

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

Marzo 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 utilizado y su verificación	5
26	2.1.3 Ventajas	11
27	2.1.4 Desventajas	11
28	2.2 Librería de Algoritmos Genéticos	12
29	2.3 Código Fuente	12
30	3 Modelo Utilizado	12
31	3.1 Composición Física del Humanoide	13
32	3.2 Articulaciones	13
33	4 Actuadores	14
34	4.1 Genérico	14
35	4.2 Fourier	15
36	4.3 Extra Fourier	15
37	4.4 Doble coseno	15
38	5 Condiciones iniciales y de contorno	15
39	5.1 Función Partida	15
40	5.2 Fase sincronizada	16
41	5.3 Riel	16
42	6 Algoritmo Genético	16
43	6.1 Individuo	16
44	6.1.1 Parámetros	16
45	6.1.2 Valores	17
46	6.1.3 Implementaciones de individuos	17
47	6.1.4 Constitución del cromosoma	18
48	6.2 Fitness	18
49	6.2.1 Altura	18
50	6.2.2 Velocidad	19
51	6.2.3 Dirección	19
52	6.2.4 Simetría	19
53	6.2.5 Pies abajo	20
54	6.3 Parámetros del Algoritmo	20
55	6.3.1 Métodos de selección	20
56	6.3.2 Métodos de cruce	20
57	6.3.3 Mutación	20

58	7 Resultados Obtenidos	21
59	7.1 Evolución del <i>fitness</i> según tipo de individuo	21
60	7.2 Velocidad según tipo de individuo	21
61	7.3 Altura según tipo de individuo	22
62	7.4 Comparación de tipos de individuo	22
63	7.5 Video	22
64	8 Conclusiones	22
65	Referencias	23

66

Resumen

67

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

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

75

1. Introducción

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

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

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

86 Este problema se puede abordar de diversas maneras, involucrando: redes neuronales¹, sistemas
87 de control (*passive walkers*²), algoritmos genéticos, entre otras. Pero varios de ellos implican
88 modelos teóricos complejos de humanoide (considerando músculos con distintos materiales, etc.).

89 En este trabajo se buscó aplicar algoritmos genéticos, y lograr la caminata usando un modelo de
90 humanoide basado en cuerpos rígidos, unidos por articulaciones, y cuyo desplazamiento depende
91 de torques aplicados a dichos cuerpos.

92 El presente informe, describe y analiza pormenorizadamente: en la sección 2, las herramientas
93 aplicadas; en la 3, el modelo del humanoide utilizado; en la 4 y 5, los diferentes tipos de actuadores
94 y funciones de partida y contorno; en la 6, el algoritmo genético; y en la 7 y 8, los resultados
95 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 *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).

⁹⁶ 2. Herramientas

⁹⁷ 2.1. Motor Físico

⁹⁸ Se le llama motor físico o *physics engine* a un “*software* capaz de realizar simulaciones de
⁹⁹ ciertos sistemas físicos, como la dinámica del cuerpo rígido, el movimiento de un fluido y la
¹⁰⁰ elasticidad”[1].

¹⁰¹ Actualmente, existen muchos motores físicos: ya sea de código propietario (PhysX, Havok), como
¹⁰² *open-source* (*Bullet Physics*, *Box2D*, Newton, OGRE). Considerando análisis relacionados [2][3],
¹⁰³ y la necesidad de que el espacio simulado fuese en 3D, se decidió que *Bullet Physics*[4] es el más
¹⁰⁴ idóneo. Está implementado en C++ y ha sido utilizado en varios juegos (*Grand Theft Auto IV* y
¹⁰⁵ V, etc.); en los efectos especiales de películas (Hancock, Bolt, etc.); y proyectos científicos, como
¹⁰⁶ la herramienta *open-source Tensegrity Robotics Toolkit* de la NASA³; entre otros.

¹⁰⁷ Si bien (como se verá más adelante) *Bullet* tiene problemas asociados con el coeficiente de restitu-
¹⁰⁸ ción, posee una muy buena *performance* en la detección de colisiones, la dinámica y la resolución
¹⁰⁹ de *constraints*. Esto se debe, en parte, a diferentes algoritmos iterativos de orden lineal (donde
¹¹⁰ el más importante es *Sequential Impulse*), de *caching* y también a la utilización de un modelo
¹¹¹ de fricción de Coulomb aproximado [5]. Además, el motor físico brinda la posibilidad de regular
¹¹² la precisión requerida en estos cálculos (sin olvidar que, con iguales recursos, a mayor precisión,
¹¹³ mayor capacidad de cómputo requerida y, ergo, mayor tiempo). Dado que la construcción del hu-
¹¹⁴ manoide implica definir características y restricciones de movimiento de cada una de sus partes,
¹¹⁵ lo antes mencionado fue crucial para la elección de *Bullet Physics* en este proyecto.

¹¹⁶ 2.1.1. Funcionamiento

¹¹⁷ El motor físico se encarga de la simulación de cuerpos rígidos y la interacción entre los mismos.
¹¹⁸ En particular debe: calcular el resultado de colisiones, arreglar el solapamiento de los cuerpos
¹¹⁹ en el espacio de simulación, estimar las fuerzas producidas debido a la fricción, y mantener el
¹²⁰ cumplimiento de restricciones que puedan existir entre los cuerpos (por ejemplo, un vínculo para
¹²¹ formar una articulación). Para lograrlo, *Bullet* modela, a partir de un conjunto de ecuaciones,
¹²² las distintas restricciones que deben ser respetadas.

¹²³ Estos modelos reciben como dato la velocidad lineal y angular de cada objeto, y las fuerzas
¹²⁴ que actúan sobre los mismos. Dada esta información, se resuelve el sistema de ecuaciones, cuya
¹²⁵ solución representa las magnitudes de las fuerzas a accionar sobre el mismo, a fin de satisfacer
¹²⁶ todas las restricciones. Para encontrar esta solución, entran en juego los distintos métodos de
¹²⁷ complejidad lineal mencionados en la subsección anterior.

¹²⁸ Este procedimiento, se lleva a cabo en cada *timestep* de la simulación; donde un *timestep* es el
¹²⁹ intervalo de tiempo que transcurre entre un cálculo de magnitudes y otro. Mientras menor sea
¹³⁰ el *timestep*, el simulador será más preciso pero también consumirá más tiempo de cómputo.

¹³¹ 2.1.2. Modelo de fricción utilizado y su verificación

¹³² Hay reglas físicas relacionadas con el entorno y que son muy importantes para la caminata:
¹³³ el modelo de fricción, con sus respectivos coeficientes de fricción y restitución.

¹³⁴
¹³⁵
³<http://bulletphysics.org/Bullet/phpBB3/viewtopic.php?f=17&t=9978>

136 En base a los modelos físico-matemáticos utilizados en los dos fenómenos en cuestión (y que
 137 se explicarán a continuación), y pensando en posibles futuras simulaciones de varios humanoides
 138 chocando e interactuando entre sí; se llevaron a cabo dos experimentos para verificar que
 139 estuvieran en concordancia con los datos arrojados por *Bullet*.

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

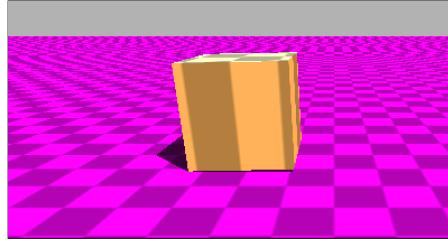


Figura 1: RENOMBRAR FIGURA

144 Para este experimento se utilizó el modelo matemático que representa la posición del cuerpo
 145 en el eje horizontal en función del tiempo, representado por la siguiente ecuación:

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

146
 147 En este caso, el cuerpo empieza su movimiento en el origen, por lo tanto la posición inicial
 148 (x_i) es cero. Debido a la fricción entre el cuerpo y el suelo, se genera una fuerza de rozamiento
 149 F_{μ_d} (ec. (2)) en la misma dirección que la velocidad del sólido y en sentido contrario.

150

$$(F_{\mu_d} = \mu_d F_N) \quad (2)$$

151 donde $F_N = mg$ es la fuerza normal que actúa sobre la caja por acción de la gravedad g , y
 152 μ_d es el coeficiente de fricción dinámico.

153
 154 Finalmente, se obtiene la aceleración:

$$a = \frac{F_{\mu_d}}{m} = \frac{-\mu_d F_N}{m} = \frac{-\mu_d mg}{m} = (\mu_d mg) \quad (3)$$

155 Considerando las ec. (1) y (3), se puede obtener el modelo matemático que predice el movi-
 156 miento de la caja:

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

157
 158 Los resultados obtenidos (ver Fig. 2, 3 y 4) exponen que posiblemente *Bullet* utilice el modelo
 159 antes expuesto a la hora de simular. No obstante, vale aclarar que, cuanto mayor sea el paso
 160 de simulación (o *timestep*) empleado, mayor es la discrepancia entre la simulación y el modelo,
 161 posiblemente porque la precisión es menor y eso lleva a cometer un error mayor.

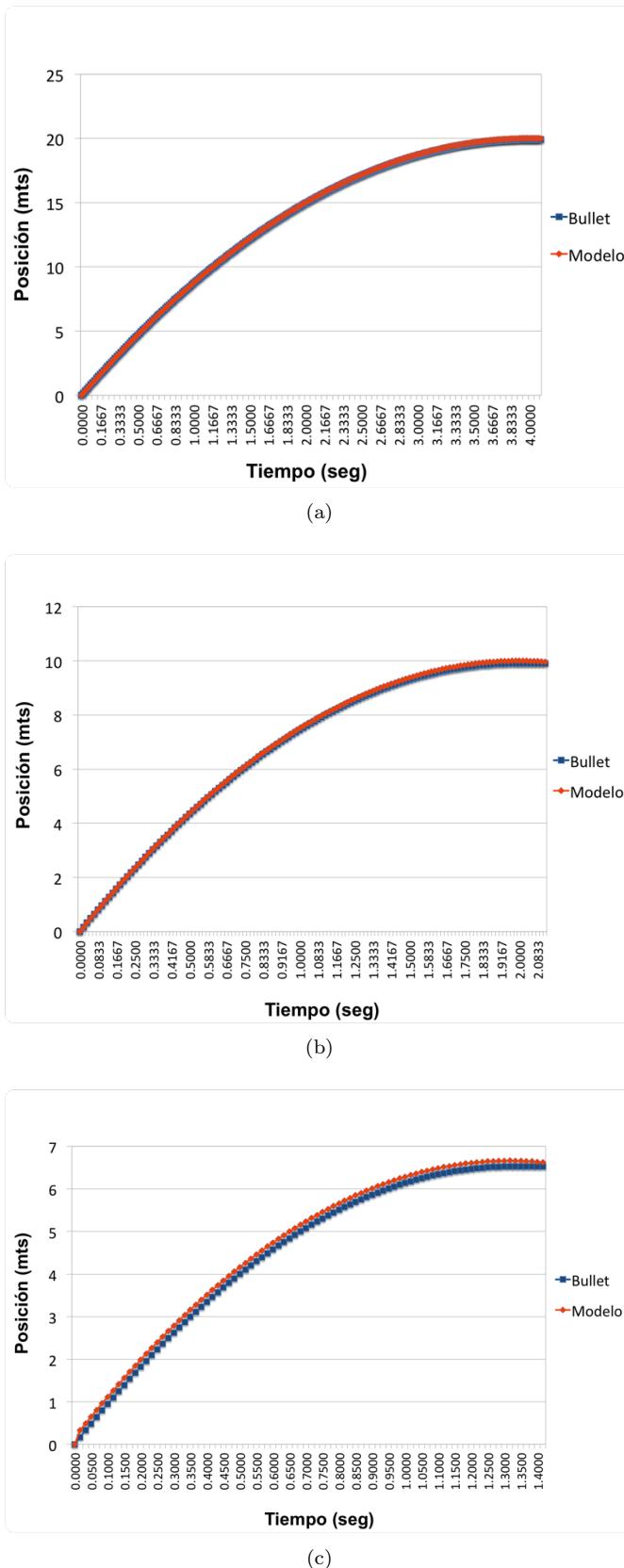


Figura 2: $v_i = 10 \frac{m}{s}$: (a) $\mu_d = 0.25$, (b) $\mu_d = 0.50$, y (c) $\mu_d = 0.75$

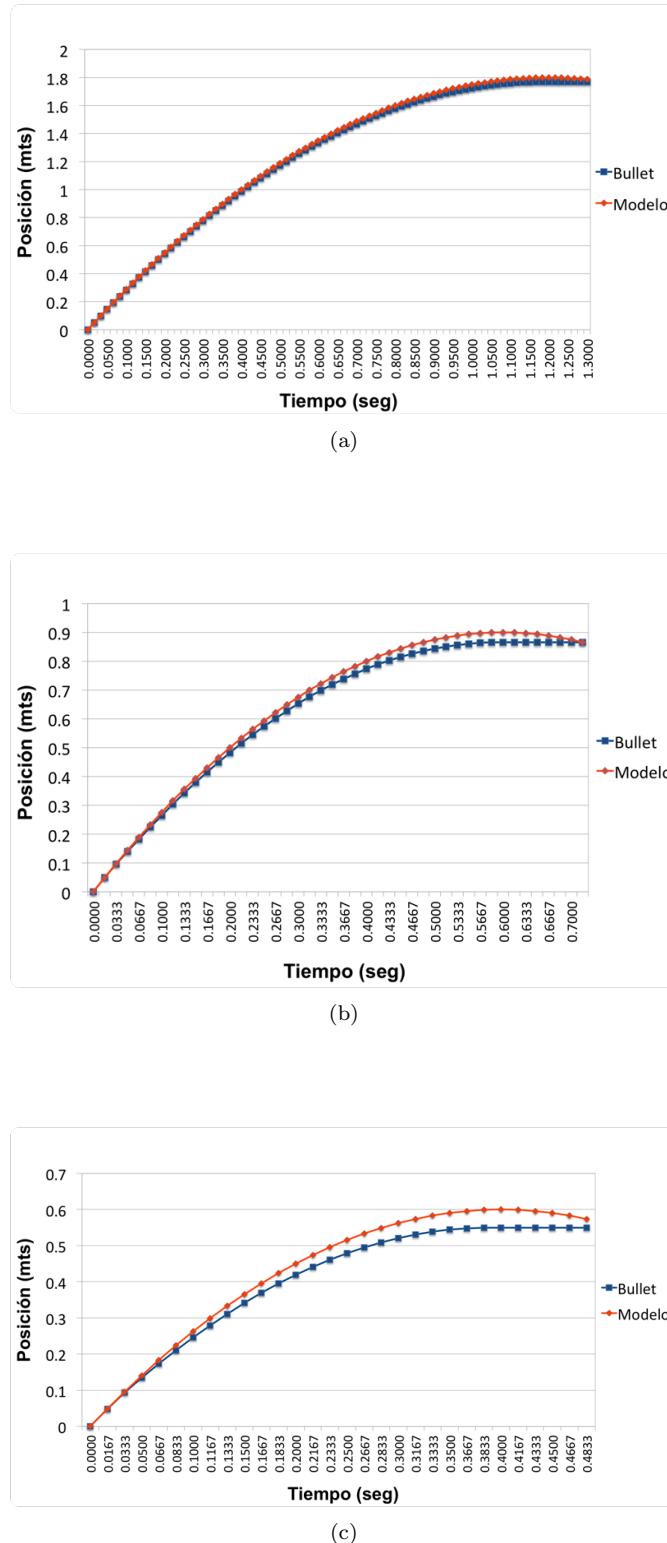


Figura 3: $v_i = 3 \frac{m}{s}$: (a) $\mu_d = 0.25$, (b) $\mu_d = 0.50$, y (c) $\mu_d = 0.75$

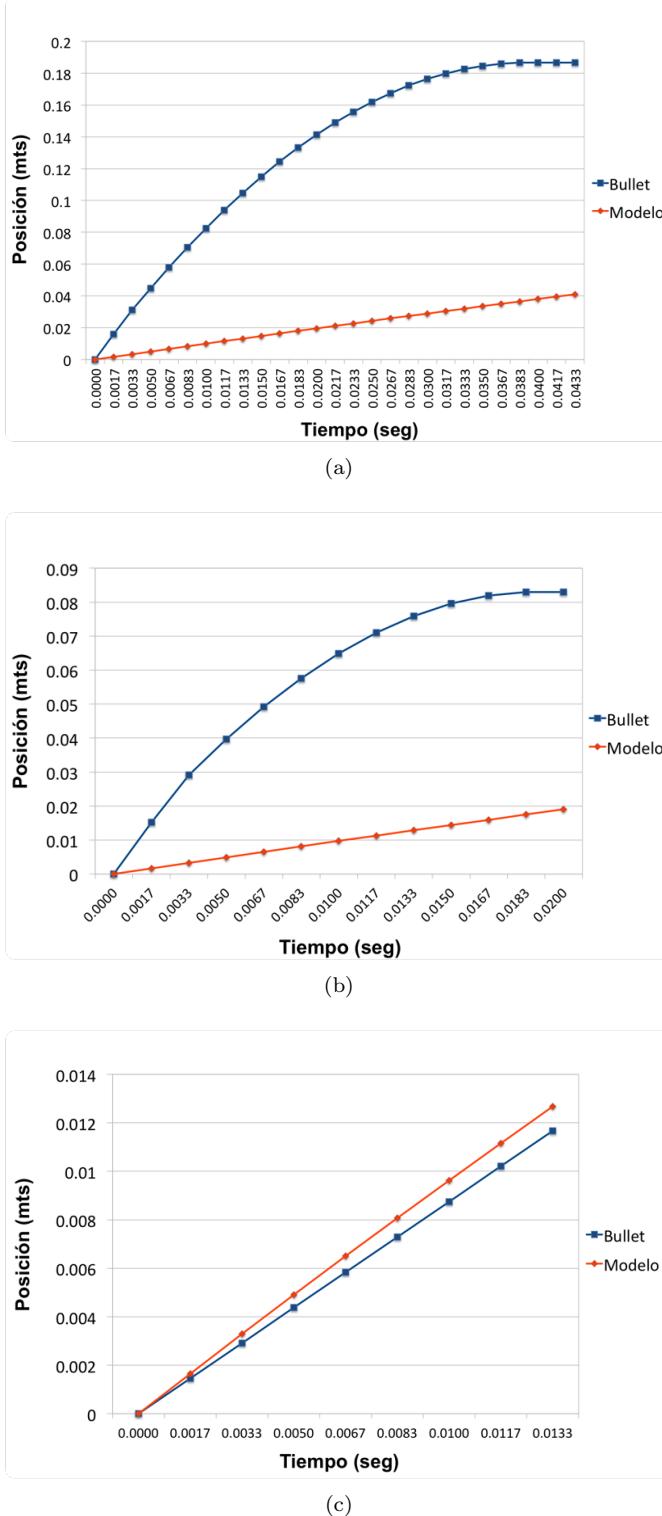


Figura 4: $v_i = 1 \frac{m}{s}$: (a) $\mu_d = 0.25$, (b) $\mu_d = 0.50$, y (c) $\mu_d = 0.75$

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

166

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

167

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 (ver Fig. 5). A la esfera se le asigna, además, un coeficiente de restitución determinado.

171

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)).

174

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 $\mu = 0.75$.

177

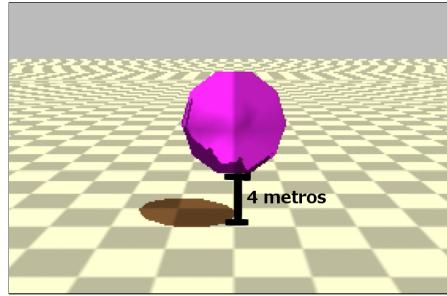


Figura 5: RENOMBRAR FIGURA

178

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 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:

182

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

183

Se muestran los experimentos numéricos realizados. En ellos, $v_i = \{-0.5, -3.5, -4, -5 \text{ y } -10\} \frac{\text{m}}{\text{s}}$ 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 de simular el sistema descripto en Fig. 5

185 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
 186 en evidencia en la Tabla 1. En cada una de ellas el error fue de casi 1.
 187
 188 La razón por la que ocurre este hecho se debe a que *Bullet* utiliza un algoritmo de colisión
 189 que frena la velocidad de un objeto que está a punto de colisionar. Haciendo esto puede evitar
 190 que los sólidos se traspasen y de esta forma se pueden realizar cálculos de fuerza más precisos.
 191 En el caso de los experimentos, las esferas poseen una velocidad muy baja, cuando están a
 192 punto de colisionar *Bullet* reduce aún más esta velocidad y eventualmente quedan con una
 193 velocidad tan baja que al chocar contra el suelo se aplica el efecto restitutivo a esta velocidad
 194 casi nula y se resuelve que la esfera debe quedar en reposo, cuando en realidad debería poseer
 195 una velocidad baja, pero no despreciable.

196 2.1.3. Ventajas

197 Las ventajas del motor físico son:

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

201 2.1.4. Desventajas

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

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

- 206 – Utilizar una librería gráfica (como *OpenGL*) acoplada a una simulación de *Bullet* puede pro-
207 ducir resultados distintos, que si se usa un programa de visualización externo (como OVITO).

208

209 **2.2. Librería de Algoritmos Genéticos**

210 Se utilizó la conocida librería de Algoritmos Genéticos para C++ GaLib, desarrollada por
211 Matthew Wall del MIT [7].

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

214

215 **2.3. Código Fuente**

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

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

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

226 **3. Modelo Utilizado**

227 Hay diversos modelos. Algunos son más genéricos [8] [9] y complejos. Sin embargo, se procuró
228 utilizar uno que fuera sencillo pero representativo a la vez.

229 Se modela al cuerpo humano, con el motor *Bullet Physics*, como un conjunto de segmentos unidos
230 por articulaciones (Fig. 6). A cada uno de ellos se les aplica una fuerza en el centro de masa
231 de cada segmento (denominada Actuador). Que la caminata se produzca o no, depende del tipo
232 de actuador utilizado (la función utilizada para la fuerza), y de sus parámetros. El objetivo,
233 entonces, se reduce a encontrar dichos parámetros. Para eso se usan los algoritmos genéticos, un
234 método de Inteligencia Artificial. De este modo, se obtiene, de forma análoga a la selección natu-
235 ral, los individuos que mejor se adapten a la caminata. Tanto los actuadores como el algoritmo
236 genético se explicarán más adelante.

237

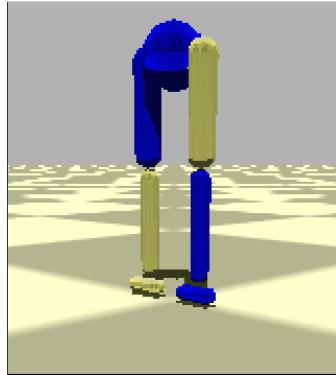


Figura 6: Humanoide diseñado

238 3.1. Composición Física del Humanoide

239 Como ya se expresó, el humanoide fue modelado en *Bullet* como un conjunto de segmentos
 240 (cuerpos rígidos), unidos por articulaciones (ver Fig. 7).

241 Se dividió al cuerpo humano en: cabeza, tronco, miembro superior, pelvis y miembro inferior
 242 (muslo, pierna y pie). Considerar la mitad superior del cuerpo implicaba mayor complejidad
 243 (manejo de equilibrio, humanoide más pesado y con mayor volumen, etc.). Ergo, solo se tomó la
 244 pelvis y el miembro inferior.

245 A continuación se presenta la composición de cada segmento (de acuerdo a la biomecánica [10]):

Parte	Cantidad	Forma	Largo (en m)	Peso (en kg)	Uniones
Pelvis	1	esférico	0.08655	9.9718	Cadera
Muslo	2	esfero-cilindro	0.4015	10.3368	Cadera y Rodilla
Pierna	2	esfero-cilindro	0.4015	3.1609	Rodilla y Tobillo
Pie	2	esfero-cilindro	-	1.0001	Tobillo

Tabla 2: Segmentos del humanoide

246 3.2. Articulaciones

247 Para unir los distintos segmentos entre sí, se utilizaron articulaciones bisagra con 1 grado de
 248 libertad: en el eje Z, donde ocurre la caminata; y en el eje Y, perpendicular al piso (ver Fig. 7).
 249 Asimismo, para cada caso en particular, se definieron cotas para los ángulos que pueden existir
 250 entre los segmentos. Esto es muy importante, no sólo porque se adecúa a datos biológicos, sino
 251 porque, de otro modo la caminata no podría lograrse: si los ángulos son demasiado altos, la
 252 caminata se produce girando las piernas por encima de la pelvis; si por el contrario, son demasiado
 253 bajos, las piernas van a estar muy rígidas, originando pocos pasos y muy cortos.
 254 Por otra parte, a la pelvis se le restringe todo tipo de rotación.

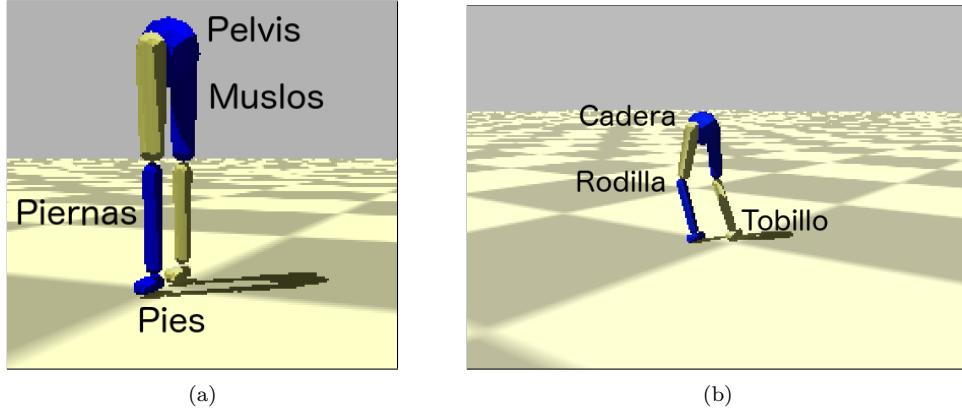


Figura 7: Humanoide diseñado: (a) segmentos, y (b) articulaciones

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. Que el torque se aplique en una sola dimensión, contribuye a que la caminata producida sea plana (en 2D, y no en 3D, como debería ser en una caminata real). Para indicar el módulo de dicho torque, se diseñaron diferentes funciones (todas ellas periódicas), mencionadas en las subsecciones que siguen.

265

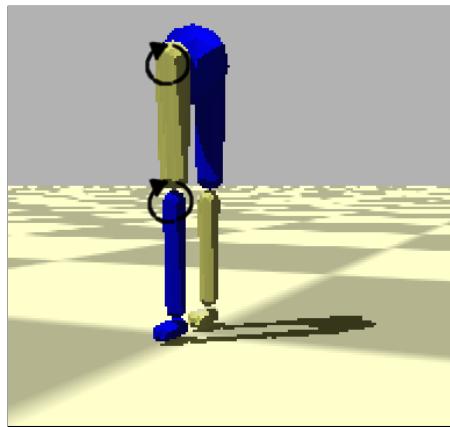


Figura 8: Aplicación de los actuadores en los segmentos del bípedo

4.1. Genérico

266 Es el actuador más sencillo, tanto matemática como computacionalmente. La fase es la misma tanto en el seno como en el coseno, para evitar que se formen otro tipo de funciones no cíclicas.

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

²⁶⁹ **4.2. Fourier**

²⁷⁰ Este actuador utiliza una serie de Fourier de dos términos.

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

²⁷¹ **4.3. Extra Fourier**

²⁷² Es una extensión del actuador anterior, pero con 9 términos. Por ser de mayor grado, brinda
²⁷³ una mayor precisión. Sin embargo, es más difícil de manejar computacionalmente; y, además,
²⁷⁴ que sea más preciso no garantiza que con él se pueda lograr una buena caminata.

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

²⁷⁵ **4.4. Doble coseno**

²⁷⁶ Esta función periódica utiliza medio ciclo de una función sinusoidal, y medio ciclo de otra
²⁷⁷ (ambas pueden tener frecuencias distintas). De esta manera, se consigue una caminata más
²⁷⁸ natural, y que no ocurre con los actuadores de Fourier, que producen una doble flexión de las
²⁷⁹ rodillas en cada ciclo.

²⁸⁰ La idea es lograr una función periódica a partir de una que no lo es (ya que t es lineal). Para
²⁸¹ eso, se utiliza $\psi(t)$ (ec. (10)) que aplica una transformación a los números reales, para que se
²⁸² encuentren dentro del rango del ciclo completo (con las dos frecuencias). ω es la frecuencia de
²⁸³ $f(t)$ (ec. (12)), que utiliza medio ciclo con frecuencia ω_1 y medio ciclo con frecuencia ω_2 .

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

²⁸⁴

²⁸⁵

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

²⁸⁶

²⁸⁷

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

²⁸⁸ **5. Condiciones iniciales y de contorno**

²⁸⁹ Como las funciones periódicas señaladas en los actuadores, no son suficientes para lograr la
²⁹⁰ caminata, se le adosaron las funciones que se detallan seguidamente.

²⁹¹ **5.1. Función Partida**

²⁹² El andar del humanoide es cíclico. Sin embargo, por la posición inicial del individuo, se requiere
²⁹³ para el tiempo del primer paso, una función distinta a la del resto de la caminata. El tipo de

función puede ser cualquiera de los actuadores vistos anteriormente (pero no necesariamente con los mismos valores de amplitud, frecuencia y fase asignados a las piernas). Empero, se utilizó la función vista en el actuador genérico.

Por otra parte, para simplificar el modelo, se decidió que el tiempo considerado para el primer paso sea fijo, y de 0.7 segundos. Dicho valor fue extraído de forma experimental.

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.

5.3. Riel

Los seres humanos aplican fuerzas constantemente en el eje perpendicular a la tierra para mantenerse erguidos. Análogamente, este comportamiento puede favorecer a que el bípedo se mantenga de pie. Para eso, se consideró un sistema de control en el que la pelvis esté sujetada a un resorte imaginario, que se traslada con ella pero ubicado a una altura determinada. Cuanto mayor sea la diferencia de la altura, mayor será la fuerza aplicada por el resorte.

De esta manera, se le puede aplicar a la pelvis, en cada paso de la simulación, un resorte en el eje Y de la forma:

$$f_r(h) = -k(h - Qh_0) \quad (15)$$

donde k es la constante elástica del resorte, h es la altura actual del humanoide, h_0 su altura inicial, y $Q = 1.15$. El término Qh_0 indica la altura del resorte.

Como contrapartida, esto imposibilita que el humanoide caiga de forma natural, ya que nunca puede caer. Éste es uno de los factores por el cual la caminata se realiza en 2D.

6. Algoritmo Genético

6.1. Individuo

La información genética de cada individuo, está compuesta por tres partes: función partida (optativa), parámetros asociados a los actuadores (obligatorios), y el riel (optativo).

6.1.1. Parámetros

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

Con respecto al riel, el único parámetro es la constante elástica (k), medida en $\frac{N}{m}$.

³²⁷ **6.1.2. Valores**

³²⁸ Cada uno de los segmentos tiene propiedades físicas distintas (masa, largo, etc.), razón por
³²⁹ la cual no necesariamente sus genes deban tener los mismos rangos de valores.

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 3: 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
Fourier	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
Extra Fourier	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
Doble Coseno	Muslo y Pierna	Amplitud	-100	100
	Muslo y Pierna	Frecuencia	0.01	10
	Muslo y Pierna	Fase	$-\pi$	π
	Muslo	Término independiente	-50	50
	Pierna	Término independiente	-100	100

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

³³⁰ Por otra parte, el parámetro k del riel puede tomar valores que oscilan entre 700 y 1600.

³³¹ **6.1.3. Implementaciones de individuos**

³³² Para favorecer el análisis de las distintas características antes implementadas, se implemen-
³³³ taron los siguientes individuos:

Tipo de Individuo	Actuador	Función Partida	Fase sincronizada	Riel
1	Genérico	No	Sí	Sí
2	Genérico	Sí	Sí	No
3	Fourier	Sí	Sí	No
4	Extra Fourier	Sí	Sí	No
5	Doble coseno	Sí	Sí	No

Tabla 5: Tipos de humanoide

334 6.1.4. Constitución del cromosoma

335 En la Tabla 6 se presenta la composición del cromosoma de cada individuo.

336 Vale aclarar que la función partida se especifica para cada segmento (las dos piernas y los dos
337 muslos). En cambio, para los actuadores, sólo se definen dos (uno para las piernas y otro para
338 los muslos).

Individuo	Amplitud	Frecuencia	Fase	Término independiente	Riel
1	12	2	2	2	1
2	12	6	6	6	0
3	16	6	6	6	0
4	44	6	6	6	0
5	8	8	6	6	0

Tabla 6: Distribución de parámetros según tipo de individuo

339 6.2. Fitness

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

$$F = H * V * D * S * PA \quad (16)$$

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

349

350 6.2.1. Altura

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

352

$$H = \frac{\sum_{n=0}^T e^{-C(h_{t_n} - h_{t_0})^2}}{N} \quad (17)$$

353

354 donde t_0 es el tiempo inicial, t_T el tiempo final, N la cantidad pasos de simulación y C una

355 constante $C = 5$.

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

361

362 6.2.2. Velocidad

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

365

$$V = \frac{\sum_{n=0}^T e^{-C(\|v_{t_n}\| - V_O)^2}}{N} \quad (18)$$

366

367 donde t_0 es el tiempo inicial, t_T el tiempo final y V_O la velocidad objetivo en el eje Z (el eje de
 368 la caminata).

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

372 6.2.3. Dirección

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

375

$$D = \frac{\sum_{n=0}^T e^{-C(v_{t_n} \cdot \vec{V}_O - 1)^2}}{N} \quad (19)$$

376

377 donde t_0 es el tiempo inicial, t_T el tiempo final, v_{t_n} el versor de la dirección del humanoide en el
 378 momento t_n y \vec{V}_O el versor de la dirección objetivo.

379 El producto escalar entre los versores responde a la Similitud Coseno: $\cos \theta = \frac{A \cdot B}{\|A\| \|B\|}$, donde A
 380 y B son vectores que no se encuentran normalizados, y θ es el ángulo formado entre ellos. Así, si
 381 $\cos \theta = 1$, significa que los vectores están paralelos entre sí (que es el efecto buscado en el caso
 382 de la dirección).

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

386

387 6.2.4. Simetría

388 Señala qué tan equidistantes se encuentran los pies de la cadera, a lo largo de la caminata.
 389 Aplicando solamente los módulos antes mencionados, provocaba resultados en donde una pierna
 390 quedaba más distante de la pelvis que la otra, lo que provocaba que el humanoide se terminara
 391 arrastrando (y posiblemente afectando a la velocidad).

392 Para mayor simplicidad, la simetría S se calculó a partir de los pies (y no de las piernas). Se

393 tomaron en cuenta sólo los ejes X y Z, porque son los relacionados a la velocidad y a la dirección,
 394 respectivamente.

$$S = \frac{\sum_{n=0}^T \frac{1}{2}[e^{-C(lf_z + rf_z)^2} + e^{-C(lf_x + rf_x)^2}]}{N} \quad (20)$$

395
 396 donde lf_X y lf_Z es la distancia desde el pie izquierdo hasta la pelvis en los ejes X y Z, respecti-
 397 vamente; y en donde rf_X y rf_Z es lo mismo, pero para el pie derecho.

398

399 6.2.5. Pies abajo

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

$$PA = \frac{\sum_{n=0}^T \frac{1}{2}(\alpha[e^{-C(ldf^2)} + e^{-C(rdf^2)}])}{N} \quad (21)$$

407
 408 donde $\alpha = \max(\min(lf, rf) - hip, 0), 1$ (es decir, vale 0 si la altura del pie izquierdo o derecho
 409 supera a la de la cadera, y 1 en otro caso); y ldf y rdf es la diferencia entre la posición inicial de
 410 los pies y la altura en el momento t_n de los pies izquierdo y derecho, respectivamente.

411

412 6.3. Parámetros del Algoritmo

413 6.3.1. Métodos de selección

414 De la vasta cantidad de métodos de selección que existen, se utilizaron: **Elite** (en donde se
 415 selecciona el individuo con mayor aptitud de la población-; y **Roulette** (método probabilístico,
 416 que selecciona un individuo de la población total al azar, con una probabilidad proporcional a
 417 su *fitness*).

418 6.3.2. Métodos de crusa

419 El método de crusa (o *crossover*) utilizado es el siguiente: De dos individuos (los padres), se
 420 originan dos nuevos individuos (los hijos). Se toma cada uno de los genes de los padres y se elige,
 421 con una probabilidad uniforme, uno de ellos para un hijo y el otro para el otro hijo.
 422 La probabilidad de que este proceso ocurra es de 0.9.

423 6.3.3. Mutación

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

427 Dicha probabilidad de mutación es de 0.3.

428 7. Resultados Obtenidos

429 Considerando los individuos definidos en el punto xx, se realizaron experimentos, corriendo
 430 el algoritmo genético, y evaluando el resultado obtenido posteriormente (ya sea numérica o
 431 gráficamente). A continuación se analizan distintos aspectos relevantes.

432 7.1. Evolución del *fitness* según tipo de individuo

433 En el caso del individuo de tipo 2, puede observarse que, aun cuando el *fitness* llega a 0.6 (y
 434 no a 1, su cota superior), tiene un buen rendimiento, ya que, suponiendo que cada uno de los
 435 módulos del *fitness* están al 90 %, se tendría que $0.9^5 = 0.59$.

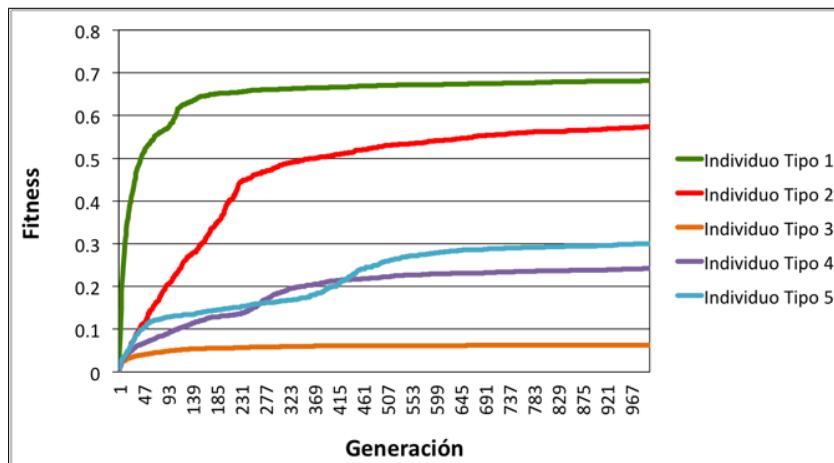


Figura 9: Evolución del *fitness* según tipo de individuo

436 7.2. Velocidad según tipo de individuo

437 A continuación, se muestra, para cada individuo, su velocidad instantánea a lo largo del
 438 tiempo.

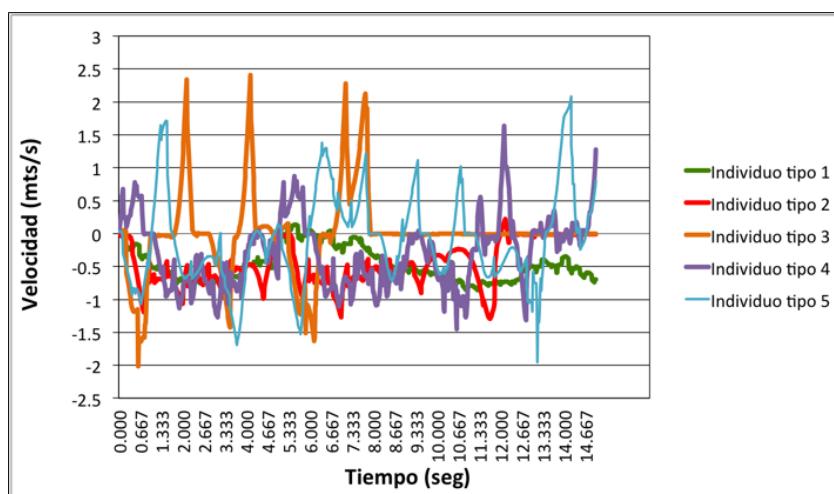


Figura 10: Velocidad según tipo de individuo

439 Como puede observarse en el caso de los individuos 3, 4 y 5 (Fourier, Extra Fourier y Doble

- 440 Coseno, respectivamente), se producen picos altos y pronunciados en la velocidad. Eso repercute
 441 en que la velocidad media no sea $1.3 \frac{m}{s}$ (que es la velocidad objetivo), y ergo, en el módulo de
 442 velocidad del *fitness* (provocando que éste sea más bajo).
 443 En el caso de los individuos 1 y 2, la velocidad instantánea oscila de forma suave.

444 **7.3. Altura según tipo de individuo**

- 445 Como puede identificarse en la figura 11, la altura de los individuos es otra característica para
 446 diferenciarlos en su rendimiento.
 447 En efecto, los individuos de tipo 3 y 4 (que utilizan actuadores de Fourier y Extra Fourier,
 448 respectivamente), son los que caen de forma más abrupta, y aunque intentan levantarse, vuelven
 449 a caer con la misma intensidad.
 450 Por su parte, el individuo de tipo 1 es el que mantiene la altura constante, a causa del riel que
 451 tiene activado (que le hace mantener la altura). El individuo de tipo 2, con actuadores genéricos
 452 y función partida, logra mantener la altura, hasta que cae. Considerando que no tiene el riel
 453 activado, y que cuando cae no vuelve a levantarse (ya que en ese movimiento suelen emplearse
 454 los brazos y manos), tiene un comportamiento similar a una caminata real.
 455 En lo que respecta al individuo de tipo 5, que utiliza como actuadores la función de doble coseno,
 456 se cae y levanta repetidamente, pero de forma más suave que los que utilizan Fourier y Extra
 457 Fourier.

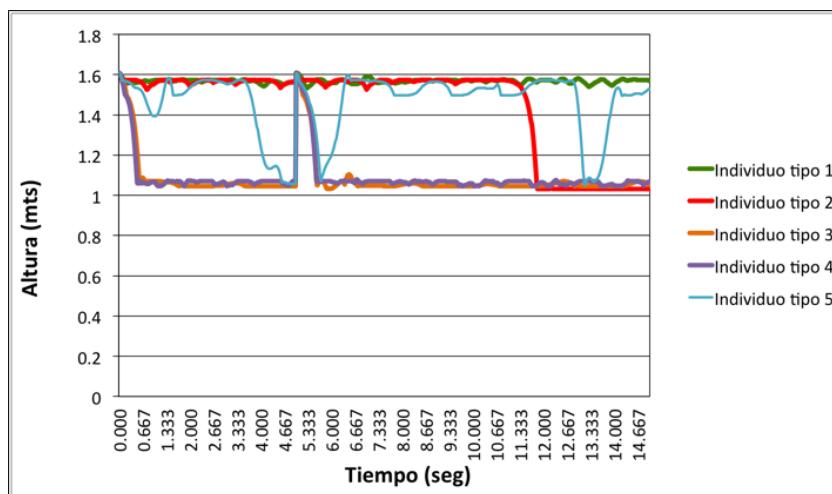


Figura 11: Altura (posición de la pelvis) según tipo de individuo

458 **7.4. Comparación de tipos de individuo**

459 **7.5. Video**

- 460 Acá iría el link del video?

461 **8. Conclusiones**

Referencias

- 463 [1] Wikipedia: https://es.wikipedia.org/wiki/Physics_engine
- 464 [2] Andreas Gerndt y otros, *An Evaluation of Open Source Physics Engines for Use in Virtual Reality Assembly Simulations*. Fecha de publicación: 2012
- 465 [3] Tom Erez y otros, *Simulation Tools for Model-Based Robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX*
- 466 [4] Sitio web de Bullet Physics: <http://www.bulletphysics.org/>
- 467 [5] Erin Catto, *Iterative Dynamic with Temporal Coherence*. Fecha de publicación: 2005
- 468 [6] Kevin Kenny, Máximo Videla y Axel Wassington, *Proyecto Final para la obtención del título: Ingeniero en Informática - ITBA, 2014*
- 469 [7] Sitio web de GaLib: <http://lancet.mit.edu/ga/>
- 470 [8] Thomas Geijtenbeek, Michiel van de Panne y A. Frank van der Stappen, *Flexible Muscle-Based Locomotion for Bipedal Creatures*, 2013
- 471 [9] Marek Wojtyra, *Multibody Simulation Model of Human Walking - Warsaw University of Technology*, 2003
- 472 [10] <http://www.exrx.net/Kinesiology/Segments.html>
- 473
- 474
- 475
- 476
- 477
- 478