

INTRODUCCIÓN A LA ROBÓTICA INTELIGENTE



TRABAJO OBLIGATORIO 2

Grupo 1:

Javier López Iniesta Díaz del Campo
Jorge Quijorna Santos
Jorge Romeo Terciado

29 de Mayo de 2020

Índice

1. Introducción	2
2. Mapa	3
3. Red neuronal artificial estática	4
3.1. Prueba 1	4
3.2. Prueba 2	7
3.3. Prueba 3	9
3.4. Prueba 4	11
3.5. Prueba 5	13
3.6. Prueba 6	15
4. Entornos aleatorios ANN	17
4.1. Posición aleatoria	17
4.2. Objetos aleatorios	18
5. Red neuronal artificial dinámica	19
5.1. Prueba 1	19
5.2. Prueba 2	21
5.3. Prueba 3	23
5.4. Prueba 4	25
6. Entornos aleatorios CTRNN	27
6.1. Objetos aleatorios	27
7. Gráficas de interés	28
8. Comparación de resultados entre ANN y CTRNN	30
9. Problemas encontrados	31
10. Futuras implementaciones	32
11. Conclusión	33
12. Bibliografía	34

1. Introducción

Actualmente, la robótica evolutiva es muy importante en robots que se entrenan para que realicen una tarea específica y que, para ello, se comportan de una manera u otra en función del entorno que les rodea, un entorno que podría estar en continuo cambio. Muchas aplicaciones requieren el uso de este tipo de robots capaces de aprender para la consecución de una serie de objetivos, teniéndolos previamente que evolucionar muchas veces en espacios aleatorios y con diversos obstáculos. El aprendizaje de robots es, por tanto, una manera de adaptarse a distintas situaciones para la realización de una tarea y los algoritmos evolutivos son una técnica a utilizar para que los robots puedan aprender por sí mismos.

Por ello, en la segunda parte de la asignatura de *“Introducción a la Robótica Inteligente”* se ha propuesto la implementación en el simulador **IRSIM** de un robot, mediante redes neuronales artificiales estática y dinámicas, para así poder comparar el funcionamiento entre ambas redes neuronales.

El planteamiento inicial del trabajo, ha sido la realización de un robot, que oscilará entre 2 luces. Este robot cuenta con una batería, a la que debe ir a recargarse si el nivel de batería se encuentra por debajo de un determinado umbral.

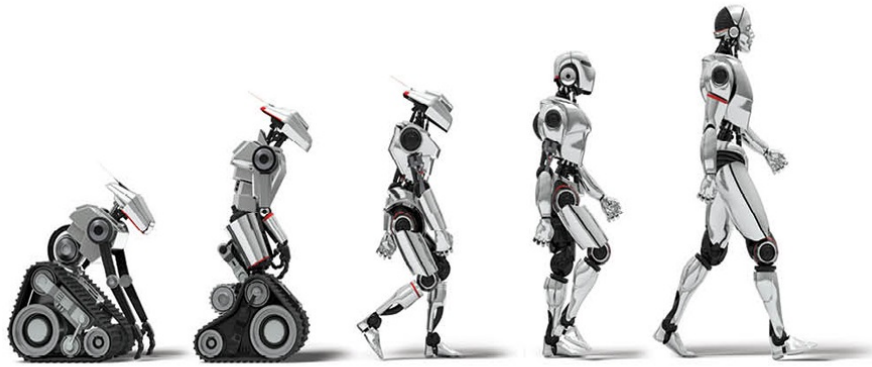


Figura 1: Robot Evolution

2. Mapa

La arena en la cual se ha evolucionado el robot no cuenta con ningún tipo de obstáculos. No obstante, hemos añadido los siguientes objetos en el fichero de parámetros, para poder resolver el problema planteado, luces de distintos colores y una batería como parámetros. En las primeras pruebas únicamente contaban con:

- 1 luz amarilla: Su función principal es actuar como recarga de la batería del robot.
- 1 luz azul: Cuando el robot esté cargado deberá ir hacia la luz, que permanecerá fija, sin ningún parpadeo.

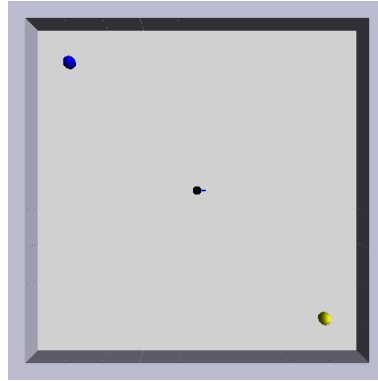
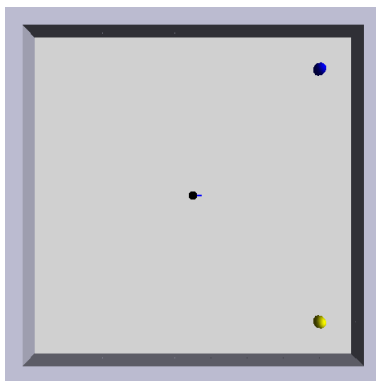


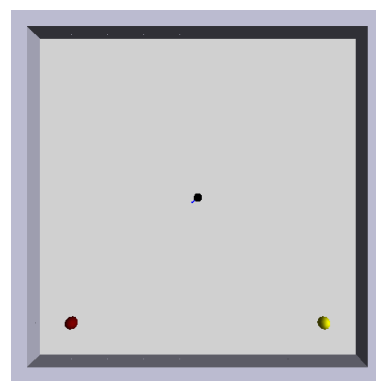
Figura 2: Mapa inicial

Posteriormente, una vez conseguido el funcionamiento básico del robot con la batería se ha rediseñado el mapa, añadiendo una luz roja y un parpadeo a la luz roja y azul:

- 1 luz amarilla: Su función principal es actuar como recarga de la batería del robot.
- 1 luz azul: Tendrá un parpadeo constante cada 500 timesteps, de forma que cuando esté encendida la luz azul, la luz roja, permanecerá apagada.
- 1 luz roja: Al igual que la luz azul, también parpadeará cada 500 timesteps, para permitir que el robot oscile entre ambas luces.



(a) Mapa con luz azul encendida



(b) Mapa con luz roja encendida

Figura 3: Mapa final

Finalmente, se ha entrenado al robot en entornos aleatorios, donde la posición inicial del robot se encontraba dentro en un cuadrado de la arena. Así como, la posición de los objetos utilizados también ha sido generada de forma aleatoria.

3. Red neuronal artificial estática

Para conseguir que el robot resuelva el objetivo planteado inicialmente mediante una “Artificial Neural Network” (ANN), se ha ido diseñando la función de fitness y la arquitectura final, en distintas etapas, elevando la complejidad poco a poco, para comprobar la influencia de cada pequeño cambio en nuestro experimento. Así para elegir el robot que resuelve nuestro objetivo, hemos realizado numerosas pruebas, como se detallan a continuación:

3.1. Prueba 1

En esta primera prueba únicamente se ha utilizado una luz amarilla, y una luz azul, que permanecen fijas (sin ningún parpadeo). La idea es sencilla, el robot deberá acercarse a la luz azul, mientras el nivel de batería se encuentre por encima de un umbral. En el momento en el que el nivel de batería, sea menor a un determinado umbral, deberá aproximarse a la luz amarilla que actuará como cargador de la batería del robot.

La arquitectura neuronal del robot implementada para esta primera prueba se presenta en la Figura 4. Está formada por cinco capas neuronales, cuatro sensoriales y una motora. Todas las neuronas de las capas sensoriales se encuentran conectadas a las neuronas de la capa motora.

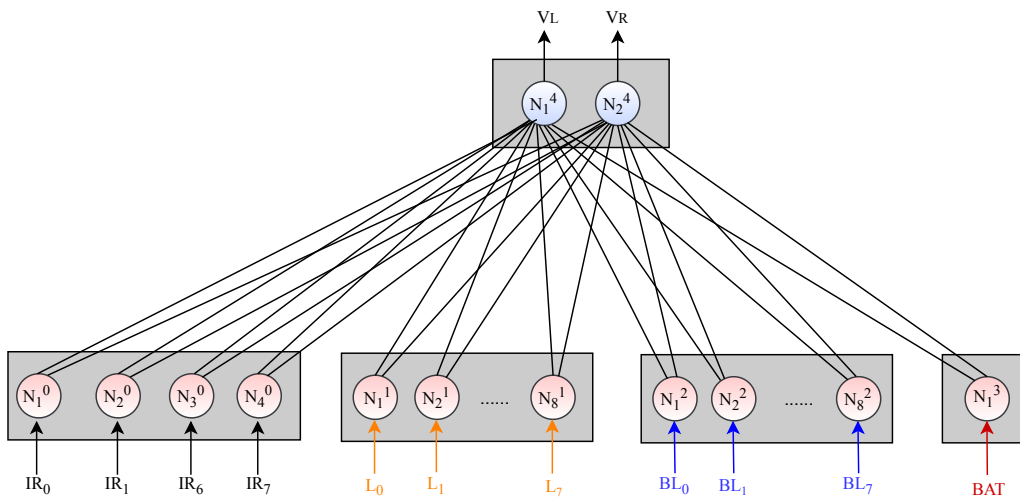


Figura 4: Arquitectura neuronal 1

Para esta prueba se ha empleado el siguiente cromosoma:

- Cromosoma: 44
- Población: 100
- Mutación: 20 %
- Tiempo evaluación: 200 s
- Elites: 6

La función de fitness utilizada para evaluar esta prueba es la siguiente:

$$F = \begin{cases} \frac{\sum_{i=0}^{N_{steps}} \left[\frac{1}{4} V (1 - \sqrt{\Delta\nu}) (M_r \cdot M_l) + \frac{1}{4} B + \frac{1}{4} (BL_0 + BL_7) \right]}{N_{steps}} \left(1 - \min \left\{ \frac{N_{coll}, 10}{10} \right\} \right) & \text{si } B > Battery_Threshold \\ \frac{\sum_{i=0}^{N_{steps}} \left[\frac{1}{4} V (1 - \sqrt{\Delta\nu}) (M_r \cdot M_l) + \frac{1}{4} B + \frac{1}{4} (L_0 + L_7) \right]}{N_{steps}} \left(1 - \min \left\{ \frac{N_{coll}, 10}{10} \right\} \right) & \text{si } B < Battery_Threshold \end{cases} \quad (1)$$

Donde V es la suma de los valores absolutos de cada una de las velocidades de cada motor menos 0.5. Por otro lado, $\Delta\nu$ es el valor absoluto de la diferencia entre las velocidades, M_r y M_l son las velocidades del motor derecho e izquierdo respectivamente, normalizadas entre 0 y 1. Por otro lado, B , representa a la batería de luz amarilla. Mientras que, BL hace referencia a la luz azul (Blue Light), y L , a la luz amarilla (Light), ambos normalizados entre 0 y 1, al tratarse de dos sensores de luz, junto con un parámetro de ajuste a la ecuación de fitness igual a 0.5. Por lo tanto, dichos parámetros son igual a:

$$V = |M_r - 0,5| + |M_l - 0,5| \quad (2)$$

$$\Delta\nu = |M_r - M_l| \quad (3)$$

Si analizamos las diferentes componentes de la función de fitness, podemos observar que:

- La componente V se maximiza, cuando la rotación de ambas ruedas es máxima, siendo el sentido de giro de éstas indiferente. Así, se pretende beneficiar el hecho de que el robot se encuentre siempre en movimiento.
- La componente $(1 - \sqrt{\Delta\nu})$ favorece el hecho de que ambas ruedas giren en el mismo sentido, siendo independiente del sentido de giro y de la velocidad de los motores.
- La componente $(M_r \cdot M_l)$, se maximiza cuando el robot se mueve hacia delante a máxima velocidad.
- El término B se maximiza cuando el nivel de batería es mayor. Así, este término maximiza la fitness cuando el robot se encuentra completamente cargado ($B=1$).
- Las componentes de $(BL_0 + BL_7)$ y $(L_0 + L_7)$, se utilizan para que el robot se aproxime a las luces azul y amarilla respectivamente. Estos términos, se maximizarán cuando el robot se encuentre de frente, con los sensores 0 y 7 apuntando hacia la luz y esté lo más cerca posible de esta.

Los resultados de la fitness obtenidos, como se puede apreciar en la Figura 5, muestran que el caso mejor empieza en un valor bajo, y en conjunto se podría considerar constante y sin apenas progresión. La media se encuentra algo inferior a 0.2. Mientras que el peor individuo es siempre igual a cero.

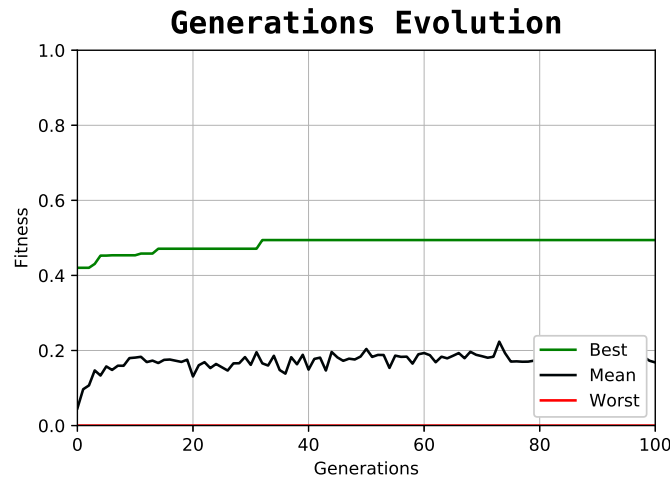
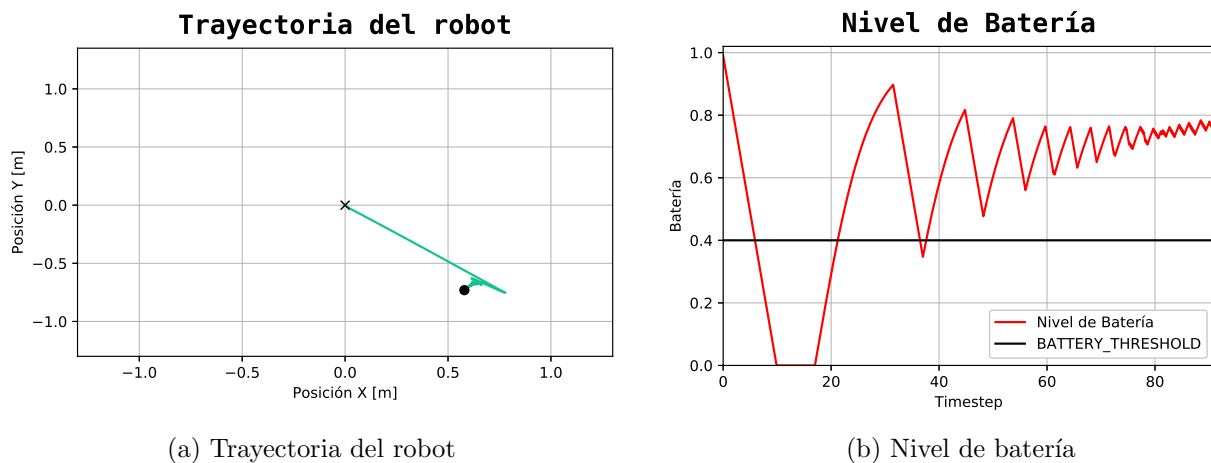


Figura 5: Evolución de las generaciones durante la prueba 1

Si ejecutamos dicho experimento, nos encontramos con un robot que partiendo de una posición inicial (marcada con una “X”) describe una trayectoria lineal, hacia la luz amarilla (Figura 6a), sin tener en cuenta la luz azul y realizando constantemente un movimiento hacia delante y detrás, de forma que nunca permanece parado. Se puede observar en la Figura 6b, que el nivel de batería, únicamente se descarga una vez, y una vez cargada la batería, esta oscila entorno a un valor próximo a 0.8. A medida que transcurren los timesteps la carga y descarga de esta es cada vez más pequeña. Este comportamiento del robot, se debe a que al sumarle a la función de fitness un término de la batería por un factor (0.25 en este caso) siempre busca estar con la máxima batería posible. Además, el robot no encuentra ningún beneficio recorriendo el mapa en busca de la luz azul, ya que prefiere estar cargado constantemente.



(a) Trayectoria del robot

(b) Nivel de batería

Figura 6: Gráficas de interés del mejor individuo de la generación 100. Prueba 1

3.2. Prueba 2

Con el fin de solucionar los problemas del robot de la prueba 1, se ha eliminado el término de la función de fitness de la batería y la componente de $M_r \cdot M_l$ con el fin de tratar aportar mayor peso al sensor de la batería y los sensores de luz amarilla. Además, se ha cambiado la arquitectura neuronal del robot. Para ello, se ha creado una capa asociativa adicional, que tiene como entrada los sensores de luz amarilla y la batería.

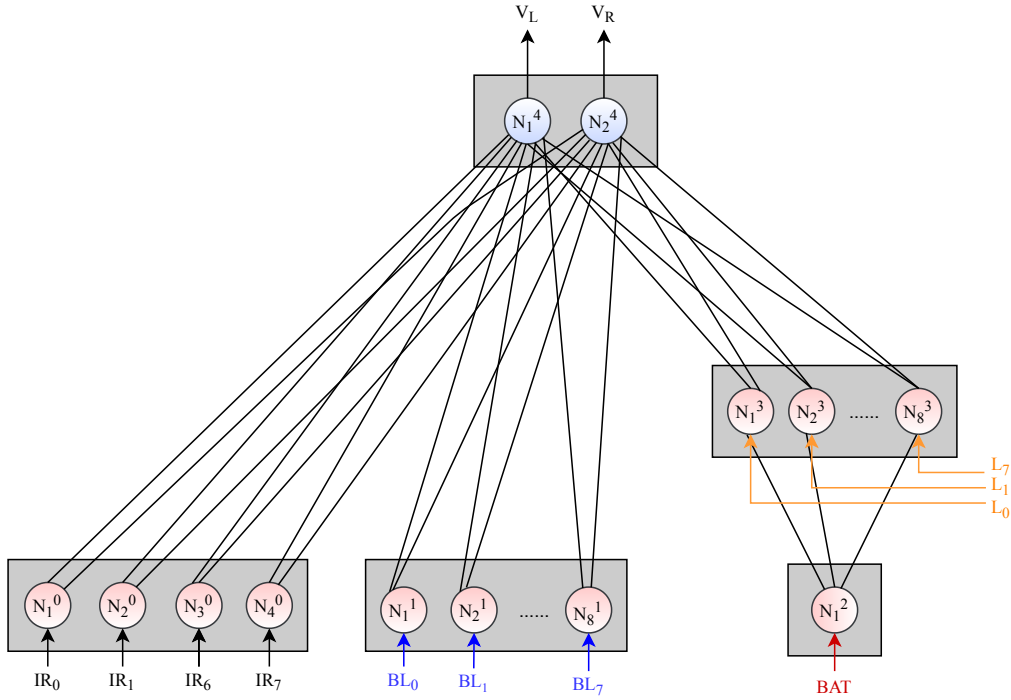


Figura 7: Arquitectura neuronal 2

Por otro lado, el cromosoma empleado para esta segunda prueba ha sido el siguiente:

- Cromosoma: 58
- Población: 100
- Mutación: 20 %
- Tiempo evaluación: 200 s
- Elites: 6

Para esta prueba se ha implementado la siguiente función de fitness, siendo similar a la de la prueba anterior. Sin embargo, se han eliminado varios términos:

$$F = \begin{cases} \frac{\sum_{i=0}^{N_{steps}} \left[\frac{1}{4}V(1 - \sqrt{\Delta\nu}) + \frac{3}{8}(BL_0 + BL_7) \right]}{N_{steps}} \left(1 - \min \left\{ \frac{N_{coll}, 10}{10} \right\} \right) & \text{si } B > Battery_Threshold \\ \frac{\sum_{i=0}^{N_{steps}} \left[\frac{1}{4}V(1 - \sqrt{\Delta\nu}) + \frac{3}{8}(L_0 + L_7) \right]}{N_{steps}} \left(1 - \min \left\{ \frac{N_{coll}, 10}{10} \right\} \right) & \text{si } B < Battery_Threshold \end{cases} \quad (4)$$

Al igual que en el caso anterior, se ha intentado buscar que el robot se mueva en línea recta y a máxima velocidad, gracias al término $V(1 - \sqrt{\Delta\nu})$. También se ha incluido un término para poder aproximarse a las luces azul o amarilla, en caso de que el nivel de batería se encuentre por encima o por debajo del umbral, respectivamente.

En la Figura 8 observamos un caso similar al de la prueba anterior. No obstante, esta vez el mejor individuo crece bastante en las primeras generaciones, y aumentando ligeramente con el paso de las generaciones, hasta llegar a algo más de 0.5. Se aprecia una media similar aunque algo inferior. Para el caso peor la fitness sigue siendo igual a 0 en todas las generaciones.

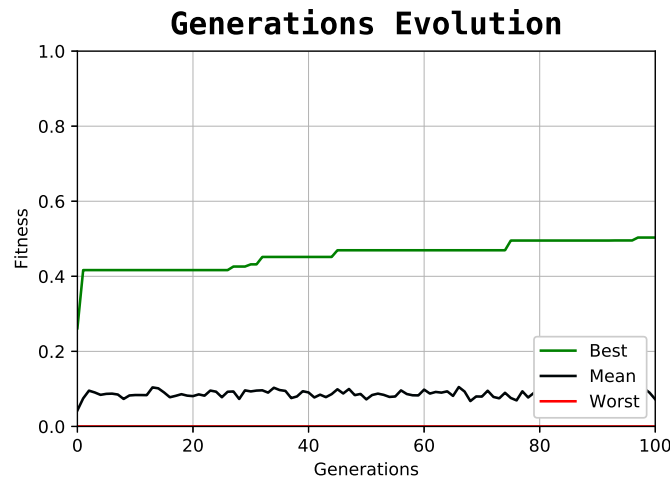


Figura 8: Evolución de las generaciones durante la prueba 2

Respecto a la trayectoria realizada por el robot, podemos apreciar que sigue una ruta similar a la descrita en la prueba 1. Aunque en este caso, comienza yendo en la dirección de la luz azul marcha atrás. Durante el trayecto, se descarga completamente la batería y se dirigirá, siguiendo una trayectoria lineal con velocidad máxima, hacia la luz amarilla (Figura 9a). Así, una vez que llega a la luz que le permite recargar la batería realiza movimientos hacia delante y hacia atrás, de forma que como se observa en la Figura 9b no consigue llegar a conseguir un nivel de batería por encima del umbral. Oscilando en torno a un nivel igual a 0.3.

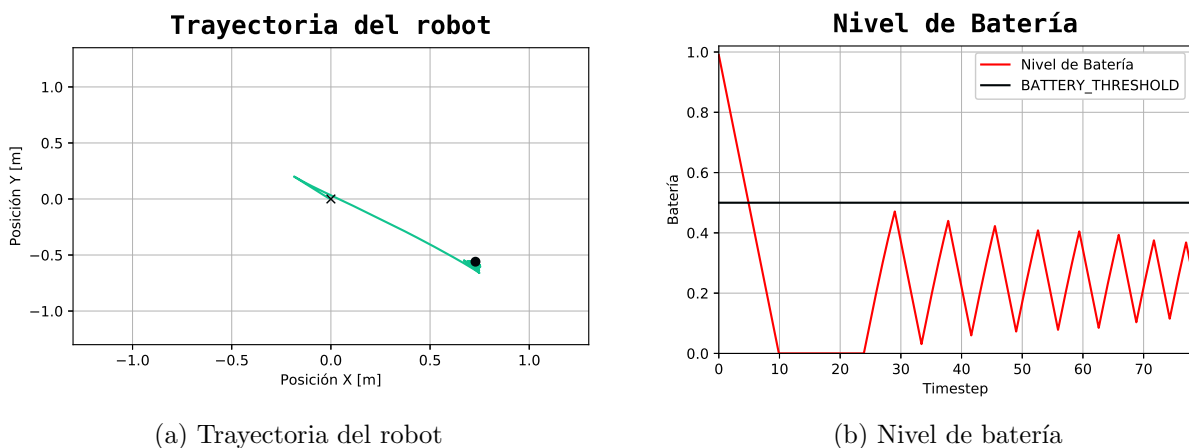


Figura 9: Gráficas de interés del mejor individuo de la generación 100. Prueba 2

Por lo tanto, con esta prueba se ha conseguido ir inicialmente a la luz azul. Aunque, al igual que en el caso anterior, una vez que llega a la luz amarilla (recarga de batería) no consigue alejarse de allí, debido a la naturaleza “vaga” de este tipo de robots, permaneciendo continuamente por debajo del umbral de batería, y por lo tanto tratando de acercarse a la luz amarilla.

3.3. Prueba 3

Con el fin de conseguir el problema planteado inicialmente, en esta prueba se cambia nuevamente la arquitectura neuronal (Figura 10), así ahora la capa asociativa tiene como entrada los sensores de luz amarillas y azules, y la batería.

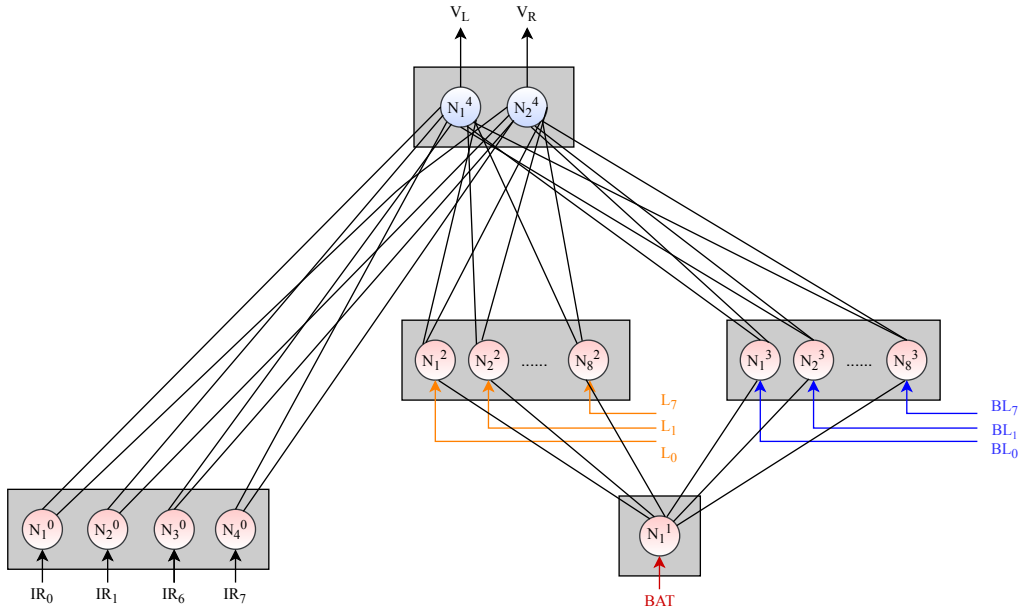


Figura 10: Arquitectura neuronal 3

Así, el cromosoma empleado para esta evolución ha sido el siguiente:

- Cromosoma: 74
- Población: 100
- Mutación: 5 %
- Tiempo evaluación: 300 s
- Elites: 4

Por otro lado, la función de fitness, es la misma que en la prueba 2 (Ecuación 4). Aunque, con el fin de conseguir que el robot se cargue completamente, se ha incorporado un flag, llamado **BatteryFlag**, que estará activo cuando el nivel de batería se encuentre por debajo del umbral y mientras esta se esté cargando, hasta que llega a 0.95. Cuando el robot termina de cargar la batería, se desactiva el flag. Como se puede observar en la Ecuación 5:

$$BatteryFlag = \begin{cases} 1 & \text{si } B < Battery_Threshold \\ 1 & \text{si } BatteryFlag_{ON} \text{ y } B < 0,95 \\ 0 & \text{resto} \end{cases} \quad (5)$$

Así, incorporando este flag a la función de fitness, quedará de la siguiente forma:

$$F = \begin{cases} \frac{\sum_{i=0}^{N_{steps}} \left[\frac{1}{4} V(1 - \sqrt{\Delta\nu}) + \frac{3}{8} (BL_0 + BL_7) \right]}{N_{steps}} \left(1 - \min \left\{ \frac{N_{coll}, 10}{10} \right\} \right) & \text{si } BatteryFlag_{OFF} \\ \frac{\sum_{i=0}^{N_{steps}} \left[\frac{1}{4} V(1 - \sqrt{\Delta\nu}) + \frac{3}{8} (L_0 + L_7) \right]}{N_{steps}} \left(1 - \min \left\{ \frac{N_{coll}, 10}{10} \right\} \right) & \text{si } BatteryFlag_{ON} \end{cases} \quad (6)$$

Los resultados tras la evolución de este robot, se muestra a continuación, en la Figura 11. Se puede ver, que el mejor individuo de cada generación, tiene una fitness aproximadamente constante en todas las generaciones, cercana a 0.4 (algo inferior a la prueba anterior). Por otro lado, la media y el peor individuo son similares a los de las pruebas previas.

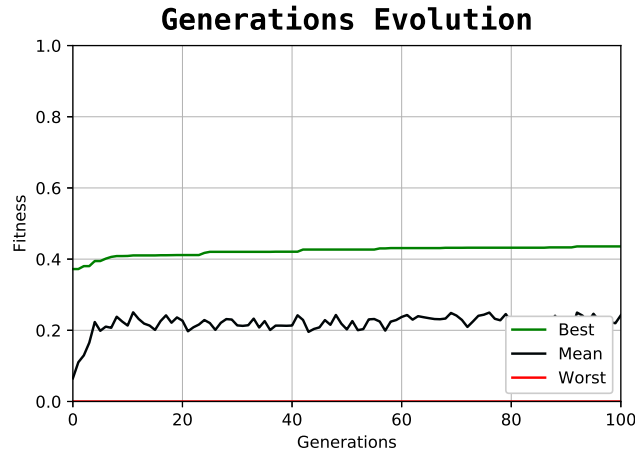


Figura 11: Evolución de las generaciones durante la prueba 3

Al ejecutar este experimento, se observan unos resultados nada satisfactorios. Inicialmente se desplaza hacia una zona donde no existe ningún tipo de luz, realizando una trayectoria extraña, hasta que se le acaba la batería. En ese momento, se dirige a la zona de carga, donde consigue cargarse hasta el nivel de 0.95 deseado. Aunque nuevamente, no consigue alejarse de la luz amarilla cuando termina de cargarse, realizando los movimientos hacia delante y hacia detrás, observados en las Figuras 6a y 9a. Por tanto, este robot también es descartado.

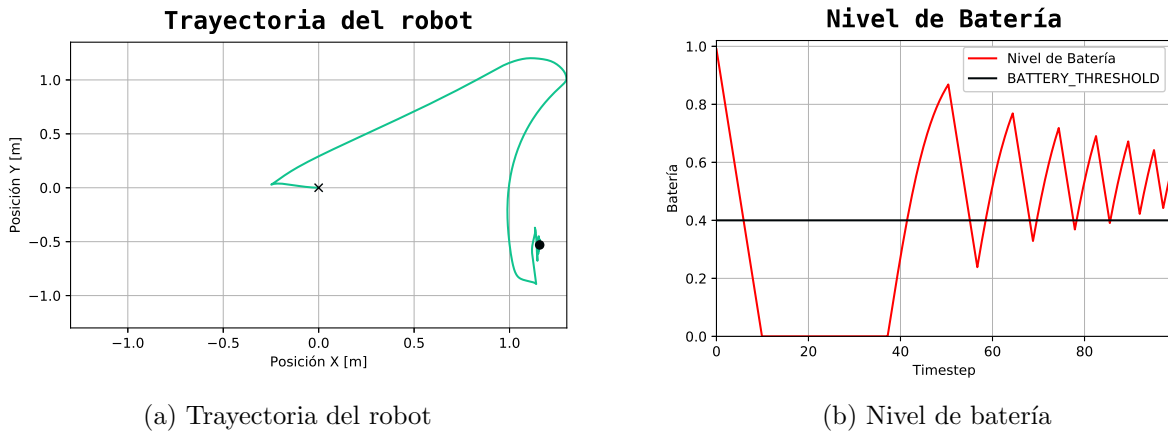


Figura 12: Gráficas de interés del mejor individuo de la generación 100. Prueba 3

3.4. Prueba 4

En la siguiente prueba, únicamente se ha cambiado la arquitectura neuronal utilizada, manteniendo la misma función de fitness que en la prueba anterior (Ecuación 6). En este caso, se ha incorporado una capa oculta, que tiene como entrada, las salidas de las capas de los sensores de luz amarillas y azules y la batería, y que no cuenta con ningún tipo de entrada sensorial.

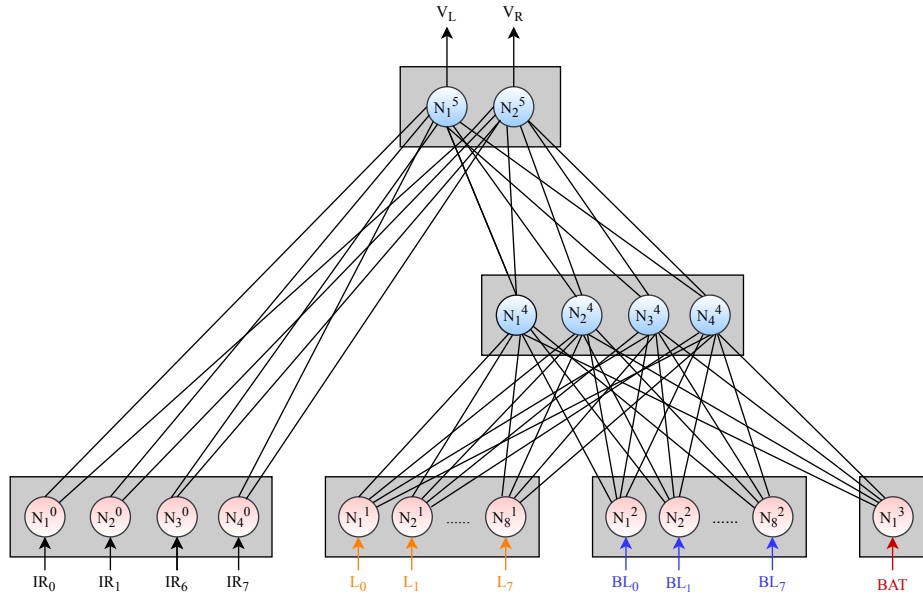


Figura 13: Arquitectura neuronal 4

El cromosoma empleado para esta prueba ha sido el siguiente:

- Cromosoma: 90
- Población: 100
- Mutación: 5 %
- Tiempo evaluación: 300 s
- Elites: 4

La evolución del individuo mejor, peor y medio de las distintas generaciones, se muestran a continuación en la Figura 14:

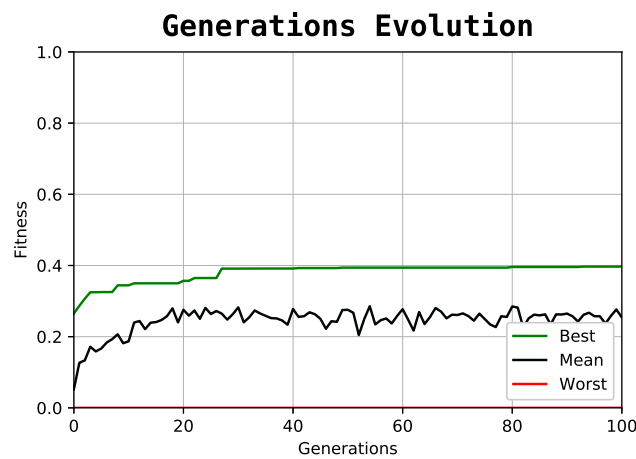


Figura 14: Evolución de las generaciones durante la prueba 4

En la Figura 14, podemos ver como el mejor individuo aunque empieza creciendo con el paso de las generaciones, a partir de la generación 25 se mantiene constante, sin apenas mejoría, siendo algo inferior al de pruebas anteriores. En lo que respecta a la media y al peor individuo se mantienen de forma similar.

Tras ejecutar el experimento, podemos comprobar en la Figura 15a como realiza el movimiento deseado, realizando elipses entre la luz azul y la luz amarilla, de tal forma le permite cargarse y descargarse, con la correcta activación del flag `BatteryFlag`. Cabe destacar el movimiento que realiza al llegar a luz de carga. Una vez conseguido, que el robot vaya correctamente a cargarse, y una vez cargado se aleje de la batería, yendo a la luz azul, en la siguiente prueba se ha añadido otra luz, para que el robot cumpla el propósito inicial del trabajo: oscilar entre dos luces y acudir a cargarse cuando no tenga batería.

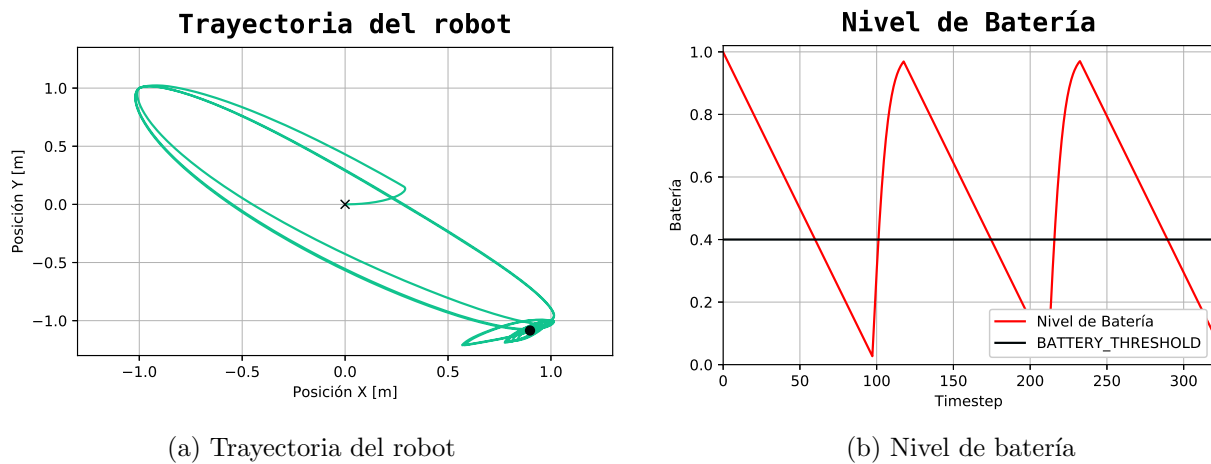


Figura 15: Gráficas de interés del mejor individuo de la generación 100. Prueba 4

3.5. Prueba 5

Como se ha comentado previamente, en esta prueba se ha añadido una luz roja al fichero de parámetros. La arquitectura neuronal del robot implementada para esta prueba se puede observar en la Figura 16. Está formada por siete capas neuronales en este caso, cinco de ellas sensoriales, una capa oculta y una motora. Las neuronas de la capa sensorial de proximidad se encuentra conectada a las neuronas de la capa motora, las neuronas del resto de capas sensoriales se encuentran conectadas a la capa oculta y la capa oculta, al igual que la capa sensorial de proximidad, se encuentra conectada a las neuronas de la capa motora.

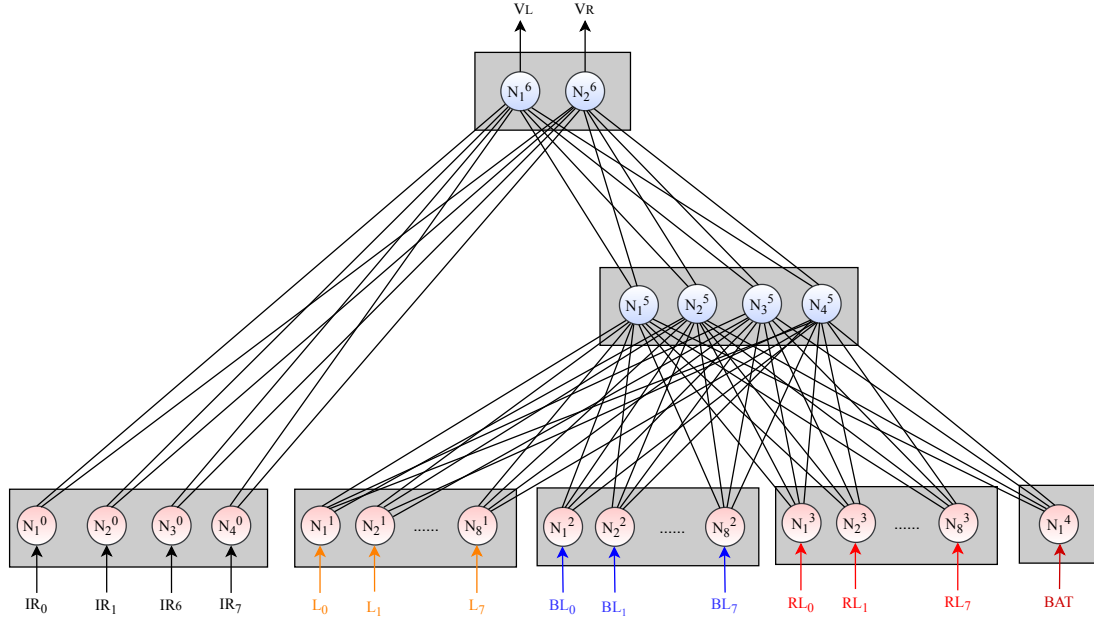


Figura 16: Arquitectura neuronal 5

El cromosoma empleado para esta prueba ha sido el siguiente:

- Cromosoma: 122
- Población: 50
- Mutación: 5 %
- Tiempo evaluación: 300 s
- Elites: 4

Por otro lado, la función de fitness utilizada en este intento responde a la siguiente expresión:

$$F = \begin{cases} \frac{\sum_{i=0}^{N_{steps}} \left[\frac{1}{4} V(1 - \sqrt{\Delta\nu}) + \frac{3}{8} (BL_{ON}(BL_0 + BL_7) + BL_{OFF}(RL_0 + RL_7)) \right]}{N_{steps}} & \text{si } BatteryFlag_{OFF} \\ \frac{\sum_{i=0}^{N_{steps}} \left[\frac{1}{4} V(1 - \sqrt{\Delta\nu}) + \frac{3}{8} (L_0 + L_7) \right]}{N_{steps}} & \text{si } BatteryFlag_{ON} \end{cases} \left(1 - \min \left\{ \frac{N_{coll}, 10}{10} \right\} \right) \left(\min \left\{ \frac{F_{bat}, 5}{5} \right\} \right) \quad (7)$$

En esta función de fitness, tras analizar los resultados de las pruebas anteriores, tenemos distintas formas de premiar al robot:

1. Se ha añadido un nuevo término, F_{bat} , que se incrementa en uno cada vez que se desactiva el flag de la batería. De esta forma, el robot deberá alejarse de la luz amarilla, para poder descargarse, incrementando el contador y así maximizar la fitness. Además, en todos los casos castigaremos al robot si colisiona más de 10 veces con las paredes de la arena.
2. Si el flag de la batería se encuentra desactivado y la luz azul se encuentra encendida, se premiará al robot por dirigirse a esta. Por el contrario si la luz azul se encuentra apagada, se favorecerá que el robot se desplace a la luz roja.
3. Si el flag de la batería se encuentra activado, premiaremos al robot por ir a la luz amarilla, para poder cargarse.

Por otro lado, como se puede observar en los resultados de la fitness (Figura 17) durante el mejor individuo del intento empieza con una fitness de entorno a 0.3 y a medida que se evoluciona llega a una fitness de 0.4.

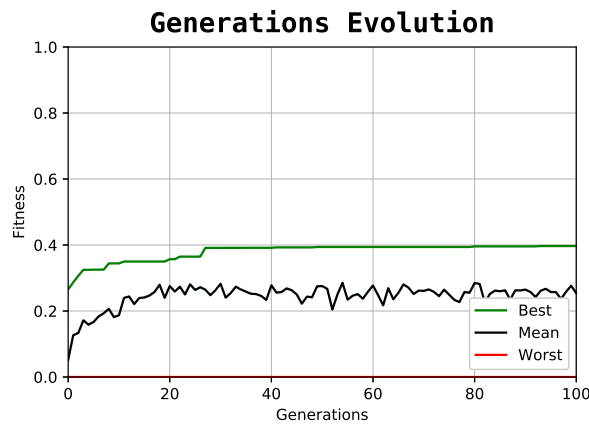


Figura 17: Evolución de las generaciones durante la prueba 5

Viendo los resultados anteriores, cabría esperar que el comportamiento del robot fuera muy diferente al que queremos que tenga, pero la realidad es muy distinta. Como se puede observar en la Figura 18a, el robot no va directamente de unas luces a otras sino que va siguiendo la pared. De esta forma, el individuo sigue un camino óptimo que le permite pasar todas las luces y no quedarse sin batería. Es un resultado correcto pero no es exactamente lo que queríamos conseguir. Por último, si observamos la Figura 18b podemos ver como el robot permanece unos 15 timesteps desde que su batería está por debajo del umbral hasta que vuelve a estar por encima de dicho umbral.

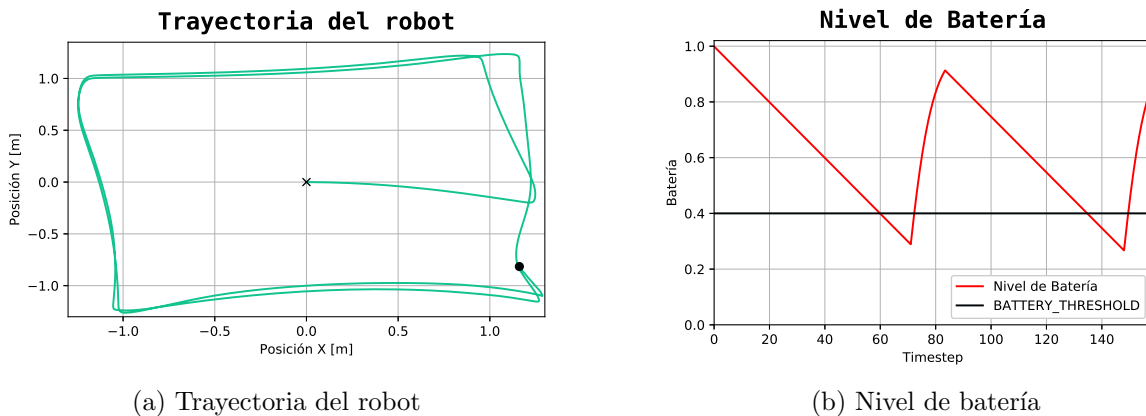


Figura 18: Gráficas de interés del mejor individuo de la generación 100. Prueba 5

3.6. Prueba 6

Finalmente, en este último intento, la arquitectura neuronal del robot implementada para esta prueba coincide con la de la prueba 5, que se puede observar en la Figura 16, ya que para mejorar el comportamiento del robot hemos decidido mantener la arquitectura y cambiar la fitness con respecto al caso anterior. Por ello, el cromosoma empleado también es el mismo que en el caso anterior.

Por otro lado, la fitness utilizada en este intento es la siguiente:

$$F = \begin{cases} \frac{\sum_{i=0}^{N_{steps}} \left[\frac{1}{4} V(1 - \sqrt{\Delta\nu})(1 - \max\{IR_i\}) + \frac{3}{8} (BL_{ON}(BL_0 + BL_7) + BL_{OFF}(RL_0 + RL_7)) \right]}{N_{steps}} & \text{si } BatteryFlag_{OFF} \\ \frac{\sum_{i=0}^{N_{steps}} \left[\frac{1}{4} V(1 - \sqrt{\Delta\nu})(1 - \max\{IR_i\}) + \frac{3}{8} (L_0 + L_7) \right]}{N_{steps}} & \text{si } BatteryFlag_{ON} \end{cases} \left(1 - \min\left\{ \frac{N_{coll}, 10}{10} \right\} \right) \left(\min\left\{ \frac{F_{bat}, 5}{5} \right\} \right) \quad (8)$$

Como se puede observar es muy similar a la de la prueba anterior pero cambiando un pequeño detalle, ahora hemos introducido un factor multiplicativo $(1 - \max\{IR_i\})$, que hace que el robot se separe de las paredes. Cuanto más cerca se encuentre alguno de los sensores de proximidad (S_i), de algún obstáculo, su valor será mayor. Al estar restado a 1, esta componente maximizará la fitness cuando el robot se aleje de las paredes y objetos.

En cuanto a la fitness, como podemos observar en la Figura 19 el caso mejor empieza con una fitness entorno a 0.3 y llega a una fitness cercana a 0.5, manteniendo un crecimiento constante a partir de la generación 10, más o menos. Por otro lado, el caso peor, sigue con un valor nulo constate y el valor medio está entorno a 0.2.

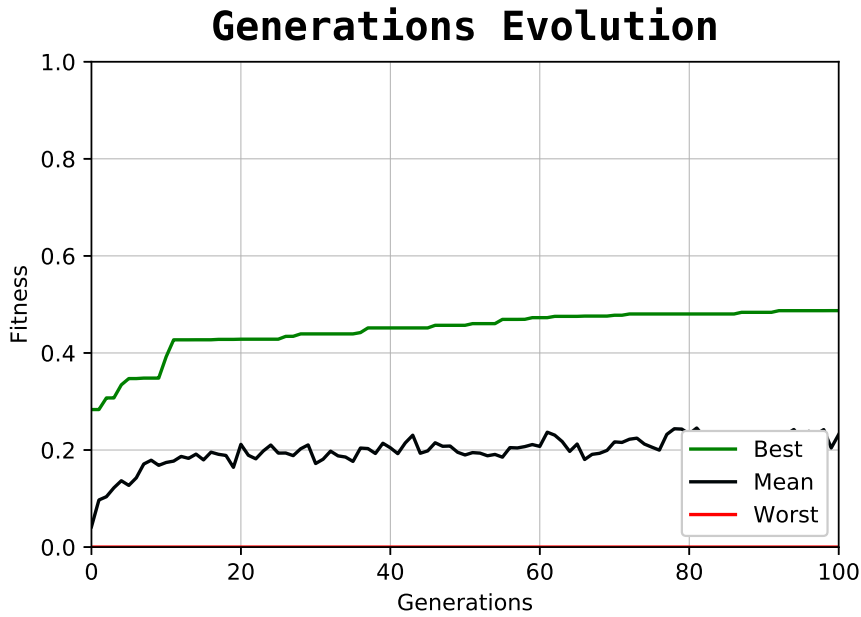


Figura 19: Evolución de las generaciones durante la prueba 6

Los resultados de la fitness nos hacen ser optimistas ya que parecen mejores que en el intento anterior.

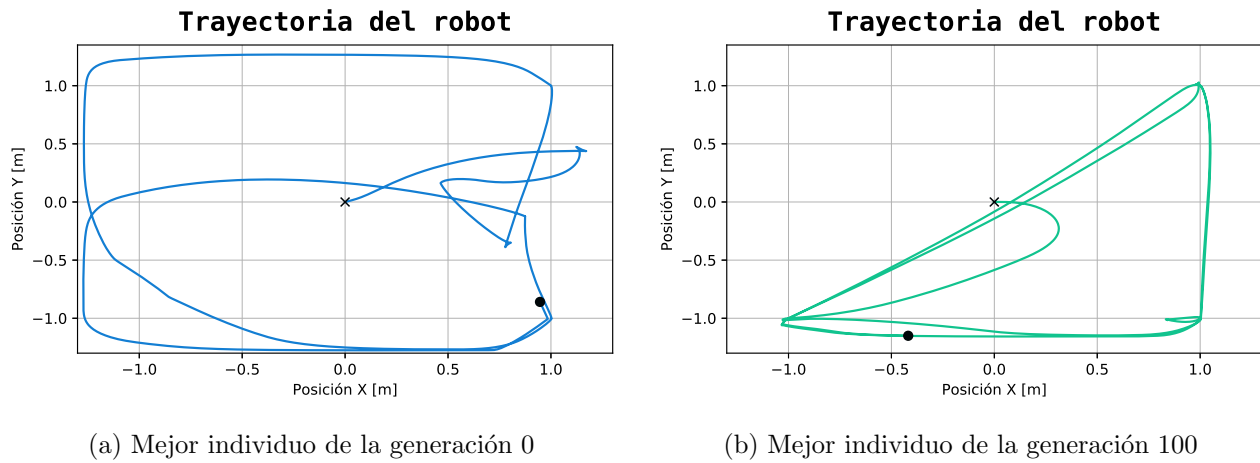


Figura 20: Trayectoria del robot

Como se puede observar en la Figura 20b, el robot inicialmente va a la luz roja ya que es la primera de las luces que se encuentra activa junto con la amarilla, luego pasa por la luz amarilla y después se dirige hacia la luz azul. A partir de este momento, el robot va oscilando entre la luz azul y la luz roja hasta que su batería está por debajo del umbral que va a la luz amarilla a cargarse y después vuelve a oscilar entre las luces roja y azul hasta que su batería vuelve a estar por debajo del umbral y así sucesivamente. Este es precisamente el resultado que queríamos conseguir.

Se puede ver la mejoría que ha experimentado el mejor individuo de la generación 0 y 100 en sus trayectorias en un entorno sin obstáculos (Figuras 20a y 20b).

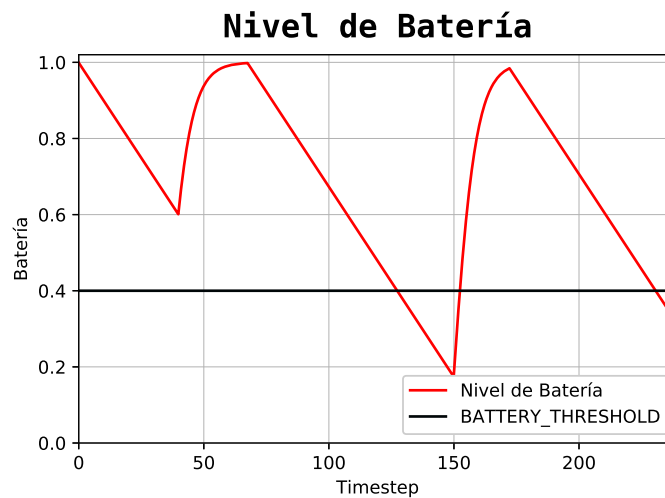


Figura 21: Nivel de batería del mejor individuo de la generación 100

Por último, si observamos la Figura 21 podemos ver ese primer momento en el que el robot pasa por la luz amarilla antes de dirigirse a la azul y como, a partir de ahí el robot solo va a cargarse cuando su batería esta por debajo del umbral. En este caso, a diferencia de en el caso anterior, cuando el robot va a cargarse su carga queda prácticamente en 1 por lo que podemos decir que el robot va cada vez que va a la luz amarilla, carga su batería completamente.

4. Entornos aleatorios ANN

Tras conseguir el correcto funcionamiento del robot en el experimento, era de esperar, que el robot había encontrado una trayectoria que le permitiera cumplir con el objetivo, mientras maximizaba la función de fitness. Por lo tanto, a continuación se pasó a evolucionar el robot en entornos aleatorios en el simulador. Primero, generando a cada individuo de la generación en una posición inicial aleatoria dentro de un cuadrado de $1m^2$, y posteriormente generando una posición aleatoria de los objetos utilizados en el experimento.

4.1. Posición aleatoria

En esta primera parte, se evolucionó al robot utilizando la arquitectura neuronal de la Figura 16 y la función de fitness de la Ecuación 8, con el siguiente cromosoma y generando para cada individuo de la generación una posición inicial dentro de un cuadrado de tamaño 1 metro por 1 metro. Por tanto, se ha utilizado el siguiente cromosoma:

- Cromosoma: 122
- Población: 100
- Número de generaciones: 1001
- Mutación: 5 %
- Tiempo evaluación: 300 s
- Elites: 4
- Muestras por cromosoma: 4

Tras evolucionar el robot, en la Figura 22, podemos observar la evolución de las distintas generaciones:

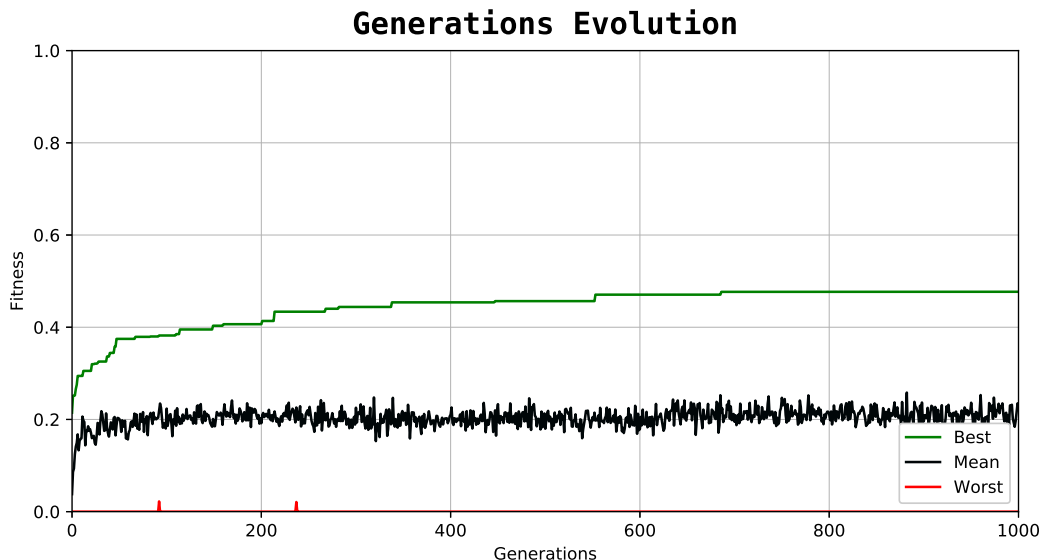


Figura 22: Evolución de las generaciones para una posición aleatoria

La fitness va creciendo de forma incremental con el paso de las generaciones, hasta llegar a valores próximos a 0.5. Por otro lado, la media y el peor individuo se mantienen de forma similar al de la prueba 6 (Figura 19). Posteriormente, se comprobó que al variar la posición inicial del robot, que este se adaptaba perfectamente a cualquier entorno, realizando con éxito su función.

4.2. Objetos aleatorios

Así, siguiendo con la evolución de las pruebas realizadas en el trabajo, para poder afirmar de que se trata de una buena función de fitness, era preciso entrenar al robot en un entorno donde los objetos se generan de forma aleatoria, así como su posición también se generara de forma aleatoria dentro de un cuadrado de 1 metro por 1 metro. Nuevamente, se ha utilizado la arquitectura neuronal de las prueba 5 y 6 (Figura 16) y la misma función de fitness de la prueba 6 (Ecuación 8). Por lo tanto, el cromosoma cuenta con las siguientes características:

- Cromosoma: 122
- Población: 50
- Número de generaciones: 1001
- Mutación: 10 %
- Tiempo evaluación: 500 s
- Elites: 6
- Muestras por cromosoma: 4

En la Figura 23, podemos observar la evolución de las distintas generaciones:

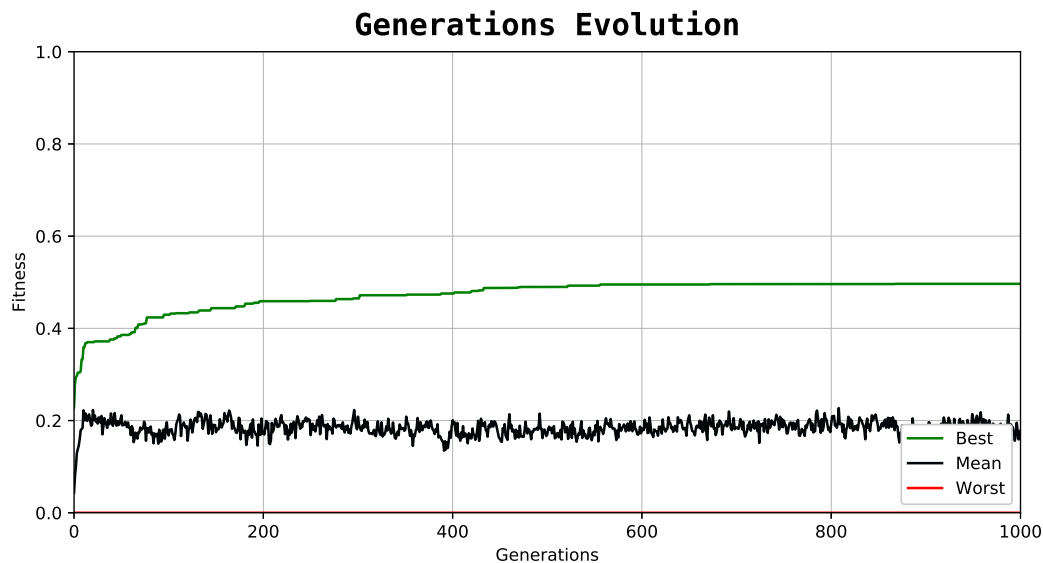


Figura 23: Evolución de las generaciones para posiciones aleatorias de los objetos

Aunque los resultados obtenidos de la fitness en este experimento son similares al anterior, el resultado es bien distinto. Al situar los tres objetos utilizados en el trabajo (luz amarilla, roja y azul) en posiciones distintas a la inicial, el robot cumple con su función. Si bien es cierto, que la trayectoria que realiza en esta situación es bien distinta, a la de casos anteriores. El robot consigue oscilar, entre la luz roja y azul, aunque para ello de vez en cuando pasa por la luz amarilla. En el apartado 7 se puede observar con más detenimiento el comportamiento de este robot (Trayectoria, nivel de batería, velocidad de los motores, etc).

5. Red neuronal artificial dinámica

A continuación, se ha extendido la red neuronal artificial estática a una red neuronal artificial dinámica. Como en el caso anterior, se detallan las pruebas más significativas hasta la consecución del objetivo deseado en el trabajo.

5.1. Prueba 1

Viendo los resultados tan positivos que obtuvimos en la prueba 6 de la red neuronal estática, nuestra primera prueba con la red neuronal dinámica se basó en utilizar tanto la misma estructura neuronal, que se puede observar en la Figura 16, como la misma fitness, que puede verse en la ecuación 8 pero cambiando los parámetros correspondientes por tratarse de una red neuronal dinámica que se detallan a continuación. El entorno del robot también decidimos utilizar el mismo que para la prueba 6, es decir, un entorno con 3 luces, una amarilla a la que acudirá a cargarse y una luz azul y otra roja entre las cuáles queremos que el robot oscile.

Para esta prueba se ha empleado el siguiente cromosoma:

- Cromosoma: 134
- Población: 50
- Mutación: 5 %
- Tiempo evaluación: 300 s
- Elites: 4

Los resultados obtenidos de la fitness se pueden ver en la Figura 24. Como se puede observar, el caso mejor empieza entorno a 0.3 y llega a cerca de 0.5. El caso peor sigue siendo 0 como en las pruebas anteriores mientras que la media es bastante baja, alrededor de 0.1. Unos resultados similares a la prueba 6 de ANN pero con la media más baja.

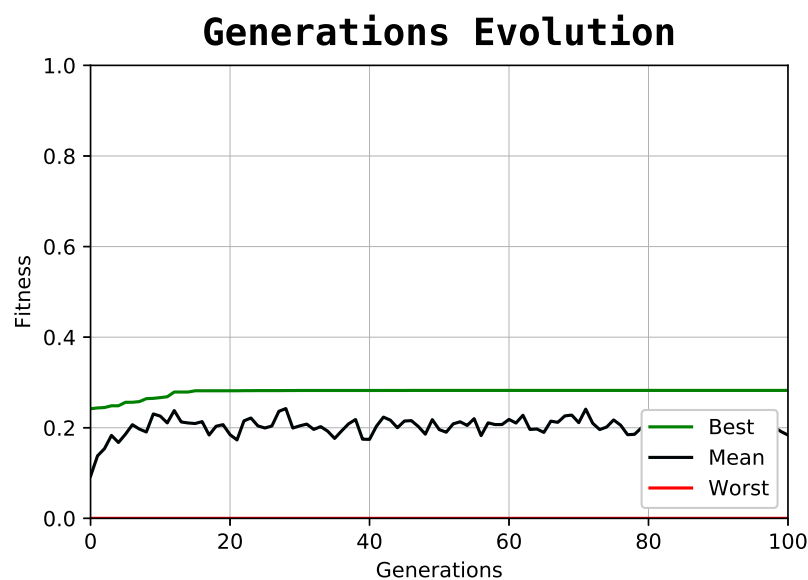


Figura 24: Evolución de las generaciones durante la prueba 1

Por otro lado, como se puede observa en la Figura 25a, el funcionamiento del robot dista mucho del esperado. En este caso, el robot se dedica a dar vueltas describiendo una elipse de tal forma que parece que ignore las luces que se encuentran a su alrededor. Ha encontrado una trayectoria óptima a seguir de tal forma que su fitness no se vea muy perjudicada. Por otro lado, como se puede ver en la Figura 25b el robot no tiene en cuenta su nivel de batería pues al estar por debajo del umbral no va a cargarse. Evidentemente este no es el resultado que queríamos conseguir.

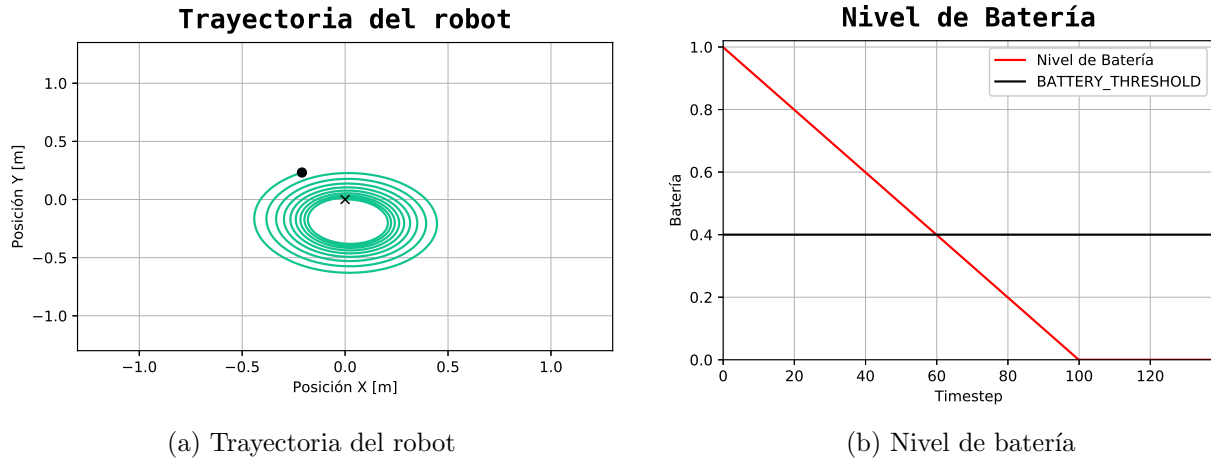


Figura 25: Gráficas de interés del mejor individuo de la generación 100. Prueba 1

5.2. Prueba 2

Con el objetivo de obtener un resultado mejor que en la prueba anterior, en esta prueba decidimos simplificar para poder conseguir que el robot con la red neuronal dinámica fuera capaz de realizar una actividad más sencilla a la que queremos llegar a conseguir y a partir de ahí intentar conseguir más cosas. En esta prueba le planteamos al robot un experimento con 2 luces, una azul y una amarilla. Si el robot tiene la batería por encima del umbral se dirigirá a la luz azul y si tiene la batería por debajo irá a la luz amarilla a cargarse. Para conseguirlo, se ha aumentado la población, el tiempo de evaluación y la mutación con respecto a casos anteriores. La arquitectura utilizada en esta prueba es la utilizada anteriormente en la prueba 1 de ANN (Figura 4). Para esta prueba se ha utilizado el siguiente cromosoma:

- Cromosoma: 48
- Población: 100
- Mutación: 20 %
- Tiempo evaluación: 400 s
- Elites: 4

La función de fitness utilizada queda representada por la ecuación 10 y previamente definimos un flag que será el que controlará la función de fitness en función de la batería del robot.

$$BatteryFlag = \begin{cases} 1 & \text{si } B < Battery_Threshold \\ 1 & \text{si } BatteryFlag_{ON} \text{ y } B < 0,95 \\ 0 & \text{resto} \end{cases} \quad (9)$$

Así, incorporando este flag a la función de fitness, quedará de la siguiente forma:

$$F = \begin{cases} \frac{\sum_{i=0}^{N_{steps}} \left[\frac{1}{4}V(1 - \sqrt{\Delta\nu}) + \frac{3}{8}(BL_0 + BL_7) \right]}{N_{steps}} \left(1 - \min \left\{ \frac{N_{coll}, 10}{10} \right\} \right) & \text{si } BatteryFlag_{OFF} \\ \frac{\sum_{i=0}^{N_{steps}} \left[\frac{1}{4}V(1 - \sqrt{\Delta\nu}) + \frac{3}{8}(L_0 + L_7) \right]}{N_{steps}} \left(1 - \min \left\{ \frac{N_{coll}, 10}{10} \right\} \right) & \text{si } BatteryFlag_{ON} \end{cases} \quad (10)$$

Los resultados de la fitness se pueden ver en la Figura 26. En ella podemos observar unos resultados que dejan mucho que desear, el caso mejor empieza algo superior a 0.2 y consigue crecer hasta algo más de 0.4, permaneciendo prácticamente constante a lo largo de las generaciones. El caso peor sigue en 0 y la media es algo superior a 0.1. Estos resultados no nos hacen ser muy optimistas en cuanto al funcionamiento del robot.

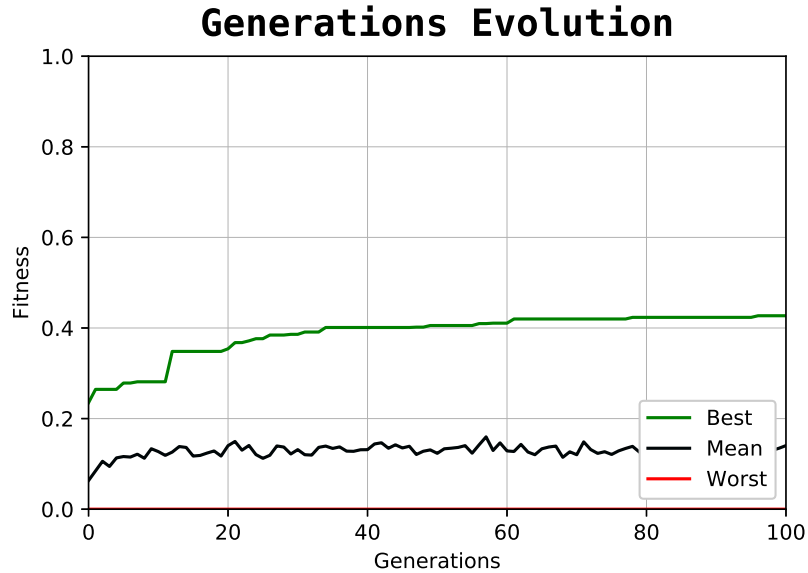


Figura 26: Evolución de las generaciones durante la prueba 2

Probando el funcionamiento del robot nos encontramos un funcionamiento que no era el que queríamos. Como se puede observar en la Figura 27a, el robot describe una trayectoria mínima. Así, si el nivel de batería está por encima del umbral, se queda apuntando a la luz azul. En cambio, si el nivel de batería está por debajo del umbral y por lo tanto el flag está activo, se queda apuntando a la luz amarilla (recarga de batería). De esta forma, consigue maximizar la fitness sin apenas moverse de la posición inicial. En cuanto al nivel de batería, vemos que una vez que se descarga, no consigue desplazarse hacia la batería, y se queda descargada, como podemos ver en la Figura 27b. Por todo esto, consideramos también esta prueba como fallida.

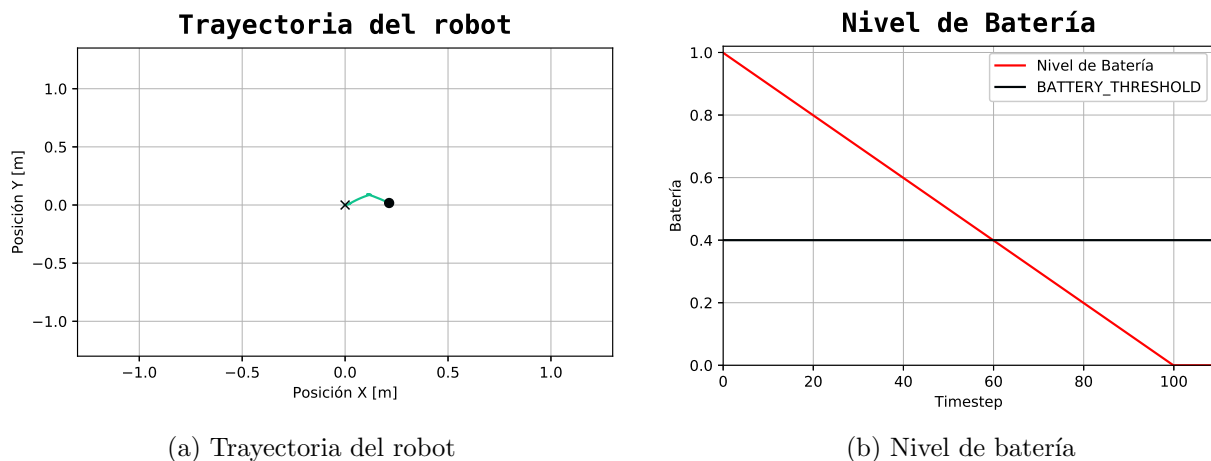


Figura 27: Gráficas de interés del mejor individuo de la generación 100. Prueba 2

5.3. Prueba 3

Sorprendidos de que la prueba anterior no funcionara ya que teníamos confianza plena en la arquitectura y fitness planteadas, decidimos hacer un nuevo intento sin cambiar estos 2 elementos pero cambiando el cromosoma, de la siguiente manera:

- Cromosoma: 48
- Población: 50
- Mutación: 5 %
- Tiempo evaluación: 400 s
- Elites: 4

Decidimos reducir la población, ya que con una población de 50 habíamos obtenido buenos resultados en los experimentos de la red neuronal estática, y la mutación, ya que consideramos que en la prueba anterior estaba alta y podía ser una de las causas de nuestro problema.

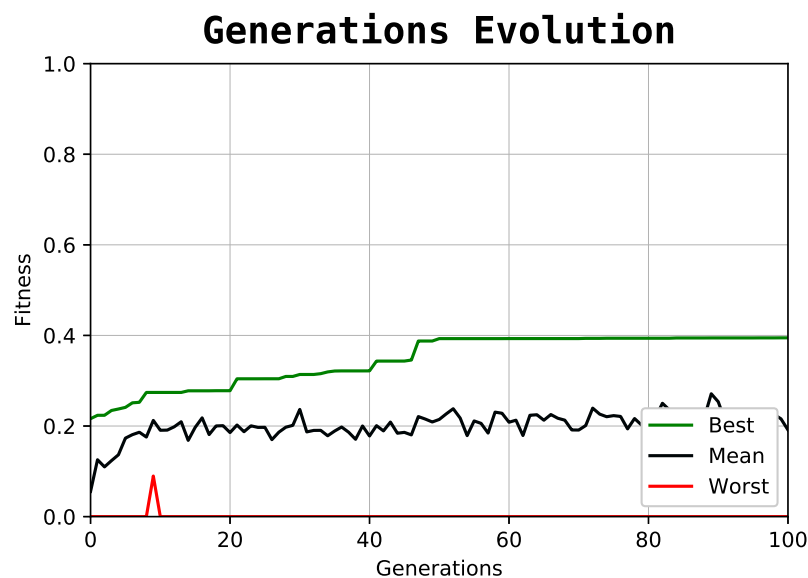


Figura 28: Evolución de las generaciones durante la prueba 3

Como podemos observar en la Figura 28, los resultados parecen mucho más prometedores que en el caso anterior pues nuestro caso mejor empieza con una fitness algo superior a 0.2 y llega a una fitness de 0.4, nuestra media es también más alta, entorno a 0.2 y nuestro caso peor es por lo general 0 aunque nos encontramos algún caso que se sale de este valor.

Si observamos la Figura 29a, vemos que el comportamiento del robot parece ser el que queríamos lograr, que el robot inicialmente fuera a la luz azul y cuando se quedara sin batería fuera a la luz amarilla para cargarse y después volver a la luz azul. Sin embargo, si miramos la Figura 29b vemos como el robot parece que se adelanta un poco a lo que queríamos, es decir, va a cargarse antes de que su batería baje del umbral, quizá porque ha encontrado este comportamiento más óptimo que el que planteábamos nosotros. De todas formas, teniendo en cuenta los casos anteriores, estamos satisfechos con el comportamiento de este robot ya que, en primer lugar, esta versión no es la definitiva sino una vía para llegar al comportamiento que planteábamos inicialmente y, por otro lado, porque aunque vaya a cargarse antes de lo que queríamos, no se queda sin batería que es el principal objetivo de que vaya a cargarse.

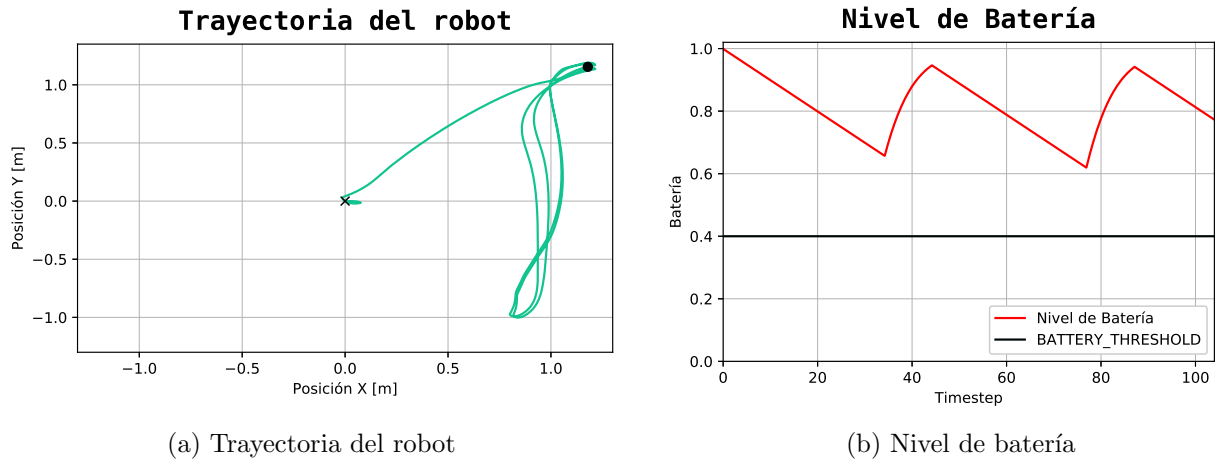


Figura 29: Gráficas de interés del mejor individuo de la generación 100. Prueba 3

5.4. Prueba 4

Una vez conseguido que el robot funcione con la luz azul y la luz amarilla, con este experimento queremos lograr que funcione también con la luz roja, es decir, con 3 luces en la arena. Para ello la arquitectura utilizada se puede observar en la Figura 30, en ella podemos ver como introducimos únicamente una capa neuronal que reciba los sensores de la luz roja y que esté conectada directamente con la capa motora.

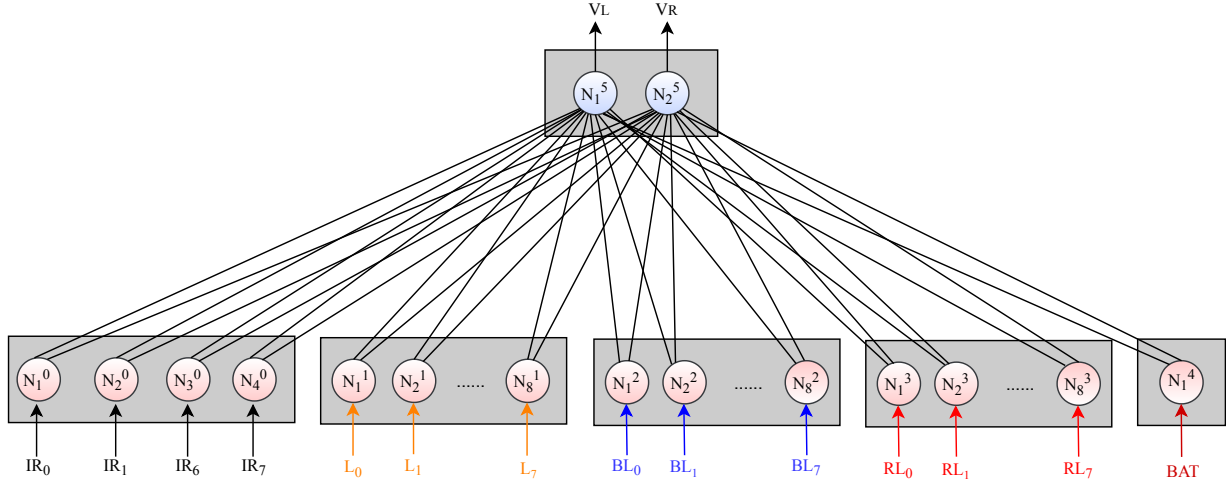


Figura 30: Arquitectura neuronal 7

El cromosoma utilizado en esta prueba es el siguiente:

- Cromosoma: 64
- Población: 50
- Mutación: 5 %
- Tiempo evaluación: 400 s
- Elites: 4

Y la fitness también cambia, ya que en ella hay que tener en cuenta también la luz roja y nos quedaría de la siguiente manera:

$$F = \begin{cases} \frac{\sum_{i=0}^{N_{steps}} \left[\frac{1}{4} V(1 - \sqrt{\Delta\nu})(1 - \max\{IR_i\}) + \frac{3}{8} (BL_{ON}(BL_0 + BL_7) + BL_{OFF}(RL_0 + RL_7)) \right]}{N_{steps}} & \text{si } BatteryFlag_{OFF} \\ \frac{\sum_{i=0}^{N_{steps}} \left[\frac{1}{4} V(1 - \sqrt{\Delta\nu})(1 - \max\{IR_i\}) + \frac{3}{8} (L_0 + L_7) \right]}{N_{steps}} & \text{si } BatteryFlag_{ON} \end{cases} \left(1 - \min \left\{ \frac{N_{coll}, 10}{10} \right\} \right) \left(\min \left\{ \frac{F_{bat}, 5}{5} \right\} \right) \quad (11)$$

Por otro lado, en la Figura 31 podemos observar la evolución de la fitness. En este caso, el caso mejor empieza con una fitness de 0.2 y llega a una fitness de cerca de 0.3, el caso peor es 0 y, en este experimento tenemos una media bastante elevada en relación al caso mejor de 0.2. Estos resultados no nos invitan a ser muy optimistas en cuanto al funcionamiento del robot.

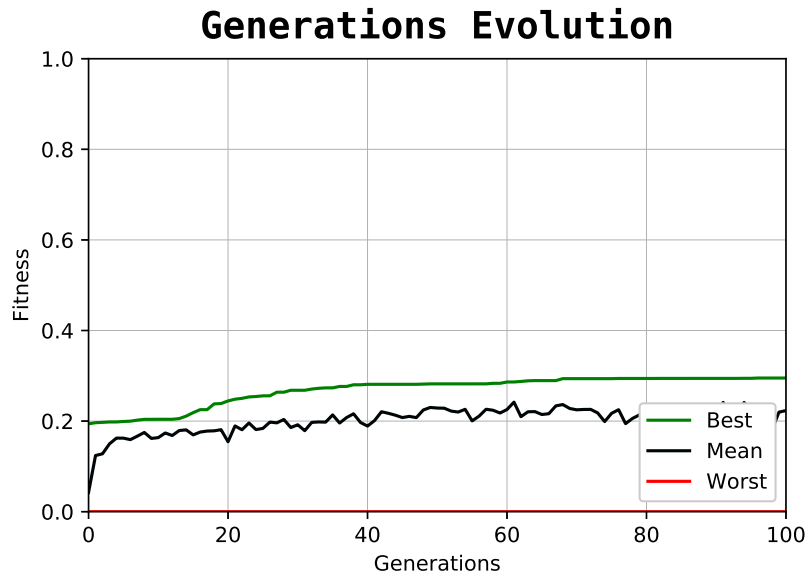


Figura 31: Evolución de las generaciones durante la prueba 4

En la Figura 32a podemos observar un comportamiento que parece algo extraño pues al principio empieza haciendo unas elipses muy pequeñas y el tamaño de esta elipse va creciendo. Según nuestro conocimiento adquirido en esta parte de la asignatura, entendemos este comportamiento como que el robot va encontrando su trayectoria óptima a medida que pasa el tiempo, de manera que pueda pasar lo más cerca posible de todas las luces. Por otro lado, en la Figura 32b observamos que en tema de batería no hemos conseguido lo que queríamos pues la batería del robot baja del umbral pero no va a cargarse, esto se debe a lo que mencionábamos antes de encontrar la trayectoria óptima, pues el robot intenta pasar lo más cerca posible de todas las luces pero no llega a la zona en la que podría cargarse, de ahí que la fitness fuera bastante baja.

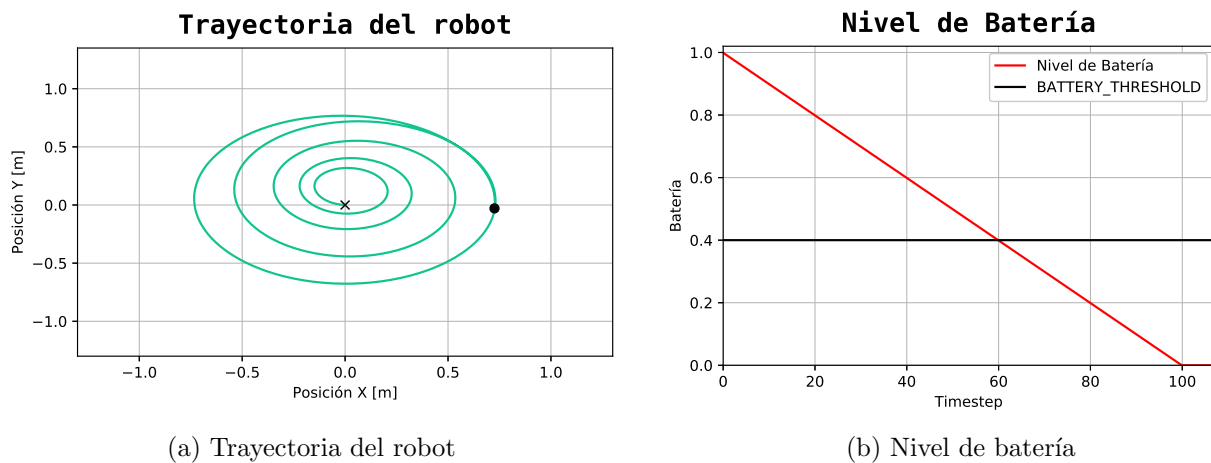


Figura 32: Gráficas de interés del mejor individuo de la generación 100. Prueba 4

6. Entornos aleatorios CTRNN

Así, siguiendo con la evolución de las pruebas realizadas en el trabajo, para poder afirmar de que se trata de una buena función de fitness, era preciso entrenar al robot en un entorno donde los objetos se generan de forma aleatoria.

6.1. Objetos aleatorios

Así, se evolucionó al robot utilizando la arquitectura neuronal de la Figura 30 y la función de fitness de la Ecuación 11. Durante la evolución, cada individuo de la generación ha sido evaluado en un entorno donde la posición de los objetos ha sido generada de forma aleatoria, así como la posición inicial del robot dentro de un cuadrado de tamaño 1 metro por 1 metro. Por lo tanto, el cromosoma cuenta con las siguientes características:

- Cromosoma: 64
- Población: 50
- Número de generaciones: 1001
- Mutación: 5 %
- Tiempo evaluación: 400 s
- Elites: 4
- Muestras por cromosoma: 4

Tras evolucionar el robot, en la Figura 33 podemos observar la evolución de las distintas generaciones:

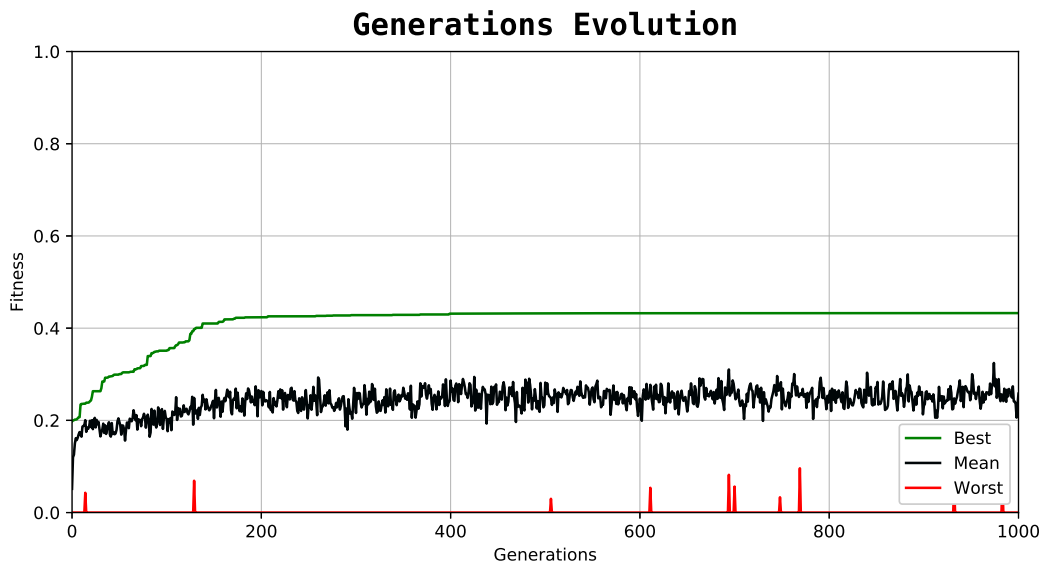


Figura 33: Evolución de las generaciones para posiciones aleatorias de los objetos

Los resultados son claramente mejores, que todas las pruebas realizadas previamente. En primer lugar, analizando la evolución de las generaciones el mejor individuo empieza aproximadamente en 0.2 y va creciendo hasta aproximadamente la generación 200, donde a partir de ahí permanece constante a lo largo de las generaciones. Por otro lado, la media es más o menos constante en torno a un valor fijo de 0.2 y el peor individuo de las generaciones es casi siempre 0, aunque hay algunos casos en los que no. En el apartado 7 se puede observar con más detenimiento el comportamiento de este robot (Trayectoria, nivel de batería, velocidad de los motores, etc).

7. Gráficas de interés

Para realizar las gráficas, utilizamos los datos generados a partir de los sensores del robot, así como la información de entrada y salida de las diferentes capas utilizadas, almacenados en los ficheros de la carpeta *outputFiles* y *geneticDataFiles*. Hemos utilizado **Python** y sus distintas librerías (pandas, matplotlib y seaborn) para generar dichas gráficas, que nos han permitido analizar las redes neuronales estáticas y dinámicas de forma más específica.

A continuación, se presentan las gráficas realizadas:

La Figura 34 muestra la comparativa de la trayectoria realizada por el mejor individuo de a) red neuronal artificial estática y b) red neuronal artificial dinámica. Se puede observar, el punto de inicio marcado por una “X”, así como el punto final marcado por un círculo.

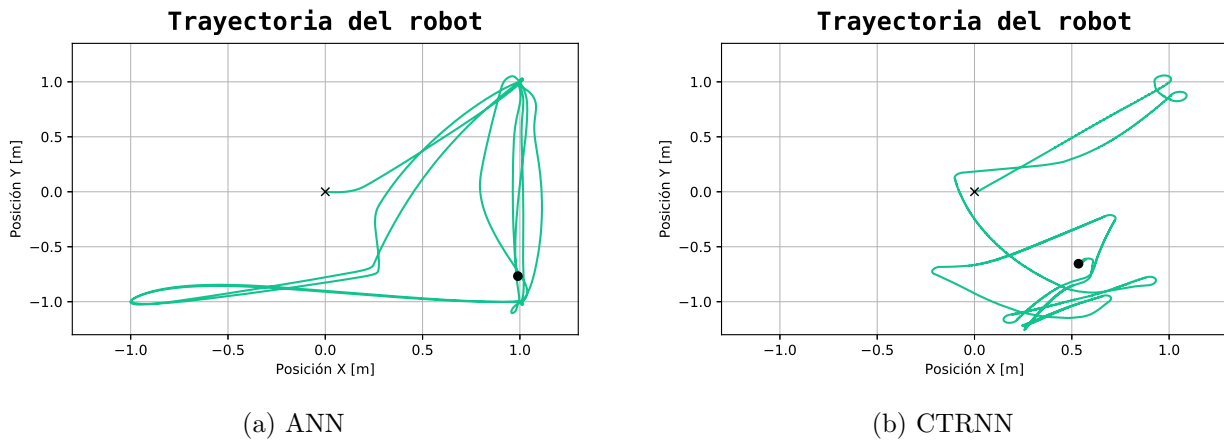


Figura 34: Trayectoria del robot

Se puede observar que la trayectoria descrita por cada de las redes neuronales es bien distinta. La red neuronal estática, realiza movimientos más rectilíneos que en el caso de la dinámica. Mientras que la red neuronal dinámica le cuesta más tratar de buscar las diferentes luces, especialmente la luz roja, en la que realiza movimientos extraños alrededor de la recarga de batería, hasta que consigue acercarse a esta.

En la Figura 35 observamos una comparación del nivel de la batería amarilla del robot, durante el tiempo que dura la simulación en a) red neuronal artificial estática y b) red neuronal artificial dinámica. El umbral de batería se encuentra fijado en 0.4 como se puede comprobar.

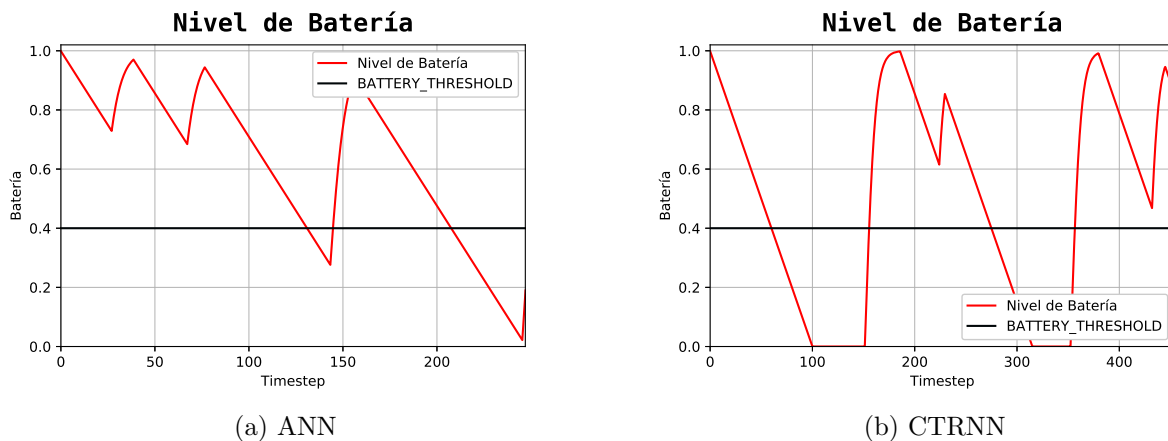


Figura 35: Nivel de batería

A continuación, se muestra la velocidad del motor de cada una de las ruedas del robot (Figura 36) en a) red neuronal artificial estática y b) red neuronal artificial dinámica.

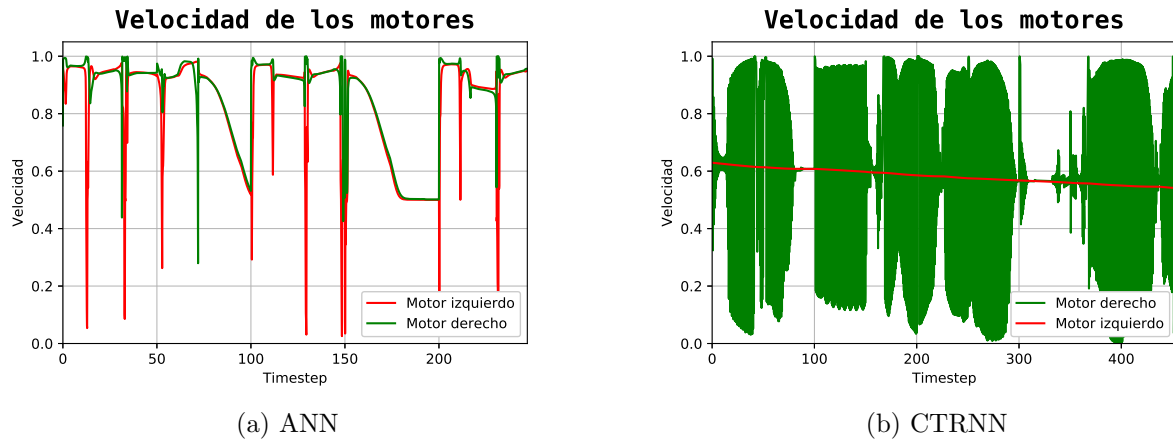


Figura 36: Velocidad de los motores

Los siguientes gráficos, están relacionados con la posición del robot durante la simulación. Hemos representado el nivel de batería (Figura 37) y velocidad de las ruedas (Figura 38), en función de la posición (coordenadas X e Y):

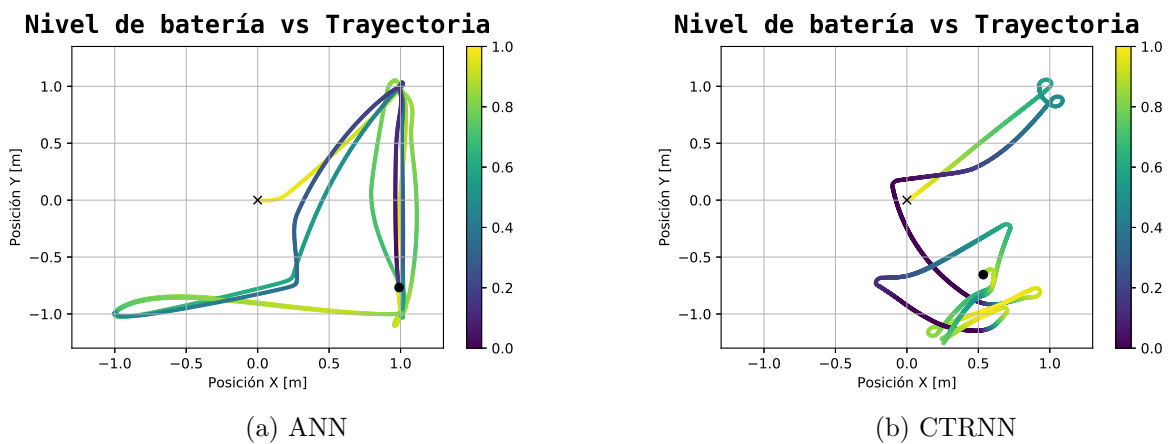


Figura 37: Nivel de batería vs Trayectoria

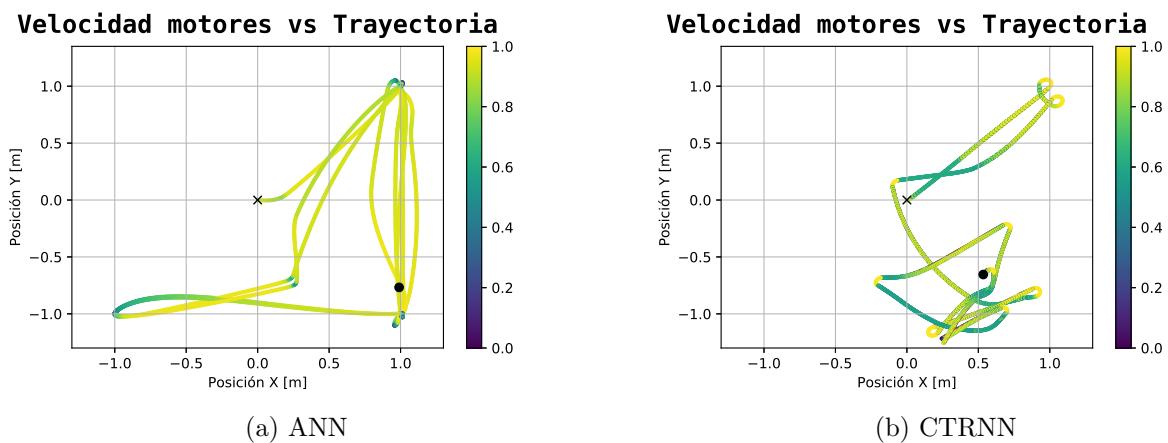


Figura 38: Velocidad motor rueda derecha vs Trayectoria

8. Comparación de resultados entre ANN y CTRNN

Durante el desarrollo del trabajo nos hemos encontrado varias diferencias entre utilizar una red neuronal estática y una red neuronal dinámica que detallamos a continuación.

La primera diferencia nos la encontramos a la hora de calcular la longitud del cromosoma ya que mientras en ANN solo teníamos que tener en cuenta los pesos y los sesgos para dicho cálculo, al calcular la longitud del cromosoma en CTRNN había que considerar pesos, sesos, τ_i , g_i y β_i , por consiguiente, la longitud del cromosoma quedaba mucho mayor.

Por otro lado, otra diferencia reseñable que nos encontramos fue que la red neuronal dinámica necesita una arquitectura más simple que la red neuronal estática para funcionar mejor. Para explicar esto vamos a fijarnos en la Sección 3.4 y en la Sección 5.3, correspondientes pruebas realizadas en ANN y CTRNN respectivamente, en concreto al intento de que el robot fuera a la luz azul si su batería estaba por encima del umbral y fue a la luz amarilla a cargarse si su batería se encontraba por debajo del mismo. Ambos experimentos funcionaban pero mientras que para hacerlo ANN necesitaba de capas ocultas, a CTRNN le bastaba con conectar las capas sensoriales a las motoras.

Por último, destacar una diferencia que se puede observar en nuestro proyecto y es que se puede conseguir un comportamiento perfecto en una es las redes neuronales y no conseguirlo en la otra. En nuestro caso, conseguimos que el robot funcionara correctamente para la red neuronal estática haciendo todo lo que propusimos al principio mientras que no conseguimos lograr del todo nuestro objetivo con la red neuronal dinámica. Sin embargo, al realizar un último entrenamiento para cada red neuronal con entornos aleatorios pudimos ver como mientras ANN empeoraba un poco su comportamiento no yendo de forma tan directa a las luces, CTRNN sin embargo mejoraba su forma de actuar dejando de realizar las elipses que hacía antes y yendo más directo a las luces dirigiéndose inicialmente a la luz azul, después a la amarilla y más tarde intentó llegar sin éxito a la luz roja, pero en conjunto funcionaba mucho mejor que sin utilizar entorno aleatorios a diferencia de ANN.

9. Problemas encontrados

A la hora de la realización de este trabajo nos hemos ido enfrentando a numerosos problemas, que hemos solventado. El más común ha sido no conseguir que el robot haga el movimiento previsto mediante la función de fitness, buscando algún punto de la arena en el cual permanecía parado o realizando otro movimiento no previsto, a la vez que intentaba maximizar la fitness. Además, también hemos observado el hecho de que el robot aprendiera una determinada trayectoria en la que conseguía realizar el objetivo planteado de forma casi perfecta. Dicho problema se solucionó evolucionando al robot en entornos donde la posición de los objetos se generaba de forma aleatoria, así como también la posición inicial del robot.

En la parte de red neuronal estática nos encontramos frente a un problema en varias pruebas, sobre el hecho de que el robot no consiguiera salir de la luz de recarga, debido a que no encontraba el estímulo suficiente por parte de las otras luces para desplazarse hacia ellas cuando el nivel de batería se encontrara cargado. Así, el robot entraba en una fase de oscilaciones sobre un nivel fijo de batería, realizando movimientos hacia delante y hacia detrás, de forma que se encontraba entrando y saliendo en la zona de recarga. Otro problema, al que nos enfrentamos fue, conseguir la correcta activación del flag de batería, `FlagBattery` y que este permaneciera activo hasta que el robot se había cargado completamente.

Por otro lado, en la parte de red neuronal dinámica nos hemos encontrado el problema de que el robot realizara movimientos circulares, mientras intentaba pasar lo más cerca posible de las distintas luces del experimento.

10. Futuras implementaciones

A lo largo del proyecto, se nos han ido ocurriendo posibles implementaciones para nuestro robot de cara a un futuro. Sin embargo, debido a las limitaciones de tiempo y conocimientos, éstas no han podido ser implementadas.

La robótica inteligente está evolucionando a gran escala en los últimos años, y es por ello por lo que creemos que existen infinitas adiciones que podrían mejorar nuestro robot, aunque todavía no dispongamos de los conocimientos necesarios para ello.

Algunas de estas mejoras podrían ser:

- Añadir distintas zonas para recoger objetos. Este procedimiento consistiría en la creación de dos nuevas zonas. Una sería una zona negra con una luz roja y la otra sería una zona gris con una luz azul. El robot tendría que coger objetos de la zona gris y llevarlos a la zona negra. El robot podría descargarse, y si se quedara sin batería en medio de este recorrido, tendría que ir a la zona de carga en medio del proceso.
- Añadir una luz más a nuestro mapa y, mediante mecanismos de prioridad, conseguir que el robot siga el recorrido que nosotros le marquemos.



11. Conclusión

Tras realizar el proyecto hemos sacado varias conclusiones acerca de la robótica evolutiva:

En primer lugar, que un robot que no haya sido entrenado en un entorno en el cuál se ejecute puede fallar, es decir, si se evoluciona a un robot en un entorno sin obstáculos y le pones a trabajar en un entorno con obstáculos, es muy posible que colisione con el mismo ya que no ha aprendido a evitarlos.

Por otro lado, consideramos que utilizar la robótica evolutiva para implementar el funcionamiento de un robot lleva mucho más tiempo que programar un comportamiento basado en una arquitectura de Subsunción o Motor Schemas ya que al utilizar redes neuronales dinámicas y estáticas se necesita un tiempo para evolucionar el robot y en función de lo complicado que sea el problema a resolver este tiempo puede ser muy grande.

En tercer lugar, nos parece mucho más complicado programar el comportamiento de un robot utilizando redes neuronales ya que diseñar una función de fitness que marque el funcionamiento de un robot es una tarea complicada. Además, no se puede saber si una fitness es la adecuada o no para un robot inmediatamente ya que para saberlo primero tienes que evolucionar al robot y ya después probarlo.

Por último consideramos que la robótica evolutiva tiene sus limitaciones ya que podría haber problemas a los que no se encuentre solución, es decir, que no consigamos que el robot realice exactamente la acción que queramos.

12. Bibliografía

Durante la realización del trabajo obligatorio 2 de IRIN ha sido necesario la consulta de los siguientes recursos:

- ROBOLABO. Manual de la asignatura y presentaciones utilizadas durante las clases de Introducción a la Robótica Inteligente.
 - <http://robolabo.etsit.upm.es/subjects.php?subj=irin&tab=tab3&lang=es>
- Información de utilidad sobre las diferentes librerías de Python.
 - <https://pandas.pydata.org/>
 - <https://seaborn.pydata.org/>
 - <https://matplotlib.org/>