

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](mailto:ealtamir@itba.edu.ar)

10

**Teresa Fontanella De Santis**

11

[tfontane@itba.edu.ar](mailto:tfontane@itba.edu.ar)

12

**Tomás Mehdi**

13

[tmehd@itba.edu.ar](mailto: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	<b>Resumen</b>	4
21	<b>1 Introducción</b>	4
22	<b>2 Herramientas</b>	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	<b>3 Modelo Utilizado</b>	12
33	3.1 Composición Física del Humanoide . . . . .	13
34	3.2 Articulaciones . . . . .	13
35	<b>4 Actuadores</b>	14
36	4.1 Genérico . . . . .	14
37	4.2 Fourier de orden 2 . . . . .	14
38	4.3 Fourier de orden 9 . . . . .	15
39	4.4 Doble seno . . . . .	15
40	<b>5 Condiciones iniciales y de contorno</b>	15
41	5.1 Función Partida . . . . .	15
42	5.2 Fase sincronizada . . . . .	16
43	<b>6 Algoritmo Genético</b>	16
44	6.1 Individuo . . . . .	16
45	6.1.1 Parámetros . . . . .	16
46	6.1.2 Valores . . . . .	16
47	6.1.3 Implementación de individuos . . . . .	17
48	6.1.4 Constitución del cromosoma . . . . .	17
49	6.2 Fitness . . . . .	18
50	6.2.1 Altura . . . . .	18
51	6.2.2 Velocidad . . . . .	19
52	6.2.3 Dirección . . . . .	19
53	6.2.4 Simetría . . . . .	19
54	6.2.5 Pies abajo . . . . .	20
55	6.3 Parámetros del Algoritmo . . . . .	20
56	6.3.1 Métodos de selección . . . . .	20
57	6.3.2 Métodos de cruce . . . . .	20
58	6.3.3 Mutación . . . . .	20

59	6.3.4	Otros parámetros . . . . .	20
60	<b>7</b>	<b>Resultados Obtenidos</b>	<b>21</b>
61	7.1	Evolución del <i>fitness</i> según tipo de individuo . . . . .	21
62	7.2	Velocidad según tipo de individuo . . . . .	21
63	7.3	Altura según tipo de individuo . . . . .	22
64	7.4	Comparación de tipo de individuos . . . . .	23
65	7.5	Video . . . . .	23
66	<b>8</b>	<b>Conclusiones</b>	<b>23</b>
67	<b>Referencias</b>		<b>23</b>

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  
81 videojuegos (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  
84 animación, sino un objeto compuesto de segmentos físicos, que interactúan entre sí y con el  
85 entorno, por medio de las leyes físicas; agregando así realismo a la situación simulada. La otra  
86 propiedad es que el bípedo aprenda por sí solo a caminar en línea recta, sin tener en su trayectoria  
87 ningún obstáculo.

88 Este problema se puede abordar de diversas maneras, involucrando: redes neuronales<sup>1</sup>, sistemas  
89 de control<sup>2</sup> (*passive walkers*<sup>3</sup>)[1], algoritmos genéticos[2], entre otras. Pero varios de ellos implican  
90 modelos teóricos complejos de humanoide (considerando músculos con distintos materiales, etc.).

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

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

---

<sup>1</sup>Una 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.

<sup>2</sup>Un 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.

<sup>3</sup>Un *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).

## <sup>98</sup> 2. Herramientas

### <sup>99</sup> 2.1. Motor Físico

<sup>100</sup> Se le llama motor físico o *physics engine* a un “*software* capaz de realizar simulaciones de  
<sup>101</sup> ciertos sistemas físicos, como la dinámica del cuerpo rígido, el movimiento de un fluido y la  
<sup>102</sup> elasticidad”[3].

<sup>103</sup> Actualmente, existen muchos motores físicos: ya sea de código propietario (*PhysX*, *Havok*), como  
<sup>104</sup> *open-source* (*Bullet Physics*, *Box2D*, *Newton*, *OGRE*). Considerando análisis relacionados [4][5],  
<sup>105</sup> y el hecho de que el espacio simulado fuese en 3D, se decidió que *Bullet Physics*[6] es el más  
<sup>106</sup> idóneo. Está implementado en C++ y ha sido utilizado en varios juegos (*Grand Theft Auto IV*  
<sup>107</sup> y *V*, etc); en los efectos especiales de películas (*Hancock*, *Bolt*, etc.); y proyectos científicos, como  
<sup>108</sup> la herramienta *open-source Tensegrity Robotics Toolkit* de la NASA<sup>4</sup>; entre otros.

<sup>109</sup> Si bien (como se verá más adelante) *Bullet* tiene problemas asociados con el coeficiente de restitu-  
<sup>110</sup> ción, posee una muy buena *performance* en la detección de colisiones, la dinámica y la resolución  
<sup>111</sup> de *constraints*. Esto se debe, en parte, a diferentes algoritmos iterativos de orden lineal (donde  
<sup>112</sup> el más importante es *Sequential Impulse*), de *caching* y también a la utilización de un modelo  
<sup>113</sup> de fricción de Coulomb aproximado [7]. Además, el motor físico brinda la posibilidad de regular  
<sup>114</sup> la precisión requerida en estos cálculos (sin olvidar que, con iguales recursos, a mayor precisión,  
<sup>115</sup> mayor capacidad de cómputo requerida y, ergo, mayor tiempo). Dado que la construcción del hu-  
<sup>116</sup> manoide implica definir características y restricciones de movimiento de cada una de sus partes,  
<sup>117</sup> lo antes mencionado fue crucial para la elección de *Bullet Physics* en este proyecto.

#### <sup>118</sup> 2.1.1. Funcionamiento

<sup>119</sup> El motor físico se encarga de la simulación de cuerpos rígidos y la interacción entre los mismos.  
<sup>120</sup> En particular debe: calcular el resultado de colisiones, arreglar el solapamiento de los cuerpos  
<sup>121</sup> en el espacio de simulación, estimar las fuerzas producidas debido a la fricción, y mantener el  
<sup>122</sup> cumplimiento de restricciones que puedan existir entre los cuerpos (por ejemplo, un vínculo para  
<sup>123</sup> formar una articulación). Para lograrlo, *Bullet* modela, a partir de un conjunto de ecuaciones,  
<sup>124</sup> las distintas restricciones que deben ser respetadas.

<sup>125</sup> Estos modelos reciben como dato la velocidad lineal y angular de cada objeto, y las fuerzas  
<sup>126</sup> que actúan sobre los mismos. Dada esta información, se resuelve el sistema de ecuaciones, cuya  
<sup>127</sup> solución representa las magnitudes de las fuerzas a accionar sobre el mismo, a fin de satisfacer  
<sup>128</sup> todas las restricciones. Para encontrar esta solución, entran en juego los distintos métodos de  
<sup>129</sup> complejidad lineal mencionados en la subsección anterior.

<sup>130</sup> Este procedimiento, se lleva a cabo en cada *timestep* de la simulación; donde un *timestep* es el  
<sup>131</sup> intervalo de tiempo que transcurre entre un cálculo de magnitudes y otro. Mientras menor sea  
<sup>132</sup> el *timestep*, el simulador será más preciso pero también consumirá más tiempo de cómputo.

#### <sup>133</sup> 2.1.2. Modelo de fricción y su verificación

<sup>134</sup> Hay reglas físicas relacionadas con el entorno y que son muy importantes para la caminata:  
<sup>135</sup> el modelo de fricción, con sus respectivos coeficientes de fricción y restitución.  
<sup>136</sup> En base a los modelos físico-matemáticos que representan a cada uno de los dos fenómenos en  
<sup>137</sup> cuestión (y que se explicarán a continuación), y pensando en posibles futuras simulaciones de

<sup>4</sup><http://bulletphysics.org/Bullet/phpBB3/viewtopic.php?f=17&t=9978>

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

140 **2.1.2.1. Verificación del coeficiente de fricción**

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

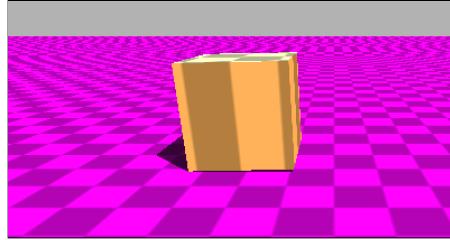


Figura 1: Visualización del sistema del cubo

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

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

147 En este caso, el cuerpo empieza su movimiento en el origen, por lo tanto la posición inicial ( $x_i$ )  
 148 es cero.  $v_i$  es la velocidad inicial, y  $a$ , la aceleración. Debido a la fricción entre el cuerpo y el  
 149 suelo, se genera una fuerza de rozamiento  $\mathbf{F}_{\mu_d}$ <sup>5</sup> (ec. (2)) en la misma dirección que la velocidad  
 150 del sólido y en sentido contrario.

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

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

153

154 Finalmente, se obtiene la aceleración:

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

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

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

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

160 Seguidamente, se muestran los gráficos obtenidos al correr los experimentos numéricos con los  
 161 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  
 162 eje Z de la caja en Bullet con su distancia en el mismo eje según la ec. (4).

<sup>5</sup>En las ecuaciones, los vectores se escriben en negrita. Y las fuerzas son vectores.



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)  $\mu_d = 0.75$

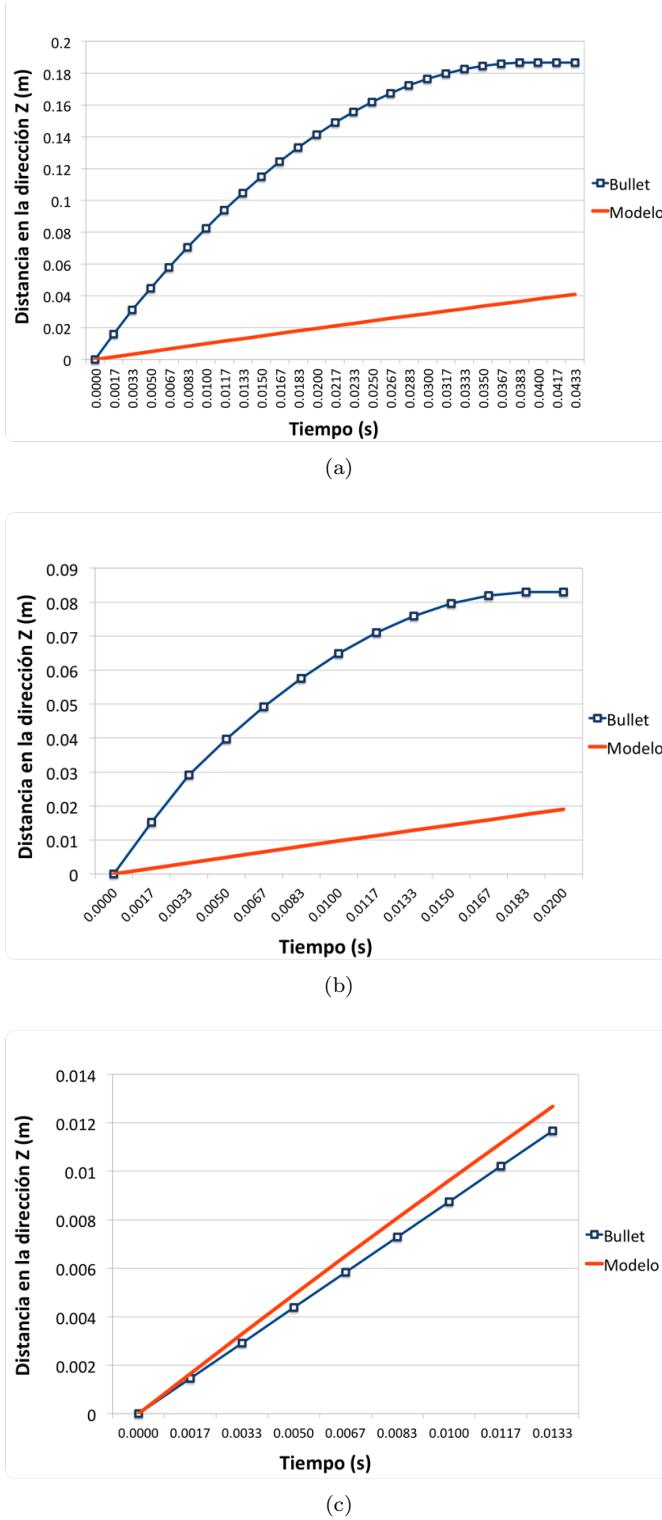


Figura 4: Resultados logrados de simular el sistema descripto en Fig. 1, usando  $v_i = 1 \frac{m}{s}$  y:

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

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

167 ( $v_i$ ), mayor es la similitud entre los dos casos.

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

171 **2.1.2.2. Verificación del coeficiente de restitución**

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

176

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

177

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

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

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

187

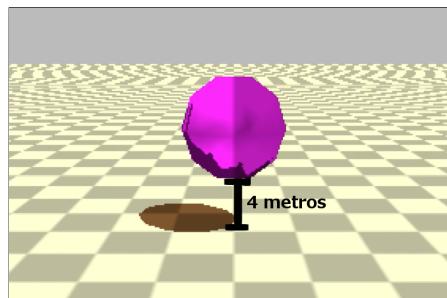


Figura 5: Visualización del sistema de la esfera

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

192

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

193 Se muestra en la Tabla 1 lo arrojado por los experimentos numéricos efectuados. En ellos,  
194  $v_i = \{-0.5, -3.5, -4, -5 \text{ y } -10\} \frac{\text{m}}{\text{s}}$  y  $e_{sim} = \{0.2, 0.5 \text{ y } 0.8\}$ .

$v_i$	$e_{sim}$	$v_f$	$e_{medida}$	$e_{rel}$
$-0.5 \frac{m}{s}$	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 descrito en Fig. 5

195 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.

201 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.

### 206 2.1.3. Ventajas

207 Las ventajas del motor físico son:

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

### 211 2.1.4. Desventajas

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

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

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

219 **2.2. Librería de Algoritmos Genéticos**

220 Se utilizó la conocida librería de Algoritmos Genéticos para C++ GaLib, desarrollada por  
221 Matthew Wall del MIT [8].

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

224 **2.3. Código Fuente**

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

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

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

235 **3. Modelo Utilizado**

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

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

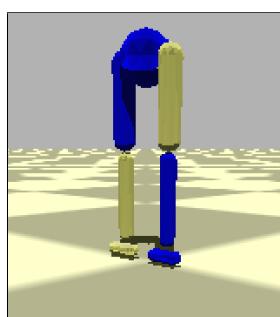


Figura 6: Humanoide diseñado

### 246 3.1. Composición Física del Humanoide

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

249 Se dividió al cuerpo humano en: cabeza, tronco, miembro superior, pelvis y miembro  
 250 inferior (muslo, pierna y pie). A los fines de este proyecto, sólo se consideraron la pelvis y el miembro  
 251 inferior (ver Fig. 7).

252 A continuación se presenta la composición de cada segmento (de acuerdo a la biomecánica [9]).

253

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

### 254 3.2. Articulaciones

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

258 Asimismo, para cada caso en particular, se definieron cotas para los ángulos que pueden existir  
 259 entre los segmentos. Esto es muy importante, no sólo porque se adecúa a datos biológicos, sino  
 260 porque, de otro modo la caminata no podría lograrse: si los ángulos son demasiado altos, la ca-  
 261 minata se produce girando las piernas por encima de la pelvis; si por el contrario, son demasiado  
 262 bajos, las piernas van a estar muy rígidas, originando pocos pasos y muy cortos.

263 Por otra parte, la pelvis no puede rotar (se le impide hacerlo); y se le restringe la amplitud con  
 264 la que puede moverse la cadera (de  $-\frac{\pi}{4}$  a  $\frac{\pi}{4}$ ). Todo esto se hace porque, en caso contrario, el  
 265 bípedo necesitaría un sistema de control para mantener el equilibrio, y eso excede el alcance de  
 266 este trabajo.

267

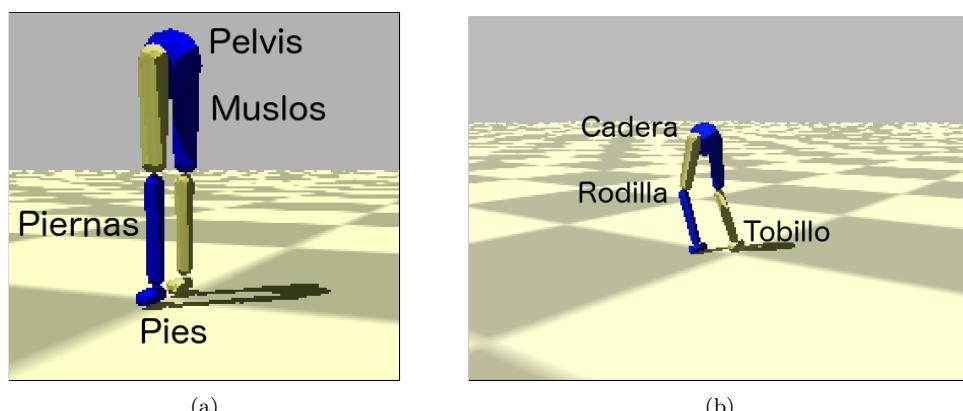


Figura 7: Humanoide diseñado: (a) segmentos, y (b) articulaciones

## 268 4. Actuadores

269 A cada uno de los segmentos correspondientes al muslo y la pierna del bípedo, se le aplica  
 270 un torque (o actuador) en el eje X (perpendicular a la trayectoria), como se ve en Fig 8. Así,  
 271 pueden moverse para arriba o para abajo (con respecto a la articulación a la que pertenecen).

272 A fin de simplificar el modelo, el humanoide tiene el mismo tipo de actuador utilizado en todos  
 273 los segmentos.

274 Es necesario aclarar que la caminata producida por el humanoide es plana (en 2D y no en 3D,  
 275 como debería ser en una caminata real). Esto se debe a que la trayectoria pensada para el bípedo  
 276 es una línea recta, y logrando que los segmentos se muevan en un solo eje es suficiente para  
 277 cumplir con dicha trayectoria. También contribuye el hecho de que el torque se aplique en una  
 278 sola dimensión. Por otra parte, los actuadores definidos a continuación son periódicos, y por eso  
 279 no se pueden aplicar en el eje Z de los segmentos (se necesitarían actuadores reactivos, para  
 280 poder detectar cuando el humanoide se está cayendo, etc.).

281 Para indicar el módulo de dicho torque, se diseñaron diferentes funciones (todas ellas periódicas),  
 282 mencionadas en las subsecciones que siguen.

283

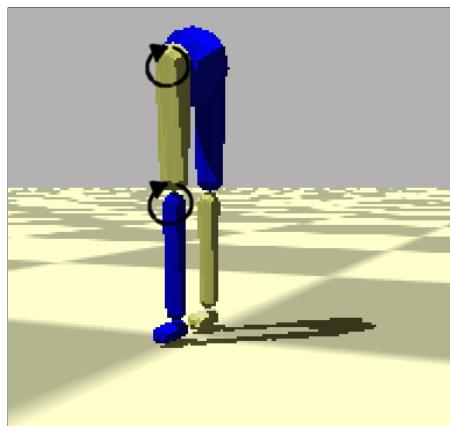


Figura 8: Aplicación de los actuadores en los segmentos del bípedo

### 284 4.1. Genérico

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

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

286 donde  $f(t)$  es la función del actuador evaluada en el instante de tiempo  $t$ ,  $A_1$  y  $A_2$  son amplitudes,  
 287  $\omega_1$  y  $\omega_2$  son frecuencias (en  $\frac{1}{s}$ ),  $\phi$  es la fase (en radianes), y  $C$  es un término independiente.

288 La fase  $\phi$  es la misma en el seno y en el coseno, para evitar que se formen otro tipo de funciones  
 289 no cíclicas.

### 290 4.2. Fourier de orden 2

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

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

292 donde  $A_1$ ,  $A_2$ ,  $B_1$  y  $B_2$  son amplitudes y  $\omega$  es frecuencia (en  $\frac{1}{s}$ ).

### 293 4.3. Fourier de orden 9

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

$$297 \quad f(t) = A_1 \sin(\omega t + \phi) + B_1 \cos(\omega t + \phi) + A_2 \sin(2\omega t + \phi) + B_2 \cos(2\omega t + \phi) \\ + A_3 \sin(3\omega t + \phi) + B_3 \cos(3\omega t + \phi) + A_4 \sin(4\omega t + \phi) + B_4 \cos(4\omega t + \phi) \\ + A_5 \sin(5\omega t + \phi) + B_5 \cos(5\omega t + \phi) + A_6 \sin(6\omega t + \phi) + B_6 \cos(6\omega t + \phi) \quad (9) \\ + A_7 \sin(7\omega t + \phi) + B_7 \cos(7\omega t + \phi) + A_8 \sin(8\omega t + \phi) + B_8 \cos(8\omega t + \phi) \\ + A_9 \sin(9\omega t + \phi) + B_9 \cos(9\omega t + \phi) + C$$

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

### 299 4.4. Doble coseno

300 Basada en [10], esta función periódica utiliza medio ciclo de una función sinusoidal, y medio  
 301 ciclo de otra (ambas pueden tener frecuencias distintas). Esto podría tener sentido porque en una  
 302 caminata, un miembro inferior primero avanza hacia adelante y luego se extiende hacia atrás, y  
 303 es razonable que esos dos movimientos se produzcan a frecuencias distintas.

304 La idea es lograr una función periódica a partir de una que no lo es (ya que  $t$  es lineal). Para  
 305 eso, se utiliza  $\psi(t)$  (ec. (10)) que aplica una transformación a los números reales, para que se  
 306 encuentren dentro del rango del ciclo completo (con las dos frecuencias).  $\omega$  es la frecuencia de  
 307  $f(t)$  (ec. (12)), que utiliza medio ciclo con frecuencia  $\omega_1$  y medio ciclo con frecuencia  $\omega_2$ .

$$308 \quad \psi(t) = t + \phi - \left\lfloor \frac{t + \phi}{\pi/\omega_1 + \pi/\omega_2} \right\rfloor (\pi/\omega_1 + \pi/\omega_2) \quad \psi : \mathbb{R} \rightarrow \left[ 0, \frac{2\pi}{\omega} \right] \quad (10)$$

$$309 \quad \omega = \frac{2\omega_1\omega_2}{\omega_1 + \omega_2} \quad (11)$$

$$310 \quad 311 \quad f(t) = \begin{cases} A \cos(\omega_1 \psi(t)) + C & \text{si } \omega_1 \psi(t) < \pi \\ A \cos(\omega_2(\psi(t) - (\pi/\omega_1) + (\pi/\omega_2))) + C & \text{en otro caso} \end{cases} \quad (12)$$

## 312 5. Condiciones iniciales y de contorno

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

### 315 5.1. Función Partida

316 El andar del humanoide es cíclico. Sin embargo, por la posición inicial del individuo, se requiere  
 317 para el tiempo del primer paso, una función distinta a la del resto de la caminata. El tipo de  
 318 función puede ser cualquiera de los actuadores vistos anteriormente (pero no necesariamente con  
 319 los mismos valores de amplitud, frecuencia y fase asignados a las piernas). Empero, se utilizó la  
 320 función vista en el actuador genérico.

<sup>321</sup> Por otra parte, para simplificar el modelo, se decidió que el tiempo considerado para el primer  
<sup>322</sup> paso sea fijo, y de 0.7 segundos. Dicho valor fue extraído de forma experimental.

## <sup>323</sup> 5.2. Fase sincronizada

<sup>324</sup> En una caminata, las piernas deben guardar simetría: mientras una va hacia adelante, la otra  
<sup>325</sup> va hacia atrás (y viceversa). Esto, de acuerdo con los actuadores definidos en la sección anterior,  
<sup>326</sup> implica que las funciones de movimiento de cada pierna estén desfasadas en medio ciclo ( $\frac{\pi}{2}$ ):

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

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

<sup>327</sup> siendo  $f(t)$  la función de movimiento (o actuador) en el momento  $t$ , y  $f_i$  y  $f_d$  las funciones de la  
<sup>328</sup> pierna izquierda y derecha, respectivamente.

## <sup>330</sup> 6. Algoritmo Genético

### <sup>331</sup> 6.1. Individuo

<sup>332</sup> La información genética de cada individuo, está compuesta por dos partes: función partida  
<sup>333</sup> (optativa) y los parámetros asociados a los actuadores (obligatorios).

#### <sup>334</sup> 6.1.1. Parámetros

<sup>335</sup> Tanto la función partida como los actuadores tienen como parámetros: amplitud ( $A$  o  $B$ );  
<sup>336</sup> fase ( $\phi$ ), que indica dónde comienza el paso y se mide en radianes; frecuencia ( $\omega$ ); y término  
<sup>337</sup> independiente ( $C$ ).

#### <sup>338</sup> 6.1.2. Valores

<sup>339</sup> Cada uno de los segmentos tiene propiedades físicas distintas (masa, largo, etc.), razón por  
<sup>340</sup> la cual no necesariamente sus genes deban tener los mismos rangos de valores, tal como puede  
<sup>341</sup> apreciarse en las Tablas 3 y 4.

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	$-\pi$	$\pi$
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	$-\pi$	$\pi$
	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	$-\pi$	$\pi$
	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	$-\pi$	$\pi$
	Muslo y Pierna	Término independiente	-10	10
Doble Coseno	Muslo	Amplitud	-50	50
	Pierna	Amplitud	-30	30
	Muslo y Pierna	Frecuencia	0.01	5
	Muslo y Pierna	Fase	$-\pi$	$\pi$
	Muslo y Pierna	Término independiente	-30	30

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

#### <sup>343</sup> 6.1.3. Implementación de individuos

<sup>344</sup> Para favorecer el análisis de las distintas características arriba indicadas, se implementaron  
<sup>345</sup> varios individuos, cada uno de ellos con propiedades distintas (Tabla 5).

Individuos	Actuador	Función Partida	Fase Sincronizada
<b>Tipo 1</b>	Genérico	No	Sí
<b>Tipo 2</b>	Genérico	Sí	Sí
<b>Tipo 3</b>	Fourier de orden 2	Sí	Sí
<b>Tipo 4</b>	Fourier de orden 9	Sí	Sí
<b>Tipo 5</b>	Doble coseno	Sí	Sí

Tabla 5: Tipo de individuos

#### <sup>346</sup> 6.1.4. Constitución del cromosoma

<sup>347</sup> La longitud del cromosoma de un individuo depende de los actuadores y la función partida  
<sup>348</sup> usados. Cada uno de ellos, a su vez, tienen los parámetros presentados en la sección 6.1.1. Sus  
<sup>349</sup> respectivas cantidades pueden verse en la Tabla 6.

<sup>350</sup> En la Tabla 7 se muestra la composición del cromosoma de cada individuo. En ella se puede  
<sup>351</sup> observar cómo según el tipo de individuo, varía la cantidad de genes, es decir, el tamaño del  
<sup>352</sup> cromosoma. Vale aclarar que la función partida se especifica para cada segmento (las dos piernas  
<sup>353</sup> y los dos muslos); en cambio, para los actuadores, sólo se definen dos (uno para las piernas y  
<sup>354</sup> otro para los muslos).

Actuadores	Parámetros			
	Amplitud	Frecuencia	Fase	Término independiente
Genérico	2	2	1	1
Fourier de orden 2	4	1	1	1
Fourier de orden 9	18	1	1	1
Doble coseno	1	2	1	1
Función partida	2	2	1	1

Tabla 6: Cantidad de parámetros según tipo de actuador o función partida

Parámetros	Individuos				
	Tipo 1	Tipo 2	Tipo 3	Tipo 4	Tipo 5
Amplitud	12	12	16	44	8
Frecuencia	2	6	6	6	8
Fase	2	6	6	6	6
Término independiente	2	6	6	6	6
<b>Totales</b>	18	30	34	62	28

Tabla 7: Cantidad de parámetros según tipo de individuos

## 355 6.2. Fitness

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

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

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

365

### 366 6.2.1. Altura

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

$$368 H = \frac{\sum_{n=0}^T e^{-C(h_{t_n} - h_{t_0})^2}}{N} \quad (16)$$

369

370 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  
 371 tiempo  $t_n$ ,  $N$  la cantidad pasos de simulación y  $C$  una constante  $C = 5$ .

372 Se calcula a partir de la diferencia entre la altura en cada instante de la simulación, con su altura  
 373 inicial (la altura está definida como la posición de la pelvis en el eje Z). Cuanto mayor sea esa  
 374 diferencia, más rápido el individuo cae, y por eso este módulo tiende a cero. Por el contrario,

375 valdrá uno si la diferencia es ínfima (lo que significa que el humanoide mantiene su misma altura  
 376 durante la caminata).

377

### 378 6.2.2. Velocidad

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

381

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

382

383 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  
 384 tiempo  $t_n$ , y  $V_O$  la velocidad objetivo en el eje Z (el eje de la caminata).

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

### 388 6.2.3. Dirección

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

391

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

392

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

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

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

402

### 403 6.2.4. Simetría

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

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

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

411

412 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.

414

#### 415 6.2.5. Pies abajo

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

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

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

423

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

428

### 429 6.3. Parámetros del Algoritmo

#### 430 6.3.1. Métodos de selección

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

#### 435 6.3.2. Métodos de crusa

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

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

#### 440 6.3.3. Mutación

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

444 Dicha probabilidad de mutación es de 0.3.

#### 445 6.3.4. Otros parámetros

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

## 448 7. Resultados Obtenidos

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

### 452 7.1. Evolución del *fitness* según tipo de individuo

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

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

460 Puede notarse que los rangos manejados en la Fig. 9 casi alcanzan el 0.7 (y no a 1, su cota  
 461 superior). Esto se debe a que la función de *fitness* está definida como un producto de ciertos  
 462 módulos: suponiendo que cada uno de ellos estuviera al 92 %, se tendría  $0.92^5 = 0.659$ .

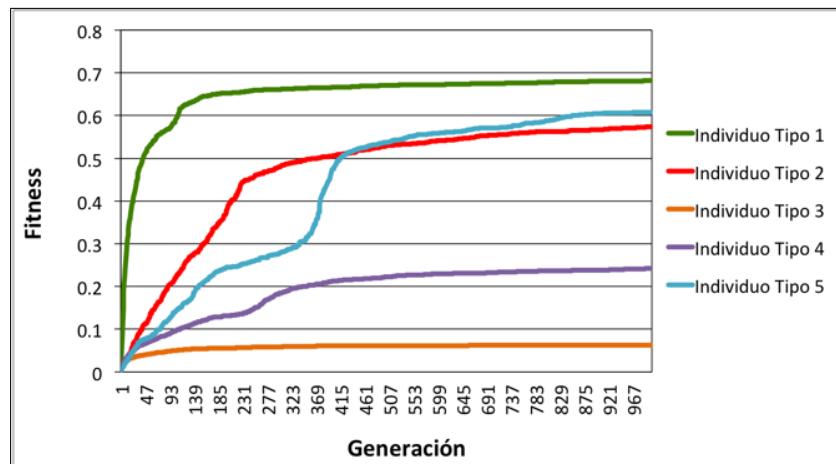


Figura 9: Evolución del *fitness* según tipo de individuo

### 463 7.2. Velocidad según tipo de individuo

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

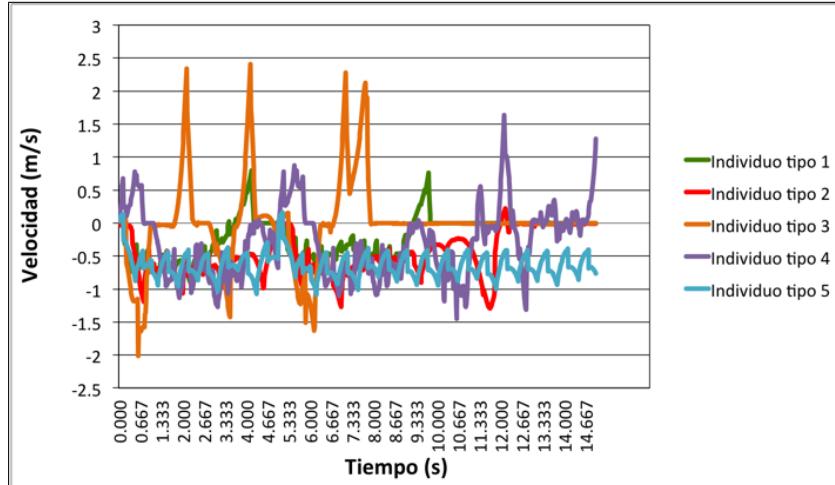


Figura 10: Velocidad según tipo de individuo

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

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

473 Por último, en el individuo de tipo 5, la velocidad oscila de forma cíclica, continuada y armoniosa  
 474 (no hay picos altos).

### 475 7.3. Altura según tipo de individuo

476 Como puede identificarse en la Figura 11, la altura de los individuos es otra característica  
 477 para diferenciarlos en su rendimiento.

478

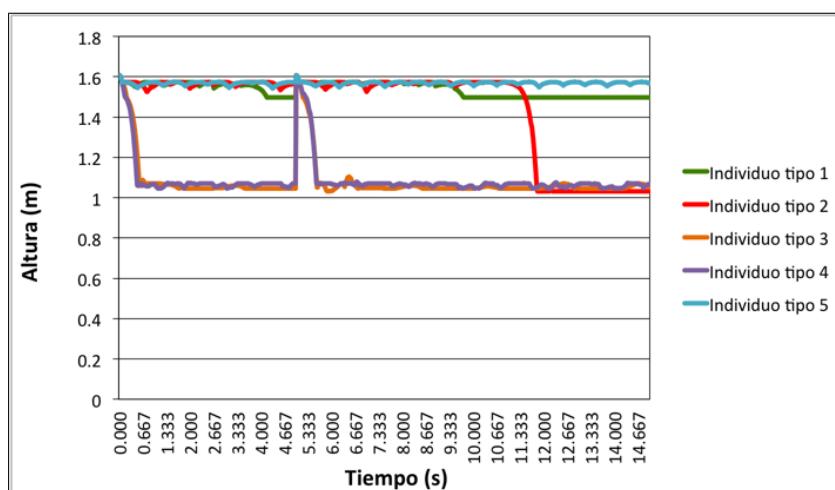


Figura 11: Altura (posición de la pelvis) según tipo de individuo

479 En efecto, los individuos de tipo 3 y 4 (que utilizan actuadores de Fourier de orden 2 y 9,

480 respectivamente), son los que caen de forma más abrupta, y aunque intentan levantarse, vuelven  
481 a caer con la misma intensidad.

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

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

#### 491 **7.4. Comparación de tipo de individuos**

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

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

#### 497 **7.5. Video**

498 Acá iría el link del video?

### 499 **8. Conclusiones**

500 Principalmente el generico y el doble coseno son los que mejor caminan, tanto en fitness como  
501 en altura.

502 Pero son dos caminatas distintas, una un poco más natural (generico) y la otra más robotica” pero  
503 estable.

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

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

509 Los actuadores fueron importantísimos.

### 510 **Referencias**

511 [1] Marek Wojtyra, *Multibody Simulation Model of Human Walking - Warsaw University of*  
512 *Technology, 2003*

513 [2] Thomas Geijtenbeek, Michiel van de Panne y A. Frank van der Stappen, *Flexible Muscle-*  
514 *Based Locomotion for Bipedal Creatures, 2013*

515 [3] Wikipedia: [https://es.wikipedia.org/wiki/Physics\\_engine](https://es.wikipedia.org/wiki/Physics_engine)

- 516 [4] Andreas Gerndt y otros, *An Evaluation of Open Source Physics Engines for Use in Virtual*  
517 *Reality Assembly Simulations*. Fecha de publicación: 2012
- 518 [5] Tom Erez y otros, *Simulation Tools for Model-Based Robotics: Comparison of Bullet, Havok,*  
519 *MuJoCo, ODE and PhysX*
- 520 [6] Sitio web de Bullet Physics: <http://www.bulletphysics.org/>
- 521 [7] Erin Catto, *Iterative Dynamic with Temporal Coherence*. Fecha de publicación: 2005
- 522 [8] Sitio web de GaLib: <http://lancet.mit.edu/ga/>
- 523 [9] <http://www.exrx.net/Kinesiology/Segments.html>
- 524 [10] Kevin Kenny, Máximo Videla y Axel Wassington, *Proyecto Final para la obtención del título:*  
525 *Ingeniero en Informática - ITBA, 2014*