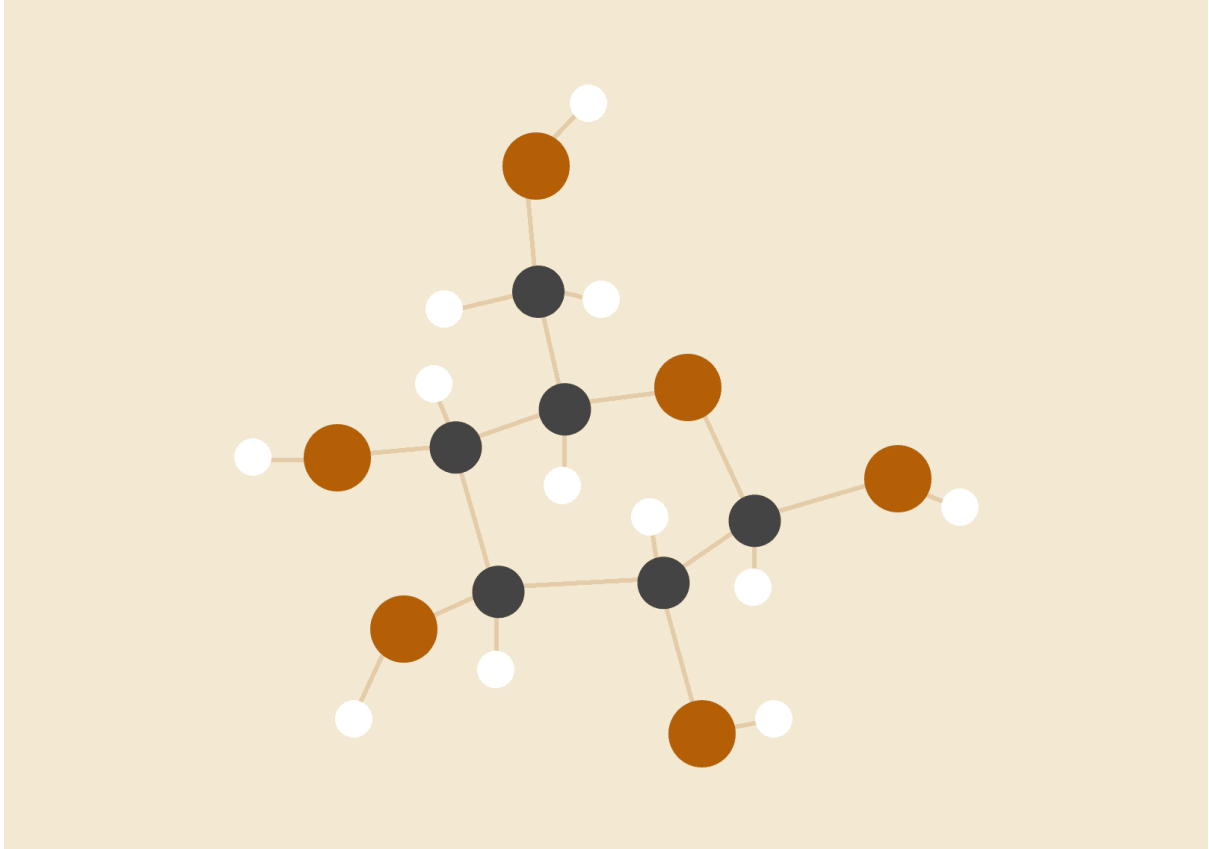


TRABAJO IRIN 02

Robótica evolutiva



Grupo 6

Samuel García Sánchez

Ana Méndez Pérez

ÍNDICE

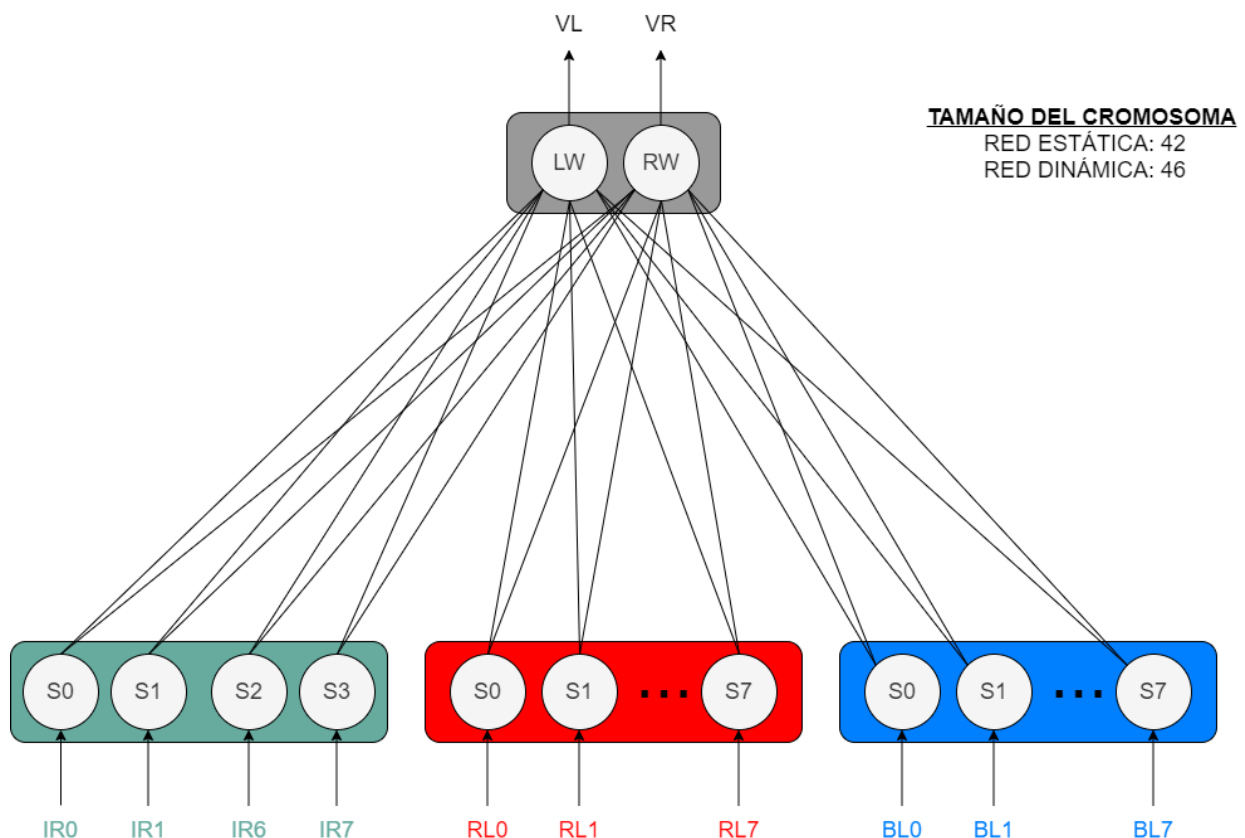
E1: LUCES	3
RED ESTÁTICA	4
RED DINÁMICA	5
E2: BATERÍA	7
E2.1.A	7
RED ESTÁTICA	8
RED DINÁMICA	10
E2.2.A	11
RED ESTÁTICA	13
RED DINÁMICA	13
E2.1.B	15
RED ESTÁTICA	16
RED DINÁMICA	16
E2.2.B	18
RED ESTÁTICA	19
RED DINÁMICA	20
E3: LUCES Y BATERÍA	22
E.3.1	21
RED ESTÁTICA	22
RED DINÁMICA	23
E.3.2	26
RED ESTÁTICA	27
RED DINÁMICA	29
E.3.3	30
RED ESTÁTICA	30
RED DINÁMICA	33
CONCLUSIONES	35
BIBLIOGRAFÍA	36

E1: LUCES

Nuestro objetivo final es llegar a un robot que vaya hacia la luz azul pero que al quedarse sin batería vaya a la luz amarilla a cargarse. En primer lugar vamos a realizar las dos funcionalidades por separado, en este primer experimento crearemos un robot que oscile entre luces, y en el segundo un robot que vaya a la batería a cargarse cuando esta caiga por debajo de cierto umbral y que se aleje de ella cuando lo supere. Una vez verificado el correcto funcionamiento de ambos, comenzaremos a integrarlos en uno solo para tratar de llegar a nuestro objetivo final, comenzando con arquitecturas y fitness más sencillas hasta las más complejas, apreciando todos los cambios que supongan en el comportamiento del robot estas modificaciones.

Como hemos mencionado, en este sencillo primer experimento intentaremos dar con un robot que logre oscilar entre las luces azul y roja. Cuando la luz azul esté encendida, la roja estará apagada y viceversa, el tiempo de encendido y apagado será de 500 time steps. Este experimento, al igual que todos los que vamos a hacer, lo vamos a llevar a cabo empleando una red estática y también dinámica para ver las diferencias.

La arquitectura de la red neuronal se presenta a continuación. Cabe destacar que en las redes dinámicas no vamos a utilizar realimentación ya que como vimos en clase no suponía un aumento del rendimiento del robot.



ECUACIÓN FITNESS:

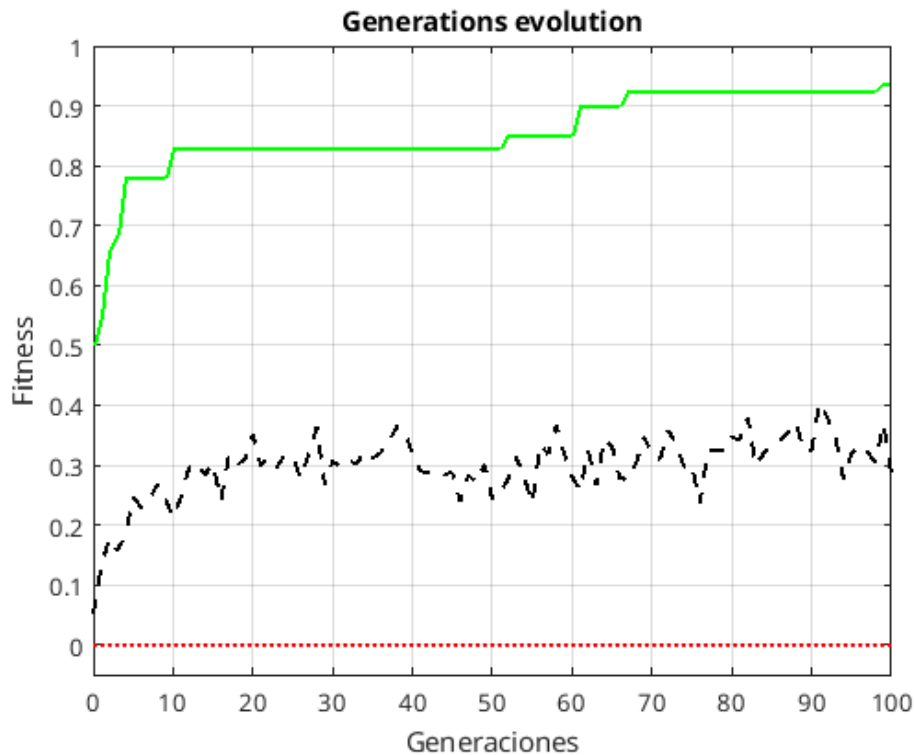
$$f = \alpha * V(1 - \sqrt{\Delta v}) + \beta [BL_{on}(BL_0 + BL_7) + RL_{on}(RL_0 + RL_7)]$$

donde V es la suma de los valores absolutos de cada una de las velocidades de cada motor menos 0.5, Δv es el valor absoluto de la diferencia entre las velocidades, α y β son 2 parámetros de ajuste de la ecuación de fitness (0.25 y 0.75 respectivamente). BL_{on} y RL_{on} indican si en ese momento la luz azul o roja están apagadas (a 0) o reciben luz través de alguno de los sensores (a 1) por lo que se dirige a la luz que está encendida en ese momento y se multiplica por los valores obtenidos por los sensores delanteros $S0$ y $S7$ de cada luz.

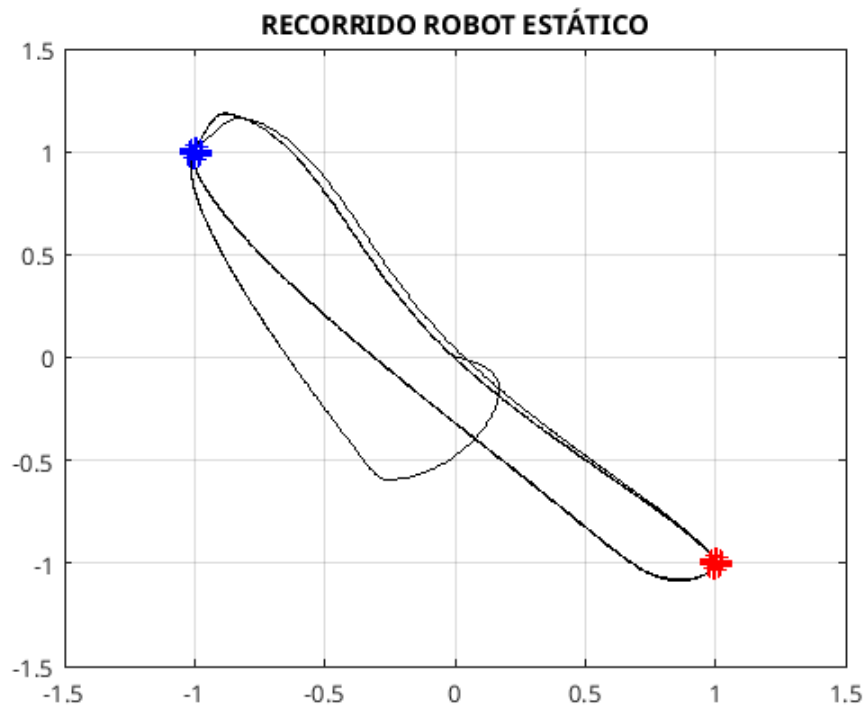
Por último, se ha tenido en cuenta las colisiones del robot con los obstáculos. La fitness total permite un máximo de 10 colisiones con los obstáculos, a partir de los cuales el robot obtiene una fitness = 0 y un tiempo de evaluación de 400 ms.

$$F_i = \frac{\sum_{t=0}^{400} f}{400} * \left(1 - \frac{\min(col, 10)}{10}\right)$$

RED ESTÁTICA

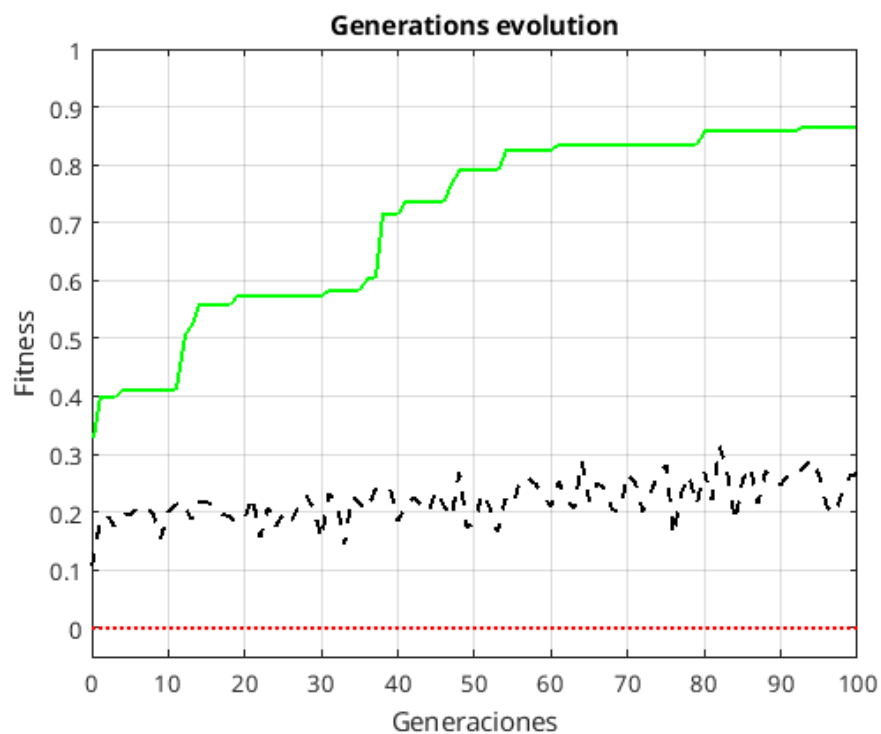


Conseguimos una fitness alta para el mejor individuo, aunque podemos apreciar que este crecimiento no acompaña en el average, donde no se produce una gran mejoría respecto a las primeras generaciones.

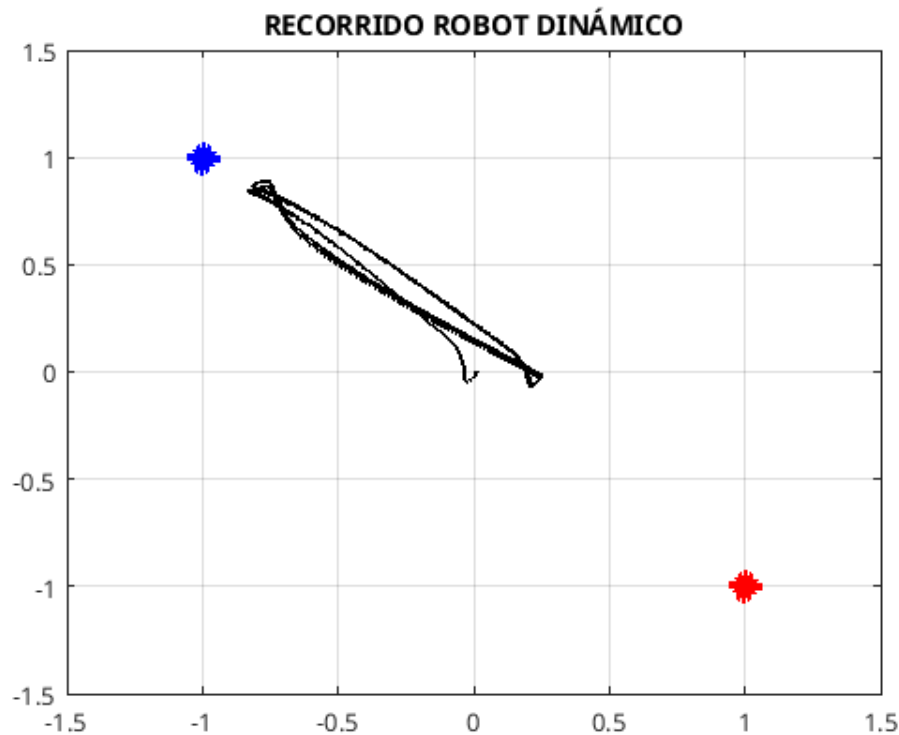


Observamos que el robot realiza una trayectoria semicircular al arrancar desde una luz a la siguiente, pero una vez se acerca bastante a la luz, donde se mantiene quieto hasta que esta se apaga. En ese momento, hace ese 'semicírculo' de nuevo para salir de la luz, se orienta y avanza hacia la siguiente luz.

RED DINÁMICA



Vemos que se alcanza una buena fitness (cerca de 0.9), muy similar al estático, pero el funcionamiento vamos a ver a continuación que es peor:

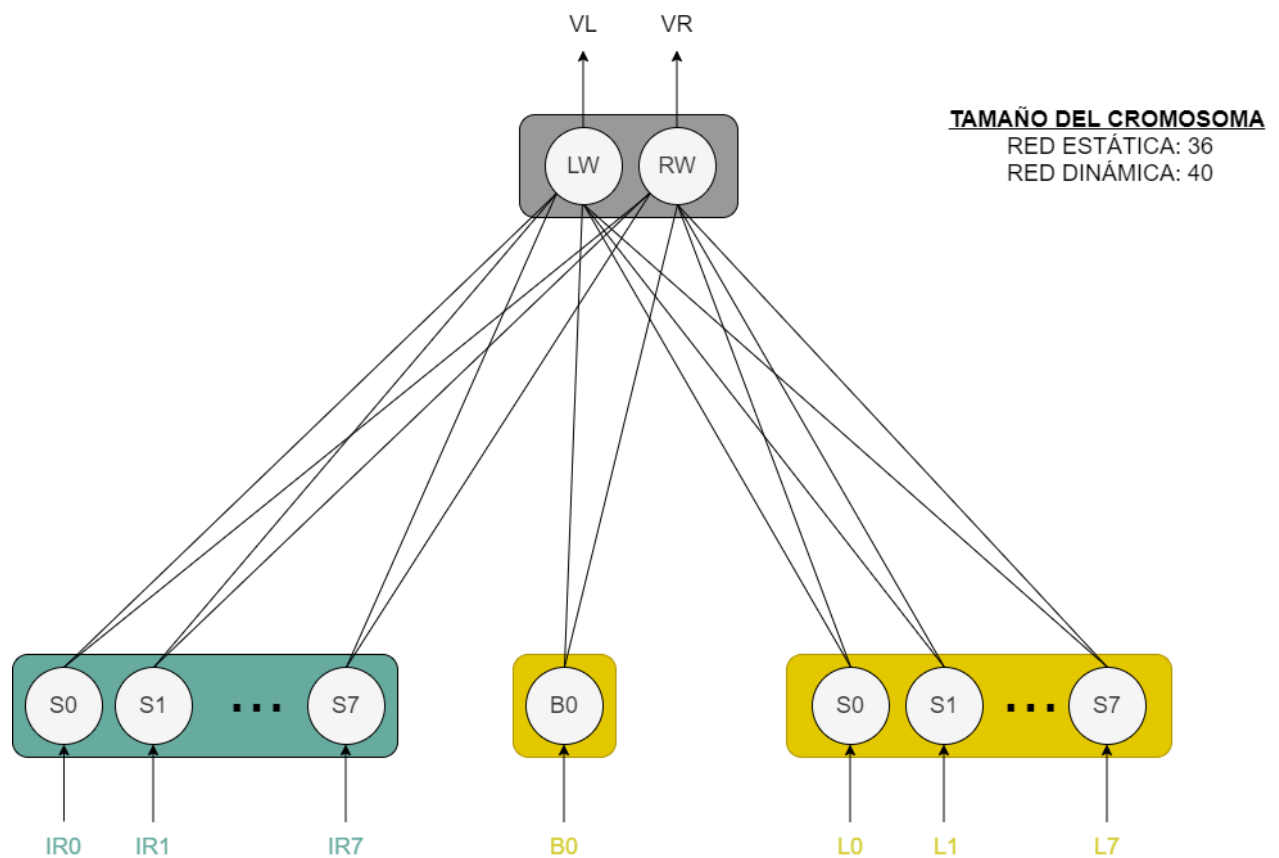


Empleando una red dinámica nos hemos encontrado con que, pese a hacer una trayectoria más 'directa' entre las luces, no llegaba a acercarse tanto a ellas como con la red estática. De hecho, nos ha sorprendido que para la azul se acercase más o menos a una distancia aceptable, pero cuando tenía que ir hacia la roja iba con una velocidad muy reducida, lo que hacía que nunca le diese tiempo a acercarse casi nada porque a mitad del camino se cumplían los 500 time steps y tenía que volver a la azul. Nos hemos quedado con la duda de por qué hacia la azul iba más rápido que hacia la roja. También cabe destacar que cuando llegaba a la luz azul, en lugar de tocarla o rodearla como en el caso anterior, el robot se quedaba parado en frente suya, y se movía de forma similar a cuando se producía una lucha de comportamientos en la primera parte de la asignatura.

E2: BATERÍA

En este experimento la finalidad es estudiar cómo se comporta el robot dependiendo de la batería que tenga. Nuestro objetivo es que vaya a la luz para cargarse y que se aleje de ella cuando ya esté cargado. Vamos a probar con dos ecuaciones de fitness diferentes (1 y 2) y con dos arquitecturas diferentes (A y B).

ARQUITECTURA A:



Usaremos los 8 sensores de proximidad, los 8 sensores de luz amarilla y el sensor de batería, que estarán todos conectados directamente a la capa motora sin capas ocultas.

E2.1.A

Para este fitness nos hemos basado en el manual de la asignatura en el capítulo 3.3.2 *Navegación y carga* perteneciente a la parte de Robótica evolutiva.

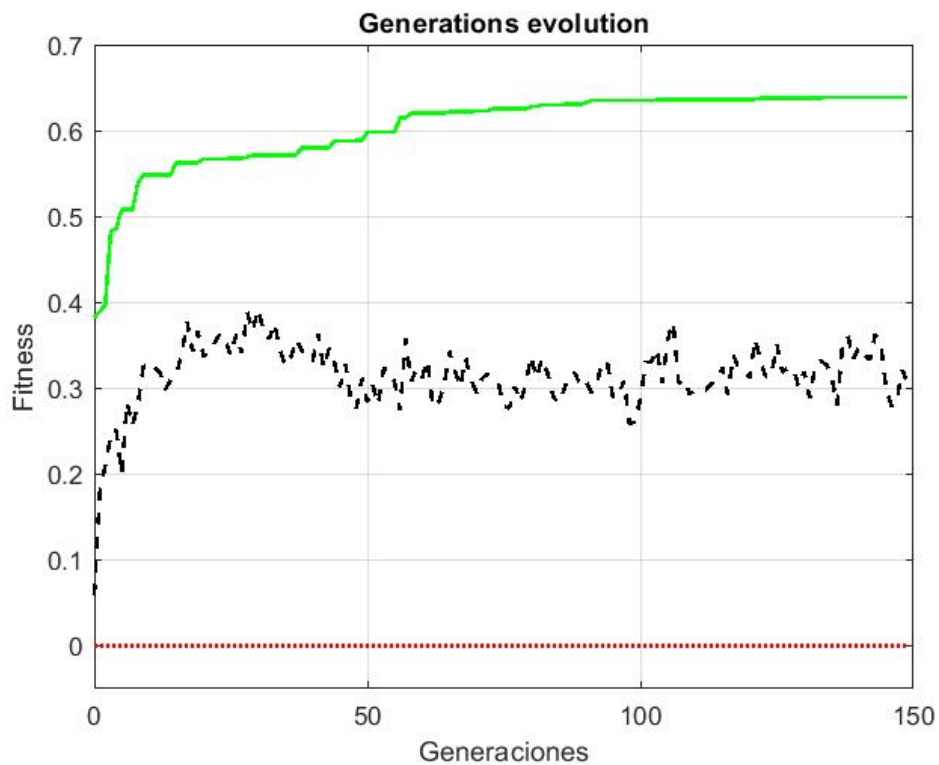
$$f = \alpha * V(1 - \sqrt{\Delta v}) * (M_r * M_l) + \beta * Battery + (1 - \alpha - \beta) * l$$

donde V es la suma de los valores absolutos de cada una de las velocidades de cada motor menos 0.5, Δv es el valor absoluto de la diferencia entre las velocidades, M_r y M_l son las velocidades del motor derecho e izquierdo respectivamente e l es la normalización entre 0 y 1 del máximo valor de los sensores infrarrojos. *Battery* es el valor real del estado de la batería del robot, l es la normalización entre 0 y 1 del máximo valor de los sensores de luz, α y β son 2 parámetros de ajuste de la ecuación de fitness (0.5 y 0.3 respectivamente).

Además se ha tenido en cuenta las colisiones del robot con los obstáculos (paredes de los bordes de la arena). La fitness total permite un máximo de 10 colisiones con los obstáculos, a partir de los cuales el robot obtiene una fitness = 0 y un tiempo de evaluación de 300 ms.

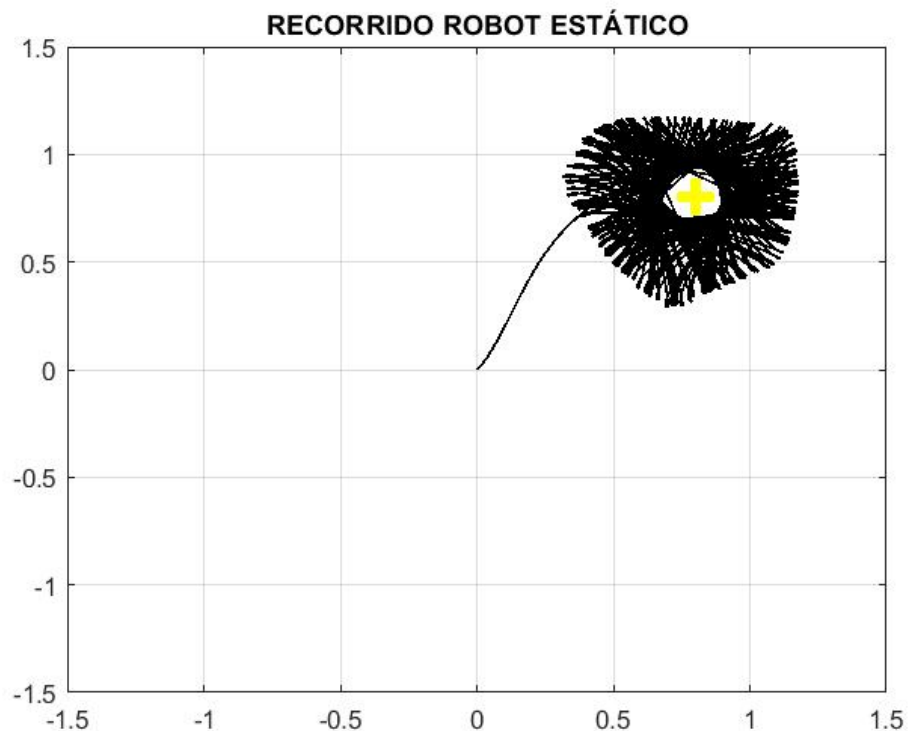
$$F_i = \frac{\sum_{t=0}^{300} f}{300} * \left(1 - \frac{\min(col, 10)}{10}\right) \quad (1)$$

RED ESTÁTICA



Al evolucionar el experimento hemos podido ver que a lo largo de 150 generaciones el experimento mejora la fitness menor a 0,4 hasta sobrepasar 0,6. Creemos que con más generaciones hubiera seguido aumentando pero no sabemos hasta qué punto.

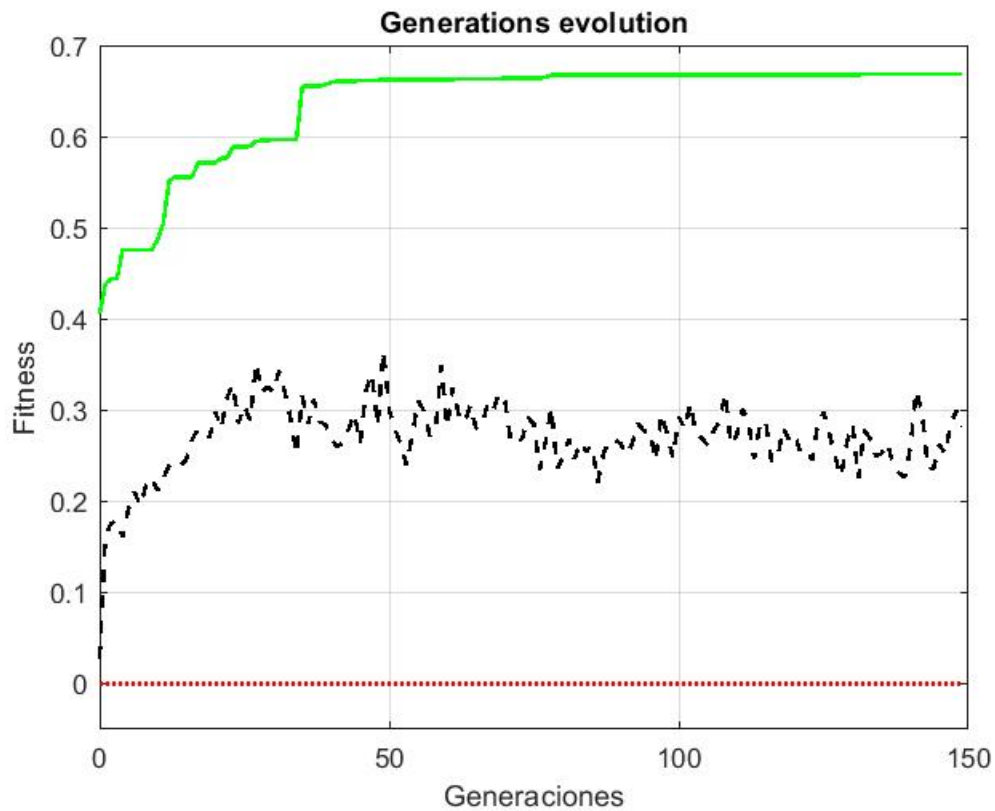
El average aumenta en un principio pero luego va bajando oscilando entre los valores cercanos a 0.3.



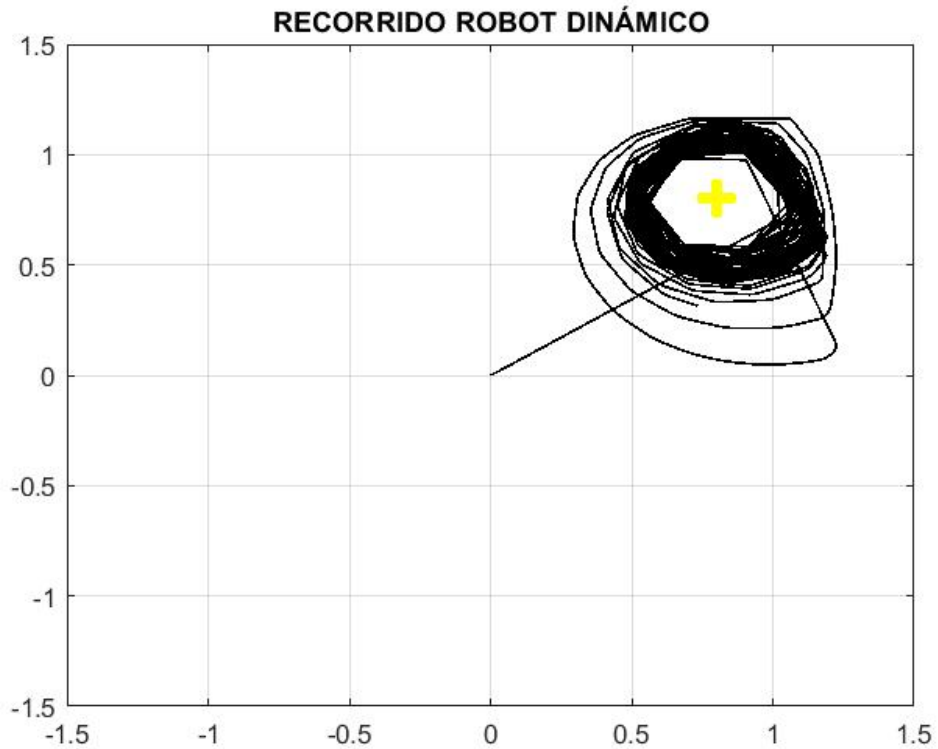
El comportamiento del robot con esta fitness es el indicado en la gráfica superior. El robot se acerca a la luz amarilla y se va acercando y alejando de la luz a partir de un determinado umbral que le impide alejarse más que se marca en el recorrido con un radio aproximadamente menor o igual a 0,5 metros que es el radio de desde el que empieza a cargarse.

RED DINÁMICA

También hemos realizado el experimento de forma dinámica para poder ver las diferencias entre ellos y ver si mejora o empeora su comportamiento. La arquitectura seguirá siendo la misma pero sus cromosomas aumentarán.



La gráfica comparada con la del experimento E2.1 estático es bastante parecida de forma pero con unos valores un poco más altos. Además, la línea del average le ocurre lo contrario, tiene la misma forma pero con valores medios más bajos del 0,3.



También podemos ver que el comportamiento del robot ha cambiado. Ahora no pasa de forma tan cercana a la luz amarilla y órbita de una forma casi hexagonal, cada vez acercándose más.

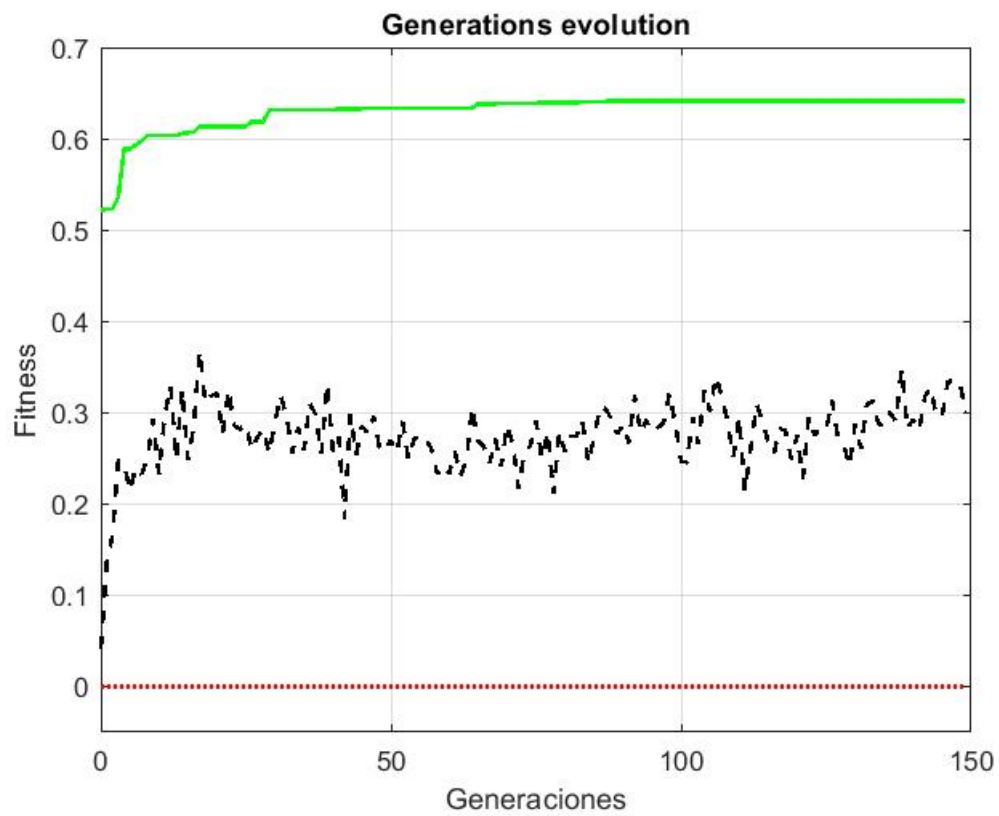
E2.2.A

Como segunda fitness hemos planteado que la fitness se multiplique cuanto más se aleje de la luz amarilla cuando la batería esté cargada (> 0.7) y si es menor que se multiplique por el valor del sensor de la luz amarilla para que se vaya acercando.

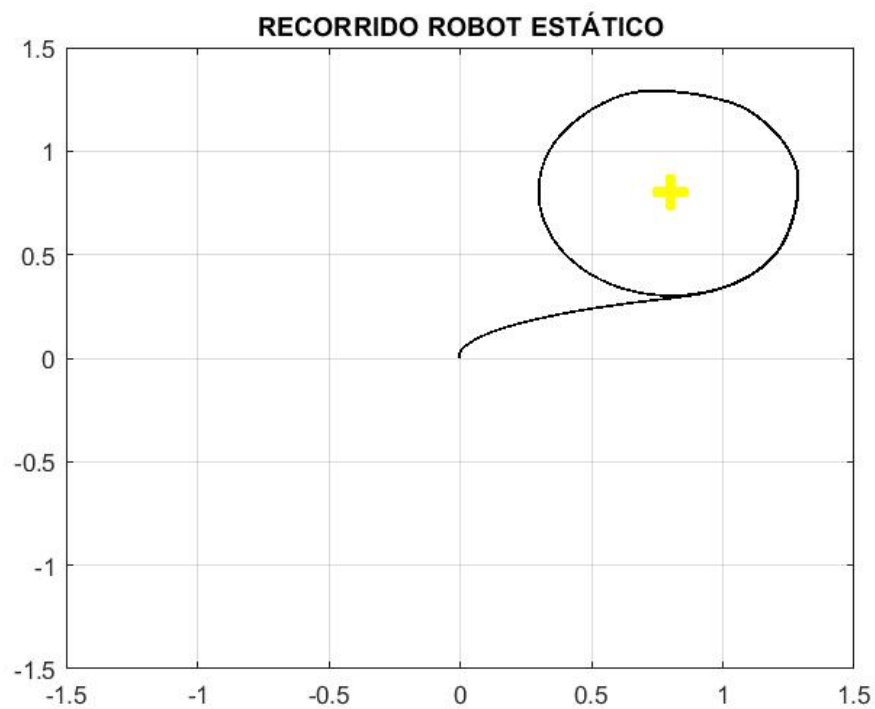
$$\begin{aligned} \text{if Battery} > 0.7 &\Rightarrow f = V(1 - \sqrt{\Delta v}) * (1 - l) \\ \text{if Battery} < 0.7 &\Rightarrow f = V(1 - \sqrt{\Delta v}) * l \end{aligned}$$

Además se ha tenido en cuenta las colisiones del robot con los obstáculos (paredes de los bordes de la arena). Para ello realizamos de nuevo la ecuación (1) pero con la nueva fitness.

RED ESTÁTICA

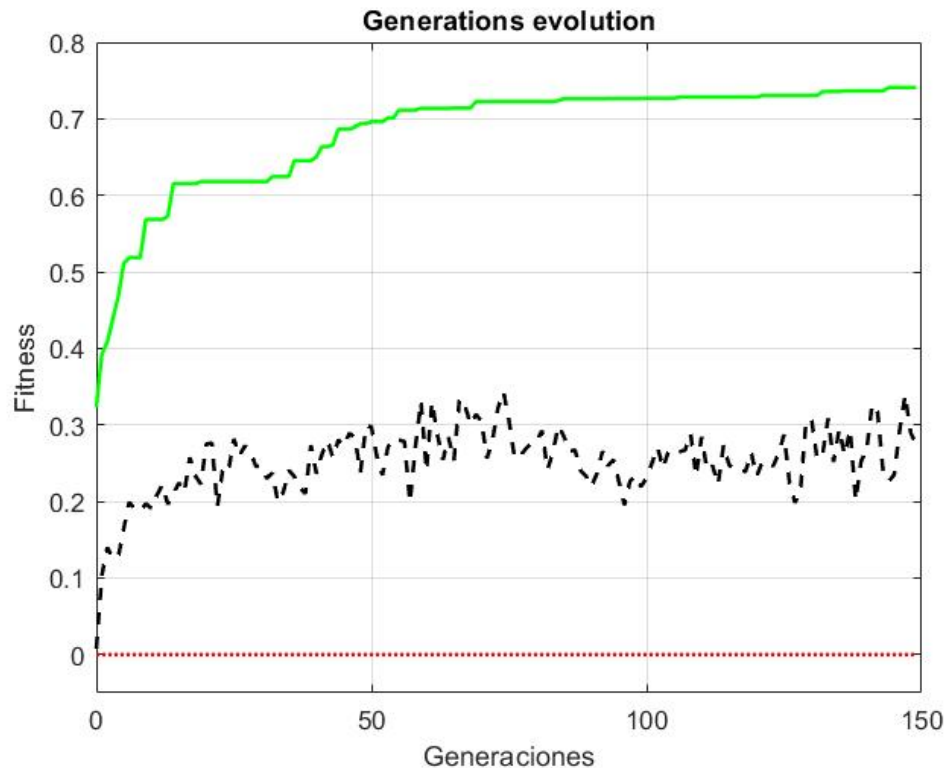


La fitness comienza en un valor más alto que el anterior pero tiene un valor parecido de fitness al final en la última generación.



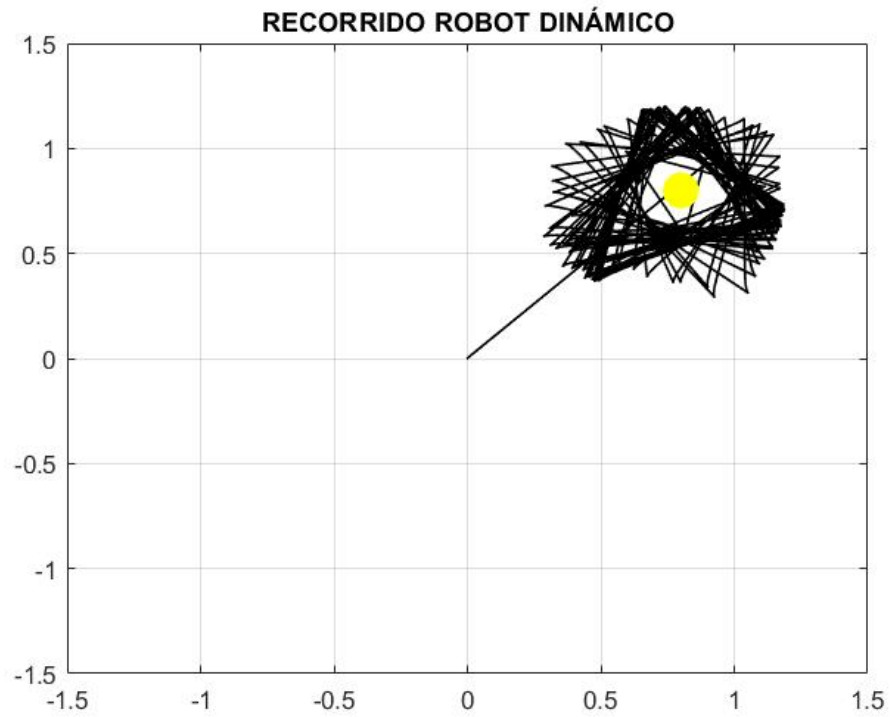
Podemos ver que el robot lo que hace es encontrar un valor óptimo que obtiene al orbitar alrededor de la luz con un radio de 0,5. Así consigue alejarse lo máximo posible de la luz amarilla pero sin descargarse.

RED DINÁMICA

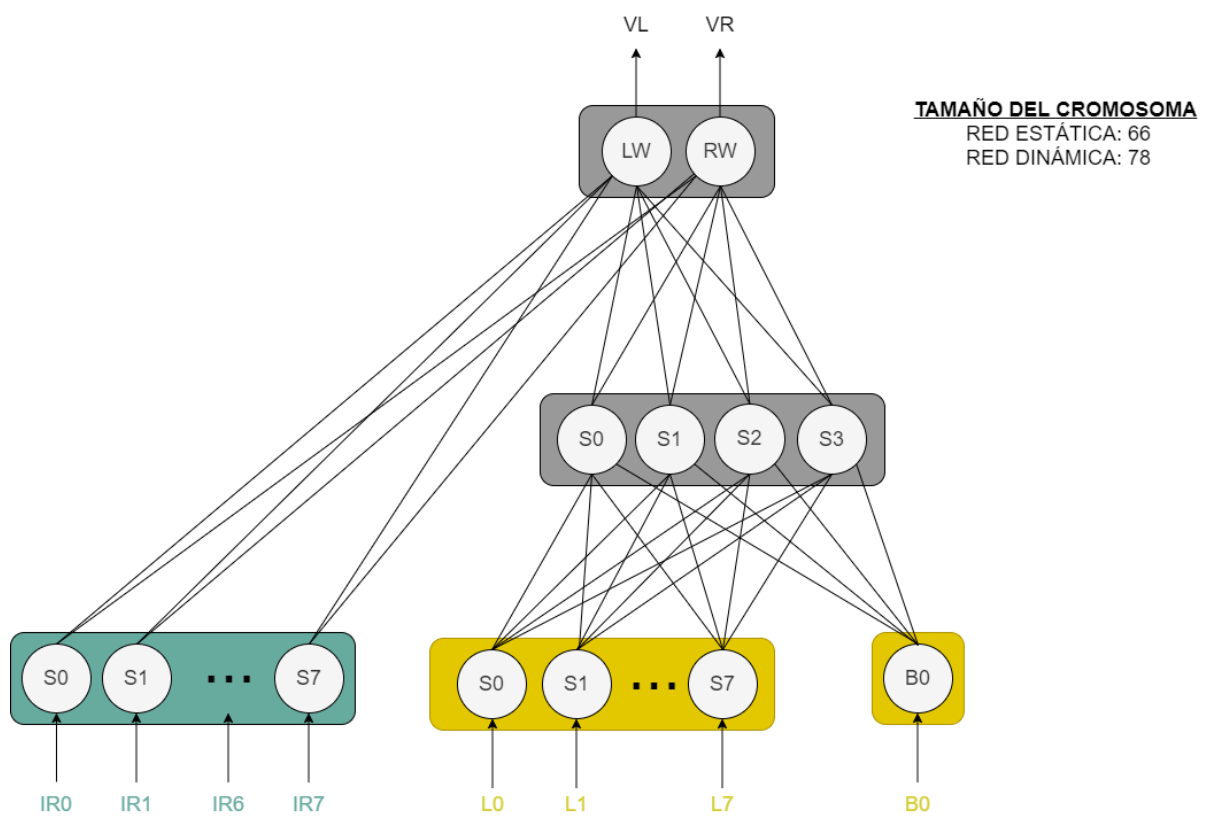


Esta es la gráfica que tiene mejor evolución por el momento: aunque empiece en un valor más bajo que las anteriores fitness pero finalmente aumenta 3 décimas su valor. El average baja su valor respecto al anterior.

Esto lo podemos comprobar al ver el comportamiento del robot: realiza casi la misma trayectoria que el anterior pero empeora el resultado de su experimento estático ya que realizaba una órbita más limpia.



ARQUITECTURA B

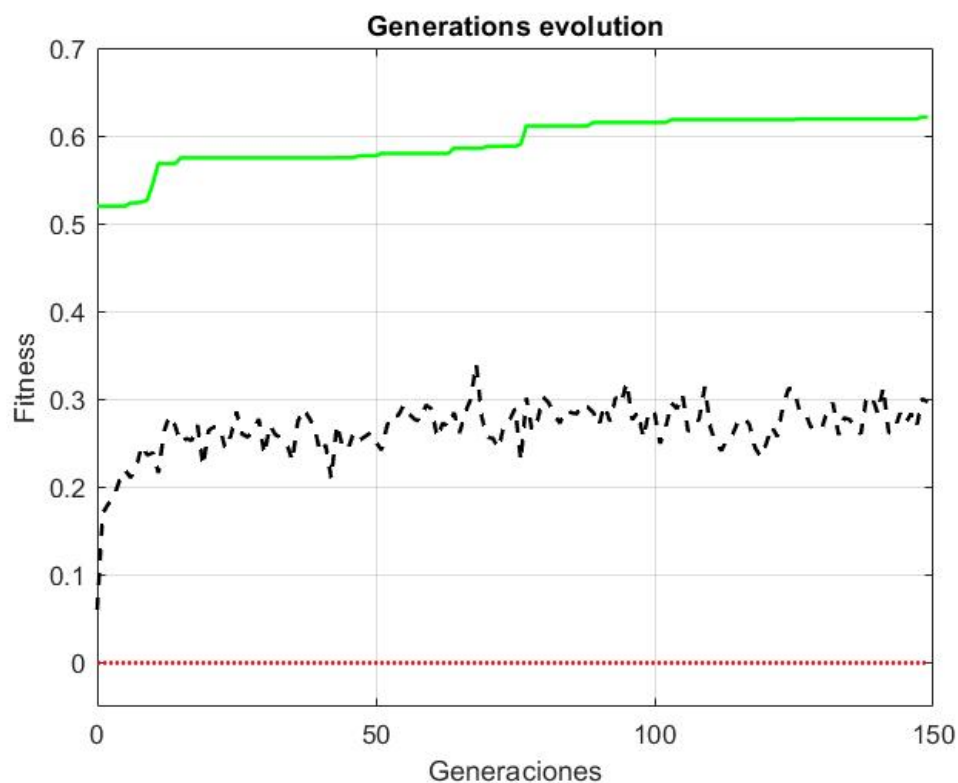


Para este experimento utilizaremos la fitness anteriores pero cambiando la arquitectura del cromosoma. Esta vez añadimos una capa oculta entre los sensores de luz y batería y la capa motora. Por lo tanto aumentará el tamaño del cromosoma a 66 en el caso de la red estática y 78 en la dinámica.

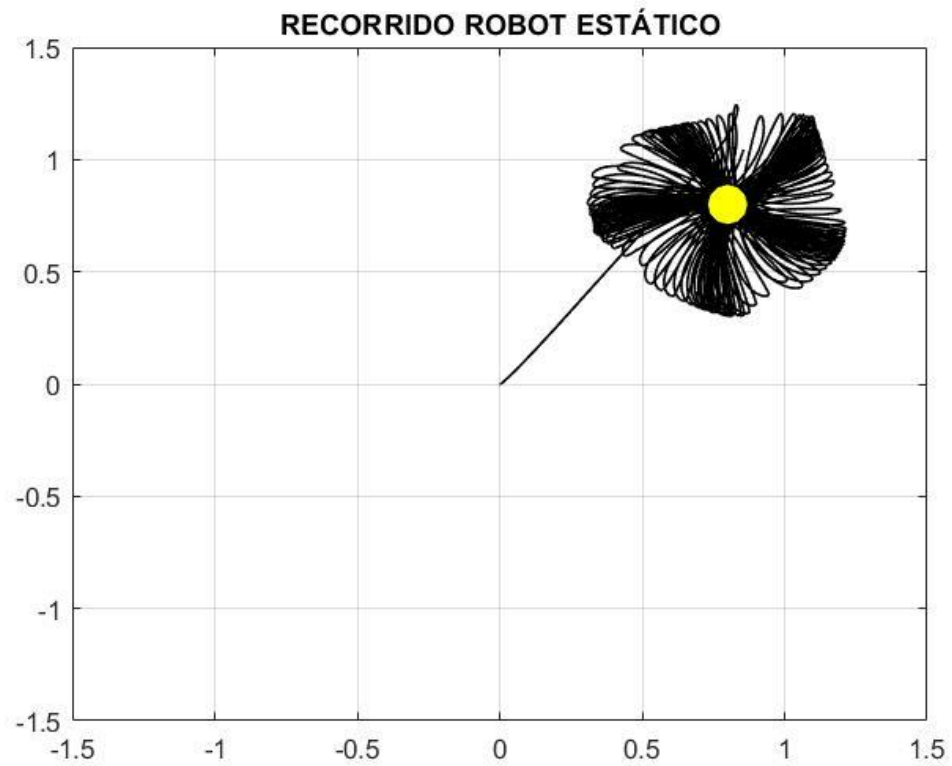
E2.1.B

RED ESTÁTICA

Se utilizará la función fitness del experimento E2.1



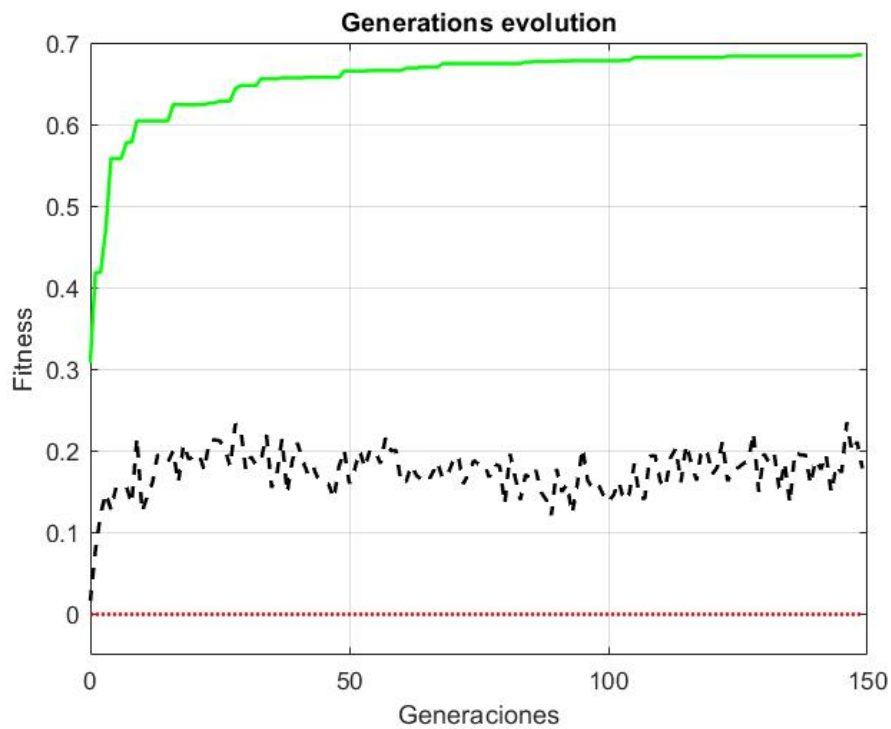
Comparando con el experimento E2.1 estático sin capa oculta el resultado es muy parecido e incluso con peor rendimiento ya que empieza en un valor más alto pero no consigue subir mucho y se queda estancado sólo una décima más alto de lo inicial.



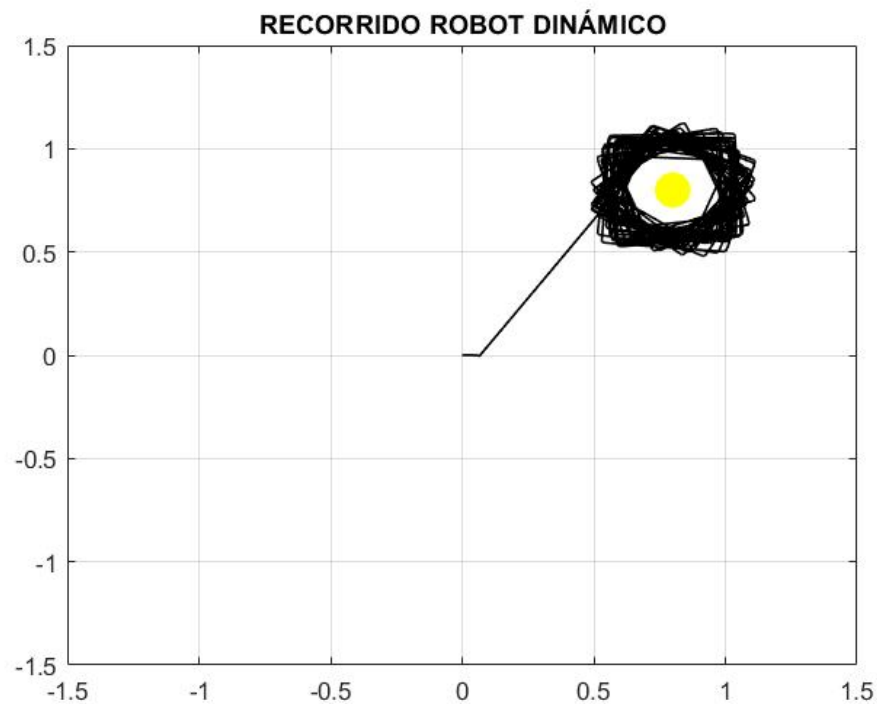
La trayectoria que realiza también es parecida al anterior experimento pero con movimientos más suaves a la hora de cambiar de sentido.

RED DINÁMICA

Se utilizará la función fitness del experimento E2.1



Las generaciones como se puede ver empieza más abajo con la de la arquitectura A y que acaba ligeramente superior.

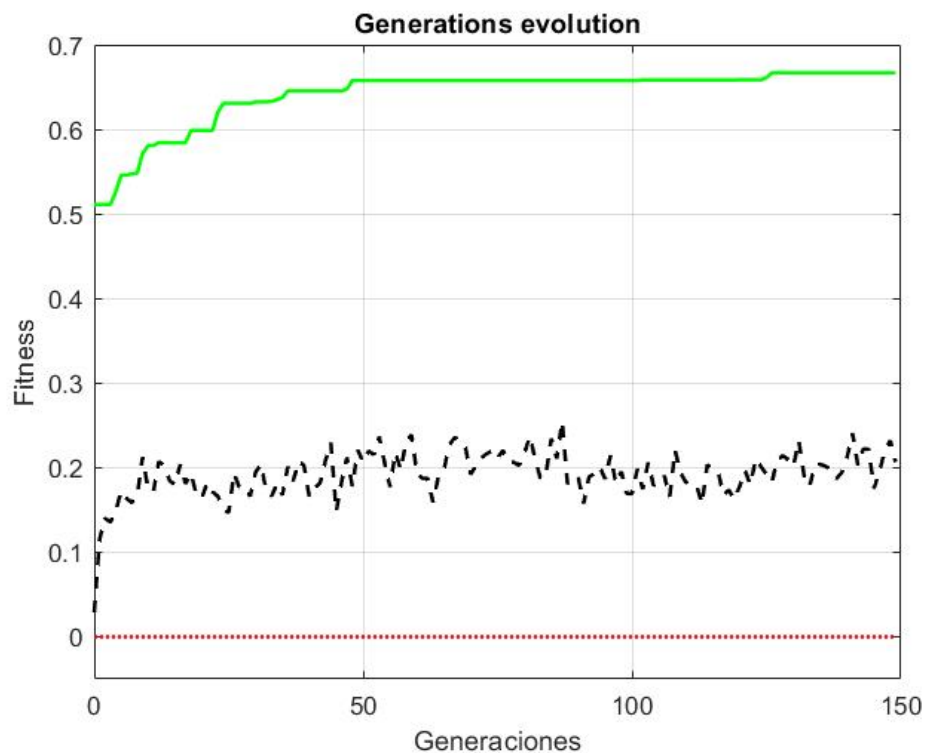


En comparación con el 2.1.A dinámico, vemos que la órbita sigue teniendo forma hexagonal pero esta es más cerrada y cercana a la luz.

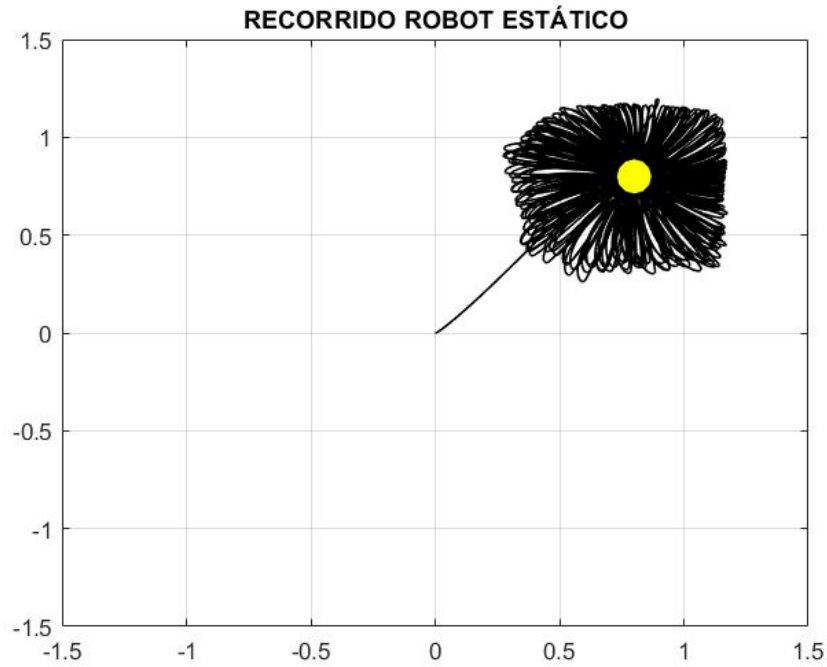
E2.2.B

RED ESTÁTICA

Se utilizará la función fitness del experimento E2.2



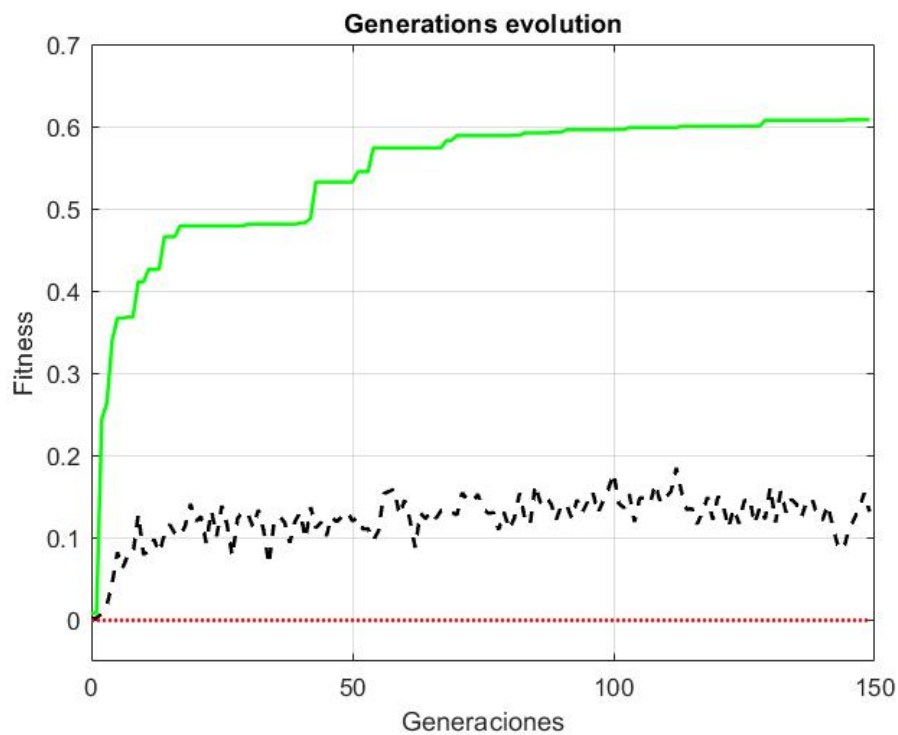
La gráfica es muy parecida a la de la arquitectura anterior para el mismo experimento sólo que ligeramente anterior, sin embargo, su trayectoria va a ser totalmente distinta:



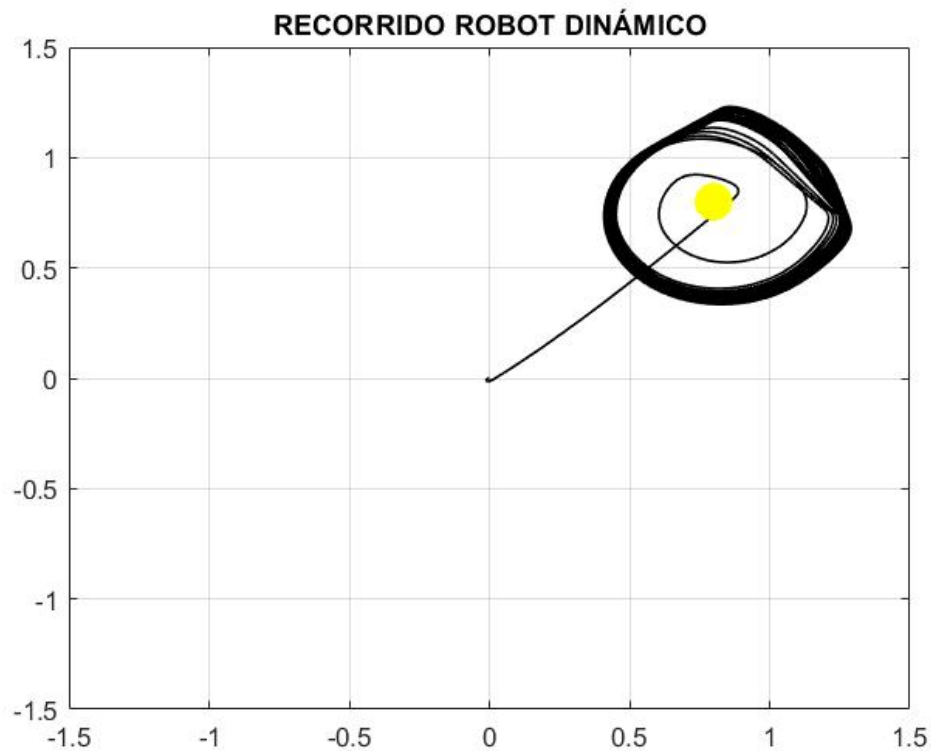
En esta ocasión el trazado del robot es muy distinto al de su versión A, en la cual era un círculo 'limpio' y ahora observamos que se acerca y aleja a la luz mientras traza ese círculo.

RED DINÁMICA

Se utilizará la función fitness del experimento E2.2



En esta gráfica podemos ver que a diferencia de los demás la fitness empieza en un valor muy bajo, prácticamente 0 y acaba en un valor de 0,6, que es peor que en casos anteriores. Además el average es también bastante bajo.



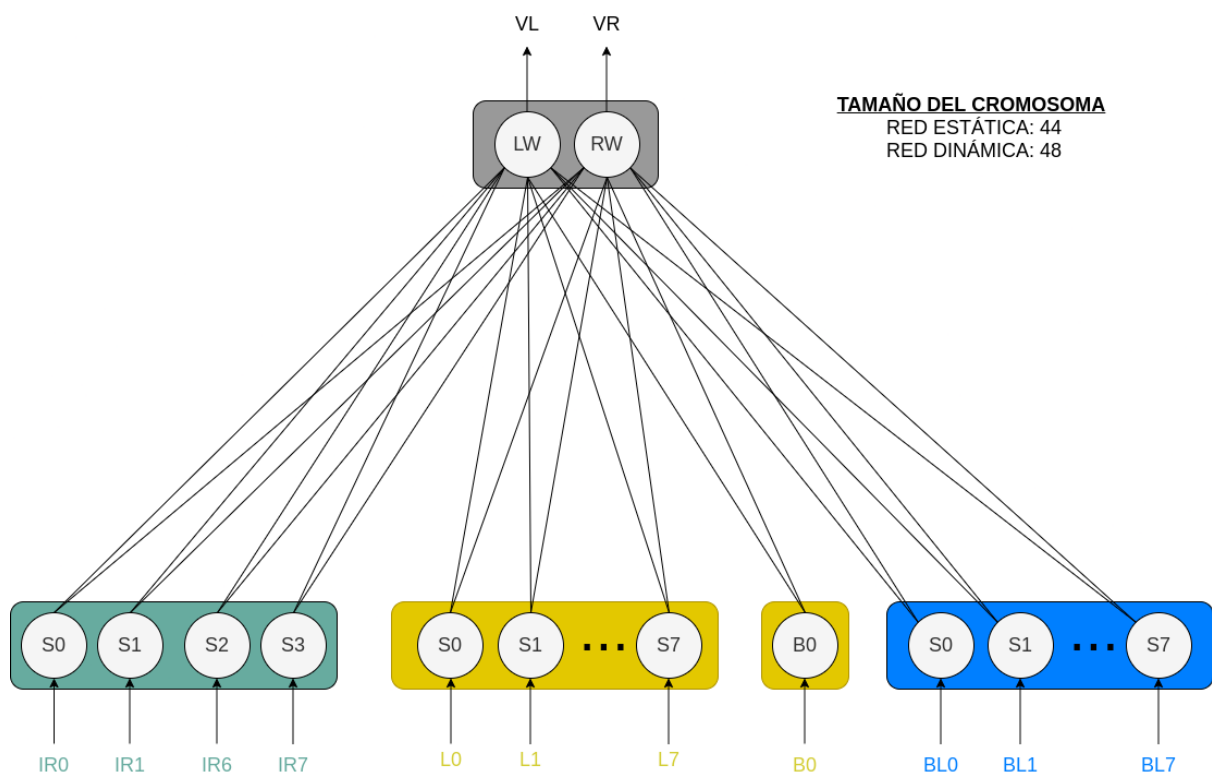
Muy similar a casos anteriores, el robot rodea la luz aunque en un principio pasa por debajo de ella hasta que encuentra la trayectoria circular que posteriormente se limita a seguir indefinidamente.

E3: LUCES Y BATERÍA

Vistos los dos primeros experimentos, vamos a tratar de integrarlos en uno mismo que sea capaz de acudir a una luz (en este caso azul) y de volver hacia la amarilla cuando se quede sin batería. En el experimento anterior nos ha gustado más la fitness 2, y será la que empleemos de aquí en adelante para la parte correspondiente a la batería.

E.3.1

En este primer experimento, vamos a empezar con una arquitectura sencilla sin capas ocultas:



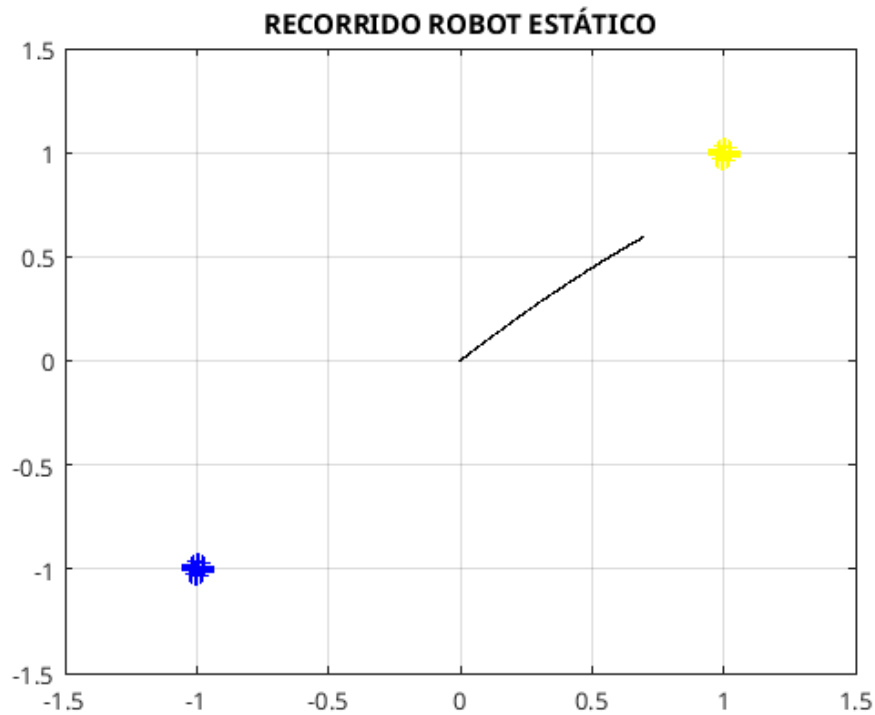
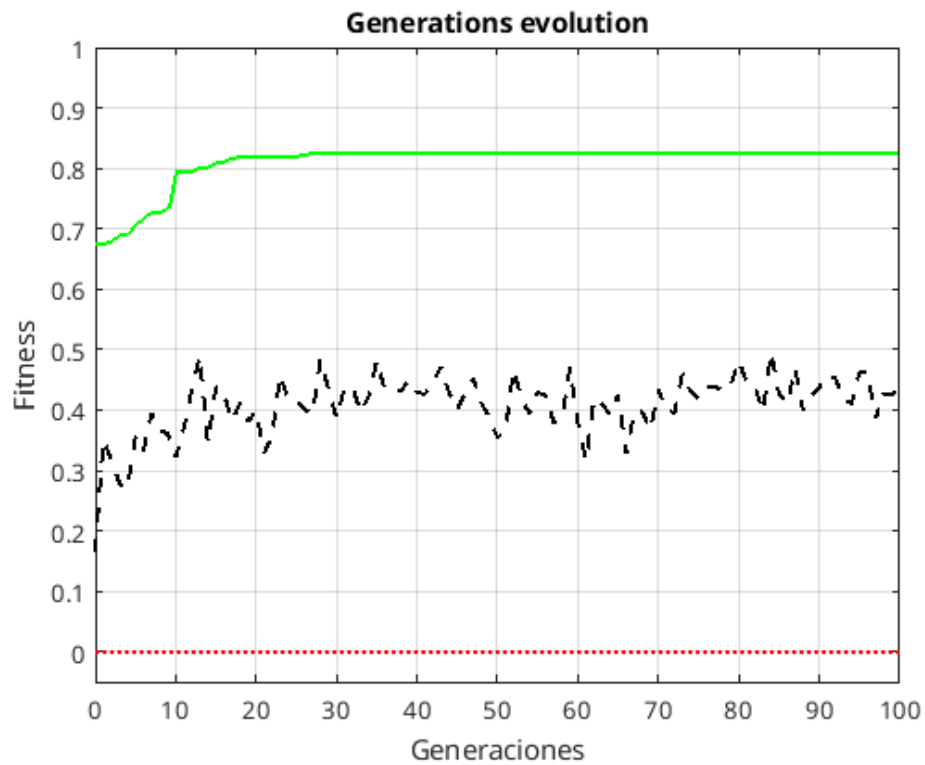
Hemos evaluado 100 generaciones con una población de 50 individuos durante 500 ms.

ECUACIÓN FITNESS:

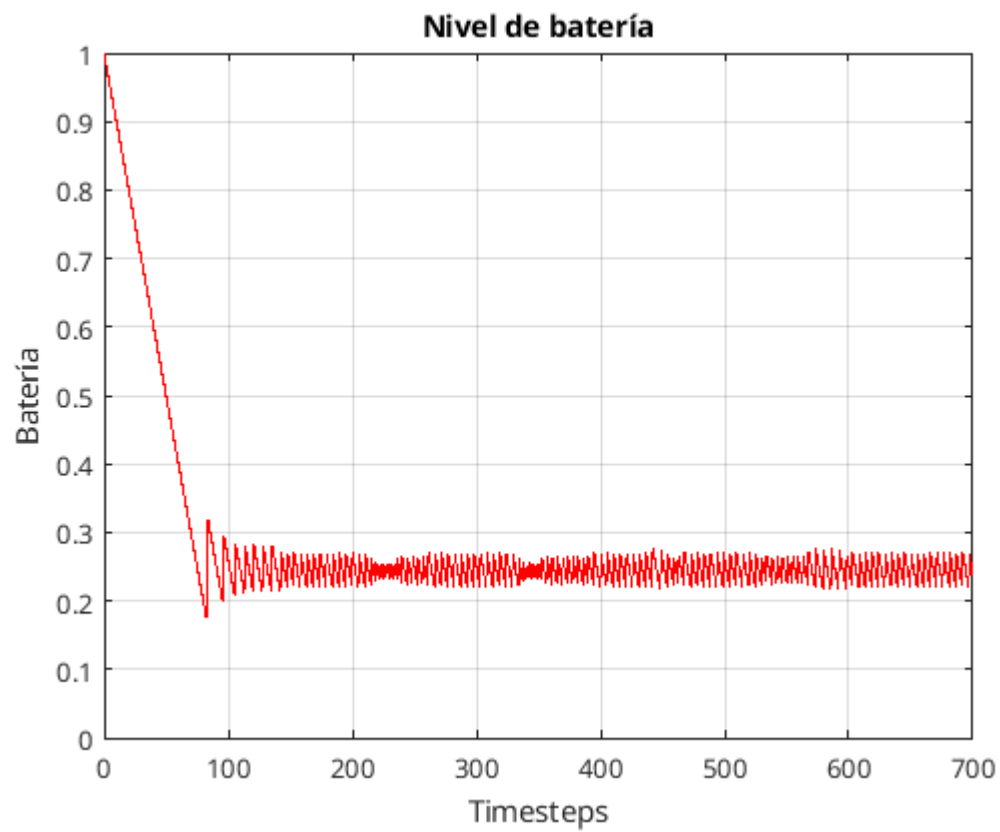
$$\begin{aligned} \text{if Battery} < 0.6 &\Rightarrow f = l \\ \text{if Battery} > 0.6 &\Rightarrow f = (1 - l) * (BL_0 + BL_7) \end{aligned}$$

Siendo l la normalización entre 0 y 1 del máximo valor de los sensores de luz y BL_0 y BL_7 los valores obtenidos por los sensores delanteros de luz azul.

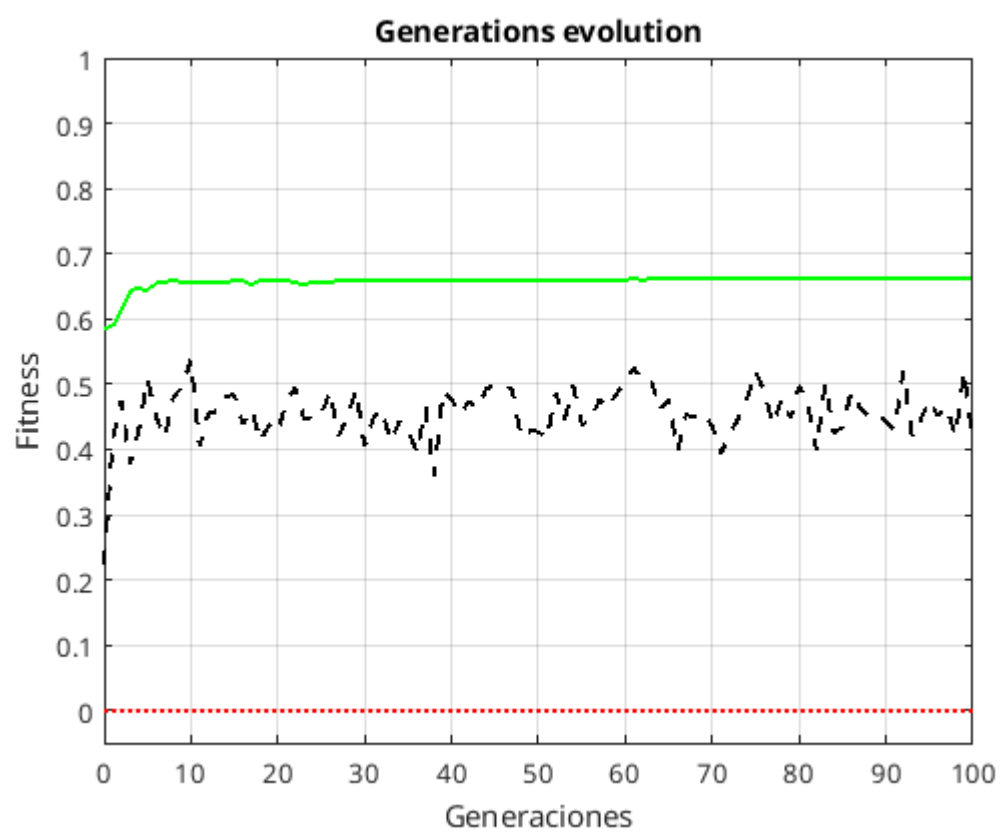
RED ESTÁTICA



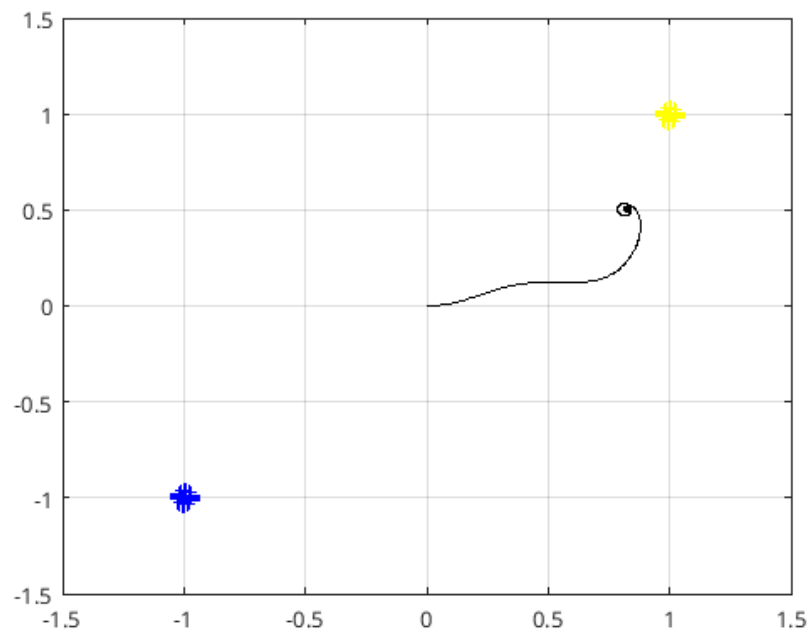
El robot avanza hacia una distancia de la luz amarilla y se detiene ahí. Aunque a priori puede parecer que no cumple con su función, si observamos el nivel de batería veremos que en realidad está situado en la frontera, y está entrando y saliendo de ella para cargarse y descargarse.



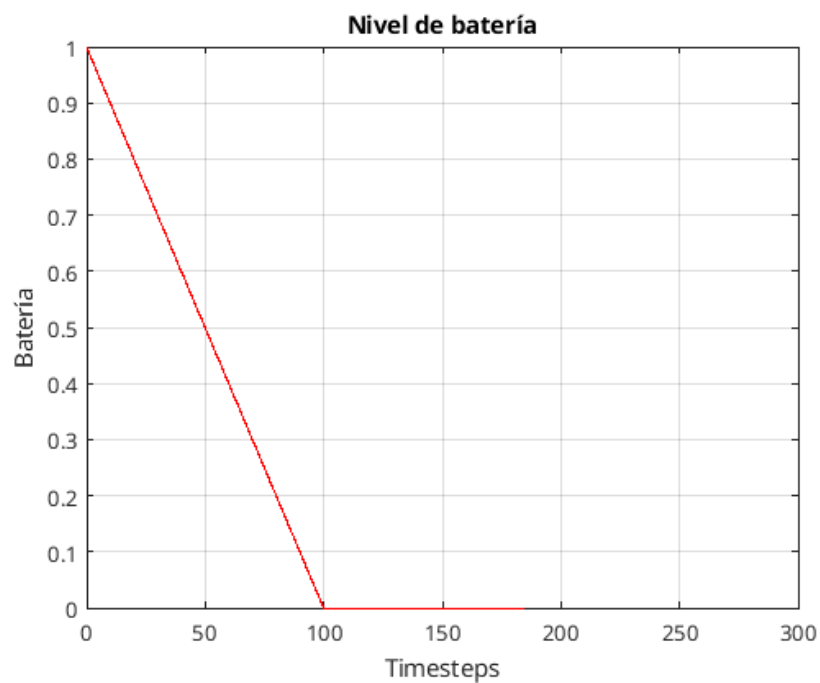
RED DINÁMICA



De entrada observamos una fitness ligeramente peor, pese a estar simulado bajo las mismas condiciones. Vamos a ver la trayectoria que traza el robot:



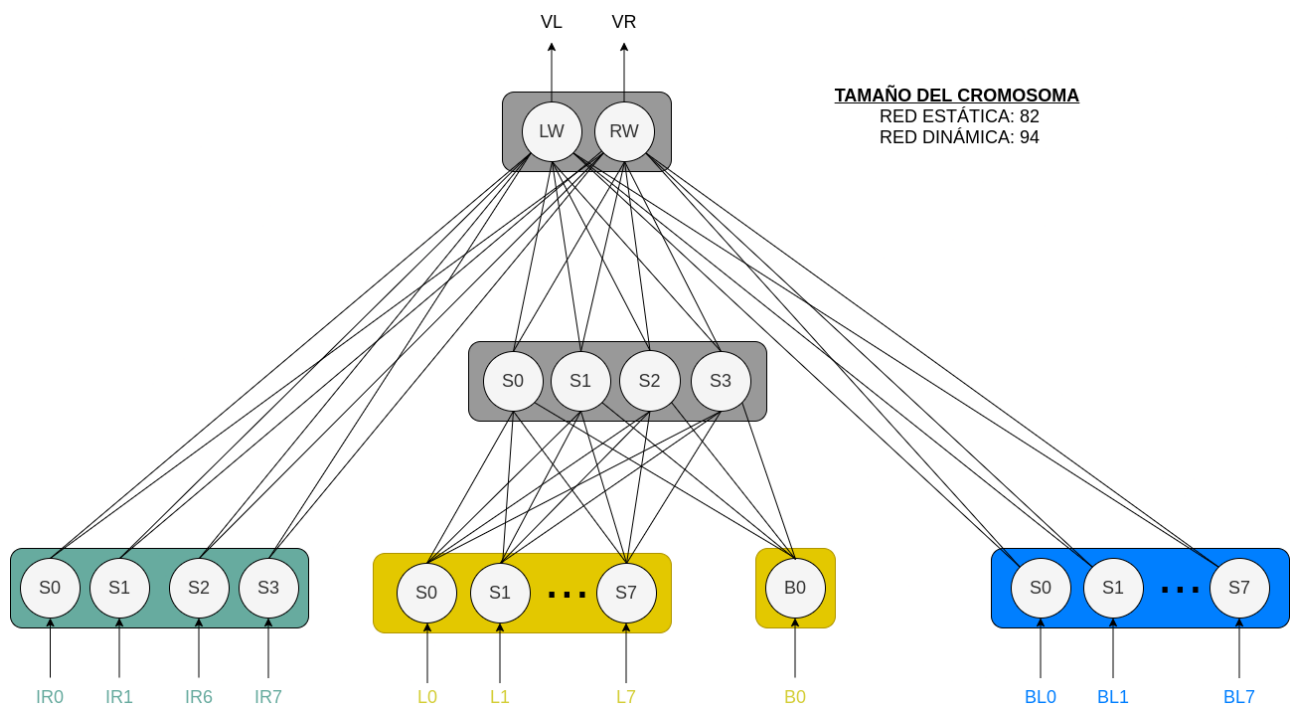
Vemos una trayectoria muy similar respecto a la red estática. Sin embargo, nos hemos encontrado con una diferencia muy importante respecto al caso anterior. La vemos a continuación.



Como vemos, en este caso el robot no entra a la zona a cargar su batería, sino que se limita a dar vueltas sobre sí mismo, cerca de la frontera pero sin entrar a la zona de carga. Por tanto verificamos que en una red dinámica se comporta peor.

E3.2

En el experimento anterior hemos comprobado que la luz azul pasaba un poco desapercibida, por ello vamos a añadir una capa oculta a la arquitectura anterior con la idea de que el robot haga algo más allá de ir hacia la luz amarilla. La función de fitness es la del experimento anterior pero multiplicada por el obstacle avoidance de Floreano. También hemos aumentado el número de generaciones a 500 ya que al tener un cromosoma más complejo pensamos que habrá más margen de mejora. La arquitectura es la que sigue:

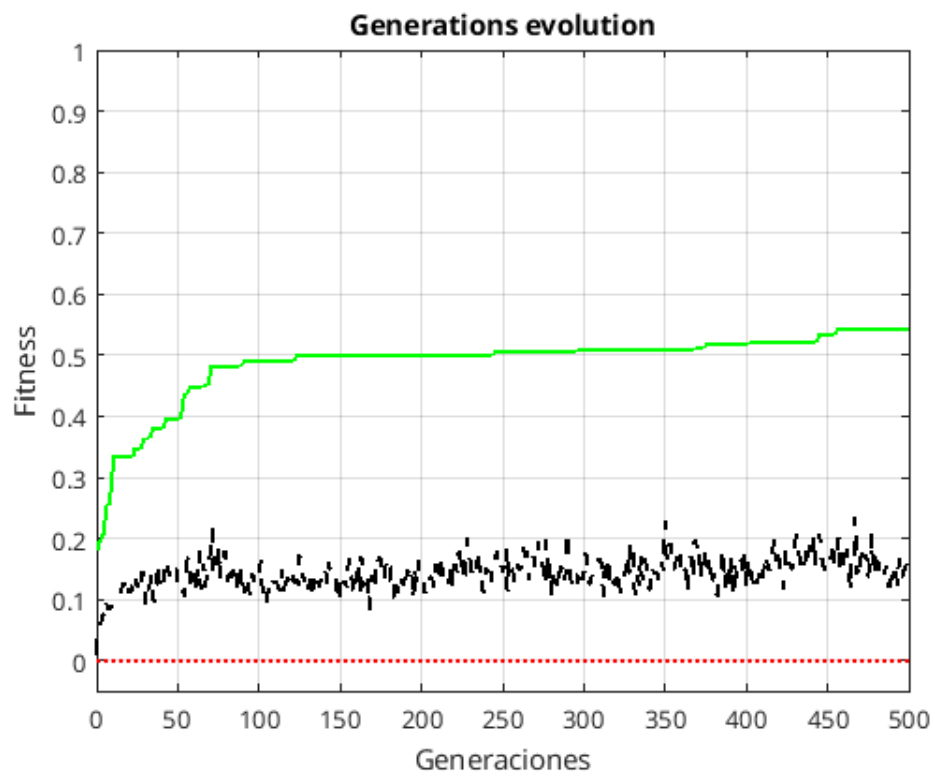


ECUACIÓN FITNESS:

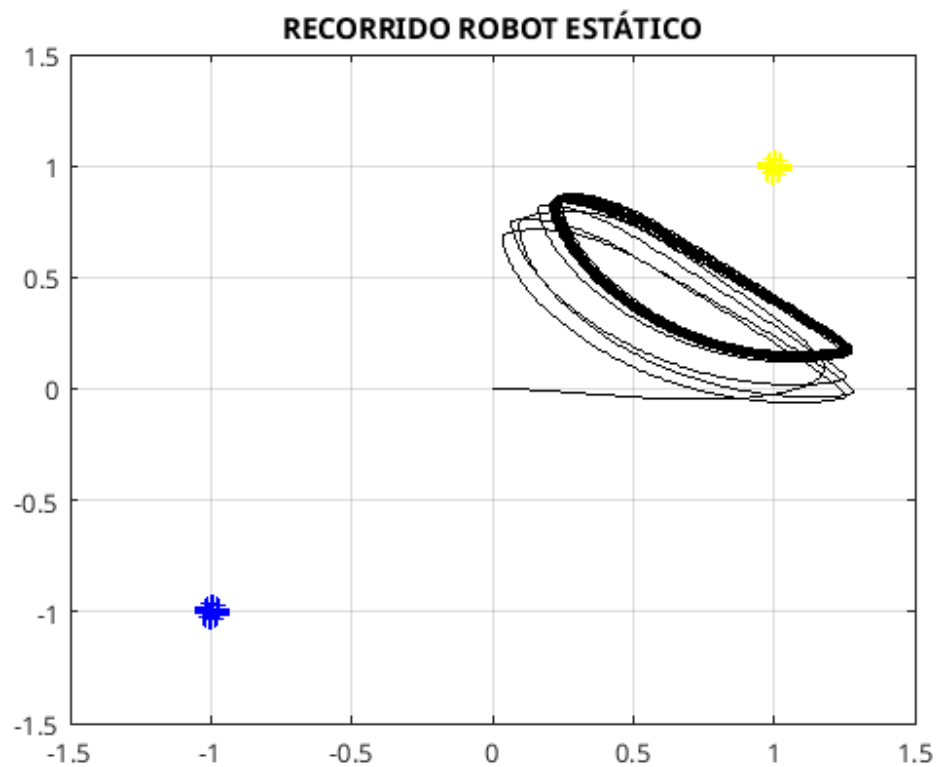
$$\begin{aligned} \text{if Battery} < 0.6 & \Rightarrow f = V(1 - \sqrt{\Delta v}) * l \\ \text{if Battery} > 0.6 & \Rightarrow f = V(1 - \sqrt{\Delta v}) * (1 - l) * (BL_0 + BL_7) \end{aligned}$$

Siendo l la normalización entre 0 y 1 del máximo valor de los sensores de luz y BL_0 y BL_7 los valores obtenidos por los sensores delanteros de luz azul y donde V es la suma de los valores absolutos de cada una de las velocidades de cada motor menos 0.5 y Δv es el valor absoluto de la diferencia entre las velocidades.

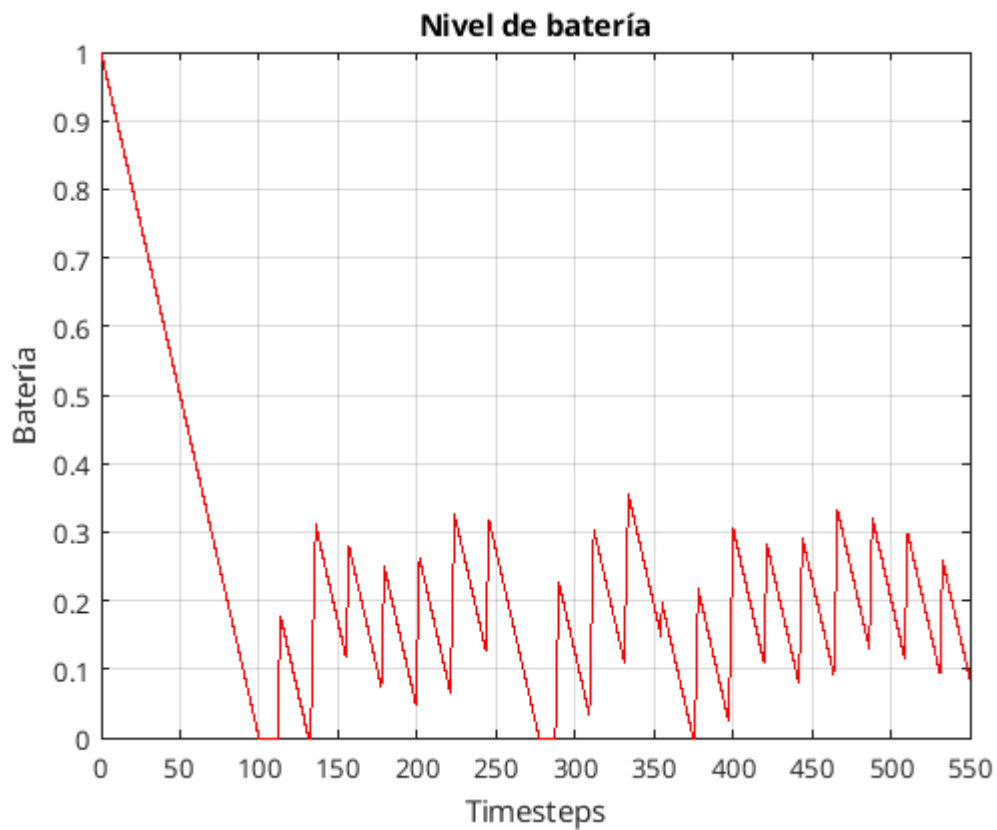
RED ESTÁTICA



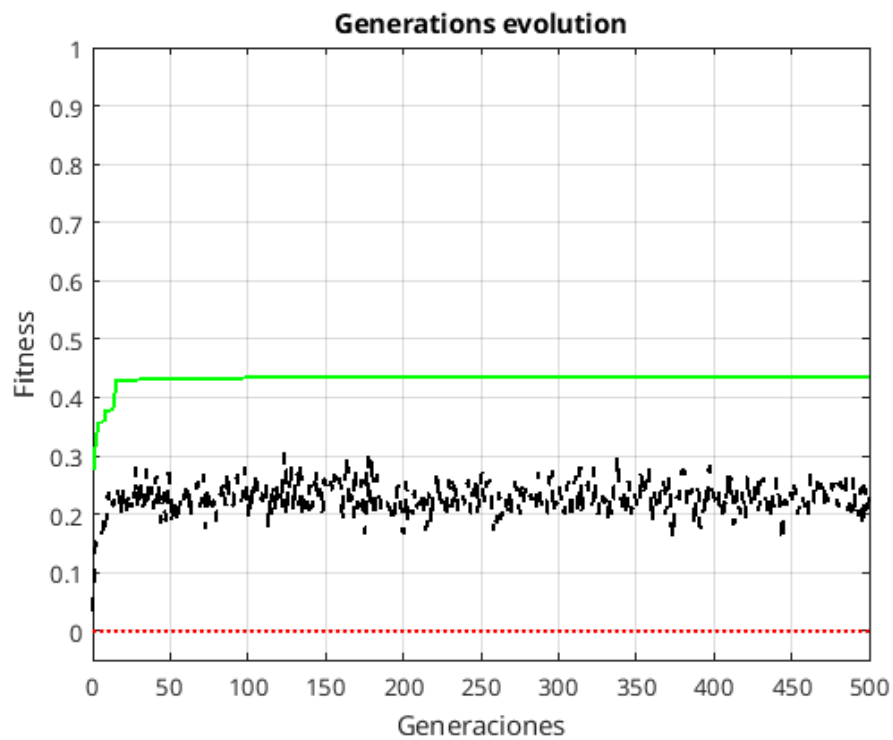
Se produce un rápido crecimiento en las primeras generaciones, que se estabiliza en torno a 0.5 durante el resto de la evolución.



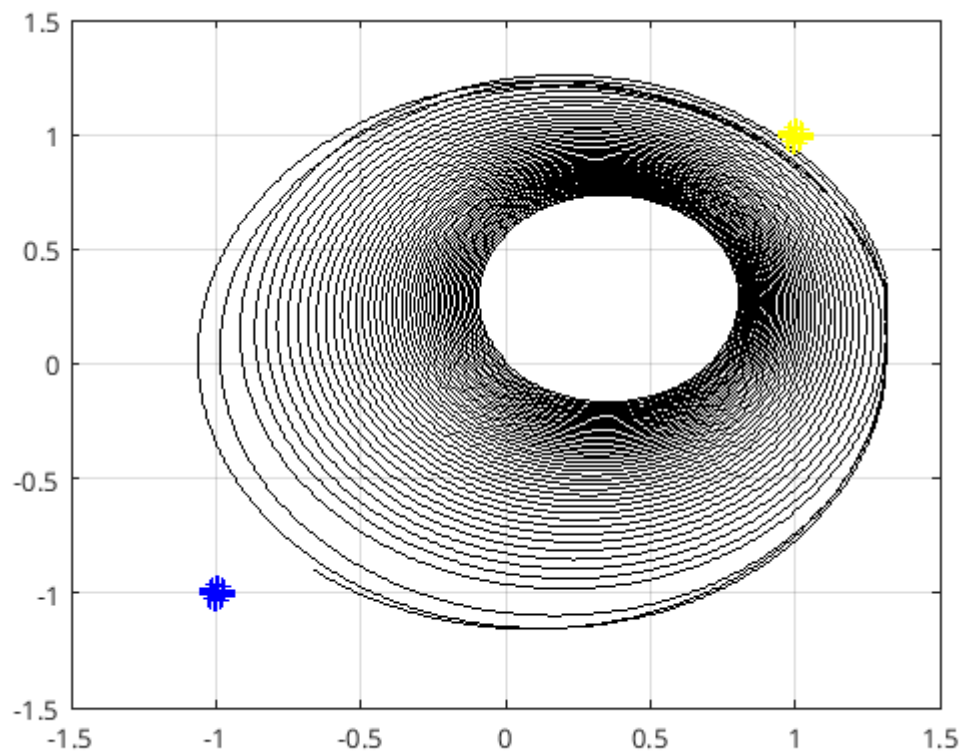
El robot se sitúa en la frontera, al igual que en el caso anterior, pero lejos de quedarse estático en una posición, ahora entra y sale de la zona de carga con una trayectoria elíptica. Como vemos a continuación, al principio se llega a producir la descarga de la batería (aunque en un periodo muy breve) pero una vez el robot alcanza la órbita fija (línea gruesa en el gráfico anterior) no vuelve a ocurrir el caso en el que la batería se descarga.



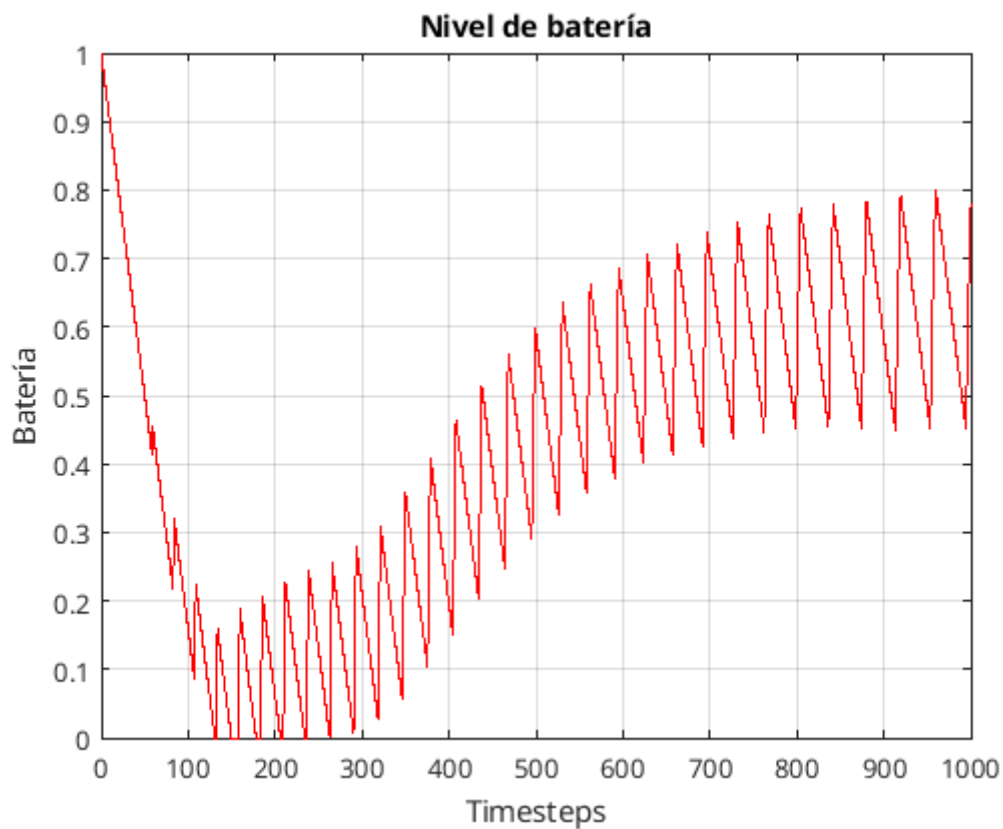
RED DINÁMICA



En la red dinámica la fitness empeora y no consigue subir en casi toda la evolución, algo que nos ha extrañado ya que en 400 generaciones apenas se ha producido una mínima mejora.

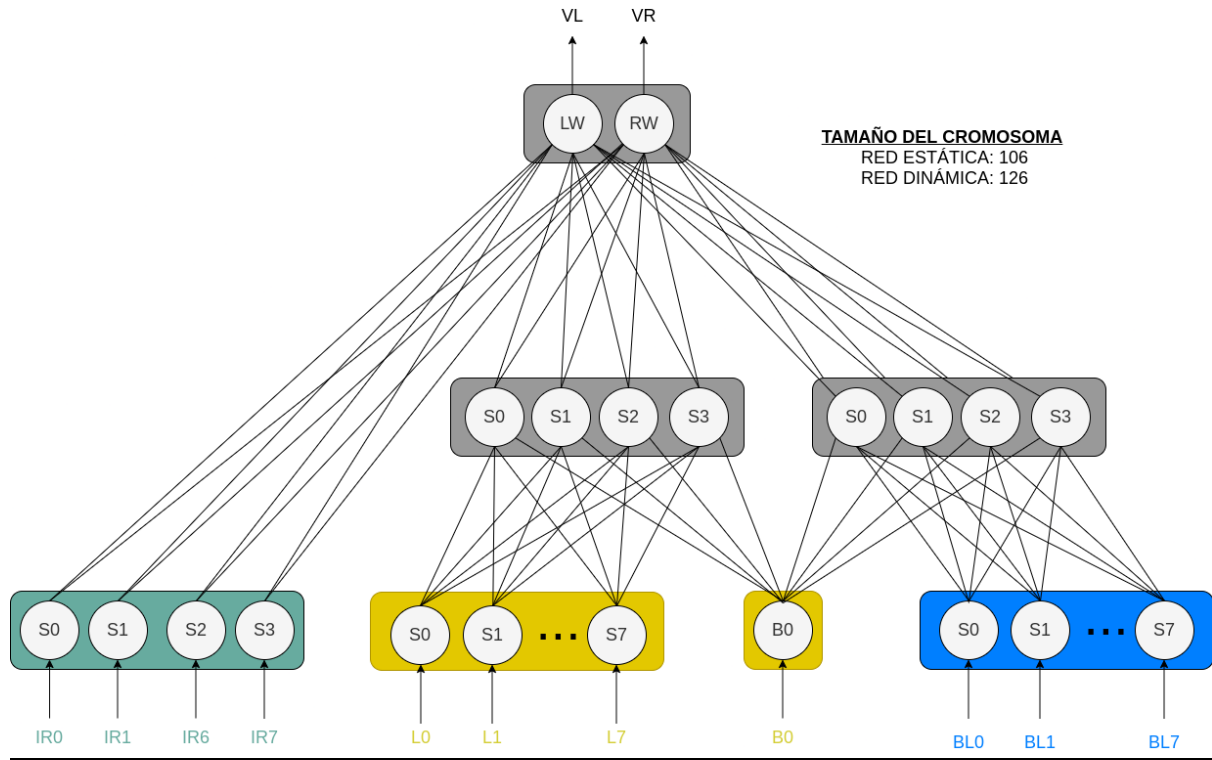


El robot realiza una trayectoria elíptica desde el comienzo, con la peculiaridad de que va aumentando su radio con cada vuelta. Esto podría explicarse como que está intentando ir a la luz azul pero haciendo círculos, lo que realmente es nuestro mejor resultado hasta ahora, ya que es la primera vez que conseguimos que entre la luz azul en juego y además cumple la función de cargarse. En las primeras vueltas, al ser el radio pequeño, hay un déficit de carga que se recupera según se va haciendo el radio más grande. Esto da lugar a una curiosa gráfica del nivel de batería:



E3.3

En el experimento 2 hemos conseguido que, aunque el funcionamiento siga sin ser el ideal, intervenga la luz azul. Ahora vamos a añadir una capa oculta adicional, buscando una mejora más notable. Además, vamos a gestionar la 'orden' de ir a cargarse mediante un ciclo de histéresis para que no sea tan brusco el cambio de fitness. Vemos la arquitectura a continuación:

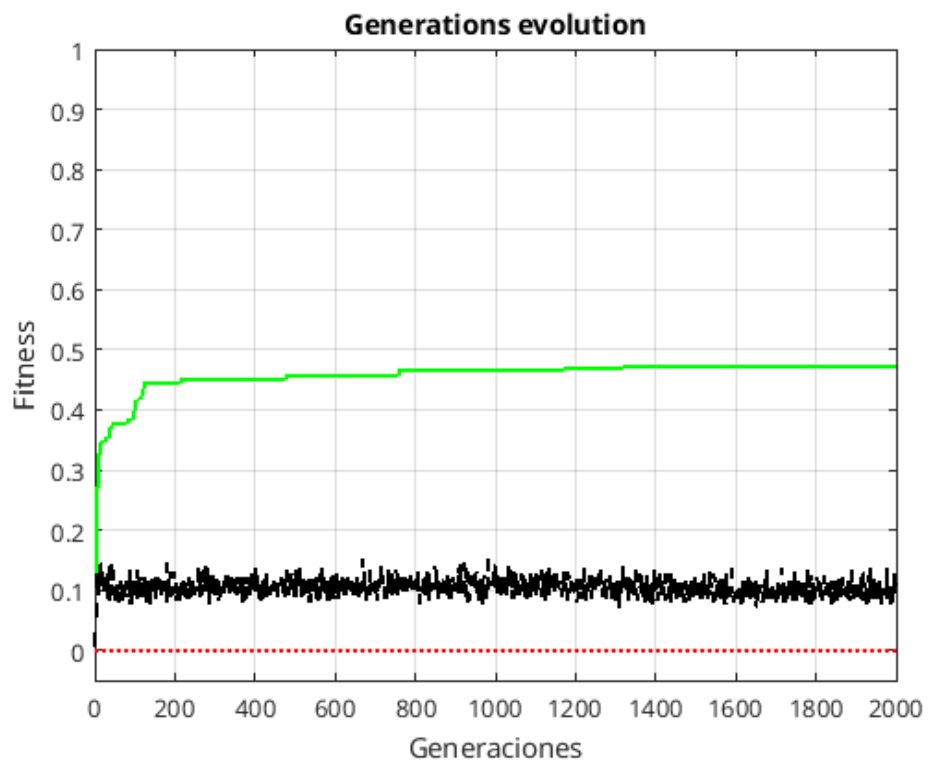


ECUACIÓN FITNESS:

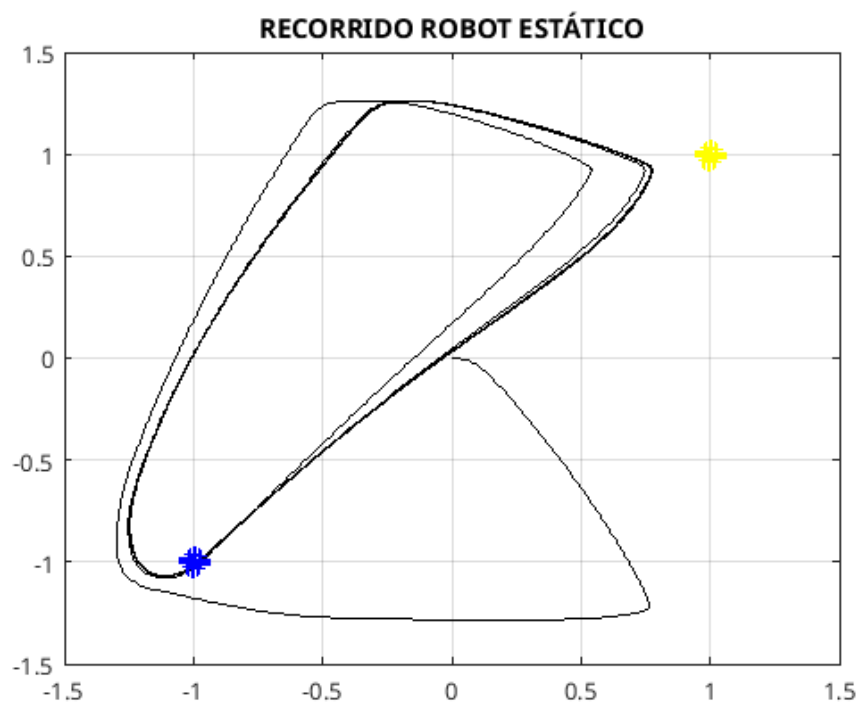
$$\begin{aligned}
 \text{if Battery} < 0.3 &\Rightarrow \text{flag} = 1 \Rightarrow f = V(1 - \sqrt{\Delta v}) * l \\
 \text{if Battery} > 0.7 &\Rightarrow \text{flag} = 0 \Rightarrow f = V(1 - \sqrt{\Delta v}) * (1 - l) * (BL_0 + BL_7) \\
 \text{if Battery} = 0.0 &\Rightarrow f = 0
 \end{aligned}$$

Donde los parámetros utilizados están descritos en el experimento anterior.

RED ESTÁTICA

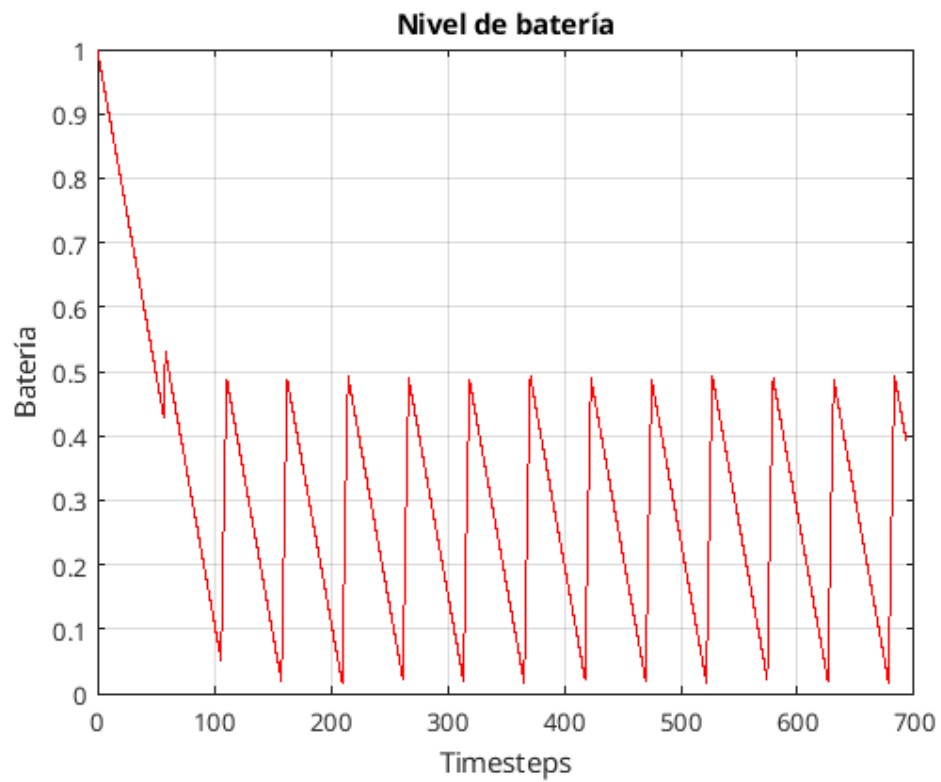


La fitness se eleva rápidamente desde casi 0 hasta cerca de 0.45, desde ahí no sufre un gran crecimiento. La fitness máxima es baja (no llega a 0.5) pero estamos ante el buque insignia de nuestro proyecto. Vamos a verlo a continuación:

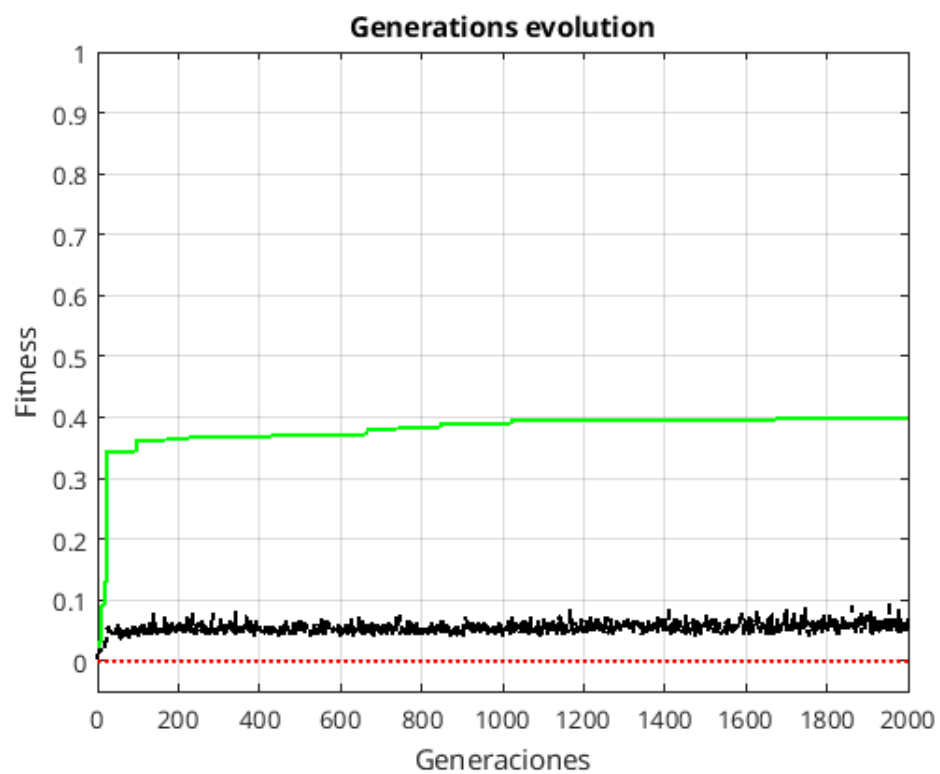


Hemos conseguido que el robot vaya directamente a la luz azul, y de ahí vuelva a la amarilla a cargarse, lo que cumple con creces lo que pretendíamos desde el principio, y es un comportamiento casi ideal.

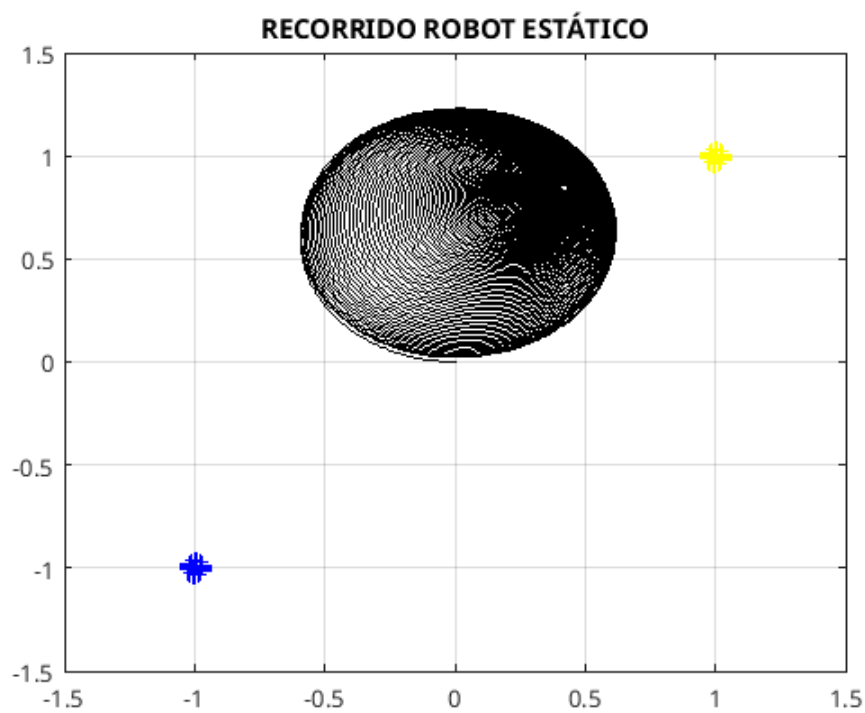
El nivel de batería realiza un ciclo constante en el que cuando está a punto de quedarse sin batería vuelve a cargarse hasta casi 0,5 y así sucesivamente.



RED DINÁMICA

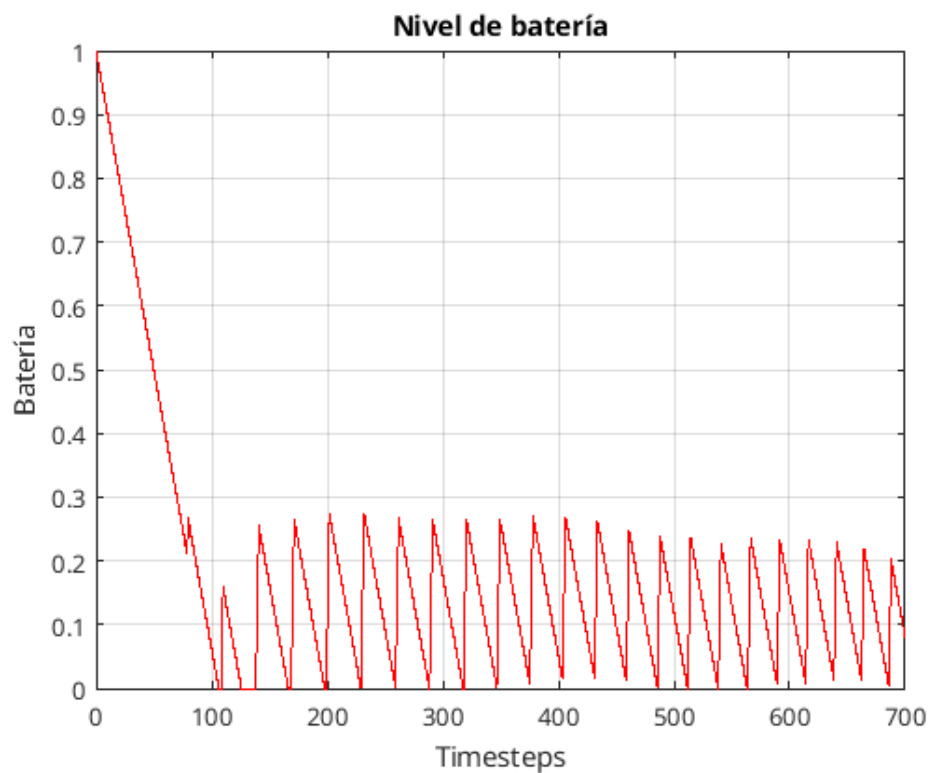


Realizando el último experimento para una red dinámica hemos obtenido el peor resultado en todos los sentidos: la fitness sólo llega a rozar el 0,4 y sin mejorar en 1000 generaciones.



No le encontramos explicación posible a la trayectoria que sigue el robot, y más teniendo en cuenta que tanto la arquitectura como la función de fitness son las mismas que en el caso anterior que tan buen resultado nos ha dado. El robot realiza trayectorias elípticas, en las que se va desviando hacia arriba a la derecha, ni se acerca a la luz azul ni se acerca mucho a la batería, por lo que el resultado con una red dinámica podemos decir que es bastante malo.

En cuanto a la batería, vemos que es muy frecuente que se llegue a agotar y que cuando se carga no lo haga a un nivel muy alto.



CONCLUSIÓN

Tras el desarrollo de todos los experimentos, nuestra conclusión principal es que para nuestro caso particular las redes estáticas nos han dado un mejor rendimiento que las dinámicas. Esto nos ha sorprendido ya que nosotros esperábamos justo lo contrario, pensábamos que en las dinámicas el robot se iba a adaptar mejor.

Hemos quedado aliviados con que, tras tantas evoluciones, al menos uno de todos los experimentos (3.3 estático) haya dado un resultado tan positivo, de hecho mucho mejor de lo que pensábamos que obtendríamos cuando empezamos, ya que al comienzo no nos salía nada por más pruebas que hacíamos y fue un poco desesperante.

Entre los aspectos negativos de este trabajo es que la capacidad de computación de nuestros ordenadores a la hora de realizar las generaciones no era muy potente y para casos de muchas generaciones hemos tardado muchísimo más en un ordenador que en otro, perdiendo mucho tiempo que podríamos haber empleado en probar nuevas fitness y estructuras.

BIBLIOGRAFÍA

- [Fundamentos Teóricos de la Asignatura y Manual del Simulador IRSIM](#)
- [Robótica Evolutiva](#)
- [Ejemplos de Robótica Evolutiva. Tarea Evitar Obstáculos](#) (4/05/21)
- [Ejemplos de Robótica Evolutiva. Tarea Luces](#) (12/05/21)
- [Ejemplos de Robótica Evolutiva. Tarea Recolección](#) (14/05/21)
- Archivos dentro del simulador Irsim 2021