ISSN 1870-4069

Movimientos en pierna robótica para el pateado de un balón a través de reinforcement learning

Gerardo Franco Delgado, M. De La Rosa, C. E. De Los Santos, J. D. Carrillo

IXMATIX ROBOTICS, San Luis Potosí, S.L.P, México
gfranco@ixmatix.com, {m.delarosa, c.delossantos, dcarrillo}.ixmatix@gmail.com

Resumen. En este trabajo, se presentan los resultados obtenidos del aprendizaje de una pierna robótica para el pateado de un balón. El entrenamiento se realizó por medio de reinforcement learning (RL) utilizando el algoritmo Q-learning. Esto con el objetivo de probar la fiabilidad del aprendizaje para futuros trabajos que involucren caminado de robots cuadrúpedos y manipulación de objetos con actuadores. El entrenamiento fue realizado dentro de un entorno de simulación en Gazebo con la intención de reproducir de forma real el aprendizaje obtenido de la simulación, es decir, que el agente pueda realizar el pateado del balón en una pierna robótica real, a través de un entrenamiento simulado.

Palabras clave: reinforcement learning, pierna robótica, Q-learning, manipulación robótica.

Movements in Robotic Leg for Kicking A Ball Using Reinforcement Learning

Abstract. The results shown in this work were obtained from a robotic leg kicking a ball using reinforcement learning (RL), specifically the Q-learning algorithm. The goal of this study was to test the reliability of learning using RL, all this with the aim to apply these concepts in future work, like quadruped walking robots and object manipulation using electromechanical actuators. The training was carried out within a simulation environment in Gazebo with the intention of reproducing in a real way the learning obtained from the simulation, so in the end the agent can perform the kicking of the ball on a real robotic leg, through training simulated.

Keywords: reinforcement learning, robotic leg, Q-learning, robotic manipulation.

1. Introducción

En la literatura actual, se han visto avances sobresalientes en el campo de la inteligencia artificial mediante el uso de reinforcement learning. Una de las

principales características de este, es que imita la forma en cómo aprendemos los humanos, donde a través de la práctica se mejora en la actividad que realiza [11]. Este enfoque ha logrado dar soluciones a retos complejos, como ganarle al campeón mundial de Go [22] y la creación de un algoritmo que permite a un agente aprender a jugar StarCraft II [27], logrando vencer a humanos profesionales sin importar lo complejo del entorno.

En este trabajo se busca demostrar la factibilidad del uso de algoritmos de auto-aprendizaje con RL para el movimiento de una pierna robótica, donde esta sea capaz de patear un balón sin la necesidad de definir la posición de este ni los ángulos de las articulaciones.

Se han desarrollado diversas investigaciones para el pateado de un balón basados en el movimiento de una pierna humana, entre ellas podemos mencionar a Roboleg [21], un robot desarrollado para la evaluación de tenis y balones, y el robot de 9DOF (por sus siglas en inglés, Degrees Of Freedom) que utiliza el método Denavit-Hartenberg para el análisis de cinemática y un método recursivo langriano para definir la dinámica [26].

Por otra parte, nuevas investigaciones han tratado el pateado del balón en robots de tipo humanoide como HOAP-3 [23] y NAO [13], el cual utiliza movimientos basados en trayectorias como un conjunto de curvas de Bezier por medio de estabilización del centro de masa y un controlador PID. A diferencia de estas propuestas, nuestro trabajo se asemeja a lo realizado por Riedmiller et al. (2001) en la competencia RoboCup, creando un agente que utiliza RL para el pateado del balón, definiendo 536 acciones que el agente puede escoger por ciclo [19]. En nuestro trabajo definimos el pateado por medio de una pierna robótica de 3DOF que se define con un total de 1848 acciones disponibles por estado, el cual es entrenado en simulación y aplicado en la realidad.

2. Descripción del sistema

El entrenamiento para el pateado del balón se realiza únicamente en la simulación y los resultados obtenidos se muestran tanto en el entorno real como en el simulado. El algoritmo Q-learning [29] es con el cuál se busca maximizar una acción en un determinado estado.

En cuanto al reconocimiento del balón, para conocer sus coordenadas, se tiene un clasificador de objetos por medio de OpenCV [25] que hace uso del algoritmo de Viola-Jones. [8, 28]

2.1. Entorno

Las características para el pateado del balón pueden definirse de acuerdo a la clasificación de entornos para agentes [20] de la siguiente forma:

- Observable: La cámara observa el entorno completo, obteniendo la información necesaria para el agente.

- Determinista: Un entorno determinista es aquel en que el siguiente estado está completamente determinado por el estado actual y la última acción ejecutada por el agente.
- Episódico: La experiencia de la pierna se encuentra dividida por episodios independientes, donde al iniciar un nuevo episodio se empieza desde un nuevo estado.
- Semi-dinámico: Durante el entrenamiento el entorno se mantiene inmutable, mientras que el agente va aumentando su rendimiento.

Los objetos delimitados dentro del entorno se definen por la pierna robótica, una mesa, una cámara y un balón de fútbol estándar (diámetro entre 68 y 70 cm).

Se utiliza un ángulo en picado [12] para la cámara, de tal manera que pueda observar hacia abajo el entorno completo, con un ángulo aproximado de 45 grados bajo la horizontal. El balón tiene límites en su ubicación, definidos por el alcance de la pierna robótica.

2.2. Diseño mecánico

El diseño está basado en las patas traseras de un canino e intenta imitar su movimiento de locomoción, además de tener en consideración las especificaciones de los motores hidráulicos. Se tiene una estructura con cada una de las partes que conforman la pierna robótica, este diseño será usado posteriormente para la simulación.

El ensamblado de la misma, se realizó con impresiones en 3D y perfiles de aluminio; cabe mencionar que dicho diseño es original y es realizado por el equipo de trabajo.

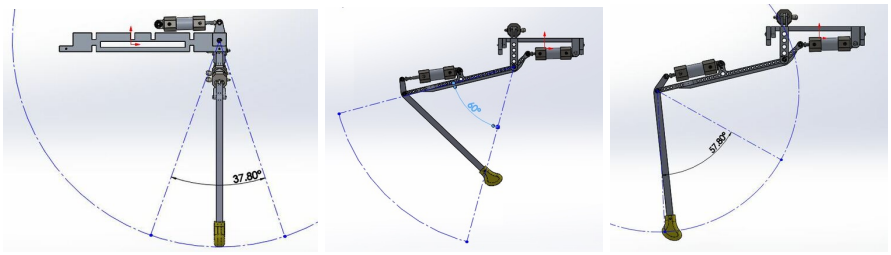
2.3. Configuración y medidas del robot

La pierna robótica consta de tres actuadores y articulaciones, que permiten tener 3DOF. Cada articulación se nombra como se muestra en la figura 2.3.

El movimiento de la pierna robótica se ve limitado debido a su estructura mecánica. Cada articulación tiene un ángulo de apertura que se calcula desde el punto máximo que permite el actuador, hasta el punto mínimo. Los ángulos de apertura para cada articulación son los siguientes: 37.8 grados para la coxa, 60 grados para el fémur y 57.8 grados para la tibia, tal como se muestra en la figura **??**.

3. Simulación

Para asegurar una aproximación acertada de la simulación de robots en entornos virtuales, es importante definir diversas características y propiedades físicas sobre cada pieza, en este caso, de la pierna del robot. Tales consideraciones incluyen el peso, centro de masa, la matriz diagonalizable del momento de



(a) Coxa
Vista lateral izquierda
37.8 grados de apertura.

(b) Fémur
Vista frontal
60 grados de apertura.

(c) Tibia
Vista frontal
57.8 grados de apertura.

Fig. 1. Vistas de los ángulos de apertura para cada actuador en la pierna robótica.

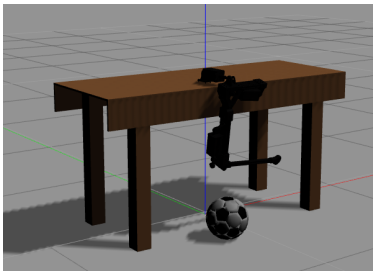


Fig. 2. Simulación de la pierna del robot.

inercia, así como una representación de mallas 3D para definir las colisiones y su visualización. En la figura 2, se muestra el entorno de simulación.

Estas propiedades se definen en un archivo con formato URDF [24] para poder utilizarlas en Gazebo [10] y ROS [18] simultáneamente. El entorno de simulación se muestra en el video [3] (accesible desde este url[1]).

3.1. Controladores

Los motores hidráulicos de cada articulación, cumplen con las características en esfuerzo y velocidad (rad/s) definidos en la ficha de datos del proveedor, con el modelo: 32TRCHEBU9A65 [14].

Estos límites se definen mediante la siguiente expresión: $-k_v(v - v^+)$. Donde k_v determina la escala del límite del esfuerzo, v expresa velocidad y v^+ el límite de la velocidad [9].

[1] `https://www.youtube.com/watch?v=F34zIs0cI-0`

4. Entorno real

4.1. Características del entorno

La simulación y el entorno real tienen las mismas condiciones para el aprendizaje del pateado del balón. La pierna robótica se encuentra en una mesa de 82 cm de altura, dejando un espacio de 20 cm desde el suelo hasta el extremo inferior de la tibia, como se muestra en la figura 3.

Cada articulación se controla mediante un sistema hidráulico como se muestra en el video [2] (accesible desde este url[2]). Los grados de movimiento coinciden con los descritos en la sección 2.3.

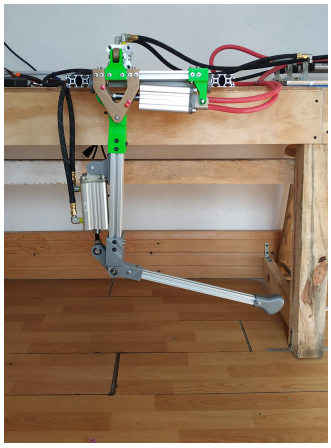


Fig. 3. Pierna robótica en entorno real.

4.2. Reconocimiento del balón en tiempo real

Se entrena un clasificador en cascada [30] para el reconocimiento del objeto mediante el algoritmo Viola-Jones utilizando OpenCV. En el entrenamiento se usan 644 imágenes positivas donde se incluye el objeto que se quiere detectar (en este caso, el balón), también se utilizan 3,019 imágenes negativas, es decir que no contienen el objeto deseado.

En cuanto a las imágenes positivas se indica la posición donde se localiza el objeto, finalmente con las herramientas proporcionadas de OpenCV se realiza el entrenamiento en cascada con un total de 20 episodios [15].

Los resultados del entrenamiento para la detección del balón y la obtención de sus coordenadas, se pueden observar en el video [4] (accesible desde este url[3]).

[2] `https://www.youtube.com/watch?v=qmeyC5y9fQE`
[3] `https://www.youtube.com/watch?v=_Cl9SZCBb0E`

5. Misión a ejecutar

El objetivo consiste en que el agente logre aprender a patear un balón dentro de un rango definido, a través de dos movimientos en sus articulaciones. En un posible caso, el agente podría hacer un movimiento para acomodarse y un segundo para generar la patada con la mayor fuerza posible.

El balón cambia su posición inicial en cada episodio; un episodio consiste en reiniciar la posición del balón y finaliza cuando la pierna ejecuta los movimientos en los motores hidráulicos. En las primeras pruebas la posición del balón se mantiene, con la intención de enfocarse y validar que es posible patear el balón a pesar del gran campo de acciones disponibles.

6. Mecanismos para el aprendizaje y control de motores

6.1. Estados

Los estados son representados por las posiciones del balón. Al cambiar sus coordenadas, las acciones que deben realizar los pistones cambian. El rango de la posición del balón (figura 4) en el eje x es de 0.1 a -0.1 m y en el eje y es de 0.1 a -0.1 m. Las posiciones cambian cada 0.05 m, por lo que se tiene un total de 25 estados, que representan las posibles posiciones del balón dentro del entorno. (El eje z siempre mantiene su mismo valor).

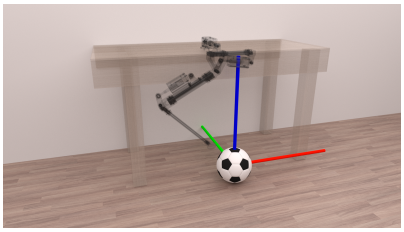


Fig. 4. Representación tridimensional dónde la línea roja representa el eje x, la verde el eje y y la azul el eje z.

6.2. Acciones

Por acciones se refiere a los movimientos que pueden ser ejecutados por los actuadores. Si las acciones se representan por cada grado que puede mover cada articulación de la pierna, existirían 126,540 posiciones diferentes, para reducir este número la articulación se mueve cada 5 grados lo que da un total de 924 posibles acciones para representar todas las posiciones. Tomando en cuenta que una acción conlleva dos movimientos, se estaría hablando de 853,776 acciones,

sin embargo, se utilizaron dos matrices de pesos donde se busca la mejor acción en determinado tiempo para el pateado del balón y el acomodo de la pierna, dando así un total de 1,848 acciones.

6.3. Q-learning

Q-learning es el algoritmo ocupado para el entrenamiento del pateado del balón, su funcionamiento se puede describir de la siguiente forma:

$$Q^{new}(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \lambda maxQ(s_{t+1}, a)]. \tag{1}$$

La función Q se va actualizando conforme pasa el tiempo, esta ayuda al agente a realizar la mejor acción con la experiencia adquirida durante el entrenamiento.

Dependiendo del estado (s) se ejecuta cierta acción (a), en un tiempo determinado (t). Conforme avanza el tiempo, existe una actualización entre la recompensa futura y la antigua al valor Q existente, esto se puede expresar como la siguiente equivalencia:

$$A^{new} = A + B. \tag{2}$$

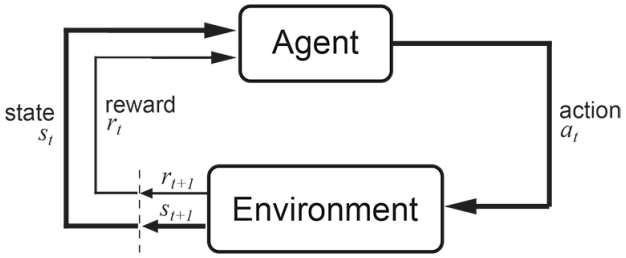


Fig. 5. Representación del entorno y el agente del algoritmo Q-learning [16].

Como se observa en el algoritmo Q-learning (figura 5), el agente recibe recompensas para indicar su desempeño en forma de retroalimentación. En este caso, las recompensas son determinadas por la magnitud que se tiene entre la posición final del balón (x_f^{ball},y_f^{ball}) con respecto a la posición en la que se inició el episodio (x_i^{ball},y_i^{ball}):

$$reward_t = \sqrt{(x_f^{ball} - x_i^{ball})^2 + (y_f^{ball} - y_i^{ball})^2}. \tag{3}$$

En cuanto a la selección de los mejores movimientos para el pateado del balón, se realiza con base a la experiencia adquirida de la primera y segunda acción, estas son determinas por medio de las ecuaciones 4 y 5:

$$W_1^{new}(s_t, a_t) \leftarrow W_1(s_t, a_t) + \alpha r_t^{reward} + \lambda max(W_2(s_{t+1}, a_t)), \tag{4}$$

$$W_2^{new}(s_t, a_t) \leftarrow W_2(s_t, a_t) + \alpha r_t^{reward}, \tag{5}$$

donde α representa el ritmo del aprendizaje de cada nueva experiencia, es decir, qué tanto el agente puede aprender de la acción realizada, y λ indica el factor de descuento que valora las recompensas de las acciones pasadas, donde 0 significa que sólo importa la recompensa por la acción realizada y no toma en cuenta la experiencia adquirida del agente. [20]

En las gráficas de la sección 7.3 se toma el promedio de los pesos de la primera y segunda acción ejecutada, se muestra que al pasar los episodios se van seleccionando solo las mejores acciones con base a sus pesos:

$$\frac{W_1(s_t, a_t) + W_2(s_t, a_t)}{2}. \tag{6}$$

7. Resultados

7.1. Estados base

En las primeras pruebas el balón siempre mantiene la misma posición. El objetivo del agente es buscar y seleccionar la mejor acción que pueda realizar cada uno de los motores hidráulicos, según el sistema de recompensas.

Actualmente se está entrenado para que sin importar la posición del balón (dentro de los 25 estados posibles como se describe en la sección 6.1) siempre pueda hacer contacto con este de la mejor manera posible y de esta forma, sin importar el estado, se ejecute una acción, que a través de los movimiento de los motores hidráulicos, realice una patada que pueda alejar al balón a una mayor distancia con respecto a las coordenadas iniciales del episodio.

En las secciones 7.2 y 7.4 se muestran los resultados obtenidos con el entrenamiento tanto de forma simulada como real.

7.2. Entrenamiento del agente en el entorno de simulación

En la figura 6, se observa la ponderación de las acciones realizadas (weights), conforme pasan los episodios se ejecutan las que tienen mayor peso (sección 6.3).

En la gráfica se puede observar que al principio del entrenamiento los resultados llegan a ser extremos entre puntos, esto se debe a que el agente se encuentra en fase de búsqueda, probando diferentes acciones hasta encontrar la mejor, en un episodio se puede tener una recompensa alta y en otro no hacer contacto con el balón.

El agente evalúa por medio de su matriz de pesos, el desempeño de la acción ejecutada y, de esta forma, considera cual acción futura sería la más conveniente. En dicha gráfica (figura 6), se muestra que conforme transcurren los episodios, se seleccionan los movimientos que tienen mayor peso con base a la experiencia adquirida. En este caso, a partir del episodio 250, se ejecutan las mejores acciones hasta el punto de llegar a ser constante con los movimientos que se realiza.

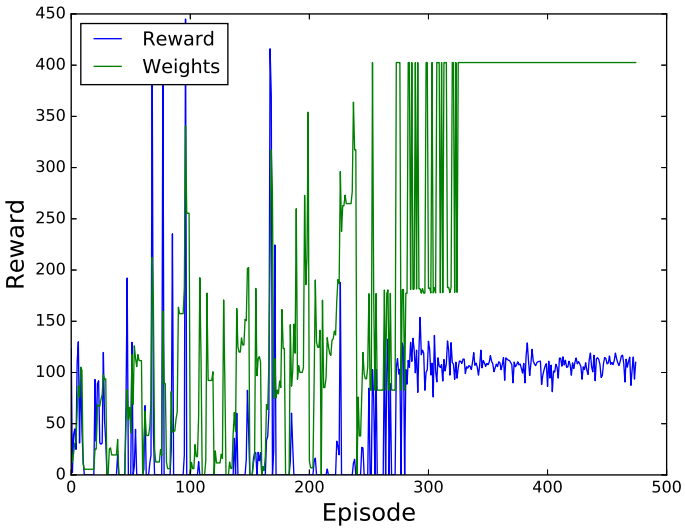


Fig. 6. El trazado con línea verde representa los pesos de las acciones y el azul indica la recompensa final que obtuvo el agente en cada episodio.

La recompensa (rewards) indica el desempeño del pateado del balón (el desplazamiento recorrido con respecto a su punto inicial); esta representación de datos es la más relevante para medir los resultados del agente, ya que muestra el aprendizaje e indica como mejora el pateado al transcurrir los episodios.

En la figura 7, se observa la recompensa por cada acción que realiza el agente. En los primeros 200 episodios se visualizan picos repentinos de recompensas mayores a 400 puntos, seguidos de caídas abruptas; esto se debe a que ciertas acciones llevan al agente a recibir puntuaciones altas, sin embargo, si se realiza la misma acción y se obtiene una recompensa baja, se genera poca probabilidad para volver a seleccionarla, por lo que el agente opta por los movimientos que siempre producen un contacto con el balón de forma constante. De esta forma se mantiene una recompensa constante de aproximadamente 100 puntos, lo que demuestra que siempre se está pateando el balón, variando el desplazamiento de este en cada episodio.

Se puede observar en el video [1] (accesible desde este url[4]) el entrenamiento realizado en el motor de simulaciones de Gazebo, se muestra como la pierna robótica ejecuta el pateado del balón al transcurrir los episodios, con el fin de ir mejorando su desempeño. Gazebo no cuenta con la propiedad de simular los coeficientes de fricción de rodadura para el balón por lo que cambia el desempeño dentro del entorno virtual con respecto al real.

[4] `https://www.youtube.com/watch?v=NlCBaihAQ9k`

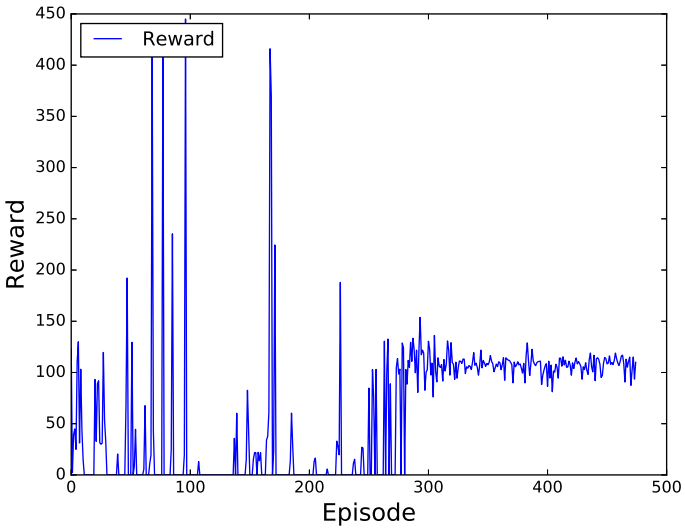


Fig. 7. Recompensas obtenidas por las acciones del agente.

7.3. Análisis del aprendizaje

Se ejecutaron 6 entrenamientos con 450 episodios cada uno. En estos ocurren variaciones, como se muestra en la tabla 1, dependiendo de las diferentes acciones ejecutadas por el agente.

Cada entrenamiento tiene una duración aproximada de una hora y 28 minutos, las características del equipo de computo utilizado son:

- Modelo del procesador: I5-7200u,
- Familia del procesador: 7ª generación de procesadores Intel Core,
- Número de núcleos del procesador: 2,
- Memoria: 8 GB DDR4-SDRAM,
- Almacenamiento: 100 GB HDD,
- Gráficos: Intel HD Graphics 620.

En las gráficas de aprendizaje (figura 9 a 14), se observa un comportamiento similar, donde el agente al finalizar su entrenamiento, siempre selecciona las mejores acciones con base a su experiencia de los episodios pasados y de esta forma se mejora el pateado del balón. La recompensa en todos los casos se mantiene sobre un rango, lo que indica que se hace contacto con el balón. Generalmente a partir del episodio 250 tiene una mejora notable, dónde el agente realiza movimientos en los actuadores que logran el pateado del balón de una forma casi constante.

El promedio de la puntuación fue de 97.33 unidades y, a partir del episodio 303.33, generalmente, se empezaba a tener un mejoramiento de forma continua.

Tabla 1. Datos de aprendizaje.

Episodio	Promedio desempeño del agente	Episodio con mejora continua
Entrenamiento 1	82	320
Entrenamiento 2	125	280
Entrenamiento 3	85	270
Entrenamiento 4	95	305
Entrenamiento 5	105	370
Entrenamiento 6	92	275

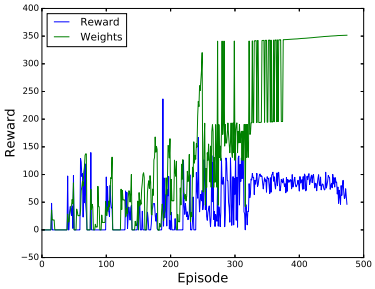


Fig. 8. Entrenamiento 1.

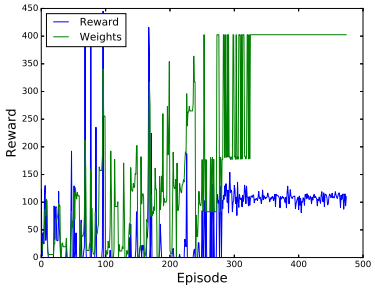


Fig. 9. Entrenamiento 2.

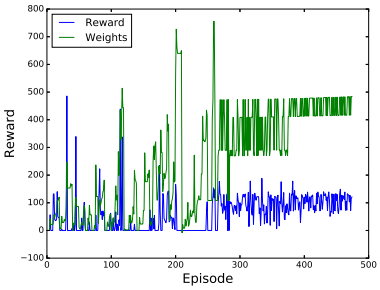


Fig. 10. Entrenamiento 3.

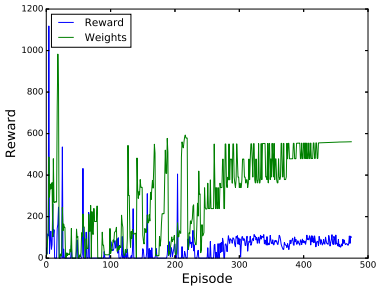


Fig. 11. Entrenamiento 4.

7.4. Resultados pateado del balón

Entorno simulado. Los resultados de la sección 7.3, para el entorno simulado se observan en el video [7] (accesible desde ete url[5]). En dicho video se muestran las ejecuciones de cada uno de los entrenamientos para el pateado del balón.

Entorno real. La matriz de pesos resultante de los entrenamientos, se trasladó a la pierna robótica real para evaluar si esta puede ejecutar el pateado del

[5] `https://www.youtube.com/watch?v=KItPI11y9cQ`

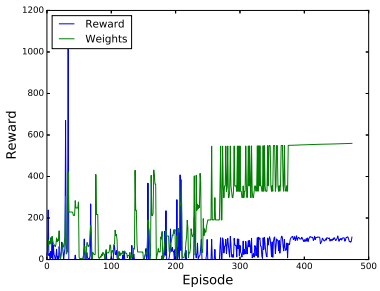


Fig. 12. Entrenamiento 5.

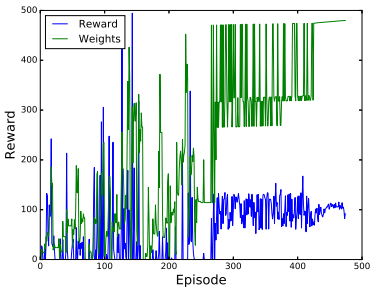


Fig. 13. Entrenamiento 6.

balón de manera efectiva, a partir del entrenamiento simulado y a través de la acción ejecutada por los motores hidráulicos. Estos resultados se pueden observar en video, [5] (accesible desde este url[6]), en el cual, se muestra los resultados del pateado del balón en el entorno real, lo que confirma que a través de la simulación, se puede entrenar de manera correcta el pateado del balón para poder ser aplicado en la realidad.

8. Conclusión

Los resultados de la sección 7.3 indican que en todos los casos, al finalizar los entrenamientos, la recompensa se muestra constante. Esto es debido a que el agente opta por movimientos que siempre generan contacto con el balón, en lugar de arriesgarse a tomar acciones que puedan llegar a tener recompensas más altas a costa de estabilidad. Un punto crítico en nuestra investigación es el uso de la matriz de pesos en el entorno real. Los resultados demuestran una viabilidad en el uso de simulaciones para obtener los pesos y, de este modo, utilizarlos en la pierna robótica real sin necesidad de un entrenamiento o adaptación; sin embargo, esto no es garantía de que sean los pesos óptimos a utilizar. Mejorar la simulación permitiría al agente tener recompensas cercanas a las esperadas en la realidad, rompiendo de este modo la brecha que pueda existir entre los pesos del entorno real y el simulado. Para mejorar la simulación se tendrían que considerar diversas cuestiones físicas como resistencia a la rodadura, y de esta forma, obtener comportamientos fidedignos. Además de esto, el aumento de calidad y realismo gráfico de la simulación, permitiría al algoritmo de detección de coordenadas un traslado rápido de aprendizaje del mundo simulado al mundo real.

Los resultados demuestran una viabilidad en el uso de estos algoritmos para movimiento de piernas robóticas. Futuros trabajos incluirían el uso de los algoritmos mencionados en este trabajo para el caminado de un cuadrúpedo y manipulación de objetos con robots industriales de 7DOF. Actualmente se

[6] `https://www.youtube.com/watch?v=tU4fGfTBfgI`

está trabajando para que la pierna robótica realice el pateado sin importar las posiciones del balón, los avances se pueden observar en el video [6] (accesible desde este url[7]), así como también se están generando cambios en el aprendizaje del agente para obtener un mejor desempeño de este.

Referencias

1. Entrenamiento en entorno simulado. `https://www.youtube.com/watch?v=NlCBaihAQ9k` (2019), último acceso: 2019/05/11
2. Pruebas de movimiento. `https://www.youtube.com/watch?v=qmeyC5y9fQE` (2019), último acceso: 2019/05/11
3. Pruebas de simulación. `https://www.youtube.com/watch?v=F34zIs0cI-0` (2019), último acceso: 2019/05/11
4. Reconociminto de balón con opencv. `https://www.youtube.com/watch?v=_Cl9SZCBb0E` (2019), último acceso: 2019/05/11
5. Resultados entorno real. `https://www.youtube.com/watch?v=tU4fGfTBfgI` (2019), último acceso: 2019/05/11
6. Resultados entorno real en diferentes posiciones. `https://www.youtube.com/watch?v=BUe4uNiYSR0` (2019), último acceso: 2019/05/11
7. Resultados entorno simulado. `https://www.youtube.com/watch?v=KItPI11y9cQ` (2019), último acceso: 2019/05/11
8. Castrillón, M., Déniz, O., Hernández, D., Lorenzo, J.: A comparison of face and facial feature detectors based on the viola–jones general object detection framework. Machine Vision and Applications 22(3), 481–494 (2011)
9. Chitta, S., Marder-Eppstein, E., Meeussen, W., Pradeep, V., Tsouroukdissian, A.R., Bohren, J., Coleman, D., Magyar, B., Raiola, G., Lüdtke, M., et al.: ros_control: A generic and simple control framework for ros. The Journal of Open Source Software 2(20), 456–456 (2017)
10. Koenig, N., Howard, A.: Design and use paradigms for gazebo, an open-source multi-robot simulator. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566). vol. 3, pp. 2149–2154. IEEE (2004)
11. Kormushev, P., Calinon, S., Caldwell, D.: Reinforcement learning in robotics: Applications and real-world challenges. Robotics 2(3), 122–148 (2013)
12. Lachat Leal, C., et al.: Didáctica de la traducción audiovisual: enseñar a mirar (2011)
13. Müller, J., Laue, T., Röfer, T.: Kicking a ball – modeling complex dynamic motions for humanoid robots. In: Ruiz-del Solar, J., Chown, E., Plöger, P.G. (eds.) RoboCup 2010: Robot Soccer World Cup XIV. pp. 109–120. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)
14. Parker: Compact hydraulic cylinders. `http://www.parker.com/literature/Industrial%20Cylinder/cylinder/cat/english/HY08-1137-7NA_che-chd.pdf` (2008), último acceso: 2019/05/11
15. Puttemans, S.: Cascade classifier training. `https://docs.opencv.org/3.3.0/dc/d88/tutorial_traincascade.html` (2017), último acceso: 2019/05/11
16. Qicui Yan, Quan Liu, Daojing Hu: A hierarchical reinforcement learning algorithm based on heuristic reward function. Reproducido como en la Figura 1 en [17] (2010), último acceso: 2019/05/11

[7] `https://www.youtube.com/watch?v=BUe4uNiYSR0`

17. Qicui Yan, Quan Liu, Daojing Hu: A hierarchical reinforcement learning algorithm based on heuristic reward function. In: 2010 2nd International Conference on Advanced Computer Control. vol. 3, pp. 371–376 (March 2010)
18. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA workshop on open source software. vol. 3, p. 5. Kobe, Japan (2009)
19. Riedmiller, M., Merke, A., Meier, D., Hoffmann, A., Sinner, A., Thate, O., Ehrmann, R.: Karlsruhe brainstormers - a reinforcement learning approach to robotic soccer. In: Stone, P., Balch, T., Kraetzschmar, G. (eds.) RoboCup 2000: Robot Soccer World Cup IV. pp. 367–372. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
20. Russell, S.J., Norvig, P.: Inteligencia Artificial: un enfoque moderno. No. 04; Q335, R8y 2004. (2004)
21. Schempf, H., Kraeuter, C., Blackwell, M.: Roboleg: a robotic soccer-ball kicking leg. pp. 1314 – 1318 vol.2 (06 1995)
22. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. nature 529(7587), 484 (2016)
23. Sung, C., Kagawa, T., Uno, Y.: Planning of kicking motion with via-point representation for humanoid robots (11 2011)
24. Theobald, M., Sozio, M., Suchanek, F., Nakashole, N.: Urdf: Efficient reasoning in uncertain rdf knowledge bases with soft and hard rules (2010)
25. Turk, M.: Computer vision in the interface. Communications of the ACM 47(1), 60–67 (2004)
26. Vahidi, M., Moosavian, S.: Dynamics of a 9-dof robotic leg for a football simulator. pp. 314–319 (10 2015)
27. Vinyals, O., Ewalds, T., Bartunov, S., Georgiev, P., Vezhnevets, A.S., Yeo, M., Makhzani, A., Küttler, H., Agapiou, J., Schrittwieser, J., et al.: Starcraft ii: A new challenge for reinforcement learning. arXiv preprint arXiv:1708.04782 (2017)
28. Viola, P., Jones, M.J.: Robust real-time face detection. International journal of computer vision 57(2), 137–154 (2004)
29. Watkins, C.J., Dayan, P.: Q-learning. Machine learning 8(3-4), 279–292 (1992)
30. Wilson, P.I., Fernandez, J.: Facial feature detection using haar classifiers. Journal of Computing Sciences in Colleges 21(4), 127–133 (2006)