<u>PROYECTO</u> <u>FINAL</u>

INGENIERÍA INFORMÁTICA - ITBA

SIMULACIÓN Y ANIMACIÓN BIOMECÁNICA DE UN HUMANOIDE

Alumnos:

Enzo Altamiranda Graterol
`ealtamir@itba.edu.ar`

Teresa Fontanella De Santis
`tfontane@itba.edu.ar`

Tomás Mehdi
`tmehdi@itba.edu.ar`

Tutor:

Dr. Daniel Ricardo Parisi

Instituto Tecnológico de Buenos Aires - ITBA
Departamento de Ingeniería Informática

Marzo 2016

Índice

Resumen

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

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

1. Introducción

Siempre ha sido de interés la simulación biomecánica de seres vivos, especialmente en las ciencias naturales (zoología, medicina, etc.).
Ahora bien, últimamente se ha incrementado el interés en otras áreas de aplicación, como los videojuegos (creación de personajes con reacciones más reales), y la ingeniería (verbigracia: diseño de espacios cerrados, con mayores medidas de seguridad).
Una característica muy importante de este trabajo es que el humanoide no es fruto de una animación, sino un objeto compuesto de segmentos físicos, que interaccionan entre sí y con el entorno, por medio de las leyes físicas; agregando así realismo a la situación simulada. La otra propiedad es que el bípedo aprenda por sí solo a caminar en línea recta, sin tener en su trayectoria ningún obstáculo.
Este problema se puede abordar de diversas maneras, involucrando: redes neuronales[1], sistemas de control[2] (*passive walkers*[3]), algoritmos genéticos, entre otras. Pero varios de ellos implican modelos teóricos complejos de humanoide (considerando músculos con distintos materiales, etc.).
En este proyecto se buscó aplicar algoritmos genéticos, y lograr la caminata usando un modelo de humanoide basado en cuerpos rígidos, unidos por articulaciones, y cuyo desplazamiento depende de torques aplicados a dichos cuerpos.
El presente informe, describe y analiza pormenorizadamente: en la sección 2, las herramientas aplicadas; en la 3, el modelo del humanoide utilizado; en la 4 y 5, los diferentes tipos de actuadores y funciones de partida y contorno; en la 6, el algoritmo genético; y en la 7 y 8, los resultados obtenidos con sus respectivas conclusiones.

[1] 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.

[2] definir sistema de control

[3] 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).

2. Herramientas

2.1. Motor Físico

Se le llama motor físico o *physics engine* a un "*software* capaz de realizar simulaciones de ciertos sistemas físicos, como la dinámica del cuerpo rígido, el movimiento de un fluido y la elasticidad"[1].
Actualmente, existen muchos motores físicos: ya sea de código propietario (PhysX, Havok), como *open-source* (*Bullet Physics*, *Box2D*, Newton, OGRE). Considerando análisis relacionados [2][3], y la necesidad de que el espacio simulado fuese en 3D, se decidió que *Bullet Physics*[4] es el más idóneo. Está implementado en C++ y ha sido utilizado en varios juegos (*Grand Theft Auto IV* y V, etc); en los efectos especiales de películas (Hancock, Bolt, etc.); y proyectos científicos, como la herramienta *open-source Tensegrity Robotics Toolkit* de la NASA[4]; entre otros.
Si bien (como se verá más adelante) *Bullet* tiene problemas asociados con el coeficiente de restitución, posee una muy buena *performance* en la detección de colisiones, la dinámica y la resolución de *constraints*. Esto se debe, en parte, a diferentes algoritmos iterativos de orden lineal (donde el más importante es *Sequential Impulse*), de *caching* y también a la utilización de un modelo de fricción de Coulomb aproximado [5]. Además, el motor físico brinda la posibilidad de regular la precisión requerida en estos cálculos (sin olvidar que, con iguales recursos, a mayor precisión, mayor capacidad de cómputo requerida y, ergo, mayor tiempo). Dado que la construcción del humanoide implica definir características y restricciones de movimiento de cada una de sus partes, lo antes mencionado fue crucial para la elección de *Bullet Physics* en este proyecto.

2.1.1. Funcionamiento

El motor físico se encarga de la simulación de cuerpos rígidos y la interacción entre los mismos. En particular debe: calcular el resultado de colisiones, arreglar el solapamiento de los cuerpos en el espacio de simulación, estimar las fuerzas producidas debido a la fricción, y mantener el cumplimiento de restricciones que puedan existir entre los cuerpos (por ejemplo, un vínculo para formar una articulación). Para lograrlo, *Bullet* modela, a partir de un conjunto de ecuaciones, las distintas restricciones que deben ser respetadas.
Estos modelos reciben como dato la velocidad lineal y angular de cada objeto, y las fuerzas que actúan sobre los mismos. Dada esta información, se resuelve el sistema de ecuaciones, cuya solución representa las magnitudes de las fuerzas a accionar sobre el mismo, a fin de satisfacer todas las restricciones. Para encontrar esta solución, entran en juego los distintos métodos de complejidad lineal mencionados en la subsección anterior.
Este procedimiento, se lleva a cabo en cada *timestep* de la simulación; donde un *timestep* es el intervalo de tiempo que transcurre entre un cálculo de magnitudes y otro. Mientras menor sea el *timestep*, el simulador será más preciso pero también consumirá más tiempo de cómputo.

2.1.2. Modelo de fricción y su verificación

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

[4]http://bulletphysics.org/Bullet/phpBB3/viewtopic.php?f=17&t=9978

En base a los modelos físico-matemáticos que representan a cada uno de los dos fenómenos en cuestión (y que se explicarán a continuación), y pensando en posibles futuras simulaciones de varios humanoides chocando e interactuando entre sí; se llevaron a cabo dos ensayos para verificar el funcionamiento del simulador físico *Bullet*.

2.1.2.1. Verificación del coeficiente de fricción

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

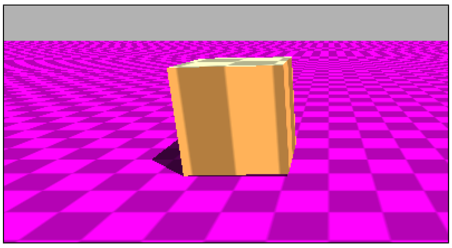


Figura 1: Visualización del sistema del cubo

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

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

En este caso, el cuerpo empieza su movimiento en el origen, por lo tanto la posición inicial (x_i) es cero. v_i es la velocidad inicial, y a, la aceleración. Debido a la fricción entre el cuerpo y el suelo, se genera una fuerza de rozamiento $\boldsymbol{F_{\mu_d}}$[5] (ec. (2)) en la misma dirección que la velocidad del sólido y en sentido contrario.

$$-\boldsymbol{F_{\mu_d}} = \mu_d \boldsymbol{F_N} \tag{2}$$

donde $\boldsymbol{F_N} = mg$ es la fuerza normal que actúa sobre la caja de masa m por acción de la gravedad g, y μ_d es el coeficiente de fricción dinámico.

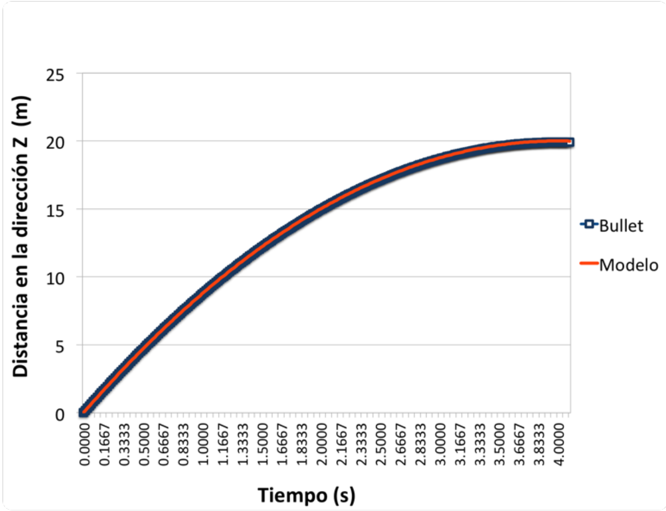
Finalmente, se obtiene la aceleración:

$$a = \frac{\boldsymbol{F_{\mu_d}}}{m} = \frac{-\mu_d \boldsymbol{F_N}}{m} = \frac{-\mu_d mg}{m} = -\mu_d mg \tag{3}$$

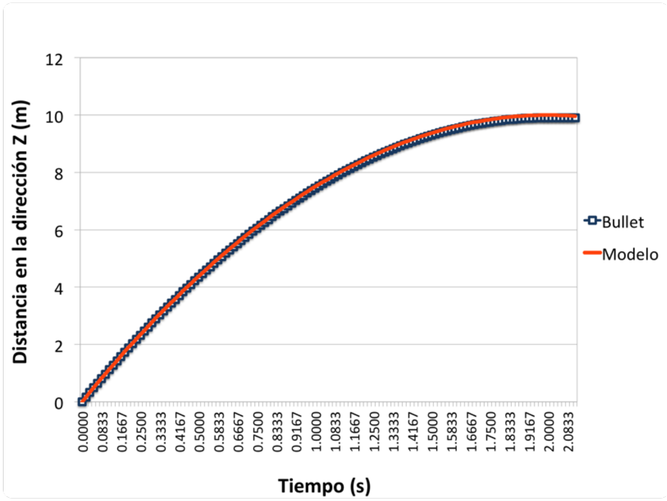
Considerando las ec. (1) y (3), se puede obtener el modelo matemático que predice el movimiento de la caja:

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

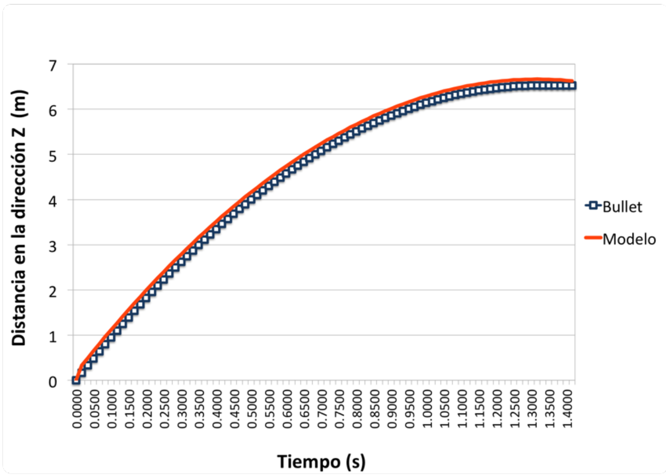
[5] En las ecuaciones, los vectores se escriben en negrita. Y las fuerzas son vectores.



(a)

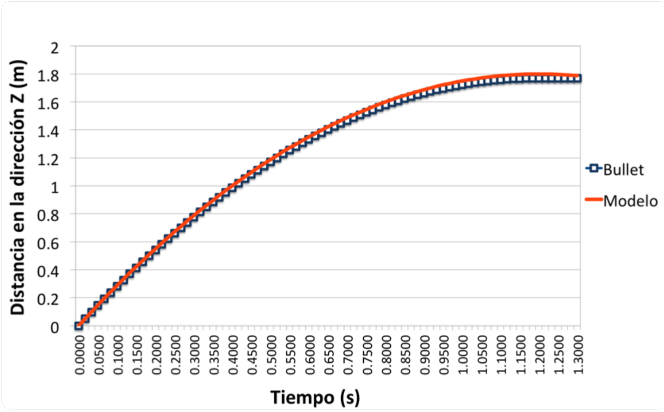


(b)

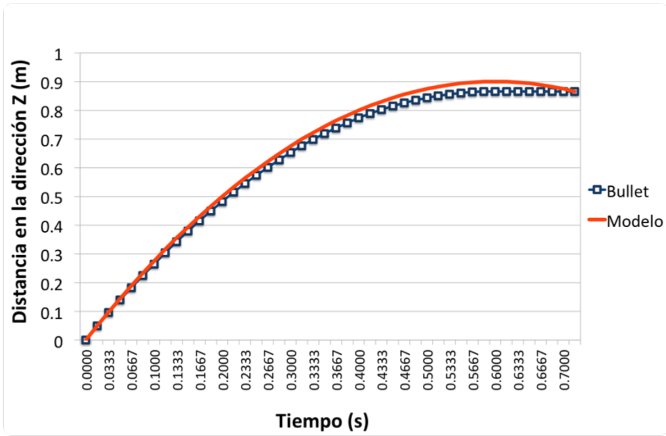


(c)

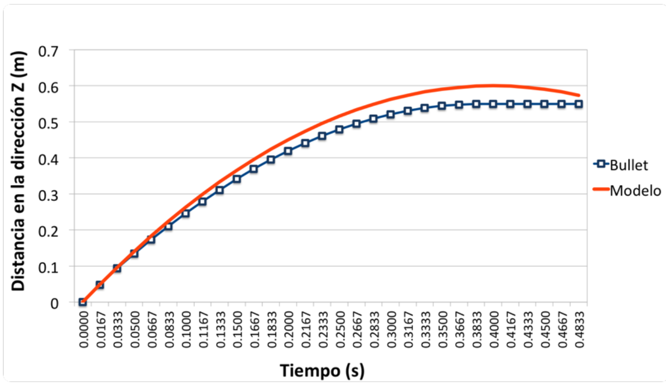
Figura 2: Resultados logrados de simular el sistema descripto en Fig. 1, usando $v_i = 10\frac{m}{s}$ y: (a) μ_d =0.25, (b) μ_d =0.50, y (c) μ_d =0.75



(a)

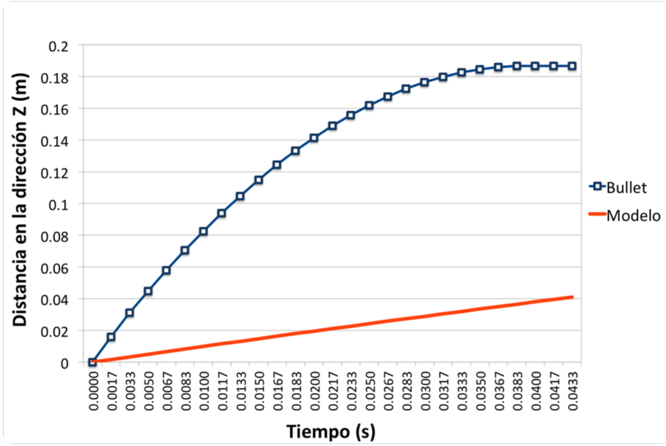


(b)

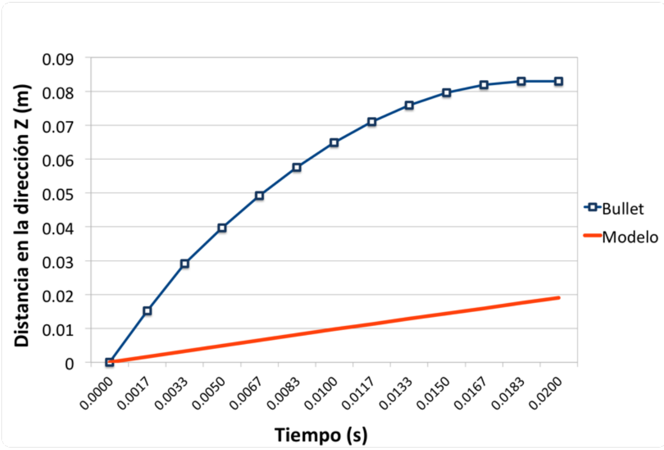


(c)

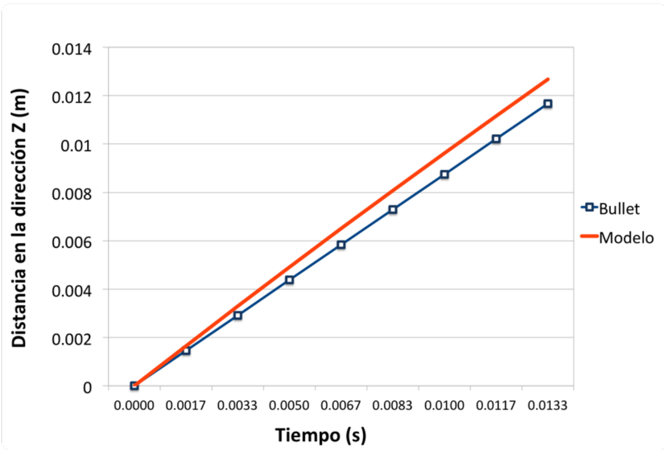
Figura 3: Resultados logrados de simular el sistema descripto en Fig. 1, usando $v_i = 3\frac{m}{s}$ y: (a) μ_d =0.25, (b) μ_d =0.50, y (c) μ_d =0.75



(a)



(b)



(c)

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

Los gráficos (Fig. 2, 3 y 4) demuestran que los resultados de las pruebas fueron favorables. Los valores obtenidos a partir de la simulación se corresponden con los alcanzados a partir del modelo matemático. Esto es un indicador de que *Bullet* debe estar usando dichos modelos para ejecutar las simulaciones.

No obstante, cabe destacar que, los gráficos que corresponden a la Fig. 4, presentan una discrepancia mayor entre la simulación y el modelo. Este hecho puede deberse a que cuanto menor sea la cantidad de puntos tomados, menor será la resolución de la simulación, y mayor será el error cometido.

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 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:

$$e = \frac{v_f}{v_i} \tag{5}$$

Para efectuar la colisión con el suelo, se empleó una esfera sólida ubicada a 4 metros del suelo, cuya masa y radio son $m_{sphere} = 1\ kg$ y $r_{sphere} = 1\ m$, respectivamente (ver Fig. 5). A la esfera se le asigna, además, un coeficiente de restitución determinado.
Se eligió un ambiente sin gravedad ($g = 0\frac{m}{s^2}$). De esta forma, se podrá tener en cuenta sólo la velocidad inicial (v_i) y la velocidad final (v_f) para el cálculo del coeficiente de restitución (e) (ver ec. (5)).
El intervalo de tiempo físico (o *timestep*) utilizado es Δt =0.001 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 μ =0.75.

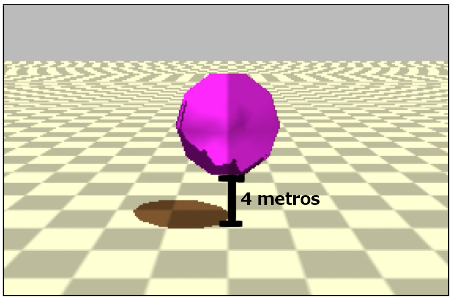


Figura 5: Visualización del sistema de la esfera

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

$$\epsilon_{rel} = \frac{e_{sim} - e_{medida}}{e_{sim}} \tag{6}$$

Se muestran los experimentos numéricos realizados. En ellos, $v_i = \{$ -0.5, -3.5, -4, -5 y -10$\}\frac{m}{s}$ y $e_{sim} = \{0.2, 0.5 \text{ y } 0.8\}$.

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

Tabla 1: Coeficientes de restitución obtenidos de simular el sistema descripto en Fig. 5

Los resultados exponen una limitación del motor físico: no representa correctamente las colisiones elásticas entre esferas y cuerpos rígidos, que ocurren a velocidades bajas. Esto queda en evidencia en la Tabla 1. En cada una de ellas el error fue de casi 1. La razón por la que ocurre este hecho se debe a que *Bullet* utiliza un algoritmo de colisión que frena la velocidad de un objeto que está a punto de colisionar. Haciendo esto puede evitar que los sólidos se traspasen y de esta forma se pueden realizar cálculos de fuerza más precisos.
En el caso de los ensayos, las esferas poseen una velocidad muy baja, cuando están a punto de colisionar *Bullet* reduce aún más esta velocidad y eventualmente quedan con una velocidad tan baja que al chocar contra el suelo se aplica el efecto restitutivo a esta velocidad casi nula y se resuelve que la esfera debe quedar en reposo, cuando en realidad debería poseer una velocidad baja, pero no despreciable.

2.1.3. Ventajas

Las ventajas del motor físico son:

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

2.1.4. Desventajas

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

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

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

2.2. Librería de Algoritmos Genéticos

Se utilizó la conocida librería de Algoritmos Genéticos para C++ GaLib, desarrollada por Matthew Wall del MIT [6].
Ofrece funcionalidades como: programación paralela, diversos métodos de selección (*elite*, *roulette*), estrategias de reemplazo (de padres, aleatorio, del peor), entre otras.

2.3. Código Fuente

Al estar *Bullet* implementado en C++, el código fuente también está desarrollado en ese lenguaje. En *Bullet*, se define un *World* (o mundo físico) en donde se puede insertar, entre otras cosas, cuerpos rígidos. En este caso en particular, el mundo consta de un plano (el suelo) y el humanoide encima (compuesto por cuerpos rígidos y otros elementos físicos).
El *software* creado incluye: creación del humanoide, siendo éste capaz de desplazarse por medio de actuadores (que se verán en la Sección 4); el desarrollo del algoritmo genético (la definición de los individuos, la función de *fitness*, métodos de selección, etc.); visualización gráfica del mejor humanoide logrado por el algoritmo genético; y la posibilidad de realizar gráficos referidos a la evolución del algoritmo genético (*fitness* por cada generación, etc.).
Se acompañan a esta presentación: el código fuente y el manual de instalación y uso.

3. Modelo Utilizado

Dentro de los diversos modelos existentes ([7][8]); en este proyecto se procuró utilizar uno que fuera sencillo pero representativo a la vez.
Se modela al cuerpo humano, con el motor *Bullet Physics*, como un conjunto de segmentos unidos por articulaciones (Fig. 6). A cada uno de ellos se les aplica una fuerza en el centro de masa de cada segmento (denominada Actuador). Que la caminata se produzca o no, depende del tipo de actuador utilizado (la función utilizada para la fuerza), y de sus parámetros. El objetivo, entonces, se reduce a encontrar dichos parámetros. Para eso se usan los algoritmos genéticos, un método de Inteligencia Artificial. De este modo, se obtiene, de forma análoga a la selección natural, los individuos que mejor se adapten a la caminata. Tanto los actuadores como el algoritmo genético se explicarán más adelante.

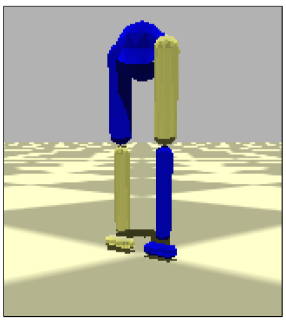


Figura 6: Humanoide diseñado

3.1. Composición Física del Humanoide

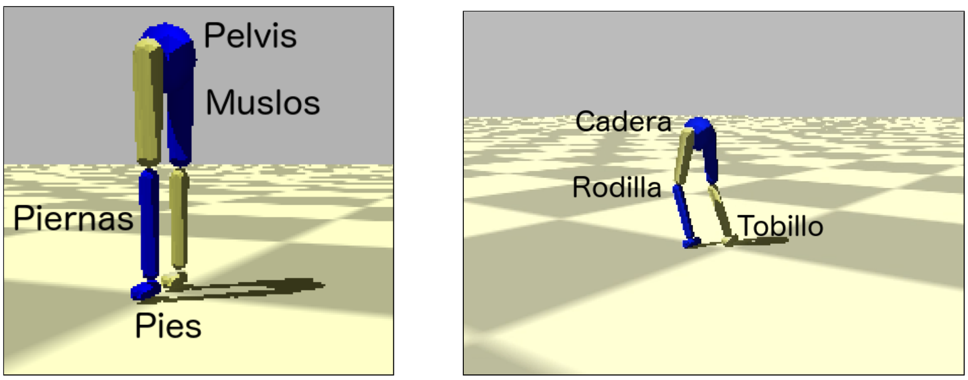
Como ya se expresó, el humanoide fue modelado en *Bullet* como un conjunto de segmentos (cuerpos rígidos), unidos por articulaciones.
Se dividió al cuerpo humano en: cabeza, tronco, miembro superior, pelvis y miembro inferior (muslo, pierna y pie). Considerar la mitad superior del cuerpo implicaba mayor complejidad (manejo de equilibrio, humanoide más pesado y con mayor volumen, etc.). Ergo, solo se tomó la pelvis y el miembro inferior (ver Fig. 7).
A continuación se presenta la composición de cada segmento (de acuerdo a la biomecánica [9]).

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

3.2. Articulaciones

Para unir los distintos segmentos entre sí, se utilizaron articulaciones bisagra con 1 grado de libertad: en el eje Z, donde ocurre la caminata; y en el eje Y, perpendicular al piso (ver Fig. 7).
Asimismo, para cada caso en particular, se definieron cotas para los ángulos que pueden existir entre los segmentos. 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.
Por otra parte, a la pelvis se le restringe todo tipo de rotación.



(a) (b)

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.
Es necesario aclarar que la caminata producida por el humanoide es plana (en 2D y no en 3D, como debería ser en una caminata real). Esto se debe a que la trayectoria pensada para el bípedo es una línea recta, y logrando que los segmentos se muevan en un solo eje es suficiente para cumplir con dicha trayectoria. También contribuye el hecho de que el torque se aplique en una sola dimensión. Por otra parte, los actuadores definidos a continuación son periódicos, y por eso no se pueden aplicar en el eje Z de los segmentos (se necesitarían actuadores reactivos, para poder detectar cuando el humanoide se está cayendo, etc.).
Para indicar el módulo de dicho torque, se diseñaron diferentes funciones (todas ellas periódicas), mencionadas en las subsecciones que siguen.

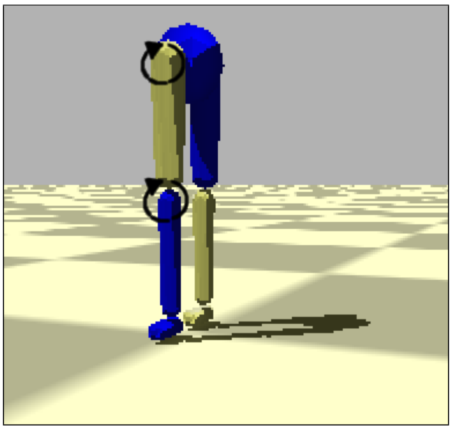


Figura 8: Aplicación de los actuadores en los segmentos del bípedo

4.1. Genérico

Es el actuador más sencillo, tanto matemática como computacionalmente.

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

donde $f(t)$ es la función del actuador evaluada en el instante de tiempo t, A_1 y A_2 son amplitudes, ω_1 y ω_2 son frecuencias (en $\frac{1}{s}$), ϕ es la fase (en radianes), y C es un término independiente.
La fase ϕ es la misma en el seno y en el coseno, para evitar que se formen otro tipo de funciones no cíclicas.

4.2. Fourier de orden 2

Este actuador utiliza una serie de Fourier de dos términos.

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

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

4.3. Fourier de orden 9

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

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

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

4.4. Doble coseno

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

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.

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

$$\omega = \frac{2\omega_1\omega_2}{\omega_1 + \omega_2} \tag{11}$$

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

5. Condiciones iniciales y de contorno

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

5.1. Función Partida

El andar del humanoide es cíclico. Sin embargo, por la posición inicial del individuo, se requiere 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 ($\frac{\pi}{2}$):

$$f_i(t) = f(t) \tag{13}$$

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

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

6. Algoritmo Genético

6.1. Individuo

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

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).

6.1.2. Valores

Cada uno de los segmentos tiene propiedades físicas distintas (masa, largo, etc.), razón por la cual no necesariamente sus genes deban tener los mismos rangos de valores, tal como puede 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	$-\pi$	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 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
Doble Coseno	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

6.1.3. Implementación de individuos

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

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

Tabla 5: Tipo de individuos

6.1.4. Constitución del cromosoma

La longitud del cromosoma de un individuo depende de los actuadores y la función partida usados. Cada uno de ellos, a su vez, tienen los parámetros presentados en la sección 6.1.1. Sus respectivas cantidades pueden verse en la Tabla **??**.
En la Tabla 7 se muestra la composición del cromosoma de cada individuo. En ella se puede observar cómo según el tipo de individuo, varía la cantidad de genes, es decir, el tamaño del cromosoma. Vale aclarar que la función partida se especifica para cada segmento (las dos piernas y los dos muslos); en cambio, para los actuadores, sólo se definen dos (uno para las piernas y otro para los muslos).

		Parámetros			
		Amplitud	**Frecuencia**	**Fase**	**Término independiente**
Actuadores	**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

		Individuos				
		Tipo 1	**Tipo 2**	**Tipo 3**	**Tipo 4**	**Tipo 5**
Parámetros	**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
Totales		18	30	34	62	28

Tabla 7: Cantidad de parámetros según tipo de individuos

6.2. Fitness

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

$$F = H * V * D * S * PA \tag{15}$$

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

6.2.1. Altura

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

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

donde t_0 es el tiempo inicial, t_T el tiempo final, h_{t_n} es la altura de la pelvis en el instante de tiempo t_n, N la cantidad pasos de simulación y C una constante $C = 5$.
Se calcula a partir de la diferencia entre la altura en cada instante de la simulación, con su altura inicial (la altura está definida como la posición de la pelvis en el eje Z). Cuanto mayor sea esa diferencia, más rápido el individuo cae, y por eso este módulo tiende a cero. Por el contrario,

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

6.2.2. Velocidad

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

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

donde t_0 es el tiempo inicial, t_T el tiempo final, v_{t_n} es la velocidad y V_O la velocidad objetivo en el eje Z (el eje de la caminata).
Sigue una lógica y cálculo similares al factor de altura: a mayor discrepancia de la velocidad real del humanoide con V_O, menor (y más cercano a cero) es el valor arrojado por el módulo de velocidad.

6.2.3. Dirección

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

$$D = \frac{\sum_{n=0}^{T} e^{-C(\boldsymbol{v_{t_n}} \cdot \boldsymbol{V_O} - 1)^2}}{N} \tag{18}$$

donde t_0 es el tiempo inicial, t_T el tiempo final, $\boldsymbol{v_{t_n}}$ el versor de la dirección del humanoide en el momento t_n y $\boldsymbol{V_O}$ el versor de la dirección objetivo.
El producto escalar entre los versores responde a la Similitud Coseno: $\cos\theta = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{AB}$, donde $\mathbf{A}$ y $\mathbf{B}$ son vectores que no se encuentran normalizados, y θ es el ángulo formado entre ellos. Así, si $\cos\theta = 1$, significa que los vectores están paralelos entre sí (que es el efecto buscado en el caso de la dirección).
Al producto escalar se le resta 1, para que el módulo sea consistente con la función exponencial utilizada y que valga 1 cuando $\theta = 0$, y 0 cuando $\theta = \pi$. Cabe aclarar que se trata al ángulo en forma simétrica, ya que, por ejemplo $\cos(-\pi/6) = \cos(\pi/6)$.

6.2.4. Simetría

Señala qué tan equidistantes se encuentran los pies de la cadera, a lo largo de la caminata. Aplicando solamente los módulos antes mencionados, provocaba resultados en donde una pierna quedaba más distante de la pelvis que la otra, lo que provocaba que el humanoide se terminara arrastrando (y posiblemente afectando a la velocidad).
Para mayor simplicidad, la simetría S se calculó a partir de los pies (y no de las piernas). Se tomaron en cuenta sólo los ejes X y Z, porque son los relacionados a la velocidad y a la dirección, respectivamente.

$$S = \frac{\sum_{n=0}^{T} \frac{1}{2}[e^{-C(lf_Z + rf_Z|^2)} + e^{-C(lf_X + rf_X|^2)}]}{N} \tag{19}$$

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

6.2.5. Pies abajo

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

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

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

6.3. Parámetros del Algoritmo

6.3.1. Métodos de selección

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

6.3.2. Métodos de cruza

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

6.3.3. Mutación

En el caso de la mutación, para cada gen del individuo, se decide con cierta probabilidad si se lo muta o no. En caso afirmativo, se cambia ese gen por un valor aleatorio (que esté dentro de su rango definido).
Dicha probabilidad de mutación es de 0.3.

7. Resultados Obtenidos

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

7.1. Evolución del *fitness* según tipo de individuo

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

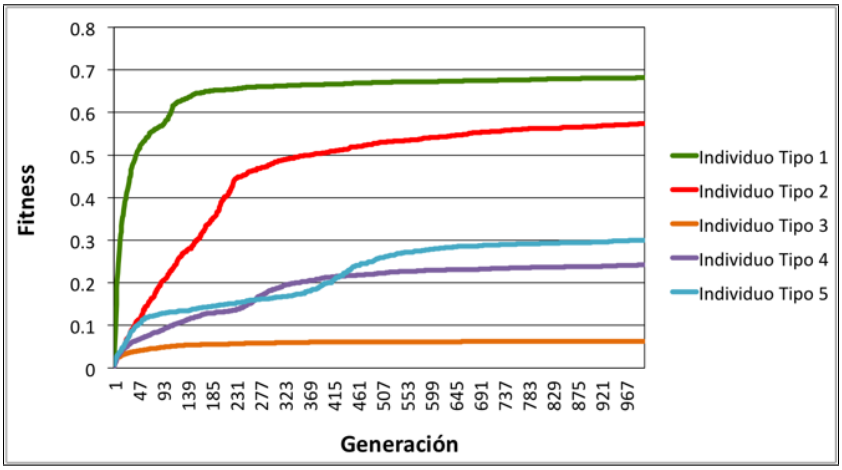


Figura 9: Evolución del *fitness* según tipo de individuo

7.2. Velocidad según tipo de individuo

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

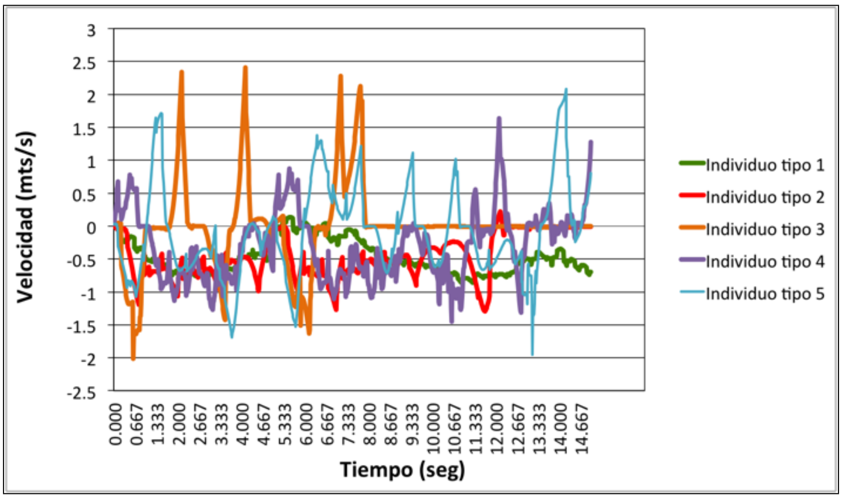


Figura 10: Velocidad según tipo de individuo

Como puede observarse en el caso de los individuos 3, 4 y 5 (Fourier de orden 2, Fourier de orden 9 y Doble Coseno, respectivamente), se producen picos altos y pronunciados en la velocidad. Eso repercute en que la velocidad media no sea 1.3 $\frac{m}{s}$ (que es la velocidad objetivo), y ergo, en el módulo de velocidad del *fitness* (provocando que éste sea más bajo).

En el caso de los individuos 1 y 2, la velocidad instantánea oscila de forma suave.

7.3. Altura según tipo de individuo

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

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

Por su parte, el individuo de tipo 1 es el que mantiene la altura constante, a causa del riel que tiene activado (que le hace mantener la altura). El individuo de tipo 2, con actuadores genéricos y función partida, logra mantener la altura, hasta que cae. Considerando que no tiene el riel activado, y que cuando cae no vuelve a levantarse (ya que en ese movimiento suelen emplearse los brazos y manos), tiene un comportamiento similar a una caminata real.

En lo que respecta al individuo de tipo 5, que utiliza como actuadores la función de doble coseno, se cae y levanta repetidamente, pero de forma más suave que los que utilizan Fourier de orden 2 y 9.

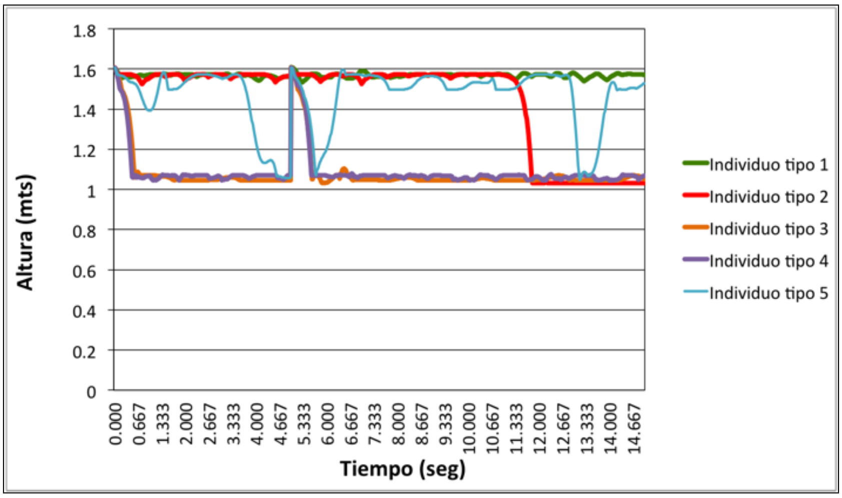


Figura 11: Altura (posición de la pelvis) según tipo de individuo

7.4. Comparación de tipos de individuo

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

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

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

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

Los actuadores fueron importantísimos.

7.5. Video

Acá iría el link del video?

8. Conclusiones

xxxxxxxx

Referencias

[1] Wikipedia: *https : //es.wikipedia.org/wiki/Physics_engine*

[2] Andreas Gerndt y otros, *An Evaluation of Open Source Physics Engines for Use in Virtual Reality Assembly Simulations. Fecha de publicación: 2012*

[3] Tom Erez y otros, *Simulation Tools for Model-Based Robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX*

[4] Sitio web de Bullet Physics: *http : //www.bulletphysics.org/*

[5] Erin Catto, *Iterative Dynamic with Temporal Coherence. Fecha de publicación: 2005*

[6] Sitio web de GaLib: *http : //lancet.mit.edu/ga/*

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

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

[9] *http : //www.exrx.net/Kinesiology/Segments.html*

[10] Kevin Kenny, Máximo Videla y Axel Wassington, *Proyecto Final para la obtención del título: Ingeniero en Informática - ITBA, 2014*