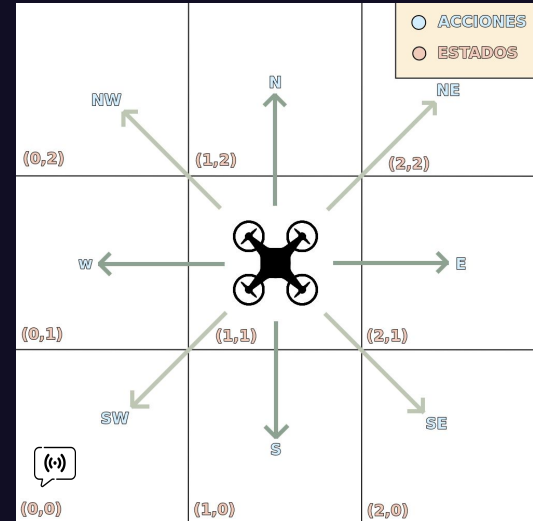
$$R = (P(t + 1) - P(t)) \times \text{Factor}$$

Positivas:

- Moverse hacia mayor potencia de señal.

Negativas:

- Moverse hacia menor potencia de señal.
- Salir del mapa de calor.



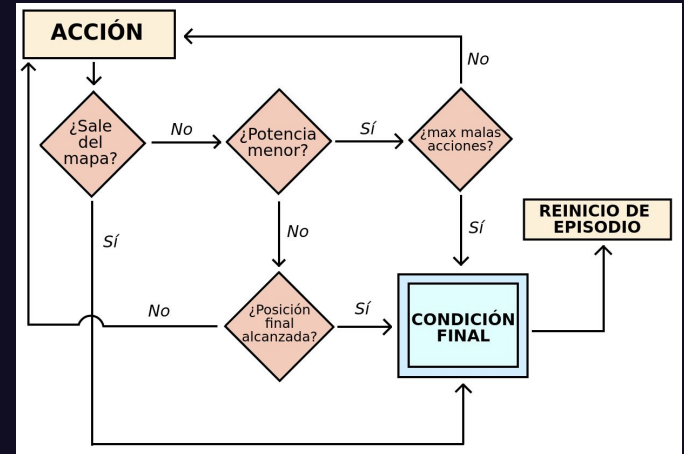
► ALGORITMO RL

Las condiciones de finalización de episodio son:

- Salir del mapa
- Llegar al transmisor
- (Opcional) n-malas acciones consecutivas

Cuando se reinicia el episodio, se selecciona unas nuevas coordenadas (o nuevo punto de entrenamiento), con el fin de evitar el *overfitting*.

En la fase de inferencia, se prueba al agente en posiciones nuevas, no empleadas durante el entrenamiento.

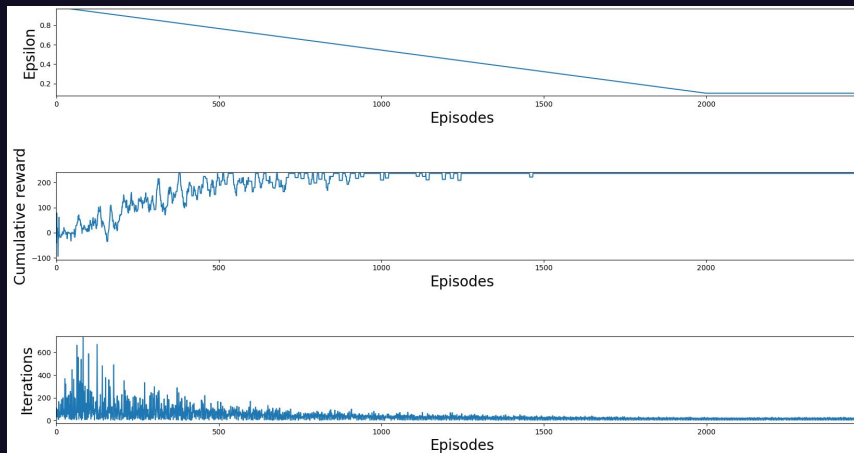


► ALGORITMO RL

Gráficas extraídas tras cada entrenamiento:

- **Epsilon**, determina la fase de exploración, donde el agente toma acciones aleatorias o acciones determinadas.
- **Recompensa acumulada**, que tiende a aumentar siempre que la acción acerque al dron al transmisor.
- **Iteraciones**, o número de acciones tomadas hasta finalizar el episodio.

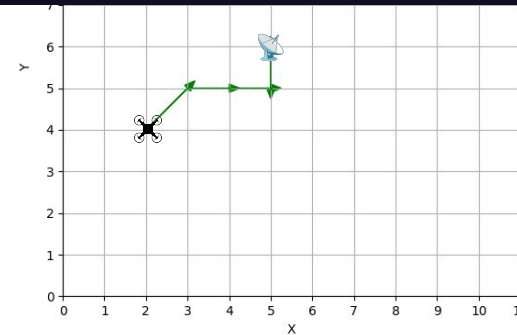
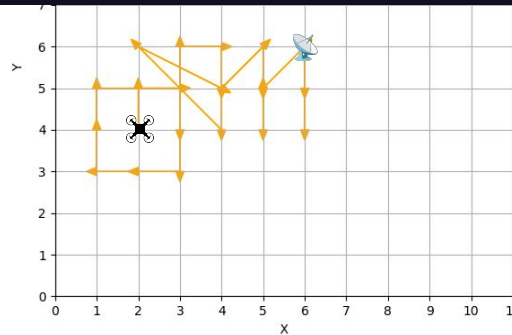
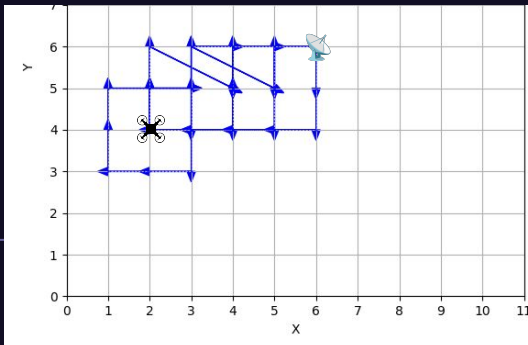
Por ello **converge** cuando la recompensa acumulada se estabiliza de forma creciente (se mueve hacia más potencia) y cuando el número de iteraciones se reduce (llega al objetivo en menos pasos).



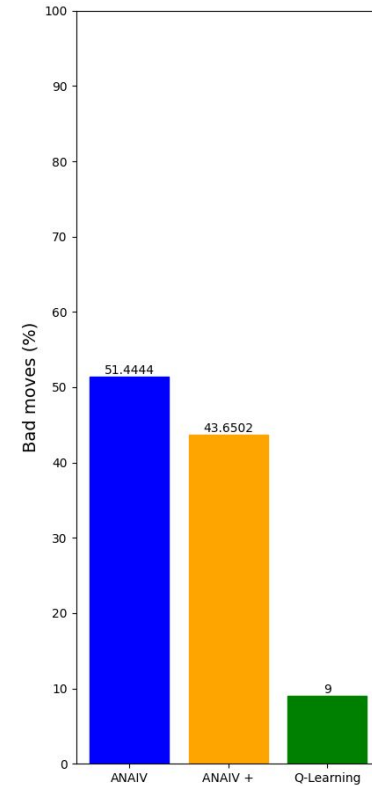
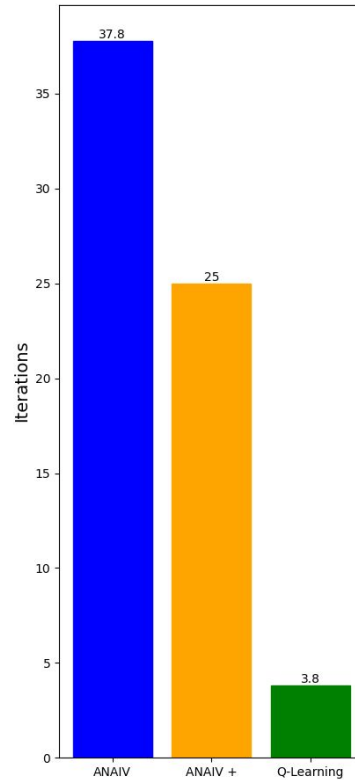
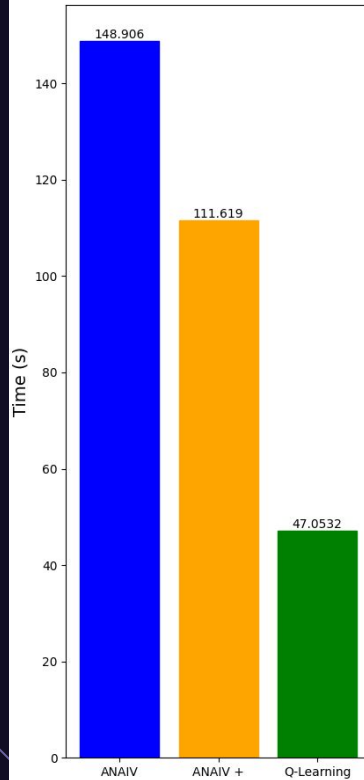
► PRIMER EXPERIMENTO

Se comparan los algoritmos mencionados en distintos escenarios:

- Transmisor en el centro y en la esquina.
- Mapas de distinto tamaño (12x12 y 30x30).



► PRIMER EXPERIMENTO



► PRIMER EXPERIMENTO



Universidad
Rey Juan Carlos

Escuela de Ingeniería
de Fuenlabrada

Navegación autónoma de un dron para localizar un transmisor de radio frecuencia basado en aprendizaje por refuerzo

Cristian Sánchez Rodríguez

Tutor: Roberto Calvo Palomino

► ENTORNOS DINÁMICOS



OBSTÁCULOS

En entornos reales, cabe esperar la presencia de elementos obstructivos durante la navegación.



NUEVO ENFOQUE

Puesto que el dron aprende el escenario concreto, es inviable predecir la posición de los obstáculos en el escenario final, por ello se debe encontrar nuevas soluciones.



EFEECTO EN LA SEÑAL

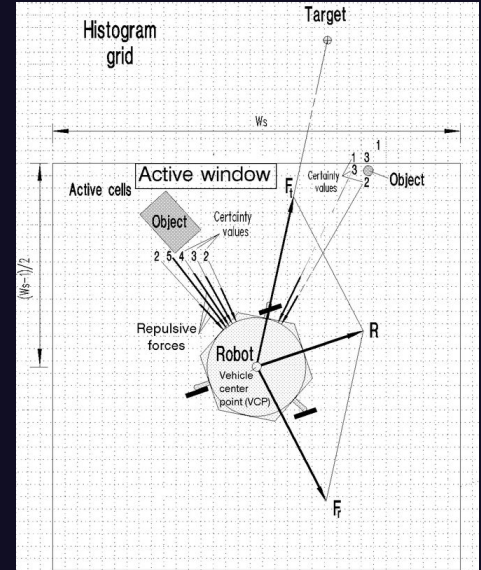
Estos, o bien por otras causas, pueden provocar interferencias en la señal, y por ende en su propagación.



ALGORITMO HÍBRIDO

Con sensorización (p.ej LiDAR) se plantea dotar al dron de un comportamiento para esquivar obstáculos cuando estén cerca.

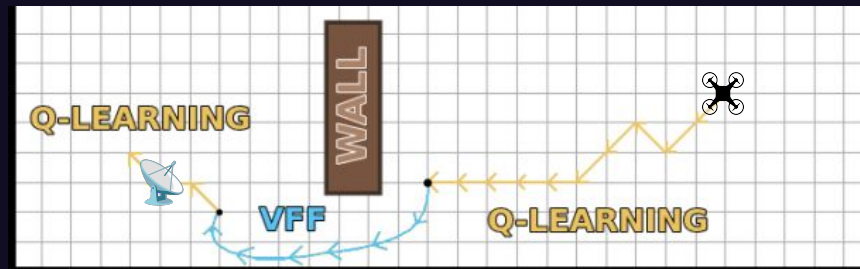
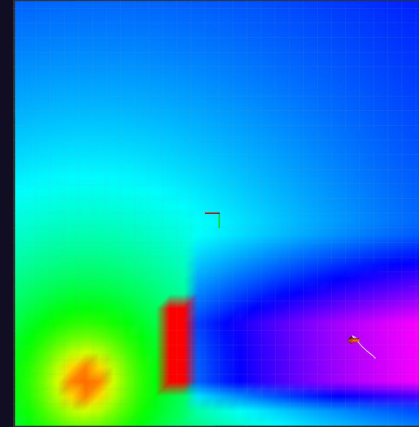
Por ello se decide opta por VFF, que permite sortear obstáculos en base a fuerza virtual resultante que proviene de fuerzas atractivas y repulsivas.



ALGORITMO HÍBRIDO

Combina Q-Learning con VFF:

- Fuerza atractiva, se genera siguiendo la tendencia de Q-Learning.
- Fuerza repulsiva, de cada celda ocupada por un obstáculo.
- Distancia de activación (3 m)



► ALGORITMO HÍBRIDO



► CONCLUSIONES

Tras la realización de este trabajo se puede afirmar que:

- Se ha desarrollado con éxito una interfaz gráfica capaz de interactuar con el dron.
- Se ha creado un nodo ROS open source destinado a trabajar con un modelo de propagación de señales.
- Se han desarrollado algoritmos de navegación autónoma tanto para escenarios sin obstáculos como para escenarios con ellos.
- Se ha determinado en base a métricas de rendimiento, que algoritmo rinde mejor.

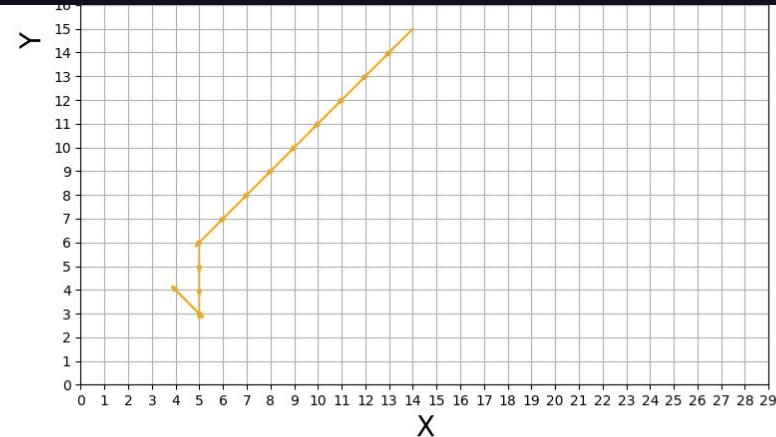
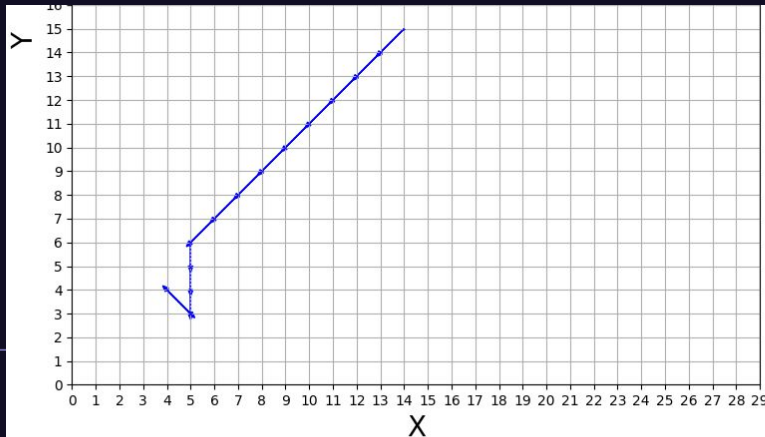
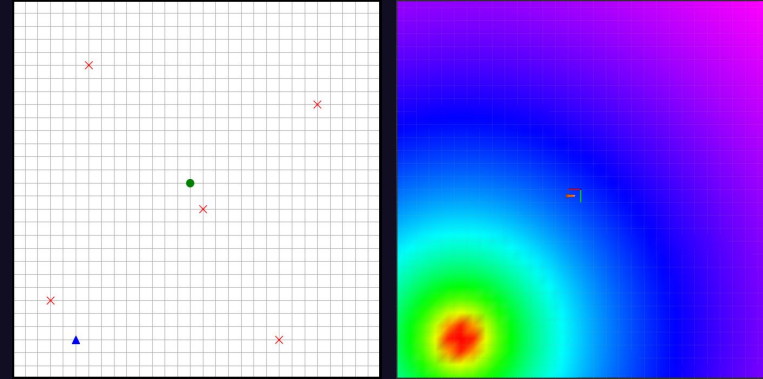
► **¡MUCHAS GRACIAS!**



► SEGUNDO EXPERIMENTO

Consiste en comparar el funcionamiento de un agente entrenado con una señal determinada en inferencia (Q-Learning), tras modificar las características de señal (Q-Learning 2):

- Potencia transmisor X2.
- Frecuencia de 2.4 GHz a 5 GHz.



► SEGUNDO EXPERIMENTO

