

PROYECTO FINAL

INGENIERÍA INFORMÁTICA - ITBA

***SIMULACIÓN Y ANIMACIÓN
BIOMECÁNICA
DE UN HUMANOIDE***

Alumnos:

Enzo Altamiranda Graterol

ealtamir@itba.edu.ar

Teresa Fontanella De Santis

tfontane@itba.edu.ar

Tomás Mehdi

tmehdi@itba.edu.ar

Tutor:

Dr. Daniel Ricardo Parisi

**Instituto Tecnológico de Buenos Aires - ITBA
Departamento de Ingeniería Informática**

Mayo 2016

Índice

Resumen	4
1 Introducción	4
2 Herramientas	5
2.1 Motor físico	5
2.1.1 Funcionamiento	5
2.1.2 Modelo de fricción y su verificación	5
2.1.2.1 Verificación del coeficiente de fricción	6
2.1.2.2 Verificación del coeficiente de restitución	10
2.1.3 Ventajas	11
2.1.4 Desventajas	11
2.2 Librería de algoritmos genéticos	12
2.3 Código fuente	12
3 Modelo utilizado	12
3.1 Composición física del humanoide	12
3.2 Articulaciones	13
4 Actuadores	14
4.1 Genérico	14
4.2 Coseno doble frecuencia	15
4.3 Fourier de orden 2	16
4.4 Fourier de orden 9	16
5 Condiciones iniciales y de contorno	17
5.1 Función partida	17
5.2 Fase sincronizada	17
6 Algoritmo genético	17
6.1 Individuo	18
6.1.1 Parámetros	18
6.1.2 Valores	18
6.1.3 Implementación de individuos	19
6.1.4 Constitución del cromosoma	19
6.2 Fitness	20
6.2.1 Altura	20
6.2.2 Velocidad	21
6.2.3 Dirección	21
6.2.4 Simetría	21
6.2.5 Pies abajo	22
6.3 Operadores del algoritmo	22
6.3.1 Métodos de selección	22
6.3.2 Métodos de cruce	22
6.3.3 Mutación	22

6.3.4	Otras características	23
7	Resultados obtenidos	23
7.1	Evolución del <i>fitness</i> promedio según tipo de individuos	24
7.2	Velocidad según tipo de individuos	24
7.3	Altura según tipo de individuos	25
7.4	Comparación de tipo de individuos	25
7.5	Video	26
8	Conclusiones	26
	Referencias	27
	Agradecimientos	27

Resumen

Este proyecto tiene como objetivo crear una simulación y animación de un humano virtual, con las siguientes propiedades:

- Biomecánica: que tanto su estructura (peso, altura y posición de cada una de sus partes) como su interacción con el entorno, respondan a comportamientos físicos reales y exactos.
- Inteligencia artificial: que aprenda a caminar por sí mismo, utilizando para ello métodos de *soft computing* como algoritmos genéticos.

1. Introducción

Siempre ha sido de interés la simulación biomecánica de seres vivos, especialmente en las ciencias naturales (zoología, medicina, etc.).

Ahora bien, últimamente se ha incrementado el interés en otras áreas de aplicación, como los videojuegos (creación de personajes con reacciones más reales), y la ingeniería (verbigracia: diseño de espacios cerrados, con mayores medidas de seguridad).

Una característica muy importante de este trabajo es que el humanoide no es fruto de una animación realizada por un artesano, que mueve cada uno de los segmentos a mano; sino un objeto compuesto de segmentos físicos, que interaccionan entre sí y con el entorno por medio de las leyes físicas; agregando así realismo a la situación simulada. La otra propiedad es que el bípedo aprenda por sí solo a caminar en línea recta, sin tener en su trayectoria ningún obstáculo. Este problema se puede abordar de diversas maneras, involucrando: redes neuronales¹, sistemas de control² (*passive walkers*³)[1], algoritmos genéticos⁴[2], entre otras. Varios de ellos implican modelos teóricos complejos de humanoide (considerando músculos con distintos materiales, etc.). Asimismo, existe un trabajo previo cuyo propósito era lograr la caminata de un cuadrúpedo virtual, utilizando algoritmos genéticos[3]. Continuando con esta última línea de investigación y extendiéndola a bípedos, en este proyecto también se buscó aplicar algoritmos genéticos, y lograr la caminata usando un modelo de humanoide basado en un conjunto de segmentos (cuerpos rígidos), unidos entre sí por articulaciones, cuyo desplazamiento depende de torques aplicados a dichos cuerpos, y sus parámetros se ajustan a partir de la evolución del algoritmo.

El presente informe, describe y analiza pormenorizadamente: en la sección 2, las herramientas aplicadas; en la 3, el modelo del humanoide utilizado; en la 4 y 5, los diferentes tipos de actuadores y funciones de partida y contorno; en la 6, el algoritmo genético; y en la 7 y 8, los resultados obtenidos con sus respectivas conclusiones.

¹Una red neuronal es un paradigma de aprendizaje automático. Se trata de un sistema de interconexión de neuronas que colaboran entre sí, para producir un estímulo de salida. Dada una entrada del sistema, se produce una salida, originada por varias transformaciones intermedias.

²Un sistema de control es un dispositivo (o conjunto de) que maneja, dirige o regula el comportamiento de otros dispositivos o sistemas, para minimizar los fallos y obtener los resultados deseados.

³Un *passive walker* utiliza el movimiento natural (*swinging*) de las piernas para ahorrar energía usada por motores. Para caminar, calcula la posición de ciertos puntos (las articulaciones, mayormente).

⁴Los algoritmos genéticos son métodos adaptativos, que pueden ser utilizados para resolver problemas de búsqueda y optimización. Están basados en la teoría de la selección natural, planteada por Charles Robert Darwin en 1859.

2. Herramientas

2.1. Motor físico

Se llama motor físico o *physics engine* a un “*software* capaz de realizar simulaciones de ciertos sistemas físicos, como la dinámica del cuerpo rígido, el movimiento de un fluido y la elasticidad” [4].

Actualmente, existen muchos motores físicos: ya sea de código propietario (*PhysX*, *Havok*), como *open-source* (*Bullet Physics*, *Box2D*, *Newton*, *OGRE*). Considerando análisis relacionados [5][6], y el hecho de que el espacio simulado fuese en 3D, se decidió que *Bullet Physics* [7] es el más idóneo. Está implementado en C++ y ha sido utilizado en varios juegos (*Grand Theft Auto IV* y *V*, etc); en los efectos especiales de películas (*Hancock*, *Bolt*, etc.); y proyectos científicos, como la herramienta *open-source Tensegrity Robotics Toolkit* de la NASA⁵; entre otros.

Si bien (como se verá más adelante) *Bullet* tiene problemas asociados con el coeficiente de restitución, posee una muy buena *performance* en la detección de colisiones, la dinámica y la resolución de *constraints*. Esto se debe, en parte, a diferentes algoritmos iterativos de orden lineal (donde el más importante es *Sequential Impulse*), de *caching* y también a la utilización de un modelo de fricción de Coulomb aproximado [8]. Además, el motor físico brinda la posibilidad de regular la precisión requerida en estos cálculos (sin olvidar que, con iguales recursos, a mayor precisión, mayor capacidad de cómputo requerida y, ergo, mayor tiempo). Dado que la construcción del humanoide implica definir características y restricciones de movimiento de cada uno de sus segmentos, lo antes mencionado fue crucial para la elección de *Bullet Physics* en este proyecto.

2.1.1. Funcionamiento

El motor físico se encarga de la simulación de cuerpos rígidos y la interacción entre los mismos. En particular debe calcular el resultado de colisiones, arreglar el solapamiento de los cuerpos en el espacio de simulación, estimar las fuerzas producidas debido a la fricción, y mantener el cumplimiento de restricciones que puedan existir entre los cuerpos (por ejemplo, un vínculo para formar una articulación). Para lograrlo, *Bullet* modela, a partir de un conjunto de ecuaciones, las distintas restricciones que deben ser respetadas.

Estos modelos reciben como dato la velocidad lineal y angular de cada objeto, y las fuerzas que actúan sobre los mismos. Dada esta información, se resuelve el sistema de ecuaciones, cuya solución representa las magnitudes de las fuerzas a accionar sobre el mismo, a fin de satisfacer todas las restricciones. Para encontrar esta solución, entran en juego los distintos métodos de complejidad lineal mencionados en la subsección anterior.

Este procedimiento, se lleva a cabo en cada *timestep* de la simulación; donde un *timestep* es el intervalo de tiempo que transcurre entre un cálculo de magnitudes y otro. Mientras menor sea el *timestep*, el simulador será más preciso pero también consumirá más tiempo de cómputo.

2.1.2. Modelo de fricción y su verificación

Hay reglas físicas relacionadas con el entorno y que son muy importantes para la caminata: el modelo de fricción, con sus respectivos coeficientes de fricción y restitución.

En base a los modelos físico-matemáticos que representan a cada uno de los dos fenómenos en cuestión (y que se explicarán a continuación), y pensando en posibles futuras simulaciones de

⁵<http://bulletphysics.org/Bullet/phpBB3/viewtopic.php?f=17&t=9978>

varios humanoides chocando e interactuando entre sí; se llevaron a cabo dos ensayos para verificar el funcionamiento del simulador físico *Bullet*.

2.1.2.1. Verificación del coeficiente de fricción

Para determinar el modelo utilizado por *Bullet* para simular las fuerzas resultantes sobre un cuerpo por acción de la fricción, se simuló un cubo, de $m_{cube} = 1kg$ y $l_{cube} = 1m$, que tiene una velocidad inicial constante (v_i) en el eje horizontal, que gradualmente se detiene por acción de la fricción, hasta llegar al reposo (Fig. 1).

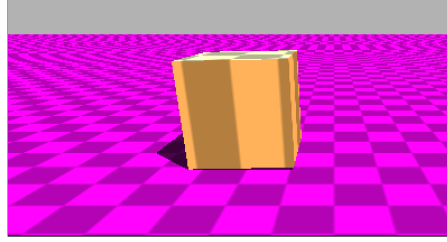


Figura 1: Visualización del sistema del cubo

Para esta prueba se utilizó el modelo matemático que representa la posición del cuerpo en el eje horizontal en función del tiempo t , representado por la siguiente ecuación:

$$x(t) = x_i + v_i t + \frac{1}{2} a t^2 \quad (1)$$

En este caso, el cuerpo empieza su movimiento en el origen, por lo tanto la posición inicial (x_i) es cero. v_i es la velocidad inicial, y a , la aceleración. Debido a la fricción entre el cuerpo y el suelo, se genera una fuerza de rozamiento \mathbf{F}_{μ_d} ⁶ (ec. (2)) en la misma dirección que la velocidad del sólido y en sentido contrario.

$$-\mathbf{F}_{\mu_d} = \mu_d \mathbf{F}_N \quad (2)$$

donde $\mathbf{F}_N = mg$ es la fuerza normal que actúa sobre la caja de masa m por acción de la gravedad $g = 10 \frac{m}{s}$, y μ_d es el coeficiente de fricción dinámico.

Finalmente, se obtiene la aceleración:

$$a = \frac{\mathbf{F}_{\mu_d}}{m} = \frac{-\mu_d \mathbf{F}_N}{m} = \frac{-\mu_d mg}{m} = -\mu_d g \quad (3)$$

Considerando las ec. (1) y (3), se puede obtener el modelo matemático que predice el movimiento de la caja:

$$x(t) = x_i + v_i t - \frac{1}{2} \mu_d g t^2 \quad (4)$$

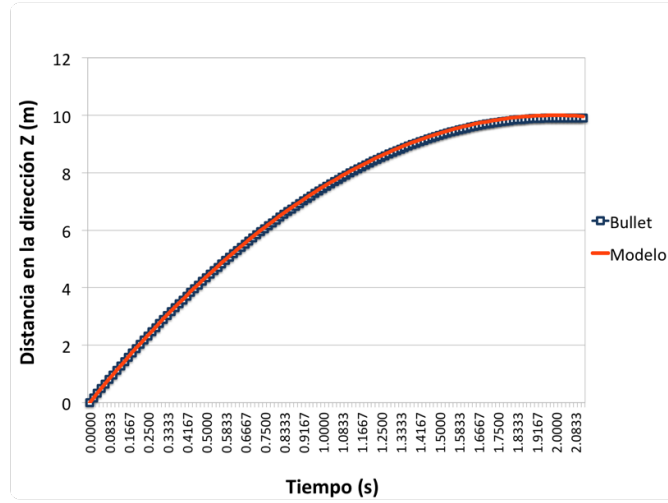
El intervalo de tiempo físico (o *timestep*) utilizado es $\Delta t = \frac{1}{60}s$. El *timestep* de animación (es decir, cada cuánto tiempo se guardan en un archivo los datos logrados) es $\Delta t' = 0.1$ y el tiempo de simulación es de $s = 100\Delta t$.

Seguidamente, se muestran los gráficos obtenidos al correr los experimentos numéricos con los siguientes valores: $v_i = \{1, 3 \text{ y } 10\} \frac{m}{s}$ y $\mu_d = \{0.25, 0.50 \text{ y } 0.75\}$. Se compara la distancia en el eje Z de la caja en *Bullet*, con su distancia en el mismo eje según la ec. (4).

⁶En las ecuaciones, los vectores se escriben en negrita.



(a)



(b)



(c)

Figura 2: Resultados logrados de simular el sistema descrito en Fig. 1, usando $v_i = 10 \frac{m}{s}$ y:

(a) $\mu_d = 0.25$, (b) $\mu_d = 0.50$, y (c) $\mu_d = 0.75$



(a)



(b)



(c)

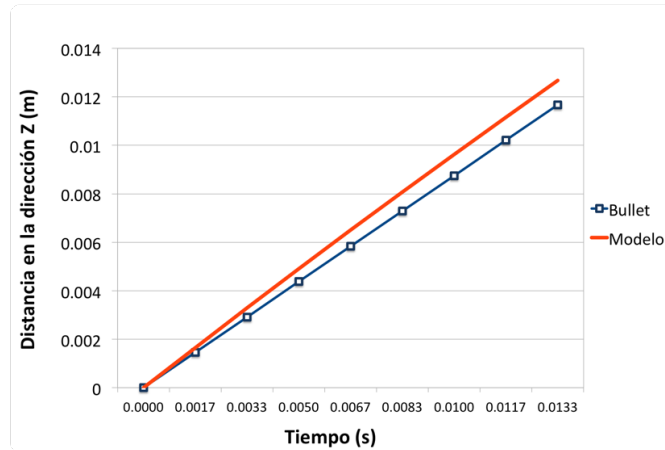
Figura 3: Resultados logrados de simular el sistema descrito en Fig. 1, usando $v_i = 3 \frac{m}{s}$ y:
(a) $\mu_d = 0.25$, (b) $\mu_d = 0.50$, y (c) $\mu_d = 0.75$



(a)



(b)



(c)

Figura 4: Resultados logrados de simular el sistema descrito en Fig. 1, usando $v_i = 1 \frac{m}{s}$ y:
 (a) $\mu_d = 0.25$, (b) $\mu_d = 0.50$, y (c) $\mu_d = 0.75$

Las Fig. 2 y 3 demuestran que los resultados de las pruebas fueron favorables. Los valores obtenidos a partir de la simulación se corresponden (en mayor o menor medida) con los alcanzados a partir del modelo matemático. Esto es un indicador de que *Bullet* debe estar usando dichos

modelos para ejecutar las simulaciones. Vale observar que, cuanto mayor es la velocidad inicial (v_i), mayor es la similitud entre los dos casos.

No obstante, los gráficos que corresponden a la Fig. 4, presentan una discrepancia mayor entre la simulación y el modelo. Este hecho puede deberse al mecanismo utilizado por *Bullet* para resolver la fricción de un cuerpo a velocidades muy bajas.

2.1.2.2. Verificación del coeficiente de restitución

El segundo ensayo simula una esfera a una altura determinada sobre el suelo, que tiene una velocidad inicial (v_i) en el eje perpendicular al piso, y que eventualmente colisiona contra el mismo. Se desea comprobar que, la colisión entre el cuerpo y el suelo respete que la velocidad final (v_f) de la esfera después del choque, sea proporcional a su coeficiente de restitución (e) dado por la ecuación:

$$e = \frac{v_f}{v_i} \quad (5)$$

Para efectuar la colisión con el suelo, se empleó una esfera sólida ubicada a 4 metros del suelo, cuya masa y radio son $m_{sphere} = 1 \text{ kg}$ y $r_{sphere} = 1 \text{ m}$, respectivamente (Fig. 5). A la esfera se le asigna, además, un coeficiente de restitución determinado.

Se eligió un ambiente sin gravedad ($g = 0 \frac{m}{s^2}$). De esta forma, se podrá tener en cuenta sólo la velocidad inicial (v_i) y la velocidad final (v_f) para el cálculo del coeficiente de restitución (e) (ver ec. (5)).

El intervalo de tiempo físico (o *timestep*) utilizado es $\Delta t = 0.001 \text{ s}$. El *timestep* de animación (es decir, cada cuánto tiempo se guardan en un archivo los datos logrados) es $\Delta t' = 0.1$ y el tiempo de simulación es de $s = 100\Delta t$. El coeficiente de fricción es $\mu = 0.75$.

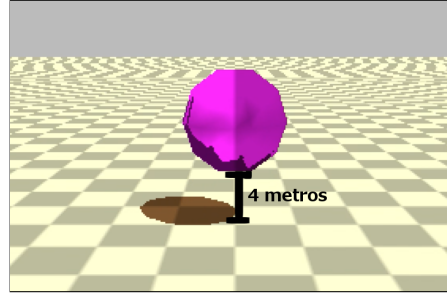


Figura 5: Visualización del sistema de la esfera

El ensayo tiene como parámetros de entrada: v_i (velocidad inicial) y e_{sim} (coeficiente de restitución esperado). Por otro lado, se obtiene v_f (velocidad de la esfera al finalizar la simulación); y luego se calculan e_{medida} (coeficiente de restitución obtenido a partir de la ec. (5)) y ϵ_{rel} , que es el error relativo entre los coeficientes e_{sim} y e_{medida} , calculado de la siguiente manera:

$$\epsilon_{rel} = \frac{e_{sim} - e_{medida}}{e_{sim}} \quad (6)$$

En la Tabla 1 se puede observar lo arrojado por los experimentos numéricos que se efectuaron; usando: $v_i = \{-0.5, -3.5, -4, -5 \text{ y } -10\} \frac{m}{s}$ y $e_{sim} = \{0.2, 0.5 \text{ y } 0.8\}$.

v_i	e_{sim}	v_f	e_{medida}	ϵ_{rel}
$-0.5 \frac{m}{s}$	0.2	$0.000249 \frac{m}{s}$	0.000498	0.997
	0.5	$0.000219 \frac{m}{s}$	0.000438	0.999
	0.8	$0.001037 \frac{m}{s}$	0.002074	0.997
$-3.5 \frac{m}{s}$	0.2	$0.000057 \frac{m}{s}$	0	1
	0.5	$0.000018 \frac{m}{s}$	0	1
	0.8	$0.3 \frac{m}{s}$	0.0857	0.893
$-4 \frac{m}{s}$	0.2	$0.000473 \frac{m}{s}$	0.00012	1
	0.5	$0.000424 \frac{m}{s}$	0.00011	1
	0.8	$1.23 \frac{m}{s}$	0.3	0.625
$-5 \frac{m}{s}$	0.2	$1 \frac{m}{s}$	0.2	0
	0.5	$2.5 \frac{m}{s}$	0.5	0
	0.8	$4 \frac{m}{s}$	0.8	0
$-10 \frac{m}{s}$	0.2	$2 \frac{m}{s}$	0.2	0
	0.5	$5 \frac{m}{s}$	0.5	0
	0.8	$8 \frac{m}{s}$	0.8	0

Tabla 1: Coeficientes de restitución obtenidos (e_{medida}) de simular el sistema descrito en Fig. 5

Los resultados exponen una limitación del motor físico: no representa correctamente las colisiones elásticas entre esferas y cuerpos rígidos, que ocurren a velocidades bajas. Esto queda en evidencia en la Tabla 1. En cada una de ellas el error fue de casi 1. La razón por la que ocurre este hecho se debe a que *Bullet* utiliza un algoritmo de colisión que frena la velocidad de un objeto que está a punto de colisionar. Haciendo esto puede evitar que los sólidos se traspasen y de esta forma se pueden realizar cálculos de fuerza más precisos.

En el caso de los ensayos, las esferas poseen una velocidad muy baja, cuando están a punto de colisionar *Bullet* reduce aún más esta velocidad y eventualmente quedan con una velocidad tan baja que al chocar contra el suelo se aplica el efecto restitutivo a esta velocidad casi nula y se resuelve que la esfera debe quedar en reposo, cuando en realidad debería poseer una velocidad baja, pero no despreciable.

2.1.3. Ventajas

Las ventajas del motor físico son:

- Código abierto: mayor conocimiento sobre las fórmulas y métodos implementados en el motor.
- Soporte de la comunidad científica.
- Licencia libre.

2.1.4. Desventajas

Como toda herramienta, *Bullet* tiene aspectos negativos, entre los que se encuentran:

- Documentación poco clara y desordenada.
- Debido a que la física se aproxima usando métodos numéricos que contienen error, las simulaciones son no determinísticas.

- Utilizar una librería gráfica como *OpenGL* acoplada a una simulación de *Bullet*, puede producir resultados distintos, que si se usa un programa de visualización externo como OVITO.

2.2. Librería de algoritmos genéticos

Se utilizó la conocida librería de algoritmos genéticos para C++ GaLib, desarrollada por Matthew Wall del MIT [9].

Ofrece funcionalidades como: programación paralela, diversos métodos de selección (*elite*, *roulette*), estrategias de reemplazo (de padres, aleatorio, del peor), entre otras.

Cabe aclarar que, antes de optar por GaLib, se había adoptado la librería de algoritmos genéticos de Mladen Jankovic, pero finalmente fue descartada, debido a errores o *bugs* en la misma (y que, cada vez, resultaban ser más inmanejables).

2.3. Código fuente

Al estar *Bullet* implementado en C++, el código fuente también está desarrollado en ese lenguaje. En *Bullet*, se define un *World* (o mundo físico) en donde se puede insertar, entre otras cosas, cuerpos rígidos. En este caso en particular, el mundo consta de un plano (el suelo), y el bípodo (compuesto por cuerpos rígidos y otros elementos físicos) ubicado sobre él.

El *software* creado incluye: construcción del humanoide, siendo éste capaz de desplazarse por medio de actuadores (que se verán en la Sección 4); el desarrollo del algoritmo genético (definición de los individuos, función de *fitness*, métodos de selección, etc.); visualización gráfica del mejor humanoide logrado por el algoritmo genético; y la posibilidad de realizar gráficos referidos a la evolución del algoritmo genético (*fitness* por cada generación, etc.).

Se acompañan a esta presentación: el código fuente y el manual de instalación y uso.

3. Modelo utilizado

Dentro de los diversos modelos existentes ([1][2]), en este trabajo se procuró utilizar uno que fuera sencillo pero representativo a la vez.

Se modela al cuerpo del bípodo, con el motor *Bullet Physics*, como un conjunto de segmentos unidos entre sí por articulaciones. A cada uno de ellos se les aplica un torque en el centro de masa de cada segmento (denominado actuador). Que la caminata se produzca o no, depende del tipo de actuador empleado (la función utilizada para el torque), y de sus parámetros. El objetivo es encontrar dichos parámetros. Para eso se usan los algoritmos genéticos, un método de inteligencia artificial. De este modo se obtiene, de forma análoga a la selección natural, los individuos que mejor se adaptan a la caminata. Tanto los actuadores como el algoritmo genético se explicarán más adelante.

3.1. Composición física del humanoide

Como ya se expresó, el humanoide fue modelado en *Bullet* como un conjunto de segmentos (cuerpos rígidos), unidos entre sí por articulaciones.

Sobre la base de la anatomía humana, se dividió el cuerpo del humano virtual en: cabeza, tronco, miembro superior (brazo, antebrazo y mano), pelvis y miembro inferior (muslo, pierna y pie).

A los fines de este proyecto, sólo se consideraron pelvis y miembro inferior (Fig. 6).



Figura 6: Segmentos del humanoide

En la Tabla 2 se presenta la composición física de cada uno de los segmentos del humano virtual, de acuerdo a la biomecánica [10].

Segmento			Cantidad	Forma	Largo (en m)	Masa (en kg)	Uniones
	Pelvis		1	Esférica	0.08655	9.9718	Cadera
Miembro inferior		Muslo	2	Esfero-cilíndrica	0.4015	10.3368	Cadera y Rodilla
		Pierna	2	Esfero-cilíndrica	0.4015	3.1609	Rodilla y Tobillo
		Pie	2	Esfero-cilíndrica	-	1.0001	Tobillo

Tabla 2: Composición física de cada segmento del humanoide

3.2. Articulaciones

Para unir los distintos segmentos entre sí, se utilizaron articulaciones bisagra con 1 grado de libertad, en el eje X, para que los segmentos puedan moverse en dos ejes: el Z, donde ocurre la caminata, y el Y, perpendicular al piso (Fig. 7(a) y (b)).

A su vez, para cada caso en particular, se definieron cotas para los ángulos que pueden existir entre los segmentos (ver Tabla 3). Esto es muy importante, no sólo porque se adecúa a datos biológicos, sino porque, de otro modo la caminata no podría lograrse: si los ángulos son demasiado altos, la caminata se produce girando las piernas por encima de la pelvis; si por el contrario, son demasiado bajos, las piernas van a estar muy rígidas, originando pocos pasos y muy cortos.

Asimismo, se le impide rotar a la pelvis, y se restringe la amplitud con la que puede moverse la cadera (de $-\frac{\pi}{4}$ a $\frac{\pi}{4}$). Esto se realiza porque, en caso contrario, el bípedo necesitaría un sistema de control para mantener el equilibrio, y eso excede el alcance de este trabajo.

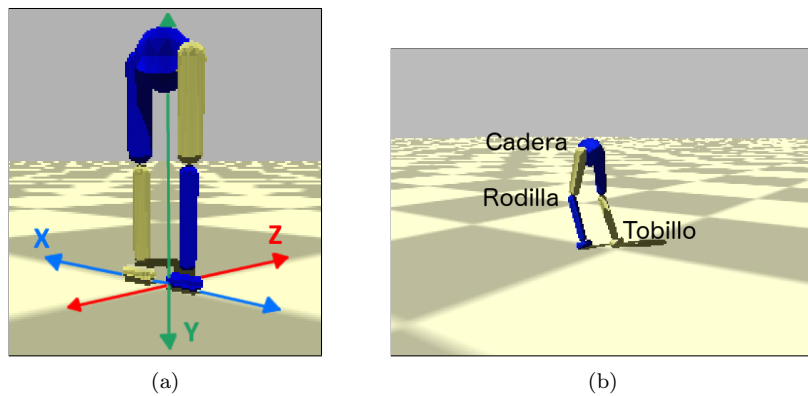


Figura 7: Humanoide diseñado: (a) ejes, y (b) articulaciones

		Ángulo mínimo	Ángulo máximo
Articulación	Cadera	$-\frac{\pi}{4}$	$\frac{\pi}{4}$
	Rodilla	$-\pi$	π
	Tobillo	0	0

Tabla 3: Rango de valores de ángulo de cada articulación del humanoide

4. Actuadores

A cada uno de los segmentos correspondientes al muslo y la pierna del bípedo, se le aplica un torque (o actuador) en el eje X (perpendicular a la trayectoria), como se ve en Fig 8. Así, pueden moverse para arriba o para abajo (con respecto a la articulación a la que pertenecen). A fin de simplificar el modelo, el humanoide tiene el mismo tipo de actuador utilizado en todos los segmentos.

Es necesario aclarar que la caminata producida por el humanoide es plana (en 2D). Esto se debe a que la trayectoria pensada para el bípedo es una línea recta, y logrando que los segmentos se muevan en un solo eje es suficiente para cumplir con dicha trayectoria. También contribuye el hecho de que el torque se aplique en una sola dimensión. Por otra parte, los actuadores definidos a continuación son periódicos, y por eso no se pueden aplicar en el eje Z de los segmentos (se necesitarían actuadores reactivos, para poder detectar cuando el humanoide se está cayendo, etc.). Para indicar el módulo de dicho torque, se diseñaron diferentes funciones (todas ellas periódicas), mencionadas en las subsecciones que siguen.

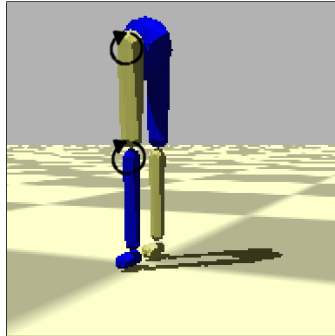


Figura 8: Aplicación de los actuadores en los segmentos del bípedo

4.1. Genérico

Es el actuador más sencillo, tanto matemática como computacionalmente.

$$f(t) = A_1 \sin(\omega_1 t + \phi) + A_2 \cos(\omega_2 t + \phi) + C \quad (7)$$

donde $f(t)$ es la función del actuador evaluada en el instante de tiempo t , A_1 y A_2 son amplitudes, ω_1 y ω_2 son frecuencias (en $\frac{1}{s}$), ϕ es la fase en radianes, y C es un término independiente (ver Fig. 9 (a) y (b)).

La fase ϕ es la misma en el seno y en el coseno, para evitar que se formen otro tipo de funciones no cíclicas.

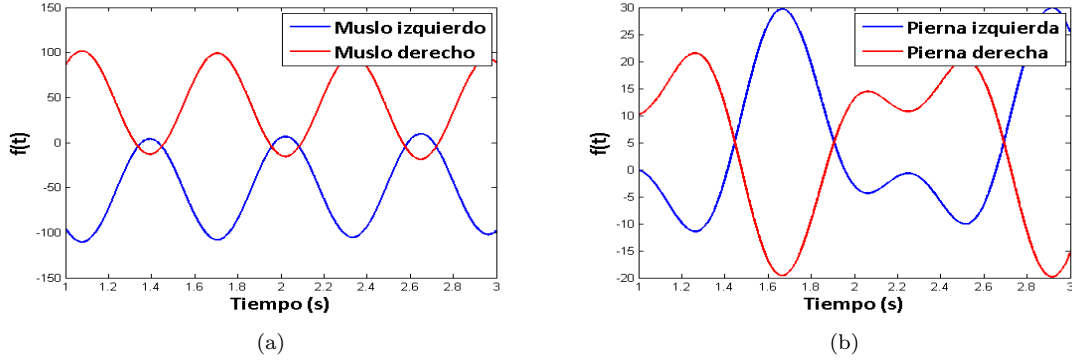


Figura 9: Ejemplo de actuador genérico aplicado en: (a) muslo, y (b) pierna

4.2. Coseno doble frecuencia

Basada en [3], esta función periódica utiliza medio ciclo de una función sinusoidal, y medio ciclo de otra (ambas pueden tener frecuencias distintas). Esto podría tener sentido porque en una caminata, un miembro inferior primero avanza hacia adelante y luego se extiende hacia atrás, y es razonable que esos dos movimientos se produzcan a frecuencias distintas (Fig. 10 (a) y (b)). La idea es lograr una función periódica a partir de una que no lo es (ya que t es lineal). Para eso, se utiliza la función $\psi(t)$ (ec. (8)) que aplica una transformación a los números reales, para que se encuentren dentro del rango del ciclo completo (con las dos frecuencias). ω es la frecuencia de $f(t)$ en $\frac{1}{s}$ (ec. (10)), que utiliza medio ciclo con frecuencia ω_1 y medio ciclo con frecuencia ω_2 .

$$\psi(t) = t + \phi - \left\lfloor \frac{t + \phi}{\pi/\omega_1 + \pi/\omega_2} \right\rfloor (\pi/\omega_1 + \pi/\omega_2) \quad \psi : \mathbb{R} \rightarrow \left[0, \frac{2\pi}{\omega}\right] \quad (8)$$

$$\omega = \frac{2\omega_1\omega_2}{\omega_1 + \omega_2} \quad (9)$$

$$f(t) = \begin{cases} A \cos(\omega_1 \psi(t)) + C & \text{si } \omega_1 \psi(t) < \pi \\ A \cos(\omega_2 (\psi(t) - (\pi/\omega_1) + (\pi/\omega_2))) + C & \text{en otro caso} \end{cases} \quad (10)$$

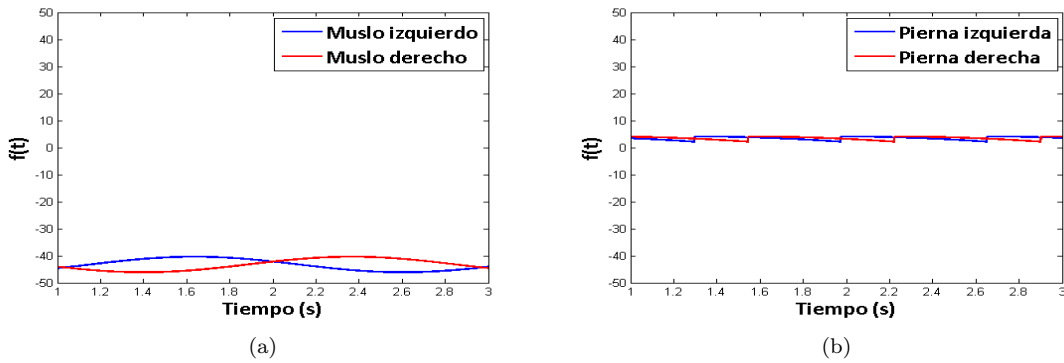


Figura 10: Ejemplo de actuador coseno doble frecuencia aplicado en: (a) muslo, y (b) pierna

4.3. Fourier de orden 2

Este actuador utiliza una serie de Fourier de dos términos. Dado que origina un resultado que, si bien es periódico, puede no ser de naturaleza sinusoidal, lo cual puede ser beneficioso para realizar otro tipo de movimientos.

$$f(t) = A_1 \sin(\omega t + \phi) + B_1 \cos(\omega t + \phi) + A_2 \sin(2\omega t + \phi) + B_2 \cos(2\omega t + \phi) + C \quad (11)$$

donde A_1 , A_2 , B_1 y B_2 son amplitudes (ver Fig. 11 (a) y (b)).

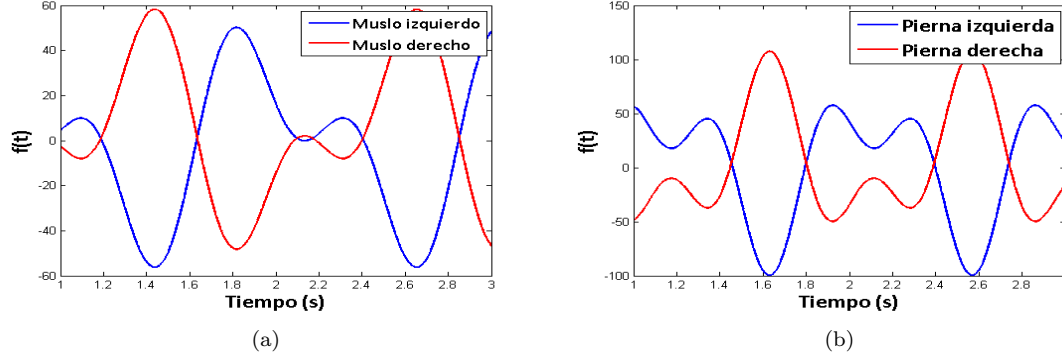


Figura 11: Ejemplo de actuador fourier de orden 2 aplicado en: (a) muslo, y (b) pierna

4.4. Fourier de orden 9

Es una extensión del actuador anterior, pero con 9 términos. Por ser de mayor grado, brinda una mayor precisión. Sin embargo, es más difícil de manejar computacionalmente; y, además, que sea más preciso no garantiza que con él se pueda lograr una buena caminata (ver Fig. 12 (a) y (b)).

$$\begin{aligned} f(t) = & A_1 \sin(\omega t + \phi) + B_1 \cos(\omega t + \phi) + A_2 \sin(2\omega t + \phi) + B_2 \cos(2\omega t + \phi) \\ & + A_3 \sin(3\omega t + \phi) + B_3 \cos(3\omega t + \phi) + A_4 \sin(4\omega t + \phi) + B_4 \cos(4\omega t + \phi) \\ & + A_5 \sin(5\omega t + \phi) + B_5 \cos(5\omega t + \phi) + A_6 \sin(6\omega t + \phi) + B_6 \cos(6\omega t + \phi) \\ & + A_7 \sin(7\omega t + \phi) + B_7 \cos(7\omega t + \phi) + A_8 \sin(8\omega t + \phi) + B_8 \cos(8\omega t + \phi) \\ & + A_9 \sin(9\omega t + \phi) + B_9 \cos(9\omega t + \phi) + C \end{aligned} \quad (12)$$

donde A_i y B_i con $1 \leq i \leq 9$ son amplitudes.

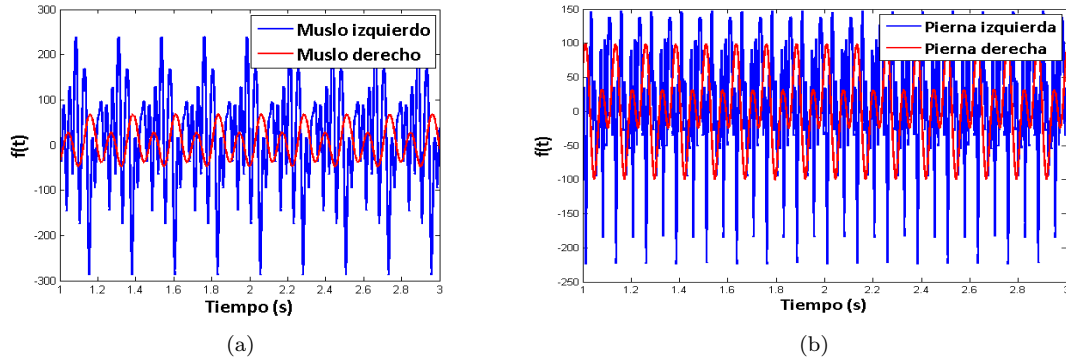


Figura 12: Ejemplo de actuador fourier de orden 9 aplicado en: (a) muslo, y (b) pierna

5. Condiciones iniciales y de contorno

Como las funciones periódicas señaladas en los actuadores, no fueron suficientes para lograr la caminata, se le adosaron las funciones que se detallan seguidamente.

5.1. Función partida

La caminata se divide en dos partes bien definidas:

La primera denominada función partida, es opcional y representa el primer paso dado por el bípedo. En ella, el humanoide comienza con una posición (las dos extremidades bien extendidas) diferente a la que tendrá en el resto de la caminata. Se aplican actuadores genéricos en cada uno de los cuatro segmentos movibles (muslos y piernas). Para simplificar el modelo, se decidió que esta etapa durara 0.7 segundos. Dicho valor fue extraído de forma experimental.

La segunda es el resto de la caminata, y debe ser cíclica y periódica (para poder eventualmente repetirse *ad infinitum*). Aquí se puede utilizar cualquiera de los actuadores vistos en la sección anterior, pero solo un tipo de ellos es aplicado a los segmentos.

5.2. Fase sincronizada

En una caminata, las piernas deben guardar simetría: mientras una va hacia adelante, la otra va hacia atrás (y viceversa). Esto, de acuerdo con los actuadores definidos en la sección anterior, implica que las funciones de movimiento de cada pierna estén desfasadas en medio ciclo ($\frac{\pi}{2}$):

$$f_i(t) = f(t) \quad (13)$$

$$f_d(t) = f(t + \frac{\pi}{2}) \quad (14)$$

siendo $f(t)$ la función de movimiento (o actuador) en el momento t , y f_i y f_d las funciones de la pierna izquierda y derecha, respectivamente.

6. Algoritmo genético

Los algoritmos genéticos son métodos adaptativos, que pueden ser utilizados para resolver problemas de búsqueda y optimización. Están inspirados en la teoría darwiniana de la selección natural. La entidad a optimizar constituye un individuo dentro de una población; que puede ser cruzado con otros individuos para obtener un “hijo”, que presenta características de sus “padres”. A través de una función de *fitness* se decide cuáles de ellos son los más “aptos”, que luego pasarán a una nueva generación de individuos. Este proceso se repite hasta obtener un individuo lo suficientemente apto, que representa a la entidad optimizada.

El objetivo de este proyecto es optimizar la caminata del humanoide. Ergo, en el algoritmo genético aplicado, el individuo está definido por rasgos del bípedo, y la función de *fitness* se focaliza en mejorar diversos aspectos de dicha caminata.

6.1. Individuo

La información genética de cada individuo, está definida a partir de un vector que contiene de forma contigua, los parámetros de la función partida (optativa) y los asociados a los actuadores (obligatorios), que actúan sobre cada uno de los segmentos (muslos y piernas) del humano virtual.

6.1.1. Parámetros

Tanto la función partida como los actuadores tienen como parámetros: amplitud (A o B); frecuencia (ω); fase (ϕ), que indica dónde comienza el paso y se mide en radianes; y término independiente (C).

La Fig. 13 muestra la estructura del individuo y su composición para un humanoide con función partida y actuador genérico.

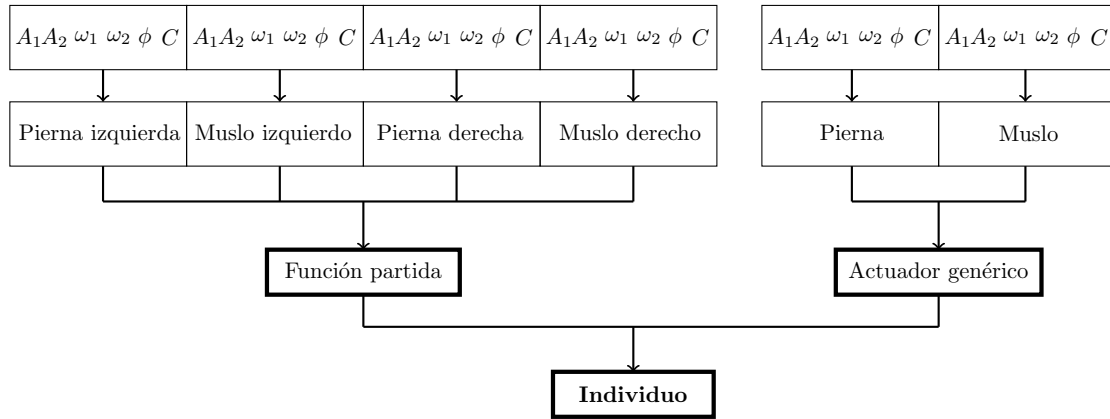


Figura 13: Esquema de un individuo (Ejemplo con actuador genérico)

6.1.2. Valores

Cada uno de los segmentos tiene una composición física distinta (largo, masa, etc.), razón por la cual no necesariamente sus genes deban tener los mismos rangos de valores, tal como puede apreciarse en las Tablas 4 y 5.

Segmento	Tipo de gen	Mínimo	Máximo
Muslo	Amplitud	-30	30
Pierna	Amplitud	-60	60
Muslo y Pierna	Frecuencia	0.01	10
Muslo y Pierna	Fase	$-\pi$	π
Muslo y Pierna	Término independiente	-10	10

Tabla 4: Rango de valores que puede tomar cada gen, para la función partida

Actuador	Segmento	Tipo de gen	Mínimo	Máximo
Genérico	Muslo	Amplitud	-30	30
	Pierna	Amplitud	-60	60
	Muslo y Pierna	Frecuencia	0.01	10
	Muslo y Pierna	Fase	$-\pi$	π
	Muslo y Pierna	Término independiente	-10	10
Coseno doble frecuencia	Muslo	Amplitud	-50	50
	Pierna	Amplitud	-30	30
	Muslo y Pierna	Frecuencia	0.01	5
	Muslo y Pierna	Fase	$-\pi$	π
	Muslo y Pierna	Término independiente	-30	30
Fourier de orden 2	Muslo	Amplitud	-60	60
	Pierna	Amplitud	-30	30
	Muslo y Pierna	Frecuencia	0.01	10
	Muslo y Pierna	Fase	$-\pi$	π
	Muslo y Pierna	Término independiente	-10	10
Fourier de orden 9	Muslo y Pierna	Amplitud	-60	60
	Muslo y Pierna	Frecuencia	0.1	2
	Muslo y Pierna	Fase	$-\pi$	π
	Muslo y Pierna	Término independiente	-10	10

Tabla 5: Rango de valores que puede tomar cada gen, según el tipo de actuador

6.1.3. Implementación de individuos

Para favorecer el análisis de las distintas características arriba indicadas, se implementaron varios individuos, cada uno de ellos con propiedades distintas (Tabla 6).

	Individuo				
	Tipo 1	Tipo 2	Tipo 3	Tipo 4	Tipo 5
Actuador	Genérico	Genérico	Coseno doble frecuencia	Fourier de orden 2	Fourier de orden 9
Función partida	No	Sí	Sí	Sí	Sí
Fase sincronizada	Sí	Sí	Sí	Sí	Sí

Tabla 6: Tipo de individuos

6.1.4. Constitución del cromosoma

Los distintos actuadores y la función partida tienen los parámetros presentados en la sección 6.1.1. Sus respectivas cantidades pueden verse en la Tabla 7.

En la Tabla 8 se muestra la composición del cromosoma de cada individuo, que depende de los actuadores y la función partida usados. En ella se puede observar cómo según el tipo de individuo, varía la cantidad de genes, es decir, la longitud del cromosoma. Vale aclarar que la función partida se especifica para cada segmento (los dos muslos y las dos piernas); en cambio, para los

actuadores, sólo se definen dos (uno para los muslos y otro para las piernas).

		Parámetro			
		Amplitud	Frecuencia	Fase	Término independiente
Actuador	Genérico	2	2	1	1
	Coseno doble frecuencia	1	2	1	1
	Fourier de orden 2	4	1	1	1
	Fourier de orden 9	18	1	1	1
Función partida		2	2	1	1

Tabla 7: Cantidad de parámetros según tipo de actuador y función partida

		Individuo				
		Tipo 1	Tipo 2	Tipo 3	Tipo 4	Tipo 5
Parámetro	Amplitud	4	12	8	16	44
	Frecuencia	2	6	8	6	6
	Fase	2	6	6	6	6
	Término independiente	2	6	6	6	6
Totales		10	30	28	34	62

Tabla 8: Cantidad de parámetros según tipo de individuos

6.2. Fitness

El papel de la función de *fitness* (F) en un algoritmo genético es evaluar qué tan bueno es un individuo. En este caso, está definida como un producto de cinco módulos o propiedades: altura (H), velocidad (V), dirección (D), simetría (S) y pies abajo (PA):

$$F = H \cdot V \cdot D \cdot S \cdot PA \quad (15)$$

Los cinco tienen la misma importancia y por eso, como se verá a continuación, están definidos de forma similar (con una función exponencial y pueden valer entre 0 y 1). Con todo esto, dado que el *fitness* está pensado como un producto, basta con que uno de los módulos sea muy chico para “anular” al individuo (es decir, otorgarle un valor que tiende a cero). Sin embargo, los diferentes módulos no son completamente independientes entre sí: por ejemplo, si la altura es demasiado baja, posiblemente la velocidad y la dirección no sean adecuadas.

6.2.1. Altura

Es un factor relacionado con la altura del individuo en toda la simulación, y se expresa:

$$H = \frac{\sum_{n=0}^T e^{-C(h_{t_n} - h_{t_0})^2}}{N} \quad (16)$$

donde t_0 es el tiempo inicial, t_T el tiempo final, h_{t_n} es la altura de la pelvis en el instante de tiempo t_n , N la cantidad pasos de simulación y C una constante $C = 5$.

Se calcula a partir de la diferencia entre la altura en cada instante de la simulación, con su altura inicial (la altura está definida como la posición de la pelvis en el eje Z). Cuanto mayor sea esa diferencia, más rápido el individuo cae, y por eso este módulo tiende a cero. Por el contrario, valdrá uno si la diferencia es ínfima (lo que significa que el humanoide mantiene su misma altura durante la caminata).

6.2.2. Velocidad

Indica qué tan cercana es la velocidad del individuo con respecto a una velocidad objetivo (en este caso, es de 1.2 m/h), y se expresa de la siguiente forma:

$$V = \frac{\sum_{n=0}^T e^{-C(v_{t_n} - V_O)^2}}{N} \quad (17)$$

donde t_0 es el tiempo inicial, t_T el tiempo final, v_{t_n} es la velocidad de la pelvis en el instante de tiempo t_n , y V_O la velocidad objetivo en el eje Z (el eje de la caminata).

Sigue una lógica y cálculo similares al factor de altura: a mayor discrepancia de la velocidad real del humanoide con V_O , menor (y más cercano a cero) es el valor arrojado por el módulo de velocidad.

6.2.3. Dirección

Señala qué tan similares son la dirección objetivo (un vector unitario, que en este caso se encuentra en el eje Z) y la dirección con la que camine el humanoide. Se calcula como sigue:

$$D = \frac{\sum_{n=0}^T e^{-C(\mathbf{v}_{t_n} \cdot \mathbf{V}_O - 1)^2}}{N} \quad (18)$$

donde t_0 es el tiempo inicial, t_T el tiempo final, \mathbf{v}_{t_n} el versor de la dirección del humanoide en el momento t_n y \mathbf{V}_O el versor de la dirección objetivo.

El producto escalar entre los versores responde a la Similitud Coseno: $\cos \theta = \frac{\mathbf{A} \cdot \mathbf{B}}{AB}$, donde \mathbf{A} y \mathbf{B} son vectores que no se encuentran normalizados, y θ es el ángulo formado entre ellos. Así, si $\cos \theta = 1$, significa que los vectores están paralelos entre sí (que es el efecto buscado en el caso de la dirección).

Al producto escalar se le resta 1, para que el módulo sea consistente con la función exponencial utilizada y que valga 1 cuando $\theta = 0$, y 0 cuando $\theta = \pi$. Cabe aclarar que se trata al ángulo en forma simétrica, ya que, por ejemplo $\cos(-\pi/6) = \cos(\pi/6)$.

6.2.4. Simetría

Este indicador marca qué tan equidistantes se encuentran los pies de la cadera, a lo largo de la caminata. Aplicando solamente los módulos antes mencionados, provocaba resultados en donde una pierna quedaba más distante de la pelvis que la otra, lo que producía que el humanoide se terminara arrastrando, posiblemente afectando a la velocidad.

Para mayor simplicidad, la simetría S se calculó a partir de los pies (y no de las piernas). Se tomaron en cuenta sólo los ejes X y Z, porque son los relacionados a la velocidad y a la dirección, respectivamente.

$$S = \frac{\sum_{n=0}^T \frac{1}{2} [e^{-C(lf_z + rf_z|^2)} + e^{-C(lf_x + rf_x|^2)}]}{N} \quad (19)$$

donde lf_X y lf_Z es la distancia desde el pie izquierdo hasta la pelvis en los ejes X y Z, respectivamente; y en donde rf_X y rf_Z es lo mismo, pero para el pie derecho.

6.2.5. Pies abajo

Con los módulos señalados anteriormente, se busca que el humanoide camine con una velocidad y dirección determinadas, que no se caiga y que mantenga simetría mientras ejecuta sus movimientos. Pero, todo esto daría, en el mejor de los casos, una caminata estilo “estrella”. Sin embargo, una característica fundamental en una caminata normal es que las piernas (ergo, los pies también) no sobrepasen la cadera. Si bien ésta es una propiedad negativa (expresa lo que no debe tener una caminata), y se puede correr el riesgo de restringir demasiado, su ausencia da resultados peores.

$$PA = \frac{\sum_{n=0}^T \frac{1}{2}(\alpha[e^{-C(ldf^2)} + e^{-C(rdf^2)}])}{N} \quad (20)$$

donde ldf y rdf son las diferencias entre la posición inicial de los pies y la altura en el momento t_n de los pies izquierdo y derecho, respectivamente, y $\alpha = \max(\min(lf, rf) - hip, 0, 1)$, siendo lf , rf y hip las alturas del pie izquierdo, pie derecho, y la cadera (es decir, vale 0 si la altura del pie izquierdo o derecho supera a la de la cadera, y 1 en otro caso).

6.3. Operadores del algoritmo

Permiten controlar en detalle el proceso de optimización. En particular, se busca un balance entre la diversidad de los individuos, el aumento del *fitness* a lo largo del algoritmo, y evitar la convergencia hacia una población sobre la cual no se puede seguir mejorando.

6.3.1. Métodos de selección

De la vasta cantidad de métodos de selección que existen se utilizó **Roulette**, método estocástico que selecciona un individuo de la población total al azar, con una probabilidad proporcional a su *fitness*.

Para no perder la mejor solución encontrada en cada generación, se activó el **Elitism**, técnica en la cual el individuo con mayor *fitness* permanece intacto en la siguiente generación.

6.3.2. Métodos de cruce

El método de cruce (o *crossover*) utilizado es el siguiente: De dos individuos (los padres), se originan dos nuevos individuos (los hijos). Se toma cada uno de los genes de los padres y se elige, con una probabilidad uniforme, uno de ellos para un hijo y el otro para el otro hijo.

La probabilidad de que este proceso ocurra es de 0.9.

6.3.3. Mutación

En el caso de la mutación, para cada gen del individuo, se decide con cierta probabilidad si se lo muta o no. En caso afirmativo, se cambia ese gen por un valor aleatorio (que esté dentro de su rango definido).

Dicha probabilidad de mutación es de 0.3.

6.3.4. Otras características

En el algoritmo genético se utilizan 1000 generaciones, de 50 individuos cada una. Además, la simulación de cada individuo (necesaria para calcular el *fitness*) es de 4 segundos.

7. Resultados obtenidos

En la Fig. 14 se representan los individuos definidos con sus características.

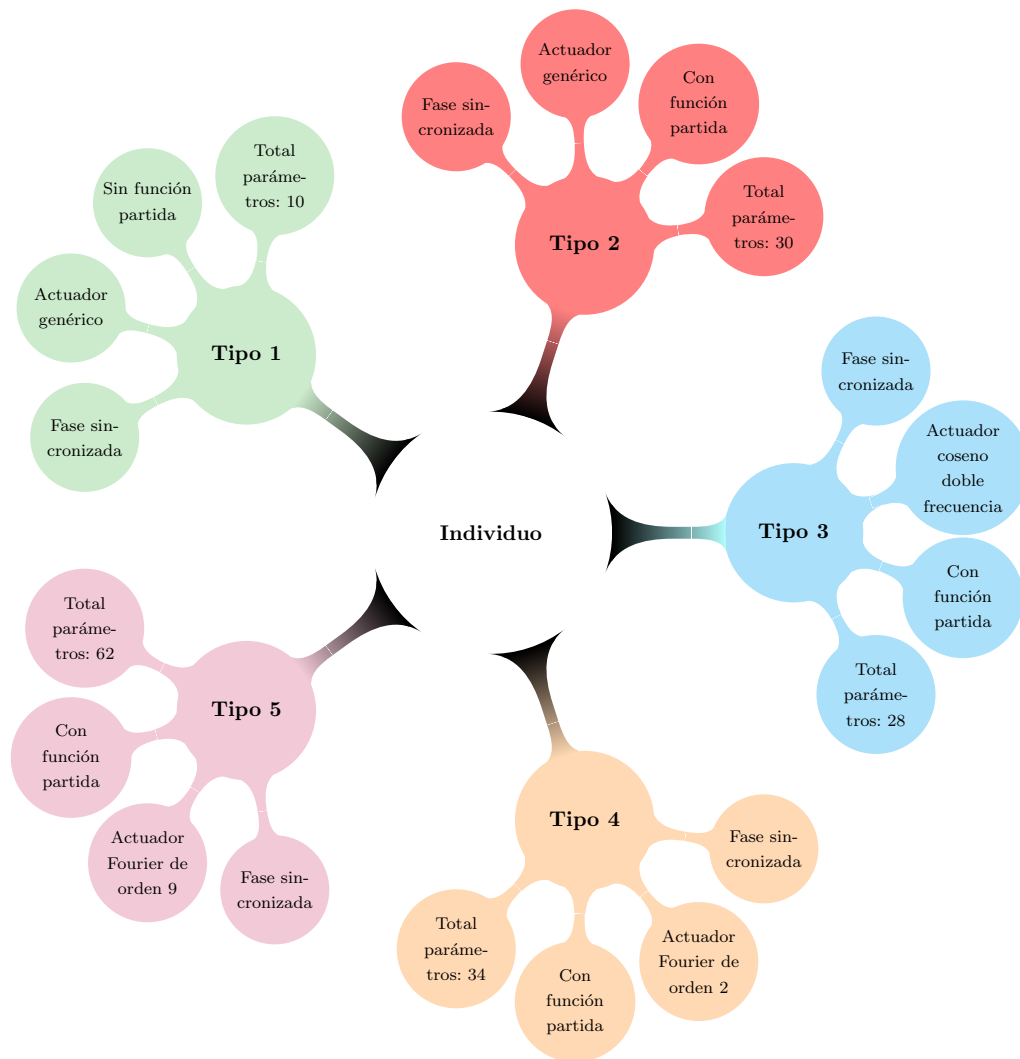


Figura 14: Individuos definidos en el algoritmo genético

Sobre ellos se realizaron pruebas, corriendo el algoritmo genético, y evaluando el resultado alcanzado posteriormente (ya sea numérica o gráficamente).

A continuación se analizan distintos aspectos relevantes. Vale aclarar que, en el caso de la altura y la velocidad, se muestran los resultados de los mejores caminantes de cada tipo; y en el caso del *fitness*, los resultados de corridas típicas del algoritmo genético para cada tipo.

7.1. Evolución del *fitness* promedio según tipo de individuos

En la Fig. 15 se muestra el *fitness* promedio por generación, producido por corridas típicas del algoritmo genético, según los tipos de individuo definidos. Si bien los resultados no son concluyentes, sí vale la pena decir que los individuos con actuador genérico (tipo 1 y 2) y coseno doble frecuencia (tipo 3), logran alcanzar un *fitness* más alto que el resto.

Por otra parte, puede notarse que los rangos manejados casi alcanzan el 0.7 (y no a 1, su cota superior). Esto se debe a que la función de *fitness* está definida como un producto de ciertos módulos: suponiendo que cada uno de ellos estuviera al 92 %, se tendría $0.92^5 = 0.659$.

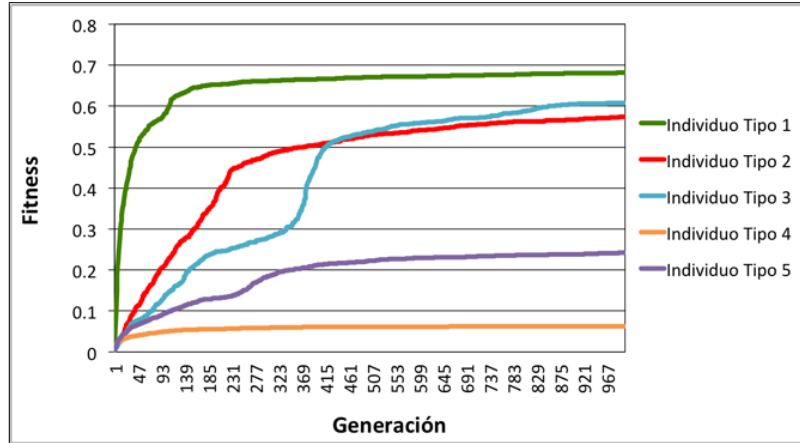


Figura 15: Evolución del *fitness* promedio según tipo de individuos

7.2. Velocidad según tipo de individuos

Seguidamente, se muestra para cada tipo de individuo, su velocidad instantánea de la pelvis a lo largo del tiempo.

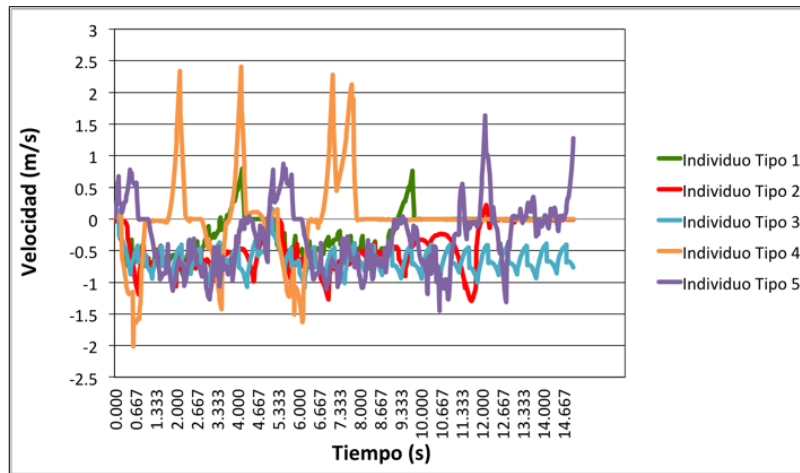


Figura 16: Velocidad instantánea según tipo de individuos

Como puede observarse en el caso de los individuos 4 y 5 (Fourier de orden 2 y 9, respectivamente), se producen picos altos y pronunciados en la velocidad. Eso repercute en que la velocidad media no sea $1.3 \frac{m}{s}$ (que es la velocidad objetivo), y ergo, en el módulo de velocidad del *fitness* (provocando que éste sea más bajo).

En el individuo de tipo 1, también se producen picos un poco menos pronunciados, pero luego de los 10 segundos, no se registra ninguna velocidad. Eso sucede porque el humanoide queda suspendido cuando intenta caer hacia atrás.

La velocidad del individuo de tipo 2 oscila de forma irregular, pero no tiene picos muy elevados. Por último, en el individuo de tipo 3, la velocidad oscila de forma cíclica, continuada y armoniosa (no hay picos altos).

7.3. Altura según tipo de individuos

Como puede identificarse en la Figura 17, la altura de los individuos es otra característica para diferenciarlos en su rendimiento.

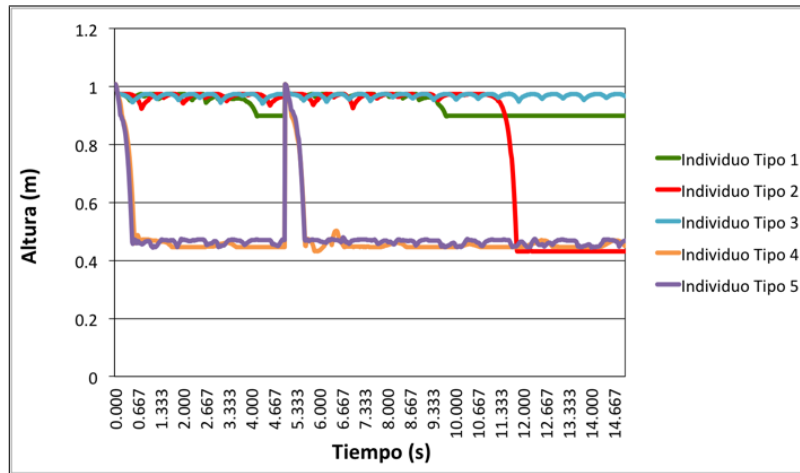


Figura 17: Altura (posición de la pelvis) según tipo de individuos

En efecto, los individuos de tipo 4 y 5 (que utilizan actuadores de Fourier de orden 2 y 9, respectivamente), son los que caen de forma más abrupta, y aunque intentan levantarse, vuelven a caer con la misma intensidad.

A su vez, el individuo de tipo 1 (actuador genérico sin función partida) mantiene su altura, hasta que a los 10 segundos queda suspendido a una altura levemente menor. Esto ocurre porque se cae para atrás, y las limitaciones de la cadera impiden que caiga. El individuo de tipo 2, con actuador genérico y función partida, logra mantener la altura, hasta que cae hacia adelante, en cuclillas. Considerando que cuando cae no vuelve a levantarse (ya que en ese movimiento suelen emplearse los brazos y manos), tiene un comportamiento similar a una caminata real.

En lo que respecta al individuo de tipo 3, que utiliza como actuador la función coseno doble frecuencia, mantiene su altura de forma constante, pero con oscilaciones leves y continuadas en el tiempo. De los cinco, es el que da mayor cantidad de pasos.

7.4. Comparación de tipo de individuos

Según lo visto, queda claro que los actuadores de Fourier no dieron buenos resultados. Asimismo, es necesario incluir la función partida (es decir, separar a la caminata en un “primer paso”, y el ciclo), ya que si bien su ausencia puede originar *fitness* más altos, genera inestabilidad a los pocos segundos (como en el individuo de tipo 1).

El individuo de tipo 3 logra un movimiento cíclico y repetitivo, que deriva en una caminata *ad infinitum*, con el costo de que sea “robótica” (muy parecida a lo observado con un *passive walker*).

7.5. Video

De los cinco individuos presentados, los que mejor pudieron realizar la caminata fueron tres, a saber: tipo 1 y 2 (genérico con y sin función partida), y tipo 3 (coseno doble frecuencia).

Para poder visualizar mejor su comportamiento, se acompañan los videos.

Individuo Tipo 1: <https://www.youtube.com/watch?v=J6j8UN1bg3w>

Individuo Tipo 2: <https://www.youtube.com/watch?v=t7jtbVL7eyY>

Individuo Tipo 3: <https://www.youtube.com/watch?v=z9QRa7wH1QI>

8. Conclusiones

El objetivo principal de este proyecto fue lograr producir la simulación y animación biomecánica de la caminata de un humano virtual. Para eso, se eligió el motor físico *Bullet Physics*, lo que requirió no solo aprender sobre su funcionamiento y los métodos físicos implementados, sino también realizar pruebas para verificar qué tan próximos eran el modelo fisico-matemático ideado y el utilizado por Bullet.

Una vez modelado el humanoide, se implementaron individuos con diferentes características (en especial, actuadores), para facilitar la comparación. En la sección 7 se verifica que las funciones utilizadas en los actuadores son decisivas para lograr una caminata.

Los actuadores que mejores resultados dieron, fueron aquellos en donde se empleaba dos frecuencias ω en vez de una.

Se comprobó que los individuos con actuadores genérico y coseno doble frecuencia son los que mejor caminan, manteniendo su altura por más tiempo y con *fitness* más alto. Sin embargo, son caminatas muy distintas: la del primero resulta ser más natural, pero se cae más rápido; mientras que la del segundo es más “robótica” (ya que parece un *passive walker*), pero más estable (no se cae ni se queda quieto).

Si hay que elegir como mejor a uno de los dos, habría que decidir entre realismo (el humanoide se cae, pero es más natural), y estabilidad (no se cae, pero parece un *passive walker*).

La función de *fitness* que se plantea (con sus módulos) indica propiedades ideales para una caminata, pero que no impliquen realismo en la misma. Y entonces dos individuos con *fitness* parecidos dieron como fruto caminatas muy distintas.

Por otra parte, el haber seguido la línea de investigación de los cuadrúpedos, si bien logró una caminata, no fue suficiente. Un sistema de control para manejar el equilibrio sería necesario.

Entre los trabajos a futuro para integrar a este proyecto, se encuentran lograr que la caminata se produzca en 3 dimensiones; y analizar el comportamiento de varios humanoides chocando e interactuando entre sí.

Se puede concluir que, la caminata de una persona, algo que parece simple y sencillo, muestra su verdadera complejidad cuando debe ser simulada por medio de actuadores aplicados a un conjunto de segmentos interconectados.

Referencias

- [1] Marek Wojtyra, *Multibody Simulation Model of Human Walking - Warsaw University of Technology, 2003*
- [2] Thomas Geijtenbeek, Michiel van de Panne y A. Frank van der Stappen, *Flexible Muscle-Based Locomotion for Bipedal Creatures, 2013*
- [3] Kevin Kenny, Máximo Videla y Axel Wassington, *Proyecto Final para la obtención del título Ingeniero en Informática: Simulación y animación de un cuadrúpedo virtual - ITBA, 2014*
- [4] Wikipedia: https://es.wikipedia.org/wiki/Physics_engine
- [5] Andreas Gerndt y otros, *An Evaluation of Open Source Physics Engines for Use in Virtual Reality Assembly Simulations. Fecha de publicación: 2012*
- [6] Tom Erez y otros, *Simulation Tools for Model-Based Robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX*
- [7] Sitio web de Bullet Physics: <http://www.bulletphysics.org/>
- [8] Erin Catto, *Iterative Dynamic with Temporal Coherence. Fecha de publicación: 2005*
- [9] Sitio web de GaLib: <http://lancet.mit.edu/ga/>
- [10] <http://www.exrx.net/Kinesiology/Segments.html>

Agradecimientos

A la Dra. Silvia Alicia Gómez, directora de la carrera de Ingeniería Informática, por su apoyo incondicional durante todos estos años. Asimismo, a todos los docentes que tuvimos a lo largo de la carrera, por la calidad y excelencia académica recibidas, que nos permitió realizar un proyecto final que integra los conocimientos y capacidades adquiridos. Y muy especialmente a nuestro tutor Dr. Daniel Ricardo Parisi, a quien también le agradecemos su valiosa colaboración y su experta ayuda en la preparación de este proyecto.