

1

2

PROYECTO FINAL

3

INGENIERÍA INFORMÁTICA - ITBA

4

5

SIMULACIÓN Y ANIMACIÓN BIOMECÁNICA DE UN HUMANOIDE

6

7

Alumnos:

8

Enzo Altamiranda Graterol

9

ealtamir@itba.edu.ar

10

Teresa Fontanella De Santis

11

tfontane@itba.edu.ar

12

Tomás Mehdi

13

tmehd@itba.edu.ar

14

Tutor:

15

Dr. Daniel Ricardo Parisi

16

Instituto Tecnológico de Buenos Aires - ITBA

17

Departamento de Ingeniería Informática

18

Abril 2016

Índice

19

20	Resumen	4
21	1 Introducción	4
22	2 Herramientas	5
23	2.1 Motor físico	5
24	2.1.1 Funcionamiento	5
25	2.1.2 Modelo de fricción y su verificación	5
26	2.1.2.1 Verificación del coeficiente de fricción	6
27	2.1.2.2 Verificación del coeficiente de restitución	10
28	2.1.3 Ventajas	11
29	2.1.4 Desventajas	11
30	2.2 Librería de algoritmos genéticos	12
31	2.3 Código fuente	12
32	3 Modelo utilizado	12
33	3.1 Composición física del humanoide	12
34	3.2 Articulaciones	13
35	4 Actuadores	14
36	4.1 Genérico	14
37	4.2 Fourier de orden 2	15
38	4.3 Fourier de orden 9	15
39	4.4 Coseno doble frecuencia	16
40	5 Condiciones iniciales y de contorno	17
41	5.1 Función partida	17
42	5.2 Fase sincronizada	17
43	6 Algoritmo genético	17
44	6.1 Individuo	17
45	6.1.1 Parámetros	18
46	6.1.2 Valores	18
47	6.1.3 Implementación de individuos	19
48	6.1.4 Constitución del cromosoma	19
49	6.2 Fitness	20
50	6.2.1 Altura	20
51	6.2.2 Velocidad	20
52	6.2.3 Dirección	20
53	6.2.4 Simetría	21
54	6.2.5 Pies abajo	21
55	6.3 Operadores del algoritmo	22
56	6.3.1 Métodos de selección	22
57	6.3.2 Métodos de cruce	22
58	6.3.3 Mutación	22

59	6.3.4 Otras características	22
60	7 Resultados obtenidos	22
61	7.1 Evolución del <i>fitness</i> según tipo de individuos	22
62	7.2 Velocidad según tipo de individuos	23
63	7.3 Altura según tipo de individuos	24
64	7.4 Comparación de tipo de individuos	24
65	7.5 Video	24
66	8 Conclusiones	25
67	Referencias	25

68

Resumen

69

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

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

77 1. Introducción

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

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

83 Una característica muy importante de este trabajo es que el humanoide no es fruto de una anima-
84 ción, sino un objeto compuesto de segmentos físicos, que interactúan entre sí y con el entorno,
85 por medio de las leyes físicas; agregando así realismo a la situación simulada. La otra propiedad
86 es que el bípedo aprenda por sí solo a caminar en línea recta, sin tener en su trayectoria ningún
87 obstáculo.

88 Este problema se puede abordar de diversas maneras, involucrando: redes neuronales¹, sistemas
89 de control² (*passive walkers*³)[1], algoritmos genéticos⁴[2], entre otras. Pero varios de ellos impli-
90 can modelos teóricos complejos de humanoide (considerando músculos con distintos materiales,
91 etc.).

92 En este proyecto se buscó aplicar algoritmos genéticos, y lograr la caminata usando un modelo
93 de humanoide basado en un conjunto de segmentos (cuerpos rígidos), unidos entre sí por articu-
94 laciones, cuyo desplazamiento depende de torques aplicados a dichos cuerpos, y sus parámetros
95 se ajustan a partir de la evolución del algoritmo.

96 El presente informe, describe y analiza pormenorizadamente: en la sección 2, las herramientas
97 aplicadas; en la 3, el modelo del humanoide utilizado; en la 4 y 5, los diferentes tipos de actuado-
98 res y funciones de partida y contorno; en la 6, el algoritmo genético; y en la 7 y 8, los resultados
99 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.

¹⁰⁰ 2. Herramientas

¹⁰¹ 2.1. Motor físico

¹⁰² Se llama motor físico o *physics engine* a un “*software* capaz de realizar simulaciones de
¹⁰³ ciertos sistemas físicos, como la dinámica del cuerpo rígido, el movimiento de un fluido y la
¹⁰⁴ elasticidad”[3].

¹⁰⁵ Actualmente, existen muchos motores físicos: ya sea de código propietario (*PhysX*, *Havok*), como
¹⁰⁶ *open-source* (*Bullet Physics*, *Box2D*, *Newton*, *OGRE*). Considerando análisis relacionados [4][5],
¹⁰⁷ y el hecho de que el espacio simulado fuese en 3D, se decidió que *Bullet Physics*[6] es el más
¹⁰⁸ idóneo. Está implementado en C++ y ha sido utilizado en varios juegos (*Grand Theft Auto IV*
¹⁰⁹ y *V*, etc); en los efectos especiales de películas (*Hancock*, *Bolt*, etc.); y proyectos científicos, como
¹¹⁰ la herramienta *open-source Tensegrity Robotics Toolkit* de la NASA⁵; entre otros.

¹¹¹ Si bien (como se verá más adelante) *Bullet* tiene problemas asociados con el coeficiente de restitu-
¹¹² ción, posee una muy buena *performance* en la detección de colisiones, la dinámica y la resolución
¹¹³ de *constraints*. Esto se debe, en parte, a diferentes algoritmos iterativos de orden lineal (donde
¹¹⁴ el más importante es *Sequential Impulse*), de *caching* y también a la utilización de un modelo
¹¹⁵ de fricción de Coulomb aproximado [7]. Además, el motor físico brinda la posibilidad de regular
¹¹⁶ la precisión requerida en estos cálculos (sin olvidar que, con iguales recursos, a mayor preci-
¹¹⁷ sión, mayor capacidad de cómputo requerida y, ergo, mayor tiempo). Dado que la construcción
¹¹⁸ del humanoide implica definir características y restricciones de movimiento de cada uno de sus
¹¹⁹ segmentos, lo antes mencionado fue crucial para la elección de *Bullet Physics* en este proyecto.

¹²⁰ 2.1.1. Funcionamiento

¹²¹ El motor físico se encarga de la simulación de cuerpos rígidos y la interacción entre los mismos.
¹²² En particular debe calcular el resultado de colisiones, arreglar el solapamiento de los cuerpos en
¹²³ el espacio de simulación, estimar las fuerzas producidas debido a la fricción, y mantener el
¹²⁴ cumplimiento de restricciones que puedan existir entre los cuerpos (por ejemplo, un vínculo para
¹²⁵ formar una articulación). Para lograrlo, *Bullet* modela, a partir de un conjunto de ecuaciones,
¹²⁶ las distintas restricciones que deben ser respetadas.

¹²⁷ Estos modelos reciben como dato la velocidad lineal y angular de cada objeto, y las fuerzas
¹²⁸ que actúan sobre los mismos. Dada esta información, se resuelve el sistema de ecuaciones, cuya
¹²⁹ solución representa las magnitudes de las fuerzas a accionar sobre el mismo, a fin de satisfacer
¹³⁰ todas las restricciones. Para encontrar esta solución, entran en juego los distintos métodos de
¹³¹ complejidad lineal mencionados en la subsección anterior.

¹³² Este procedimiento, se lleva a cabo en cada *timestep* de la simulación; donde un *timestep* es el
¹³³ intervalo de tiempo que transcurre entre un cálculo de magnitudes y otro. Mientras menor sea
¹³⁴ el *timestep*, el simulador será más preciso pero también consumirá más tiempo de cómputo.

¹³⁵ 2.1.2. Modelo de fricción y su verificación

¹³⁶ Hay reglas físicas relacionadas con el entorno y que son muy importantes para la caminata:
¹³⁷ el modelo de fricción, con sus respectivos coeficientes de fricción y restitución.
¹³⁸ En base a los modelos físico-matemáticos que representan a cada uno de los dos fenómenos en
¹³⁹ 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>

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

142 2.1.2.1. Verificación del coeficiente de fricción

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

147

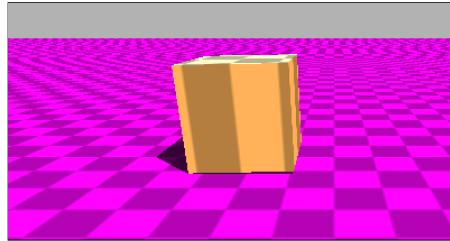


Figura 1: Visualización del sistema del cubo

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

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

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

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

154 donde $\mathbf{F}_N = mg$ es la fuerza normal que actúa sobre la caja de masa m por acción de la gravedad
 155 $g = 10 \frac{m}{s^2}$, y μ_d es el coeficiente de fricción dinámico.

156

157 Finalmente, se obtiene la aceleración:

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

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

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

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

163 Seguidamente, se muestran los gráficos obtenidos al correr los experimentos numéricos con los
 164 siguientes valores: $v_i = \{1, 3 y 10\} \frac{m}{s}$ y $\mu_d = \{0.25, 0.50 y 0.75\}$. Se compara la distancia en el
 165 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.



Figura 2: Resultados logrados de simular el sistema descripto en Fig. 1, usando $v_i = 10 \frac{m}{s}$ y:
 (a) $\mu_d = 0.25$, (b) $\mu_d = 0.50$, y (c) $\mu_d = 0.75$



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

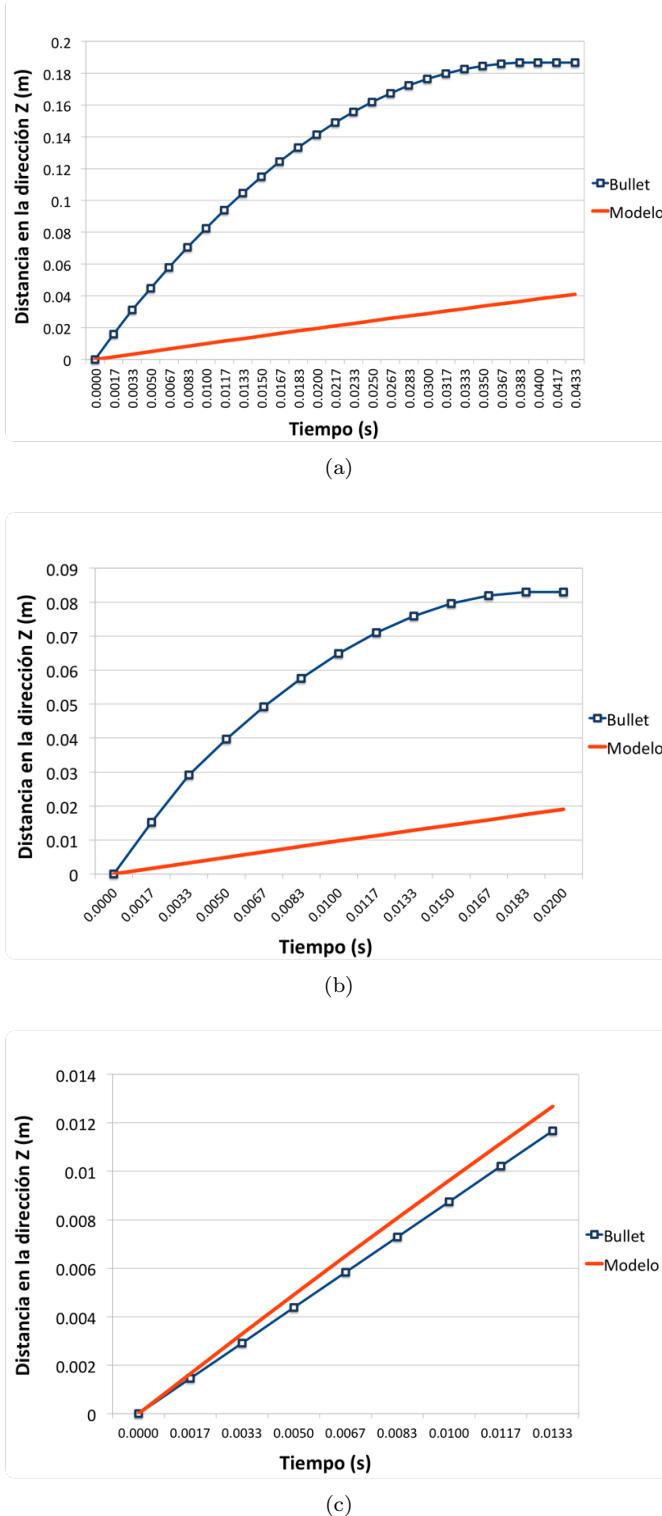


Figura 4: Resultados logrados de simular el sistema descripto en Fig. 1, usando $v_i = 1 \frac{m}{s}$ y:

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

¹⁶⁶ Las Fig. 2 y 3 demuestran que los resultados de las pruebas fueron favorables. Los valores
¹⁶⁷ obtenidos a partir de la simulación se corresponden (en mayor o menor medida) con los alcanzados
¹⁶⁸ 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 ($g = 0 \frac{\text{m}}{\text{s}^2}$). 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 $\mu = 0.75$.

190

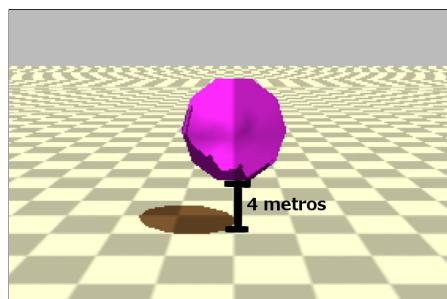


Figura 5: Visualización del sistema de la esfera

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:

195

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

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

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

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

209 2.1.3. Ventajas

210 Las ventajas del motor físico son:

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

214 2.1.4. Desventajas

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

- 216 – Documentación poco clara y desordenada.
- 217 – Debido a que la física se aproxima usando métodos numéricos que contienen error, las simu-
 218 laciones son no determinísticas.

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

222 **2.2. Librería de algoritmos genéticos**

223 Se utilizó la conocida librería de algoritmos genéticos para C++ GaLib, desarrollada por
224 Matthew Wall del MIT [8].

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

227 **2.3. Código fuente**

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

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

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

238 **3. Modelo utilizado**

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

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

249 **3.1. Composición física del humanoide**

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

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

254 A los fines de este proyecto (lograr la caminata), sólo se consideraron pelvis y miembro inferior
255 (Fig. 6).

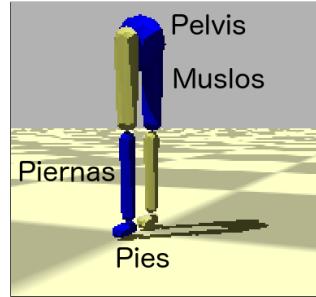


Figura 6: Segmentos del humanoide

256 En la Tabla 2 se presenta la composición física de cada uno de los segmentos del humano virtual,
257 de acuerdo a la biomecánica [9].

Segmento		Cantidad	Forma	Largo (en m)	Masa (en kg)	Uniones
Miembro inferior	Pelvis	1	Esférica	0.08655	9.9718	Cadera
	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

258 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)).

259 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.
260 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.

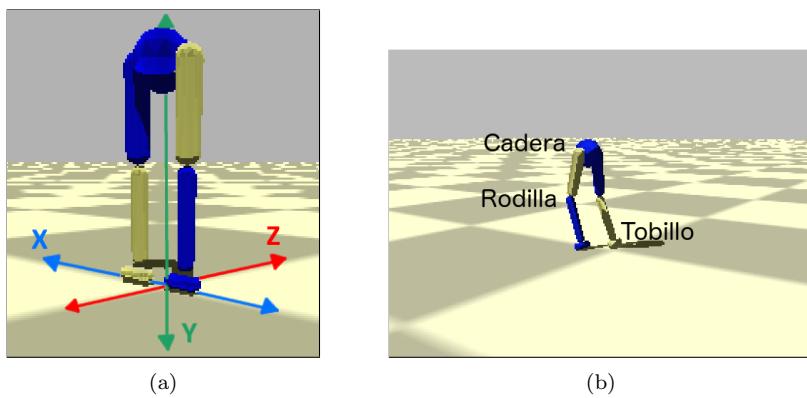


Figura 7: Humanoide diseñado: (a) ejes, y (b) articulaciones

Articulación	Ángulo mínimo	Ángulo máximo
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.

284

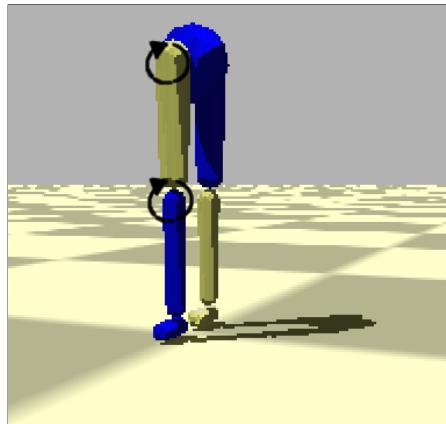


Figura 8: Aplicación de los actuadores en los segmentos del bípedo

4.1. Genérico

286 Es el actuador más sencillo, tanto matemática como computacionalmente.

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

287 donde $f(t)$ es la función del actuador evaluada en el instante de tiempo t , A_1 y A_2 son amplitudes, 288 ω_1 y ω_2 son frecuencias (en $\frac{1}{s}$), ϕ es la fase en radianes, y C es un término independiente (ver 289 Fig. 9 (a) y (b)).

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

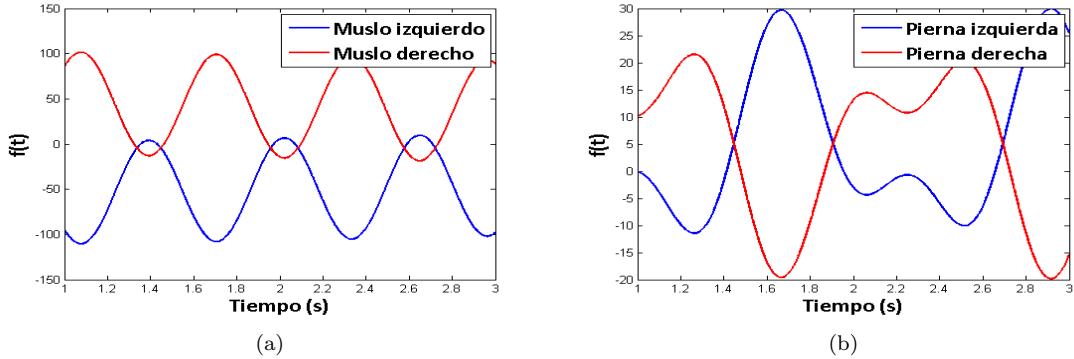


Figura 9: Ejemplo de actuador genérico aplicado en: (a) muslo, y (b) pierna

292 4.2. Fourier de orden 2

293 Este actuador utiliza una serie de Fourier de dos términos.

$$f(t) = A_1 \sin(\omega t + \phi) + B_1 \cos(\omega t + \phi) + A_2 \sin(2\omega t + \phi) + B_2 \cos(2\omega t + \phi) + C \quad (8)$$

294 donde A_1, A_2, B_1 y B_2 son amplitudes y ω es frecuencia (en $\frac{1}{s}$) (ver Fig. 10 (a) y (b)).

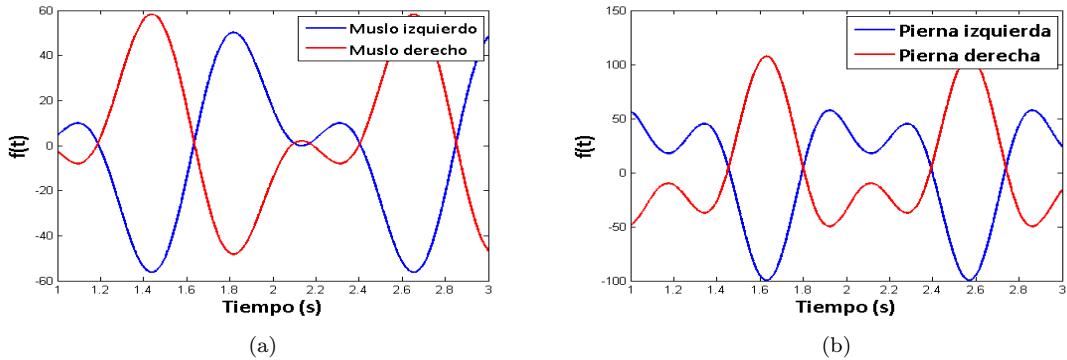


Figura 10: Ejemplo de actuador fourier de orden 2 aplicado en: (a) muslo, y (b) pierna

295 4.3. Fourier de orden 9

296 Es una extensión del actuador anterior, pero con 9 términos. Por ser de mayor grado, brinda
297 una mayor precisión. Sin embargo, es más difícil de manejar computacionalmente; y, además, que
298 sea más preciso no garantiza que con él se pueda lograr una buena caminata (ver Fig. 11 (a) y (b)).

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

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

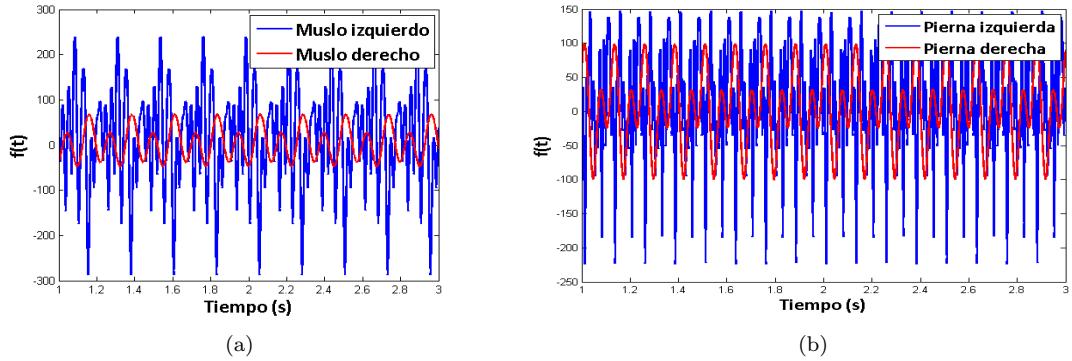


Figura 11: Ejemplo de actuador fourier de orden 9 aplicado en: (a) muslo, y (b) pierna

4.4. Coseno doble frecuencia

Basada en [10], 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. 12 (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 $\psi(t)$ (ec. (10)) 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 $f(t)$ (ec. (12)), que utiliza medio ciclo con frecuencia ω_1 y medio ciclo con frecuencia ω_2 .

$$\psi(t) = t + \phi - \left\lfloor \frac{t + \phi}{\pi/\omega_1 + \pi/\omega_2} \right\rfloor (\pi/\omega_1 + \pi/\omega_2) \quad \psi : \mathbb{R} \rightarrow \left[0, \frac{2\pi}{\omega}\right] \quad (10)$$

$$\omega = \frac{2\omega_1\omega_2}{\omega_1 + \omega_2} \quad (11)$$

$$f(t) = \begin{cases} A \cos(\omega_1 \psi(t)) + C & \text{si } \omega_1 \psi(t) < \pi \\ A \cos(\omega_2(\psi(t) - (\pi/\omega_1) + (\pi/\omega_2))) + C & \text{en otro caso} \end{cases} \quad (12)$$

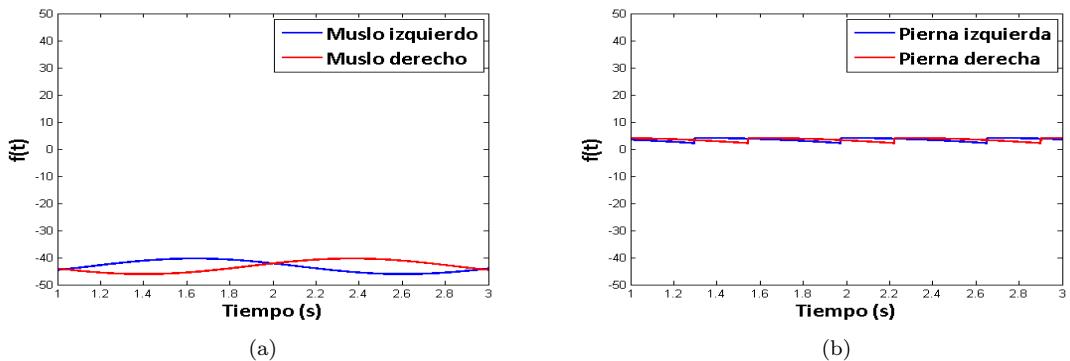


Figura 12: Ejemplo de actuador coseno de doble frecuencia aplicado en: (a) muslo, y (b) pierna

³¹⁴ 5. Condiciones iniciales y de contorno

³¹⁵ Como las funciones periódicas señaladas en los actuadores, no fueron suficientes para lograr
³¹⁶ la caminata, se le adosaron las funciones que se detallan seguidamente.

³¹⁷ 5.1. Función partida

³¹⁸ El andar del humanoide es cíclico. Sin embargo, por la posición inicial del individuo, se requiere
³¹⁹ para el tiempo del primer paso, una función distinta a la del resto de la caminata. El tipo de
³²⁰ función puede ser cualquiera de los actuadores vistos anteriormente (pero no necesariamente con
³²¹ los mismos valores de amplitud, frecuencia y fase asignados a las piernas). Empero, se utilizó la
³²² función vista en el actuador genérico.
³²³ Por otra parte, para simplificar el modelo, se decidió que el tiempo considerado para el primer
³²⁴ paso sea fijo, y de 0.7 segundos. Dicho valor fue extraído de forma experimental.

³²⁵ 5.2. Fase sincronizada

³²⁶ En una caminata, las piernas deben guardar simetría: mientras una va hacia adelante, la otra
³²⁷ va hacia atrás (y viceversa). Esto, de acuerdo con los actuadores definidos en la sección anterior,
³²⁸ implica que las funciones de movimiento de cada pierna estén desfasadas en medio ciclo ($\frac{\pi}{2}$):

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

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

³³⁰ siendo $f(t)$ la función de movimiento (o actuador) en el momento t , y f_i y f_d las funciones de la
³³¹ pierna izquierda y derecha, respectivamente.

³³² 6. Algoritmo genético

³³³ Los algoritmos genéticos son un conjunto de técnicas de optimización, inspiradas en la teoría
³³⁴ darwiniana de la selección natural. La entidad a optimizar constituye un individuo dentro de
³³⁵ una población; que puede ser cruzado con otros individuos para obtener un “hijo”, que presenta
³³⁶ características de sus “padres”. A través de una función de *fitness* se decide cuáles de ellos son
³³⁷ los más “aptos”, que luego pasarán a una nueva generación de individuos. Este proceso se repite
³³⁸ hasta obtener un individuo lo suficientemente apto, que representa a la entidad optimizada.
³³⁹ El objetivo de este proyecto es optimizar la caminata del humanoide. Ergo, en el algoritmo
³⁴⁰ genético aplicado, el individuo está definido por rasgos del bípedo, y la función de *fitness* se
³⁴¹ focaliza en mejorar diversos aspectos de dicha caminata.

³⁴² 6.1. Individuo

³⁴³ La información genética de cada individuo, está definida como un vector y se compone de
³⁴⁴ dos partes, ubicadas en forma contigua: función partida (optativa) y los parámetros asociados a
³⁴⁵ los actuadores (obligatorios).

346 6.1.1. Parámetros

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

350 6.1.2. Valores

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

354

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
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
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

Tabla 4: Rango de valores que puede tomar cada gen, según el tipo de actuador

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

355 6.1.3. Implementación de individuos

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

358

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

Tabla 6: Tipo de individuos

359 6.1.4. Constitución del cromosoma

360 Los distintos actuadores y la función partida tienen los parámetros presentados en la sección
 361 6.1.1. Sus respectivas cantidades pueden verse en la Tabla 7.

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

367

	Parámetro			
	Amplitud	Frecuencia	Fase	Término independiente
Actuador	Genérico	2	2	1
	Fourier de orden 2	4	1	1
	Fourier de orden 9	18	1	1
	Coseno doble frecuencia	1	2	1
Función partida		2	2	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	12	12	16	44
	Frecuencia	2	6	6	6
	Fase	2	6	6	6
	Término independiente	2	6	6	6
Totales		18	30	34	62
					28

Tabla 8: Cantidad de parámetros según tipo de individuos

368 6.2. Fitness

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

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

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

378

379 6.2.1. Altura

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

381

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

382

383 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
 384 tiempo t_n , N la cantidad pasos de simulación y C una constante $C = 5$.

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

390

391 6.2.2. Velocidad

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

394

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

395

396 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
 397 tiempo t_n , y V_O la velocidad objetivo en el eje Z (el eje de la caminata).

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

401 6.2.3. Dirección

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

404

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

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

407 El producto escalar entre los versores responde a la Similitud Coseno: $\cos \theta = \frac{\mathbf{A} \cdot \mathbf{B}}{AB}$, donde \mathbf{A} y
408 \mathbf{B} son vectores que no se encuentran normalizados, y θ es el ángulo formado entre ellos. Así, si
409 $\cos \theta = 1$, significa que los vectores están paralelos entre sí (que es el efecto buscado en el caso
410 de la dirección).

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

414 6.2.4. Simetría

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

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

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

422

423 donde lf_z y lf_z es la distancia desde el pie izquierdo hasta la pelvis en los ejes X y Z, respecti-
424 vamente; y en donde rf_x y rf_z es lo mismo, pero para el pie derecho.

425 6.2.5. Pies abajo

426 Con los módulos señalados anteriormente, se busca que el humanoide camine con una ve-
427 locidad y dirección determinadas, que no se caiga y que mantenga simetría mientras ejecuta sus
428 movimientos. Pero, todo esto daría, en el mejor de los casos, una caminata estilo “estrella”.

429 Sin embargo, una característica fundamental en una caminata normal es que las piernas (ergo,
430 los pies también) no sobrepasen la cadera. Si bien ésta es una propiedad negativa (expresa lo que
431 no debe tener una caminata), y se puede correr el riesgo de restringir demasiado, su ausencia da
432 resultados peores.

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

433

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

438

439 6.3. Operadores del algoritmo

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

443 6.3.1. Métodos de selección

444 De la vasta cantidad de métodos de selección que existen, se utilizaron: **Elite** (en donde
445 se elige el individuo con mayor aptitud de la población); y **Roulette** (método estocástico, que
446 selecciona un individuo de la población total al azar, con una probabilidad proporcional a su
447 *fitness*).

448 6.3.2. Métodos de crusa

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

452 La probabilidad de que este proceso ocurra es de 0.9.

453 6.3.3. Mutación

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

457 Dicha probabilidad de mutación es de 0.3.

458 6.3.4. Otras características

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

461 7. Resultados obtenidos

462 Considerando los individuos definidos en el punto 6.1.3, se realizaron pruebas, corriendo
463 el algoritmo genético, y evaluando el resultado alcanzado posteriormente (ya sea numérica o
464 gráficamente). A continuación se analizan distintos aspectos relevantes.

465 7.1. Evolución del *fitness* según tipo de individuos

466 Los individuos que usan actuadores de Fourier son los que producen *fitness* más bajos; el de
467 orden 2 (individuo de tipo 3) se estanca a las pocas generaciones (al igual que el de tipo 1); el
468 *fitness* del individuo con actuadores Fourier de orden 9 se “ameseta” progresivamente (después
469 de 500 generaciones).

470 El individuo de tipo 1 es el individuo con *fitness* más alto, pero que se obtuvo a las pocas
471 generaciones (es decir que es un máximo local). El individuo de tipo 2 (con actuadores genéricos
472 y función partida) va aumentando su *fitness* progresivamente, aunque sin superar al de tipo 1.

473 Por otra parte, el individuo de tipo 5 (con actuadores coseno doble frecuencia), va mejorando su
474 *fitness* paulatinamente, lo que impide estancarse en un máximo local. Además, tiene el segundo

475 mejor *fitness*.

476 Puede notarse que los rangos manejados en la Fig. 13 casi alcanzan el 0.7 (y no a 1, su cota
477 superior). Esto se debe a que la función de *fitness* está definida como un producto de ciertos
478 módulos: suponiendo que cada uno de ellos estuviera al 92 %, se tendría $0.92^5 = 0.659$.

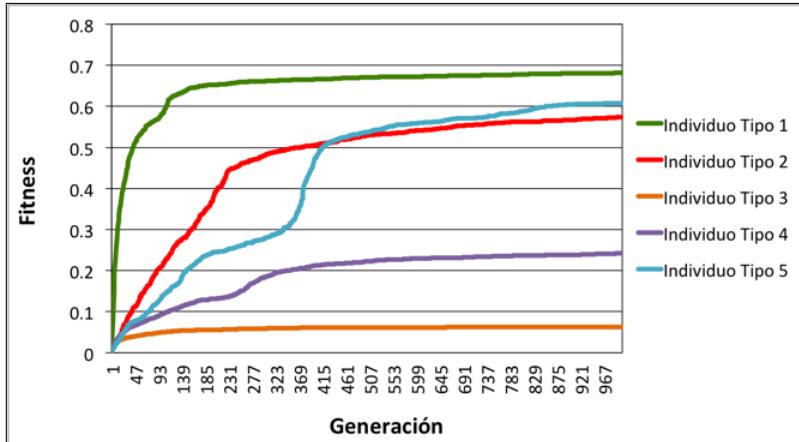


Figura 13: Evolución del *fitness* según tipo de individuos

479 7.2. Velocidad según tipo de individuos

480 Seguidamente, se muestra para cada individuo, su velocidad instantánea a lo largo del tiempo.

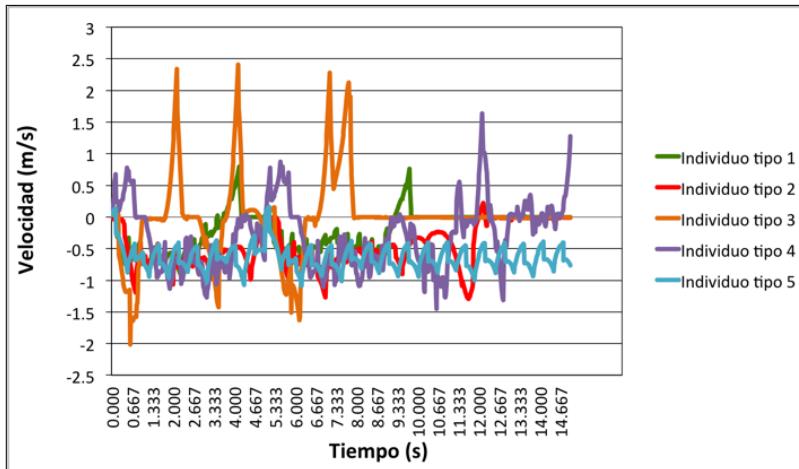


Figura 14: Velocidad según tipo de individuos

481 Como puede observarse en el caso de los individuos 3 y 4 (Fourier de orden 2 y 9, respectivamente),
482 se producen picos altos y pronunciados en la velocidad. Eso repercute en que la velocidad media no
483 sea $1.3 \frac{m}{s}$ (que es la velocidad objetivo), y ergo, en el módulo de velocidad del *fitness* (provocando
484 que éste sea más bajo).

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

488 La velocidad del individuo de tipo 2 oscila de forma irregular, pero no tiene picos muy elevados.
489 Por último, en el individuo de tipo 5, la velocidad oscila de forma cíclica, continuada y armoniosa
490 (no hay picos altos).

491 7.3. Altura según tipo de individuos

492 Como puede identificarse en la Figura 15, la altura de los individuos es otra característica
 493 para diferenciarlos en su rendimiento.

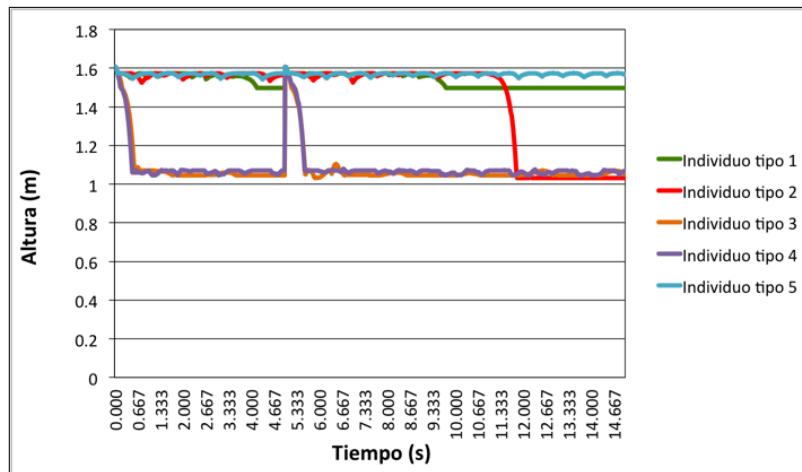


Figura 15: Altura (posición de la pelvis) según tipo de individuos

494 En efecto, los individuos de tipo 3 y 4 (que utilizan actuadores de Fourier de orden 2 y 9,
 495 respectivamente), son los que caen de forma más abrupta, y aunque intentan levantarse, vuelven
 496 a caer con la misma intensidad.

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

503 En lo que respecta al individuo de tipo 5, que utiliza como actuadores la función coseno doble
 504 frecuencia, mantiene su altura de forma constante, pero con oscilaciones leves y continuadas en
 505 el tiempo. De los cinco, es el que da mayor cantidad de pasos.

506 7.4. Comparación de tipo de individuos

507 En base a lo visto anteriormente, queda claro que los actuadores de Fourier no dieron buenos
 508 resultados tanto en fitness, como en altura y velocidad.

509 Por otra parte, es necesario incluir la función partida (es decir, separar a la caminata en un
 510 “primer paso”, y el ciclo), ya que si bien su ausencia puede originar fitness más altos, genera
 511 inestabilidad a los pocos segundos (como en el individuo de tipo 1).

512 El individuo de tipo 5 logra un movimiento cíclico y repetitivo, que deriva en una caminata *ad infinitum*, pero con el costo de que sea “robótica”(muy parecida a lo visto con un *passive walker*).

514 7.5. Video

515 Para una mejor visualización de los distintos individuos obtenidos, y de su evolución a lo
 516 largo de las generaciones del algoritmo genético, se acompaña video.

517 [Acá va el link del video](#)

518 8. Conclusiones

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

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

527 Los actuadores que mejores resultados dieron, fueron aquellos en donde se empleaba dos frecuen-
528 cias ω en vez de una.

529 Lo ocurrido con el individuo de tipo 1 (genérico y sin función partida), que es el que tiene un
530 *fitness* más alto, aunque no produce una caminata acorde, posiblemente se deba a que el tiempo
531 de simulación empleado en el algoritmo genético, fuera corto.

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

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

540 Si hay que elegir como mejor a uno de los dos, habría que decidir entre realismo (el humanoide
541 se cae, pero es más natural), y estabilidad (no se cae nunca, pero parece un *passive walker*).
542 Entre los trabajos a futuro para integrar a este proyecto, se encuentran lograr que la caminata
543 se produzca en 3 dimensiones; y analizar el comportamiento de varios humanoides chocando e
544 interactuando entre sí.
545 Se puede concluir que, la caminata de una persona, algo que parece simple y sencillo, muestra
546 su verdadera complejidad cuando debe ser simulada por medio de actuadores aplicados a un
547 conjunto de segmentos interconectados.

548 Referencias

- 549 [1] Marek Wojtyra, *Multibody Simulation Model of Human Walking - Warsaw University of*
550 *Technology, 2003*
- 551 [2] Thomas Geijtenbeek, Michiel van de Panne y A. Frank van der Stappen, *Flexible Muscle-
552 Based Locomotion for Bipedal Creatures, 2013*
- 553 [3] Wikipedia: https://es.wikipedia.org/wiki/Physics_engine
- 554 [4] Andreas Gerndt y otros, *An Evaluation of Open Source Physics Engines for Use in Virtual*
555 *Reality Assembly Simulations. Fecha de publicación: 2012*
- 556 [5] Tom Erez y otros, *Simulation Tools for Model-Based Robotics: Comparison of Bullet, Havok,
557 MuJoCo, ODE and PhysX*

- 558 [6] Sitio web de Bullet Physics: <http://www.bulletphysics.org/>
- 559 [7] Erin Catto, *Iterative Dynamic with Temporal Coherence*. Fecha de publicación: 2005
- 560 [8] Sitio web de GaLib: <http://lancet.mit.edu/ga/>
- 561 [9] <http://www.exrx.net/Kinesiology/Segments.html>
- 562 [10] Kevin Kenny, Máximo Videla y Axel Wassington, *Proyecto Final para la obtención del título:*
563 *Ingeniero en Informática - ITBA, 2014*