“LEARNING TO WALK WITH GENETIC ALGORITHMS”

Autores:
- Jesse Alonso Galán Tirapo
- Guillermo Daniel Montaño Gamarra
- Jason Luis Valera Espinoza

Códigos:
- 20135167
- 20151692
- 20135426

Resumen- El movimiento humano resulta de la compleja y variada coordinación de músculos, tendones, articulaciones y otros elementos fisiológicos. A lo largo de la etapa del desarrollo del ser humano, los niños aprenden a caminar, correr, escalar y saltar en sus primeros años de vida. Por otro lado, la mayoría de nosotros puede navegar en entornos complejos, como una calle concurrida o en un bus en movimiento. En la búsqueda de definir un modelo humano fisiológico para caminar y correr, se planteará una mejora del controlador entrenado con un óptimo performance y menor costo computacional que las propuestas presentadas en NIPS 2017: Learning to Run. Ello se logrará mediante técnicas de computación bioinspirada, específicamente, algoritmos genéticos. Todo ello, por medio de la definición de una articulación base, una función fitness basada en caminar sin caerse o la evolución de los parámetros del controlador.

1. Introducción

Actualmente, el desarrollo de controladores que puedan sintetizar de manera eficiente y robusta a humanos, específicamente el movimiento de caminar o correr en una variedad de entornos siguen siendo un gran desafío para los biomecánicos, los neurocientíficos y los informáticos. Los controladores actuales están confinados a un pequeño conjunto de movimientos preespecificados o impulsados por pares, en lugar de los complejos actuadores musculares que se encuentran en los humanos.

Es así que existen organizaciones como el Centro Nacional de Simulación para la Investigación en Rehabilitación (NCSRR, por sus siglas en inglés) del NIH para equipar a la comunidad de investigadores en rehabilitación con herramientas de simulación de vanguardia, que permiten a los investigadores complementar estudios experimentales de desempeño humano con software avanzado de simulación y modelos biomecánicos.que. El paquete que utilizan les permite sintetizar movimientos fisiológicamente precisos al combinar la experiencia biomecánica incorporada en el software de simulación OpenSim con estrategias de control de vanguardia mediante el Aprendizaje de refuerzo profundo.

El interés sobre el desarrollo de esas aplicaciones a permitido que se dispone de los recursos en biomecánica y soluciones de aprendizaje por refuerzo de las conferencias en NIPS 2017 y la competencia “Learning to Run NIPS 2017”. Por lo tanto, basados en los objetivos descritos anteriormente, el comportamiento esperado es que, frente a la entrada de parámetros que devuelve el ambiente, el controlador responda con un set de parámetros de salida de forma que permita que el modelo 3D tenga un movimiento coordinado que imite la caminata de un humano y no se caiga al suelo.

2. Metodología

o *Baselines*

La baseline utilizada define a la articulación del modelo. Esto implica que se defina una articulación base, eso lleva a que esta articulación comienza a cambiar, antes que las otras articulaciones. Por lo tanto, la junta de la base se considera como la base para el movimiento de todas las demás juntas. Es decir, todas las demás articulaciones comienzan el movimiento en un período de tiempo t después del movimiento de las articulaciones de la base (t> = 0) [1].

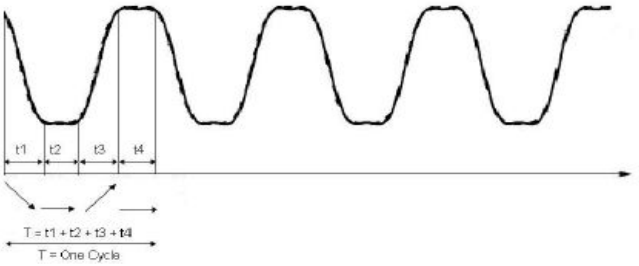

Figura 1. Diagrama de movimiento y diseño del gen [1].

o Enfoque(s) propuesto:

El comportamiento del modelo se basa en el modelo de dinámica de activación muscular de equilibrio (comúnmente utilizados para modelar músculos en simulaciones músculo-esqueléticas de parámetros concentrados). Tienen una singularidad en sus ecuaciones de estado cuando la activación es cero. De esa manera se restringe primero la activación y la excitación para que permanezcan entre el nivel de activación mínimo (0.01 milisegundos) [2].

El modelo tiene como principal función una entrada que corresponde a la acción de excitación muscular de las piernas (una lista de longitud 18 con valores continuos en [0,1]). La función de salida devuelve un diccionario que describe el estado de los músculos, articulaciones y cuerpos en el sistema biomecánico.

Como se detalló al inicio, el enfoque de IA utilizado para el presente trabajo corresponde a las técnicas de computación bioinspirada, específicamente, algoritmos genéticos. Este tipo de algoritmos son una alternativa computacionalmente menos intensiva que los algoritmos típicamente utilizados de aprendizaje por reforzamiento. Esta característica es muy conveniente cuando el entorno de aprendizaje consume muchos recursos y no es posible proveer al agente de muchas iteraciones de ejecución.

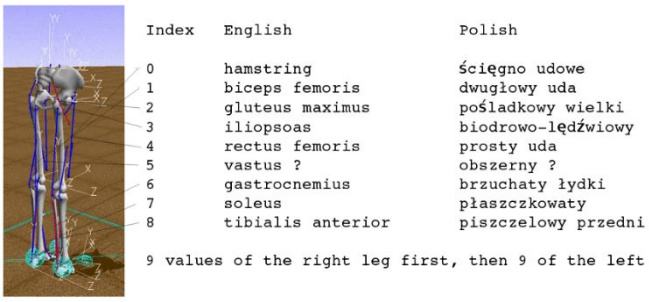

Figura 2. Infografía del detalle de las 18 activaciones musculares [3]

La técnica de entrenamiento basada en algoritmos genéticos consiste en la selección de los individuos con mejor performance de una población mayor en número. De esta manera, se imita a la proceso biológico de selección natural. Así, para este modelo musculoesquelético, a cada individuo se le asigna un arreglo de valores que caracteriza su comportamiento en el ambiente de ejecución. Este arreglo es llamado cromosoma y representa cada posible solución del espacio de búsqueda del ambiente. Para este caso, se utilizó la representación en punto flotante para los cromosomas.

Tal y como se propone en [3], el movimiento de las piernas durante una caminata es de tipo periódico, por lo que las leyes de activación de cada músculo pueden modelarse como una función periódica. Ya que la manipulación manual de tales funciones puede ser computacionalmente costosa, se decide aproximar con una serie de Fourier truncada, que puede aproximar virtualmente cualquier función periódica, usando una suma ponderada de seno y coseno.

Siguiendo la metodología de [3], cada función de activación se trunca a 4 armónicas y se definen 8 parámetros a modificar, correspondientes a la amplitud y fase de cada armónica Teniendo cada pierna 9 músculos, se usó 9 de estas diferentes funciones periódicas. Para la otra pierna, se usan las mismas funciones, pero se cambian la fase en 180º [2]. De esta forma, el cromosoma de cada individuo queda definido como una arreglo de 72 valores en punto flotante. Cada valor es denominado un alelo del cromosoma para un individuo.

El algoritmo genético utilizado sigue la estructura clásica de explotación y exploración de los individuos. La función de explotación consigue la mejor combinación de alelos y guía la búsqueda en el espacio de soluciones en la dirección indicada por los mejores individuos de la población. Complementariamente, la función de exploración introduce cierto factor de aleatoriedad en ciertos individuos, produciéndose soluciones subóptimas fuera de la dirección general de la función de explotación, pero permite que la búsqueda escape de máximos locales (o mínimos, según sea el caso). Los pasos del AG implementado son los siguientes:

1. Inicialización: Se genera cromosomas aleatorios para cada individuo en la población.
2. Cruzamiento: Sea esta la función de explotación, se agruparon todos los individuos de la población en pares y se obtuvieron 2 descendientes de cada par, determinados como:

$$child1 = father1 * 0.3 + father2 * 0.7$$
$$child2 = father1 * 0.7 + father2 * 0.3$$

   donde *childX y fatherX* son los cromosomas de los descendientes y parentales, respectivamente.

3. Mutación: Sea esta la función de exploración, para cada cromosoma se eligió un valor de cada función de activación aleatoriamente y se alteró sumando un valor también aleatorio.
4. Evaluación: Se ejecuta el controlador con los parámetros del cromosoma y se obtiene el valor de recompensa.
5. Selección: La porción de la población que formará parte de la siguiente generación de individuos es seleccionada de forma que se mantenga el mismo número de individuos de la población original, Para esto, se ordenan los individuos descendentemente de acuerdo a su valor de recompensa asignado y se elige la cantidad indicada. Por último, se retorna al punto 2.

El criterio de parada del entrenamiento consiste en observar si la recompensa del mejor individuo de cada generación llega a converger a algún valor.

- Métricas de evaluación

Para evaluar el desempeño del algoritmo, se estableció de acuerdo al baseline, una recompensa (reward) obtenida en la última iteración. La recompensa se calcula como un cambio en la posición de la pelvis a lo largo del eje X menos la penalización por el uso de ligamentos. Un flag de detención, indica si el movimiento fue el último paso del entorno. Esto sucede si se alcanzaron 1000 iteraciones o la altura de la pelvis es inferior a 0,65 metros.

- Infraestructura:

El modelo está implementado en OpenSim [4], que se basa en el motor de física Simbody. Dicho modelo permite realizar lo siguiente:

- Calcular las activaciones de los músculos a partir del vector de excitaciones proporcionado a la función step ().
- Activar los músculos según estas activaciones.
- Calcular los pares generados por las activaciones musculares.
- Calcular las fuerzas causadas por el contacto con el suelo
- Calcular velocidades y posiciones de articulaciones y cuerpos
- Generar un nuevo estado basado en fuerzas, velocidades y posiciones de las articulaciones.

3. Resultados

Como primer resultado tenemos la ejecución del baseline seleccionado, correspondiente al trabajo de Di Pablo [3]. Se ejecutó el algoritmo de entrenamiento por un total de 500 generaciones de una población de 100 individuos. El mejor cromosoma obtenido permite realizar el movimiento próximo a un paso de alrededor de 2.2 segundos antes de caer. Cabe resaltar que el entorno analiza si el agente está en condiciones de caminar con una bandera de salida, de esa manera, cuando el entorno devuelve esa salida, el proceso se detiene [3].

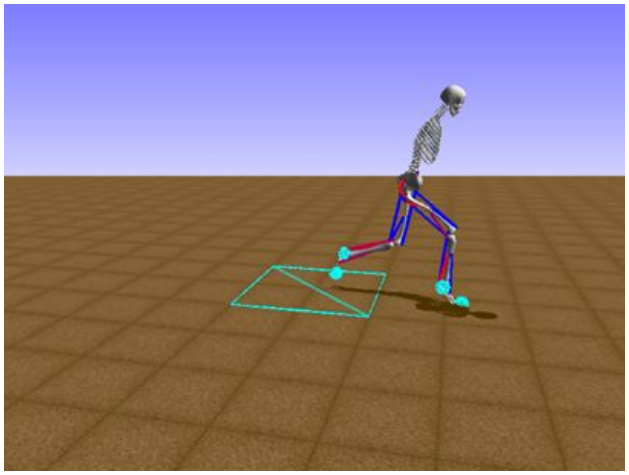

Figura 3. Agente realizando movimiento de un paso. Elaboración propia

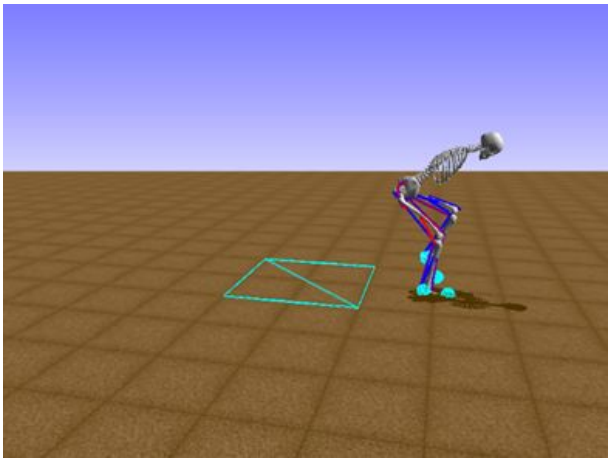

Figura 4. Agente en reposo luego que el entorno indicará que no está en condición de caminar. Elaboración propia

Los parámetros escogidos son un mínimo de evaluación en el entorno de 5 segundos, un step de 0.01 segundo en cada acción en el vector que controla los músculos de las piernas.

Los resultados alcanzados en la baseline demostraron que se puede lograr que este agente en este entorno pueda caminar utilizando algoritmos genéticos. Se inició a partir de unos valores de excitación que corresponde al vector que controla los músculos de las piernas y a partir de allí se empezó a escoger una mejor opción. Sin embargo, pudimos observar que la acción de caminar estaba incompleta y de poca duración a pesar de indicar que se realice en 5 segundos.

Esta implementación utiliza un único individuo para toda la población, por lo que no es posible definir una función de cruzamiento. Asimismo, la función de mutación actúa sobre todos los valores del cromosoma y reemplaza al individuo original si es que el nuevo cromosoma obtiene una mejor recompensa. Esto trae como consecuencia que la búsqueda de soluciones llegue a converger rápidamente en algún máximo mínimo.

A partir del enfoque anterior, se ha desarrollado un algoritmo genético con parámetros que influyen en el cruzamiento y mutación para obtener los valores que utiliza el vector que controla los músculos de las piernas para mejorar la recompensa, que para el algoritmo genético corresponde al fitness y un mayor tiempo de duración de la acción de caminar.

El algoritmo consiste en aplicar una probabilidad de mutación del 50% y una función de cruzamiento completo. Finalmente, una población de 200 individuos y 500 generaciones.

Debido a la complejidad y precisión del entorno OpenSim, el cual es altamente costoso en recursos computacionales, el sistema de entrenamiento propuesto se ha ejecutado por un total de 10 intentos, cada uno por un total de 500 generaciones. Asimismo, se diseñó el programa de forma que aproveche los núcleos disponibles de la CPU y ejecuta el entrenamiento en paralelo, de forma que se obtenga la función reward de un individuo en cada núcleo.

En base a ello, cada entrenamiento tuvo una duración promedio de aproximadamente 24 horas para 500 generaciones. El hardware utilizado corresponde a una laptop Dell Inspiron 7000 con Procesador Intel Core i7-8550U de 4 núcleos a 1.80 GHz con una memoria RAM de 12 GB y un GPU integrado Intel UHD Graphics 620. Se ejecutó el entrenamiento en el IDE Spyder 3.3.4 con lenguaje Python.

El movimiento del agente correspondiente al mejor individuo de las poblaciones obtenidas de todos los intentos se muestra en la figura 6, con un total de 4 pasos, antes de que el entorno señalase el fin de la simulación. En total, el entorno ejecutó 442 pasos, equivalente a 4,42 segundos y recorrió un total de 4.58 metros.

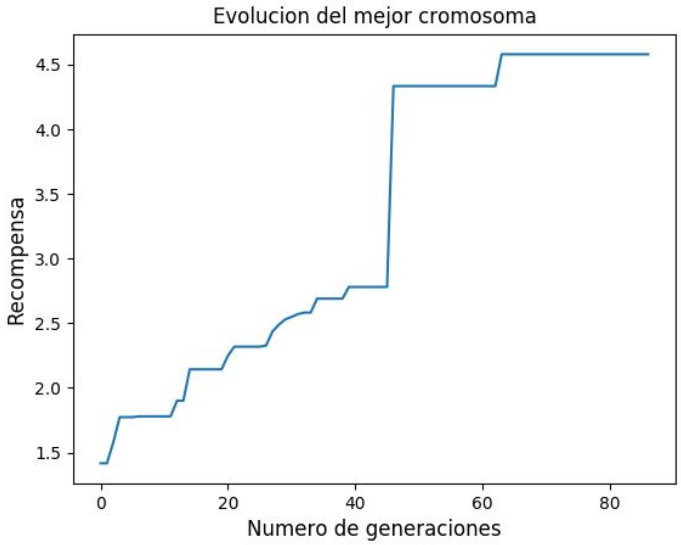

Figura 5. Gráfica de la evolución del intento correspondiente al mejor resultado obtenido. Elaboración propia

La implementación de este código se encuentra publicada en un repositorio público [9].

4. Conclusiones

En primer lugar, se logró implementar un algoritmo genético que responda a la solicitud de la competencia "Learning to Run NIPS 2017". Para el desarrollo de esta propuesta se eligió algoritmo genético debido a su relación conceptual propia del algoritmo genético (computación bioinspirada) con la aplicación a una funcionalidad de carácter humano (biológico) como es el caminar y correr.

Adicionalmente, se consiguió obtener un resultado satisfactorio considerando los recursos invertidos en el entrenamiento y la cantidad de intentos que se ejecutaron.

Para poder continuar con la mejora del control del individuo producto del entrenamiento, se necesita incrementar el tiempo de ejecución de la implementación. Para eso, se recomienda paralelizar el entrenamiento con un mayor número de núcleos o contar con un *farm* de procesadores, ya sea en equipo *on premise* o alojados en la nube.

Finalmente, consideramos que estas aplicaciones permiten que estudiantes de diversas áreas apliquen los conocimientos de nuevas tecnologías de desarrollo que están tomando un gran impacto a nivel de la industria y tecnología moderna.

5. Referencias

[1]. M. Azhdari, "Bipedal Walking Control Using Genetic Algorithm," 2007.

[2]. Thelen, D.G. "Adjustment of muscle mechanics model parameters to simulate dynamic contractions in older adults." ASME Journal of Biomechanical Engineering 125 (2003): 70–77.

[3]. N. Di Pablo, 'Learning to walk with evolutionary algorithms applied to a bio-mechanical model', 2017. [Online]. Disponible: https://towardsdatascience.com/learning-to-walk-with-evolutionary-algorithms-applied-to-a-bio-mechanical-model-1ccc094537ce [Acceso: 23-abril-2019].

[4]. Delp, Scott L., et al. "OpenSim: open-source software to create and analyze dynamic simulations of movement." IEEE transactions on biomedical engineering 54.11 (2007): 1940-1950.

[5]. J. H. Holland. Adaptation in Natural and Artificial Systems. 2nd. Edition. Cambridge MA: Mit Press, 1992.

[6]. Shinya, Aoi. "Neuromusculoskeletal model that walks and runs across a speed range with a few motor control parameter changes based on the muscle synergy hypothesis." NATURE: Scientific Reports 9 (2019).

[7]. L. Kidzi´nski, "Learning to Run challenge solutions: Adapting reinforcement learning methods for neuromusculoskeletal environments." 2018.

[8]. A. Boeing "Simulation of Biped Walking using Genetic Algorithms." 2002.

[9]. Galán, J, Montaño, G, Valera, J. https://github.com/MontanoG/IA-genetic-alg

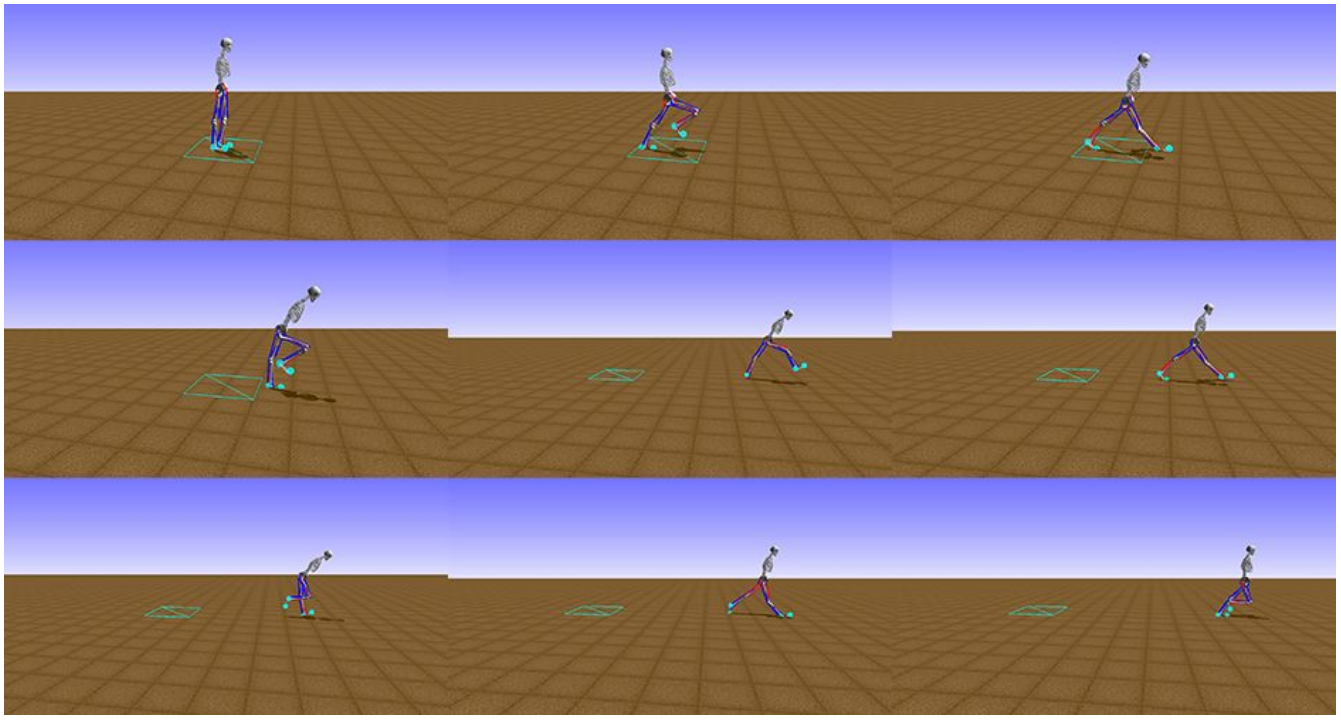

Figura 6. Movimiento del mejor resultado del entrenamiento. Elaboración propia