



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN
LABORATORIO DE BIO-ROBÓTICA

APRENDIZAJE DE MOVIMIENTOS BÁSICOS DE UN ROBOT BÍPEDO
MEDIANTE TÉCNICAS DE NEUROEVOLUCIÓN

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN CIENCIAS (COMPUTACIÓN)

PRESENTA:
PABLO ALBERTO MONROY LLAGUNO

TUTOR:
DR. JESÚS SAVAGE CARMONA
FACULTAD DE INGENIERÍA

MÉXICO, D. F. ENERO 2016



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Resumen

La presente tesis describe la integración de un sistema de aprendizaje automatizado basado en la neuroevolución (redes neuronales artificiales optimizadas a través de algoritmos genéticos) y un simulador basado en física, dicho sistema se crea con el objetivo de generar controladores de movimiento que puedan lograr que un robot humanoide bípedo se desplace en el ambiente. El caminado bípedo se explica junto con metodologías que se han desarrollado y empleado buscando resolver este complicado problema en robots actuales, también se tratan los principios detrás de la neuroevolución para luego detallar y caracterizar la técnica de aprendizaje neuroevolutiva y simulador que se va a utilizar. Posteriormente se explican los elementos involucrados en el modelo del robot virtual junto con todos los aspectos involucrados en su interacción con el simulador que provee el medio en donde los controladores se van a evaluar y medir. Los experimentos y resultados presentados, muestran que el sistema de aprendizaje logra su cometido y genera controladores que hacen que los modelos virtuales se comporten de diferente manera para moverse en el ambiente.

Abstract

This thesis describes the integration of an automated learning system based on neuroevolution (optimization of artificial neural networks via genetic algorithms) and a physics based simulator, such a system is developed to generate movement controllers that are capable of achieving that a bipedal humanoid robot moves in the environment. Bipedal walking is explained along with methodology developed and used looking to solve this complicated problem in today's robots, the principles behind neuroevolution are also explained and then the specifics of the neuroevolution

learning technique and simulator used are detailed. The elements involved in the virtual robot model along with all aspects involved in its interaction with the simulator that provides the means to evaluate and measure the controller are explained subsequently. The presented experiments and results, show that the learning system fulfills its purpose and generates controllers that make the virtual models behave differently to move in the environment.

“La evolución es más inteligente que tú”

Leslie E. Orgel, segunda ley de Orgel

Agradecimientos

A todos los tutores, compañeros de clase, administrativos y demás personal del *Posgrado en Ciencia e Ingeniería de la Computación*, con su trabajo y dedicación hacen posible que este programa de posgrado sea una experiencia sumamente enriquecedora.

Al Dr. Jesús Savage Carmona y compañeros del laboratorio de Bio-Robótica por darme la oportunidad de colaborar con ellos, pero más que nada por la amistad, el apoyo y los consejos que he recibido durante mis estudios y la realización del presente trabajo.

Al Mtro. Luis Fernando Lupián Sánchez de la *Universidad La Salle* y al Dr. Alejandro Aceves López del *Instituto Tecnológico y de Estudios Superiores de Monterrey*, así como a todos sus alumnos con quienes tuve el gusto de participar en competencias como un mismo equipo.

A todas aquellas personas de las que tengo el honor de ser un amigo sin importar las cortas o grandes distancias que nos llegan a separar.

Al *CONACyT* por la beca otorgada durante mis estudios de posgrado.

A la *DGAPA-UNAM* por el apoyo proporcionado para la realización de esta tesis a través del proyecto *PAPIIT IG100915* “Desarrollo de técnicas de la robótica aplicadas a las artes escénicas y visuales”.

A la Federación Mexicana de Robótica por la labor de promover la investigación, organizar los torneos nacionales y proporcionar apoyos para la asistencia a torneos internacionales.

Finalmente, a mis padres y hermanos de quienes he recibido lo más importante: amor, apoyo incondicional y siempre un buen ejemplo a seguir.

1. Introducción	1
1.1. Presentación	2
1.2. Objetivos	4
1.3. Motivación	5
1.4. Organización de la tesis	7
2. Marco teórico y antecedentes	8
2.1. Métodos usados para solucionar el problema de caminado bípedo	12
2.1.1. Caminado estático	15
2.1.2. Punto de momento cero	16
2.1.3. Caminador pasivo dinámico	19
2.1.4. Generador central de patrones	20
2.2. Neuroevolución y robótica evolutiva	23
2.2.1. Algoritmos genéticos	25
2.2.2. Redes neuronales artificiales	27
2.3. Trabajo relacionado	35
2.3.1. Estado del arte	36
3. Técnica neuroevolutiva y simulador	40
3.1. Algoritmo NEAT	40
3.1.1. El genoma y fenoma	40
3.1.2. El número de innovación	41

3.1.3.	Las mutaciones	42
3.1.4.	La cruza	43
3.1.5.	La especiación y reproducción	44
3.1.6.	La minimización de la dimensionalidad	47
3.1.7.	Ventajas y desventajas del algoritmo	48
3.2.	Simulador basado en física	50
3.2.1.	Simulador Gazebo	51
3.2.2.	Detalles importantes de ODE	53
4.	Implementación	55
4.1.	Modelo del robot y propiedades del ambiente simulado	55
4.1.1.	Segmentos	55
4.1.2.	Articulaciones	59
4.1.3.	Ambiente virtual	62
4.2.	Sensores, métricas y el control de los actuadores	63
4.2.1.	Sensores	63
4.2.2.	Control PID	65
4.2.3.	Métricas	66
4.3.	Integración de NEAT e interfaz con el simulador Gazebo	68
4.4.	Funciones de aptitud y su importancia	73
4.5.	Pruebas de concepto	75
4.5.1.	Especificaciones del modelo virtual del péndulo invertido simple	75
4.5.2.	Especificaciones del modelo virtual del péndulo invertido doble	76
5.	Experimentos y resultados	79
5.1.	Péndulo invertido simple	79
5.1.1.	Experimento sin perturbaciones	80
5.1.2.	Experimento con perturbaciones	85
5.2.	Péndulo invertido doble	88
5.2.1.	Experimento sin perturbaciones	89
5.2.2.	Experimento con perturbaciones	92
5.3.	Robot bípedo	96
5.3.1.	Experimento 1, movimiento por saltos	97

5.3.2. Experimento 2, movimiento por deslizamiento	107
6. Conclusiones	115
6.1. Discusión	115
6.2. Trabajo futuro	117
Apéndice	119
A. Conceptos de física	120
B. Parámetros y diagrama de clases en NEAT	126
B.1. Parámetros	126
B.2. Diagrama de clases	129
Bibliografía	130

Índice de figuras

1-1. El robot humanoide NimbRo-OP	4
2-1. Los planos principales en los que se divide el cuerpo humano.	11
2-2. Las fases del caminado desde una perspectiva sagital.	12
2-3. Clasificación del control de locomoción.	15
2-4. Modelo de control del robot ASIMO de Honda, un ejemplo del estado de arte que emplea ZMP.	16
2-5. Las fuerzas y momentos que actúan en el pie durante el soporte simple.	18
2-6. Control por CPG de una salamandra robótica.	21
2-7. Un oscilador de Matsuoka.	23
2-8. El proceso de evolución en un algoritmo genético.	27
2-9. Representación de la neurona biológica y su modelo artificial.	29
2-10. Red neuronal de alimentación progresiva de una capa oculta.	31
2-11. Red neuronal recurrente.	32
2-12. Funciones de activación.	33
2-13. Elementos del modelo de Hill.	36
2-14. Ejemplo de puntos de fijación en los cuerpos rígidos y las regiones permitidas.	37
2-15. El sistema de control muscular.	38
3-1. Genoma y fenoma de NEAT.	42
3-2. Mutación en NEAT.	44
3-3. Cruza en NEAT.	45

3-4. Ejemplos de redes evolucionadas con NEAT.	50
3-5. Ciclo de simulación.	51
3-6. Plataformas robóticas que se han simulado con Gazebo.	52
4-1. Representación del robot NimbRo-OP como cadena cinemática.	56
4-2. Modelos virtuales del caminador.	56
4-3. Algunos segmentos y su punto de origen.	57
4-4. Articulaciones del caminador.	61
4-5. Sensores del caminador.	64
4-6. Diagrama de bloques de un controlador PID.	65
4-7. Diagrama de flujo del sistema de aprendizaje.	69
4-8. Ventana principal para el control de NEAT.	71
4-9. Ventana para la interacción manual con Gazebo.	72
4-10. Ventana para la evaluación del comportamiento de individuos.	72
4-11. Clases de funciones de aptitud / Grado de conocimiento incorporado.	74
4-12. Modelo virtual del péndulo simple.	75
4-13. Modelo virtual del péndulo doble.	77
5-1. Aptitud por generación del péndulo invertido simple sin perturbaciones.	82
5-2. Nodos por generación del péndulo invertido simple sin perturbaciones.	82
5-3. Conexiones por generación del péndulo invertido simple sin perturbaciones.	83
5-4. Especies de la última generación del péndulo invertido simple sin perturbaciones.	83
5-5. Comportamiento del péndulo invertido simple bajo el control del mejor individuo de la última generación (60).	84
5-6. Aptitud por generación del péndulo invertido simple con perturbaciones.	85
5-7. Nodos por generación del péndulo invertido simple con perturbaciones.	86
5-8. Conexiones por generación del péndulo invertido simple con perturbaciones.	86
5-9. Especies de la última generación del péndulo invertido simple con perturbaciones.	87
5-10. Comportamiento del péndulo invertido simple bajo el control del mejor individuo de la última generación (200).	87
5-11. Aptitud por generación del péndulo invertido doble sin perturbaciones.	89
5-12. Nodos por generación del péndulo invertido doble sin perturbaciones.	90
5-13. Conexiones por generación del péndulo invertido doble sin perturbaciones.	90

5-14. Especies de la última generación del péndulo invertido doble sin perturbaciones.	91
5-15. Comportamiento del péndulo invertido doble bajo el control del mejor individuo de la última generación (400).	91
5-16. Aptitud por generación del péndulo invertido doble con perturbaciones.	92
5-17. Nodos por generación del péndulo invertido doble con perturbaciones.	93
5-18. Conexiones por generación del péndulo invertido doble con perturbaciones.	93
5-19. Especies de la última generación del péndulo invertido doble con perturbaciones.	94
5-20. Comportamiento del péndulo invertido doble bajo el control del mejor individuo de la última generación (536).	94
5-21. Estructura de red del mejor individuo de la última generación (536).	95
5-22. Aptitud por generación del experimento 1.	100
5-23. Nodos por generación del experimento 1.	101
5-24. Conexiones por generación del experimento 1.	102
5-25. Ruta promedio seguida por el caminador del experimento 1.	102
5-26. Respuesta promedio del acelerómetro en el caminador del experimento 1.	103
5-27. Respuesta promedio del giroscopio en el caminador del experimento 1.	104
5-28. Ruta seguida por el caminador en la mejor prueba del experimento 1.	104
5-29. Respuesta del acelerómetro en la mejor prueba del caminador del experimento 1.	105
5-30. Respuesta del giroscopio en la mejor prueba del caminador del experimento 1.	105
5-31. Secuencia visual del comportamiento del experimento 1.	106
5-32. Aptitud por generación del experimento 2.	109
5-33. Nodos por generación del experimento 2.	110
5-34. Conexiones por generación del experimento 2.	110
5-35. Ruta promedio seguida por el caminador del experimento 2.	111
5-36. Respuesta promedio del acelerómetro en el caminador del experimento 2.	112
5-37. Respuesta promedio del giroscopio en el caminador del experimento 2.	112
5-38. Ruta seguida por el caminador en la mejor prueba del experimento 2.	113
5-39. Respuesta del acelerómetro en la mejor prueba del caminador del experimento 2.	113
5-40. Respuesta del giroscopio en la mejor prueba del caminador del experimento 2.	114
5-41. Secuencia visual del comportamiento del experimento 2.	114
B-1. Diagrama de las clases principales en NEAT.	129

Índice de tablas

4-1. Esta tabla indica los atributos de los segmentos del robot virtual, todas las unidades de propiedades físicas son las estándares del sistema internacional.	58
4-2. Esta tabla indica los atributos de las articulaciones del robot virtual, todas las unidades de propiedades físicas son las estándares del sistema internacional.	60
4-3. Esta tabla indica los parámetros del simulador y motor de física.	62
4-4. Esta tabla indica los atributos de los segmentos del péndulo invertido simple, todas las unidades de propiedades físicas son las estándares del sistema internacional. . . .	76
4-5. Esta tabla indica los atributos de las articulaciones del péndulo invertido simple, todas las unidades de propiedades físicas son las estándares del sistema internacional.	76
4-6. Esta tabla indica los atributos de los segmentos del péndulo invertido doble, todas las unidades de propiedades físicas son las estándares del sistema internacional. . . .	77
4-7. Esta tabla indica los atributos de las articulaciones del péndulo invertido doble, todas las unidades de propiedades físicas son las estándares del sistema internacional. . . .	78
5-1. Parámetros de NEAT en el experimento de péndulo invertido simple.	80
5-2. Parámetros de NEAT en el experimento de péndulo invertido simple.	88
5-3. Parámetros de NEAT en el experimento 1 del caminador.	98
5-4. Parámetros extra del experimento 1 del caminador.	99
5-5. Resultados de 20 pruebas de comportamiento del mejor individuo obtenido en el experimento 1.	103
5-6. Parámetros de NEAT en el experimento 2 del caminador.	107
5-7. Parámetros extra del experimento 2 del caminador.	108

5-8. Resultados de 20 pruebas de comportamiento del mejor individuo obtenido en el experimento 2. 111

Acrónimos

- NEAT** Neuroevolución de topologías incrementales (**NeuroEvolution of Augmenting Topologies**)
- API** Interfaz de programación de aplicaciones (**Application Programming Interface**)
- ANN** Red neuronal artificial (**Artificial Neural Network**)
- RNN** Red neuronal recurrente (**Recurrent Neural Network**)
- GA** Algoritmo genético (**Genetic Algorithm**)
- CPG** Generador central de patrones (**Central Pattern Generator**)
- CMA-ES** Estrategia evolutiva de adaptación de matriz de covarianza (**Covariance Matrix Adaptation Evolution Strategy**)
- ODE** Motor de dinámicas libre (**Open Dynamics Engine**)
- DT** Tiempo discreto (**Discrete Time**)
- CT** Tiempo continuo (**Continuous Time**)
- CoM** Centro de masa (**Center of Mass**)
- CoG** Centro de gravedad (**Center of Gravity**)
- CoP** Centro de presión (**Center of Pressure**)
- CoF** Coeficiente de fricción (**Coefficient of Friction**)
- CW** Sentido de las manecillas del reloj (**Clock Wise**)

CCW Contra sentido de las manecillas del reloj (**C**ounter **C**lock **W**ise)

DoF Grado de libertad (**D**egree **o**f **F**reedom)

IMU Unidad de medición inercial (**I**nertial **M**easurement **U**nit)

PDW Caminador dinámico pasivo (**P**assive **D**ynamic **W**alker)

PD Proporcional-Derivativo (**P**roportional-**D**erivative)

PID Proporcional-Integral-Derivativo (**P**roportional-**I**ntegral-**D**erivative)

ROS Sistema operativo para robots (**R**obot **O**perating **S**ystem)

SDF Formato de descripción de la simulación (**S**imulation **D**escription **F**ormat)

ZMP Punto de momento cero (**Z**ero **M**oment **P**oint)

EANN Redes neuronales artificiales evolutivas (Evolutionary **ANN**)

TWEANN Redes neuronales artificiales de topología y pesos en evolución (**T**opology and **W**eight
Evolving **ANN**)

BoS Base de soporte (**B**ase **o**f **S**upport)

En las últimas décadas, con el avance de la ciencia y la tecnología se ha visto crecer el campo de aplicación de la robótica.

En los años 60's aproximadamente, el primer campo en beneficiarse del uso de robots fue la industria, en donde un robot capaz de realizar tareas de precisión, peligrosas, repetitivas, desagradables o que simplemente un humano no puede realizar pasó de ser una curiosidad, a ser un elemento indispensable y ampliamente utilizado en las industrias de manufactura actuales. Estos robots se denominan *robots industriales* y se caracterizan por ser en su mayoría manipuladores multifuncionales estacionarios o con desplazamiento limitado que realizan sus tareas en un ambiente estructurado, predecible y restringido. Al implementar estos robots, se procura obtener modelos dinámicos precisos del ambiente y del robot para crear un control capaz de realizar la tarea asignada de forma exacta, conforme el sistema (formado por la interacción ambiente-robot) cambia a lo largo del tiempo.

El otro campo de aplicación y que recientemente se ha popularizado, es aquel que involucra al robot para uso personal y profesional (excluyendo la automatización industrial que ya está cubierta por los robots industriales). Como ejemplos, tenemos las labores domésticas, entretenimiento, exploración de ambientes de riesgo, intervenciones médicas, asistencia en operaciones de seguridad pública y militares, por mencionar algunos. A este tipo de robots se les denomina *robots de servicio* y se enfrentan con retos más grandes, ya que estos actúan con un ambiente dinámico, impredecible y que puede ser solo parcialmente observable, junto con el hecho de que estos robots pueden diseñarse con diferente estructura física, estos aspectos hacen que la obtención de modelos completos del ambiente y del robot sea complicado o incluso poco factible. Esta complicación hace que surjan problemas interesantes que los robots deben resolver utilizando solo la información que está dis-

ponible del ambiente como lo es la localización, navegación, manipulación y el comportamiento inteligente entre otros. Hay un especial interés en cómo desarrollar sistemas automáticos que le permitan a los robots aprender a resolver estos problemas o a adaptar una solución a cambios que puedan presentarse.

1.1. Presentación

Una cualidad que puede considerarse esencial en el incremento de los robots de servicio, es que tienen la capacidad de moverse de un lugar a otro por medios propios (esto se denomina locomoción). Los robots que poseen esta capacidad se denominan *robot móviles* y atraen mucha investigación, debido al enorme potencial actual y a futuro que presenta su aplicación en el servicio personal y profesional, así como por el carácter multidisciplinario de su estudio, ya que involucra ciencias como la ingeniería mecánica para el diseño y construcción de mecanismos para el movimiento, ciencias computacionales para tratar las representaciones del entorno y robot, la planificación de acciones y los algoritmos de control y razonamiento, ingeniería eléctrica para la integración de sensores, sistemas y comunicaciones entre los elementos que componen el robot e incluso psicología cognitiva y neurociencias cuando se desea entender y buscar replicar cómo organismos biológicos formulan estrategias con las que resuelven problemas similares a los que un robot puede enfrentar [11].

La locomoción de una entidad se da como resultado de una o varias fuerzas que actúan entre esta y el entorno donde se ubica, por lo tanto la elección de mecanismos para lograr el movimiento de un robot debe realizarse tomando en consideración el ambiente en el que se va a desempeñar (aéreo, terrestre, acuático, espacio o combinaciones). En el caso de robots móviles terrestres, se puede optar por el uso de mecanismos simples como lo son las ruedas u orugas que son muy populares, puesto que se pueden usar en diferentes configuraciones y si se tienen suficientes puntos de apoyo, se pueden simplificar e incluso descartar problemas difíciles del movimiento como lo es mantener el equilibrio del robot cuando está o no en movimiento. Sin embargo, el empleo de ruedas como una solución, requiere que el ambiente cumpla restricciones con respecto al suelo, si este presenta desniveles, brechas u otras irregularidades las ruedas pueden atorarse y fallar, optar entonces por extremidades es una solución factible que nos puede otorgar ventajas para afrontar estos retos presentados por el terreno [42]. El uso de extremidades se relaciona con un enfoque llamado *bioinspirado* el cual esencialmente consiste en buscar soluciones partiendo de lo que se puede encontrar en los seres biológicos, los cuales presentan gran variedad de estructuras y

estrategias para solucionar problemas y desplazarse tanto en entornos simples y complejos [40] [14, Sección 3]. El estudio de los principios dinámicos y el diseño de mecanismos basados en los seres biológicos son objeto de estudio de la biomecánica.

Si se van a emplear extremidades para la movilidad de robots terrestres, hay que decidir cuantas hay que utilizar, lo más común que se ve en la naturaleza y también en robots son las configuraciones octópodos, hexápodos, cuadrúpedas o bípedas. La elección de una configuración particular depende del diseñador y los criterios que el considere de importancia como pueden ser el costo, la estabilidad deseada o restricciones impuestas por el mismo ambiente al cual el robot va a estar destinado. Para robots de servicio con el propósito de ayudar al ser humano o realizar actividades similares a las que un humano desempeña, parece natural la idea de que la estructura más apropiada para un robot se asemeje a la del cuerpo humano, en este caso un denominado *robot humanoide* que puede ser totalmente humanoide si toda su estructura es similar al cuerpo humano o solo parcialmente humanoide si algunos elementos estructurales son similares y otros difieren considerablemente. Algunos argumentos sólidos y favorables para esta elección de diseño son por un lado que la infraestructura y dispositivos que se encuentran en los ambientes humanos están diseñados para ser utilizados por el mismo ser humano, y por otro lado que el mismo cuerpo humano posee un diseño versátil probado por nosotros mismos y capaz de realizar acciones muy complejas, es deseable que una máquina creada con el propósito de hacer cosas similares tenga funcionalidad y forma equivalente [4].

Retomando la idea de locomoción y considerando el caso de un robot totalmente humanoide o parcialmente humanoide por estar dotado de dos piernas, entonces el robot es también un *robot bípedo*. La locomoción bípeda a pesar de parecer una tarea sencilla para nosotros que la realizamos día tras día, en un robot es un problema sumamente complejo. Esa complejidad se ve reflejada al momento que se requiere desarrollar un controlador que sea capaz de generar los movimientos de locomoción bípeda, se puede hacer en base a modelos dinámicos los cuales el diseñador debe encargarse de formular o se puede nuevamente dar un vistazo a la naturaleza y considerar que el proceso de evolución natural encontró controladores y estructura apropiados en seres vivos que resuelven el problema de la locomoción bípeda. Estos controladores se encuentran en la forma de sistemas nerviosos y la estructura en la forma del sistema esquelético-muscular, esta idea resulta muy interesante sobre todo si se considera que esto ocurrió muchos millones de años antes de la aparición del primer ser humano en la tierra. La *robótica evolutiva* [34] es un campo de investigación de rápido desarrollo que adopta la idea anterior y que mediante la utilización de técnicas de evolu-

ción artificial busca resolver problemas referentes al diseño automático de robots y el aprendizaje de controladores que permiten a un robot enfrentar un problema específico.

En el presente trabajo se describe la elaboración de un sistema de aprendizaje que integra una técnica de la robótica evolutiva denominada neuroevolución y un simulador basado en física para buscar controladores que sean capaces de resolver el problema de locomoción bípeda.

1.2. Objetivos

Este trabajo busca el desarrollo de un sistema que mediante la integración de, (a) una técnica neuroevolutiva y (b) un simulador para robots basado en física, facilite, a través de la interacción del robot y el ambiente virtual, el aprendizaje automático de controladores de movimiento que puedan realizar la locomoción bípeda del robot humanoide virtual que está basado en el robot físico “NimbRo-OP” [46], figura 1-1.

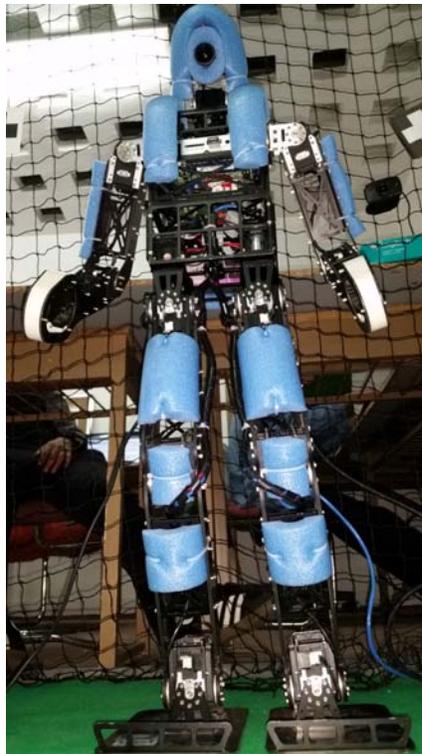


Figura 1-1: El robot humanoide NimbRo-OP

De forma específica se busca lograr lo siguiente:

- Explorar el problema de la locomoción bípeda así como enfoques y métodos que se emplean

actualmente para solucionarlo.

- Describir detalladamente la técnica evolutiva que se emplea, en este caso el algoritmo **NEAT**.
- Crear y describir los elementos de un modelo virtual del robot NimbRo-OP para que pueda interactuar con el ambiente virtual en una simulación basada en física.
- Crear una herramienta intuitiva en donde se integren y se puedan controlar tanto el algoritmo de aprendizaje neuroevolutivo y los elementos necesarios para lograr una apropiada interfaz con el simulador que debe proporcionar el entorno de prueba para el robot.
- Utilizar la herramienta desarrollada para evolucionar controladores que reciben información sensorial y generan comandos de movimiento que permitan al robot virtual moverse de forma bípeda sobre un plano horizontal sin obstáculos.
- Analizar y discutir los resultados obtenidos para definir las posibilidades de este trabajo a futuro.

1.3. Motivación

El 3 de marzo de 2011 un terremoto y tsunami ocurrieron en Japón generando graves daños en la infraestructura del país, de esto derivó una importante crisis resultado de fallas en reactores nucleares ubicados en la provincia de Fukushima. Debido al peligro que representa para un ser humano ingresar en una instalación con daños estructurales y donde los niveles de radiación son elevados, se consideró enviar un robot humanoide para poder revisar los daños y tratar de operar mecanismos que permitieran regular los niveles de radiación. Al hacerse pública esta posibilidad, la compañía japonesa Honda recibió múltiples peticiones para enviar al robot ASIMO, fruto de investigaciones en robótica humanoide y bípeda que investigadores de Honda realizan desde 1986 y considerado en ese entonces e incluso hoy día, uno de los robots humanoides más avanzados y capaces en cuestiones de manipulación y locomoción. Debido a que ASIMO no fue diseñado para moverse en un ambiente donde hay escombros e irregularidades, se tuvo que recurrir a otro tipo de robots. A raíz del desastre, la necesidad de redoblar esfuerzos con el fin de que los robots logren alcanzar los niveles de habilidad que posee un ser humano en cuestiones de locomoción y manipulación, se convirtió en foco de atención de muchos grupos de investigación y generó una iniciativa por parte de la Agencia de proyectos de investigación avanzados de defensa (DARPA

por sus siglas en inglés) conocida como el Darpa Robotics Challenge¹ que se llevó a cabo entre 2013 y 2015 con la finalidad de desarrollar robots capaces de responder a desastres asistiendo a los humanos.

A finales de enero de 2014 se recibió en el laboratorio de Bio-Robótica de la Facultad de Ingeniería de la UNAM un robot NimbRo-OP, este robot bípedo humanoide fue desarrollado por la universidad de Bonn, con el fin de ser usado como prototipo de investigación y mostró ser capaz de caminar de forma omnidireccional sobre suelo plano, siempre y cuando valores apropiados para los múltiples parámetros que utiliza el controlador de movimientos implementado en él sean proporcionados [3]. Dadas las especificaciones de este robot, se puede participar en las competencias de la RoboCup (liga de soccer), que es una competencia realizada anualmente desde que se propuso en 1997 con el fin de promover la investigación en robótica e inteligencia artificial y cuyo objetivo oficial es “Para el año 2050, un equipo de robots humanoides autónomos que juegan soccer debe ganar un juego llevado a cabo bajo las mismas reglas de la FIFA y enfrentando al equipo humano ganador de la copa mundial más reciente”. Es claro que el nivel de agilidad y movilidad que necesita un robot para cumplir con el objetivo antes descrito, todavía está lejos de ser alcanzado, es deseable que en el futuro se implementen métodos de locomoción desarrollados en el laboratorio de Bio-Robótica que sean capaces de igualar y superar los del controlador actual por lo que es muy importante iniciar la exploración de metodologías que puedan aportar a ese fin.

Una gran mayoría de los robots bípedos actuales solucionan el problema de locomoción usando métodos de control tradicionales que hacen uso del modelado cinemático y dinámico. Buscar la solución de esta manera enfrenta al investigador con la dificultad de requerir conocimiento avanzado de la metodología involucrada y del problema en cuestión para la formulación matemática de las ecuaciones y el controlador, esto es un proceso cuya dificultad se incrementa a medida que el ambiente, el robot y el comportamiento que se desea realizar se hacen más complejos. En contraste, al recurrir a la combinación de técnicas de inteligencia artificial para lograr el aprendizaje automatizado mediante la estructuración de un modelo de cómputo que funge como controlador y la respectiva optimización de sus parámetros, se busca que el comportamiento apropiado sea aprendido a través de la interacción del robot con el ambiente en la ausencia de un modelo preciso y complicado que lo describa.

Además de la robótica, otras disciplinas como la medicina y la cinematografía tienen mucho interés en el desarrollo de nuevo conocimiento y tecnología en torno al caminado bípedo, ya que puede

¹www.theroboticschallenge.org

usarse para desarrollar dispositivos de rehabilitación, prótesis o para la animación por computadora de personajes para las películas y videojuegos.

La herramienta a desarrollar en este trabajo aunque se enfoque aquí al problema de locomoción bípeda, podrá usarse en la generación de controladores que realizan otros comportamientos que pueden ayudar al avance de robots de servicio, hay que destacar que en años venideros el mercado para estos robots se espera que crezca, tal como indica un estudio de la federación internacional de robótica² (IFR de sus siglas en inglés).

1.4. Organización de la tesis

En el capítulo 2, **Marco teórico y antecedentes**, se describe el problema de locomoción bípeda haciendo énfasis en las características y dificultades que presenta, se exponen algunos enfoques y metodología que se han utilizado para resolverlo al igual que ideas y técnicas detrás de la neuroevolución y su relación con la robótica evolutiva. Finalmente se mencionan algunos trabajos relacionados y uno en especial que puede considerarse estado del arte.

En el capítulo 3, **Técnica neuroevolutiva y simulador**, se describen a detalle las peculiaridades del algoritmo **NEAT** y del simulador “Gazebo” que se utiliza para evaluar los comportamientos del robot.

En el capítulo 4, **Implementación**, se definen los elementos que se desarrollaron e integraron en el sistema experimental, la función que cumple cada uno de ellos y los retos que se presentaron.

En el capítulo 5, **Experimentos y resultados**, se muestran resultados obtenidos de los experimentos y se discute acerca de estos.

En el capítulo 6 **Conclusiones** se realiza un análisis de las posibilidades que presenta este trabajo así como que mejoras y trabajo futuro pueden realizarse.

²http://www.ifr.org/uploads/media/Executive_Summary_WR_2014.pdf

Marco teórico y antecedentes

Un robot puede describirse de diferentes maneras que pueden variar dependiendo del grado de detalle o disciplina bajo la que se analiza, en el contexto de este trabajo un robot es un sistema físico o una representación virtual que se compone de un conjunto de elementos que se relacionan entre sí, estos elementos pueden agruparse en sistemas que dotan al robot de una capacidad que se requiere para llevar a cabo el objetivo con el que fue creado el robot. En el caso de un robot móvil (de los cuales los robots bípedos son una subclase) [11] menciona que los problemas fundamentales que debe resolver son; ¿Cómo moverse en el espacio e influenciarlo?, ¿Cómo sensor el estado del espacio y el estado propio? y ¿Cómo razonar acerca de la información de los estados?, con base a estos problemas se puede considerar que un robot móvil se compone de los siguientes sistemas:

- Sistema locomotor, conformado por una estructura física que consta de segmentos (normalmente cuerpos rígidos) unidos entre si mediante articulaciones y mecanismos que le permiten al robot realizar movimientos para interactuar con otros elementos del ambiente y desplazarse en él.
- Sistema sensor, conformado por todos aquellos dispositivos sensores (producen información ante la acción de un fenómeno físico) y algoritmos de procesamiento usados para obtener mediciones del estado del ambiente (lo que se conoce como exterocepción) y del propio estado del robot (que se conoce como propiocepción).
- Sistema de razonamiento, conformado por los dispositivos de cómputo y algoritmos que realizan la toma de decisiones en base a las mediciones sensoriales y al conocimiento preprogramado o adquirido.

En la introducción se mencionaba que el movimiento de una entidad se da como resultado de fuerzas que se producen al interactuar este con el entorno, las leyes de movimiento de Newton y conceptos de dinámica y estática explican formalmente el fenómeno del movimiento (el apéndice **A** provee un breve compendio de conceptos relacionados), retomando el caso donde el sistema locomotor de un robot utiliza ruedas, los materiales del que están hechas estas y el suelo, así como la fuerza normal que se genera al contacto entre ambos por acción de la gravedad, determinan la fuerza de fricción que se produce, si esta es suficiente para evitar el deslizamiento de las ruedas sobre la superficie, entonces un mecanismo en el robot que aplique torque para hacer girar las ruedas logra que se genere un empuje contra la fuerza de fricción y por lo tanto movimiento.

El andar bípedo es un problema complejo debido a múltiples dificultades, la primera de estas dificultades que afectan al bipedismo es que es inherentemente inestable y la estabilidad es sumamente importante, ya que de esta depende que un bípedo pueda mantener el balance durante la marcha. Cuando se utilizan extremidades o piernas, estas también deben explotar las fuerzas generadas al contacto con el suelo para que se de un desplazamiento, durante la marcha, estas piernas cambian de posición en el espacio a lo largo del tiempo, lo cual se traduce en que los puntos de apoyo no siempre están presentes y esto lleva a la inestabilidad. Considérese que al tocar el suelo la parte distal de la extremidad o el pie forma lo que puede denominarse como una huella, es decir, una región que esta en contacto con la superficie sobre la que se apoya, si una o varias extremidades están tocando el suelo simultáneamente entonces existe un cierre convexo que envuelve todos los vértices de estas huellas y forma un polígono en el plano horizontal, a este polígono se le conoce como **polígono o base de soporte (BoS)**, también considérese que en las entidades que caminan normalmente el **centro de masa CoM** se encuentra en el torso pues este tiene significativamente más masa que las extremidades y si para fines prácticos también suponemos que el campo gravitacional es completamente uniforme este **CoM** es también el **centro de gravedad (CoG)** de la entidad. Lo anterior es relevante debido a que la ubicación de la proyección horizontal del **CoG** de una entidad con respecto a la base de soporte que forman sus extremidades, es un criterio para saber si se mantiene o no el balance, en base a esto se definen los siguientes tipos de estabilidad:

Estabilidad estática: El balance se mantiene siempre y cuando la proyección horizontal del **CoG** permanezca dentro del polígono de soporte en todo momento.

Estabilidad dinámica: La proyección horizontal del **CoG** suele abandonar la región definida por el polígono de soporte por lo que el balance se debe mantener aplicando movimientos contro-

lados que se basan en el modelado y control de la inercia.

Una consideración adicional sumamente importante y que se asume se cumple para realizar la síntesis de locomoción artificial con extremidades es que al contacto de un pie con el suelo, no se sufren deslizamientos gracias a la fuerza de fricción, de lo contrario el problema de la inestabilidad se torna demasiado complejo.

La segunda dificultad de un bípedo es que constituye un sistema subactuado. El término subactuado se utiliza para describir un sistema mecánico que posee más **grados de libertad (DoF)** que actuadores en su estructura y por lo tanto este sistema no puede ser comandado para seguir trayectorias arbitrarias en el espacio de configuración, la subactuación resulta evidente cuando se considera que aunque el robot tenga un actuador por cada **DoF** en sus articulaciones, al no encontrarse fijo con respecto a un punto en el espacio, el robot tiene 6 **DoF** adicionales debido a las 3 traslaciones y 3 rotaciones posibles en el espacio. Sea N el número de articulaciones actuadas en el robot, c un comando para mover esas articulaciones y q la posición en el espacio $c \in \mathbb{R}^N$ y $q \in \mathbb{R}^{N+6}$.

Una tercera dificultad es la alta dimensionalidad y no linealidad de un sistema bípedo que resulta del diseño mecánico de la estructura, esta dificultad se incrementa conforme aumenta el número de **DoF** y de elementos mecánicos implementados en el robot, complicando la labor de obtener modelos dinámicos precisos así como la tratabilidad matemática de dichos modelos para desarrollar los controladores. En [7] se menciona que 12 **DoF** (6 por cada pierna) son el estricto mínimo necesario para lograr caminados específicos de la locomoción humana en un robot mientras que en [27] se argumenta que robots con 14 **DoF** (7 por pierna) todavía no son capaces de reproducir fielmente la cinemática del caminado humano.

A pesar de las dificultades que se acaban de mencionar, el bipedismo en los seres vivos constituye una forma de movilidad energéticamente eficiente que además permitió liberar las extremidades superiores que subsecuente se especializaron para realizar otras funciones interesantes como la manipulación de herramientas en los seres humanos y el vuelo en las aves. Hay que mencionar que el bipedismo puede llevarse a cabo de diferentes modos, se puede correr, saltar o caminar. De estos modos, el caminado es el modo de andar más común en los bípedos y el que primeramente se busca realizar en los robots. El caminar se describe en [49] como “*una serie de movimientos alternantes, rítmicos, de las extremidades y del tronco que determinan un desplazamiento hacia delante del centro de gravedad*”, estos movimientos se llevan a cabo principalmente en el plano sagital y es precisamente en este plano donde se encuentran las articulaciones más influyentes, no obstante, el

2. MARCO TEÓRICO Y ANTECEDENTES

análisis del diseño de un bípedo y su movimiento puede realizarse desde los 3 planos principales que se utilizan para dividir un cuerpo humano, en la figura 2-1 se muestran estos planos principales.

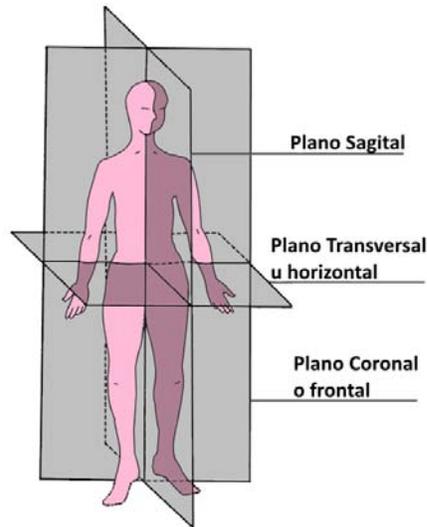


Figura 2-1: Los planos principales en los que se divide el cuerpo humano. El plano sagital divide en parte izquierda y derecha, el coronal en parte anterior y posterior, el transverso en parte superior e inferior.

Dada la repetición de los movimientos para caminar se puede considerar una secuencia determinada como un ciclo de caminado, el cual inicia cuando uno de los pies hace contacto con el suelo y finaliza cuando ese mismo pie hace su siguiente contacto, para facilitar el estudio de este ciclo se puede dividir en fases, en [7] se hace una descripción de estas fases, las cuales se mencionan a continuación y se ilustran en la figura 2-2:

1. Entre el 0 % al 15 % de un ciclo, ambos pies se encuentran en contacto con el suelo soportando el peso, esta fase se denomina de **soporte doble**.
2. Entre el 15 % al 50 % de un ciclo, el pie izquierdo deja el suelo y se columpia, mientras el pie derecho soporta el peso, esta fase se denomina de **soporte simple**.
3. Entre el 50 % al 65 % de un ciclo, el pie izquierdo aterriza realizando nuevamente contacto con el suelo, esta fase es nuevamente de soporte doble.
4. Finalmente entre el 65 % al 100 % de un ciclo, el pie derecho deja el suelo y se columpia, mientras el pie izquierdo soporta el peso, esta fase es nuevamente de soporte simple.

De acuerdo con estas fases que se mencionaron anteriormente, el bípedo se encuentra un 30 % del tiempo en una fase de soporte doble durante la cual se puede considerar que las piernas forman

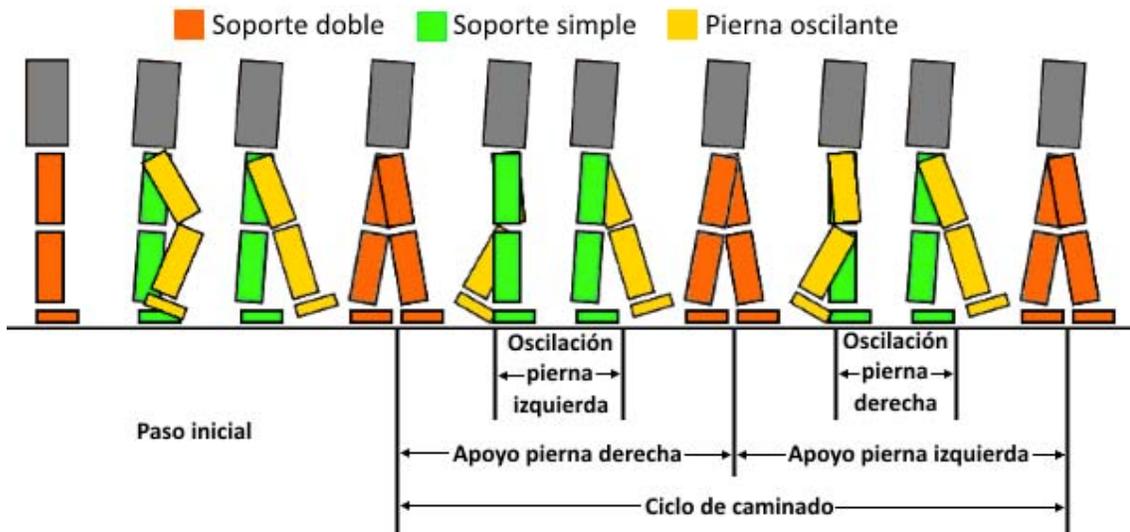


Figura 2-2: Las fases del caminado desde una perspectiva sagital.

una cadena cinemática cerrada y debido a que se cuenta con dos apoyos que forman una **BoS** apropiada normalmente se cumple el criterio de estabilidad estática, en el otro 70 % del tiempo restante se está en una fase de soporte simple en donde las piernas forman una cadena cinemática abierta.

2.1. Métodos usados para solucionar el problema de caminado bípido

La locomoción bípida es un tema que todavía interesa mucho a investigadores de todo el mundo y aunque para su solución se han propuesto múltiples métodos que han logrado avances impresionantes en la materia, especialmente en las últimas décadas, todavía los robots se encuentran lejos de realizar las proezas que los sistemas bípedos biológicos suelen demostrar.

En [61] se hace mención de 2 grandes enfoques que se han adoptado en las investigaciones orientadas al desarrollo de un controlador de caminado bípido: el basado en el modelo y el libre de modelo.

El enfoque basado en el modelo busca que el controlador sea dependiente del modelado del robot, una forma de proceder en la obtención de modelos cinemáticos y dinámicos se realiza descomponiendo en partes el ciclo de caminado analizándolo desde un plano específico y aplicar métodos de Newton-Euler o Lagrange-Euler para obtener las ecuaciones que describen el sistema, debido a que

2.1. MÉTODOS USADOS PARA SOLUCIONAR EL PROBLEMA DE CAMINADO BÍPEDO

la obtención de las ecuaciones por estos métodos es laborioso y complicado, se recurre a utilizar simplificaciones como el modelo de péndulo invertido o el modelo de carro-mesa para describir el sistema [23, 24], una vez que se cuenta con las ecuaciones se puede implementar el controlador como una máquina finita de estados donde cada estado corresponde a una de las partes en que se dividió el ciclo y que genera los movimientos de caminado correspondientes a esa parte tomando en cuenta las ecuaciones de movimiento para asegurar que los movimientos por generarse cumplen un criterio como el del punto de momento cero (ZMP) [55] para que el robot pueda caminar de forma estable.

La siguiente ecuación se utiliza frecuentemente para modelar la dinámica del movimiento en cuerpos de n articulaciones rotacionales ([7, 16, 63] muestran diferentes variaciones de la ecuación):

$$\tau = M(q)\ddot{q} + C(q, \dot{q}) + G(q, \dot{q}) \quad (2-1)$$

En donde:

τ es un vector n dimensional de torques en las articulaciones.

q es un vector n dimensional de posiciones de las articulaciones.

\dot{q} es un vector n dimensional de velocidades de las articulaciones.

\ddot{q} es un vector n dimensional de aceleraciones de las articulaciones.

$M(q)$ es una matriz $n \times n$ dimensional de inercia.

$C(q, \dot{q})$ es un vector n dimensional que agrupa los términos de efectos de coriolis y centrifugales.

$G(q, \dot{q})$ es un vector n dimensional que agrupa los términos de gravedad y otras fuerzas que actúan en las articulaciones.

El otro enfoque que es libre de modelo tiene 2 partes que hay que considerar, la primera se centra en el modelo de computación que va a servir para controlar el sistema y la segunda se centra en la optimización de los parámetros y variables del controlador a través de métodos de aprendizaje o adaptación¹. Este enfoque libre de modelo es el que se toma como base en este trabajo de tesis.

Algunas desventajas que el enfoque basado en el modelo presenta son:

¹El término de **adaptación** se refiere a la actualización de un comportamiento a través del tiempo, el término **aprendizaje** se refiere a la obtención de un comportamiento ante una situación asociada y por lo tanto esta implica adaptación.

2.1. MÉTODOS USADOS PARA SOLUCIONAR EL PROBLEMA DE CAMINADO BÍPEDO

1. La obtención de los modelos es un proceso laborioso, cambios estructurales o de las propiedades físicas del robot implicarían tener que ajustar los modelos o reformularlos en el peor de los casos, además el uso de un controlador basado en el modelo puede resultar computacionalmente costoso si todos los cálculos necesarios para la generación de trayectorias de movimiento estable se hace en línea.
2. Robots que utilizan este tipo de modelos en algunos casos deben cumplir ciertas restricciones. Por ejemplo, el uso de pies planos para asegurar un contacto total con el suelo, o en contraste, el uso de pies curvos para que el contacto se pueda llevar a cabo con rodamiento o una de las restricciones más notorias que se puede percibir es que caminan con las rodillas un poco flexionadas con el fin de evitar singularidades² que ocurren en la posición donde la rodilla está extendida.
3. El movimiento de estos robots se percibe falto de naturalidad ya que las máquinas de estado que se utilizan no suelen replicar los movimientos con fluidez.
4. Se puede extender estos modelos para utilizar técnicas de adaptación lo cual aporta mucho a la robustez del enfoque basado en el modelo pero la integración de estas técnicas añade a la laboriosidad y complejidad del controlador.

Algunas desventajas que el enfoque libre de modelo presenta son:

1. Se tiene que definir el comportamiento que se desea alcanzar en términos que puedan ser medibles en el desempeño del controlador para poder dirigir el aprendizaje en la dirección correcta, esto no es una labor precisa ni claramente definida por lo que se tiene que recurrir parcialmente a heurísticas.
2. El aprendizaje o adaptación puede ser un proceso que consume mucho tiempo, así que generalmente siempre se hace fuera de línea.
3. Dependiendo el comportamiento que se desea realizar los procesos pueden estancarse en mínimos locales que solo cumplen parcialmente con el comportamiento y que son difíciles de rectificar.

²Una singularidad es una configuración donde el comportamiento de una ecuación no está apropiadamente definido.

2.1. MÉTODOS USADOS PARA SOLUCIONAR EL PROBLEMA DE CAMINADO BÍPEDO

Además de los enfoques mencionados hay que considerar también los diversos tipos de control que se pueden utilizar para la síntesis de locomoción, la figura 2-3 muestra una clasificación de estos tipos y en seguida se describen algunos de los tipos y clases.

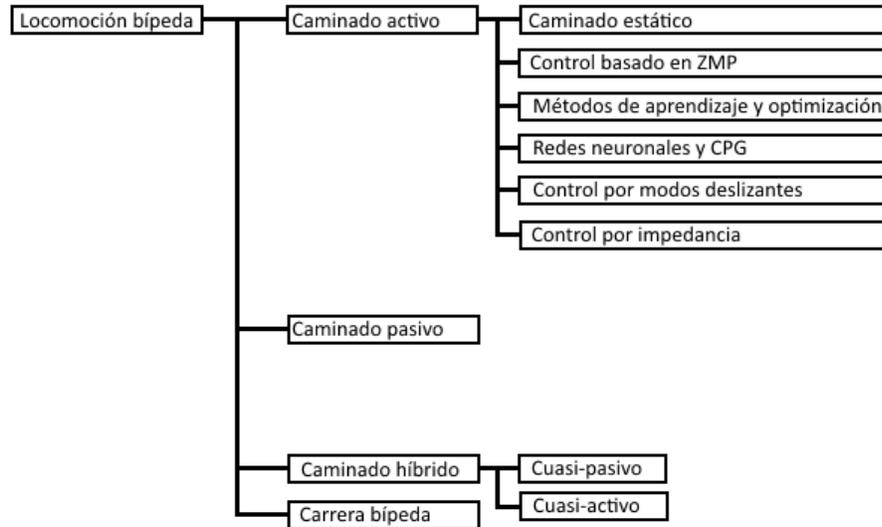


Figura 2-3: Clasificación del control de locomoción, imagen adaptada de [15].

2.1.1. Caminado estático

Este tipo de caminado se basa totalmente en el criterio de estabilidad estática por lo que los movimientos que se realizan deben siempre mantener la proyección horizontal del **CoG** dentro del polígono de soporte. El robot inicia una secuencia de caminado partiendo de una fase de doble soporte, sin abandonar este doble soporte debe moverse para que su **CoG** cambie de posición y la proyección quede dentro de la región de soporte definida por un solo pie, ahora puede pasar a la fase de soporte simple al levantar y columpiar al frente el pie que no está soportando el peso y subsecuentemente al aterrizar el pie se encuentra nuevamente en fase de doble soporte y se repite la secuencia. En este tipo de caminado no se consideran los efectos dinámicos que provoca el caminar en el robot, sin embargo esto implica los movimientos que realiza el robot sean cuidadosos para no generar aceleraciones que produzcan efectos inerciales que afectan la estabilidad y por lo tanto camina lentamente y con poca robustez.

Las trayectorias que permiten un caminado estático se pueden crear a mano si el diseñador es lo suficientemente hábil y se toma el tiempo de ajustar las posiciones cuidadosamente para no romper

el criterio de estabilidad estática.

2.1.2. Punto de momento cero

El concepto de **punto de momento cero (ZMP)** fue propuesto a inicio de los 70's por Miomir Vukobratović, sin embargo la primera demostración práctica de la efectividad de este concepto para mantener la estabilidad dinámica de un robot bípedo se dio hasta 1984 con el robot WL-10RD de la universidad Waseda de Japón, a partir de entonces el reconocimiento del significado y la función del **ZMP** marco un cambio en las tendencias de control de caminado y es actualmente el concepto más popular que se utiliza para la realización de caminado en robots bípedos, y se ha utilizado en robots que son considerados estado del arte (la figura 2-4 muestra un ejemplo). A continuación se da la explicación del concepto **ZMP** como viene descrito en [55].

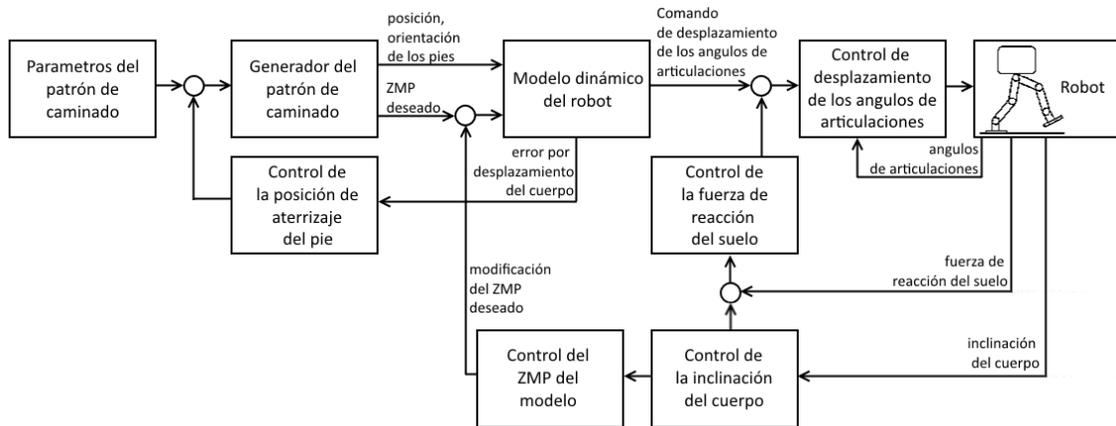


Figura 2-4: Modelo de control del robot ASIMO de Honda, un ejemplo del estado de arte que emplea ZMP, imagen adaptada de [20].

En un sistema bípedo donde todas las articulaciones están actuadas y se pueden controlar directamente, el contacto entre el pie y el suelo puede considerarse como un **DoF** adicional no actuado y es donde tiene lugar la interacción entre el sistema y el entorno, este contacto es esencial para poder realizar el caminado ya que la posición del sistema en el entorno depende de la posición de los pies con relación al piso.

El pie solo puede ser controlado indirectamente a través del mecanismo que se encuentra antes del pie, por lo tanto, el indicador general del comportamiento del sistema bípedo es el punto en donde la influencia de todas las fuerzas que actúan en el mecanismo se pueden reemplazar por una sola fuerza resultante. A este punto se le llamó de momento cero.

2.1. MÉTODOS USADOS PARA SOLUCIONAR EL PROBLEMA DE CAMINADO BÍPEDO

Una de las tareas más importantes de un sistema locomotor al andar es la de preservar la estabilidad dinámica, lo cual se logra asegurando que toda el área del pie, no solo un borde, entre en contacto con el piso. El pie se apoya libremente sobre el soporte y el único contacto con el ambiente se lleva a cabo mediante la fuerza de fricción y el componente vertical de la fuerza de reacción del piso.

Considerando el sistema bípedo en la fase de soporte simple con el pie completamente apoyado en el suelo (plano horizontal completamente nivelado) y, con la finalidad de facilitar el análisis, despreciando la parte del mecanismo que se encuentra por encima del tobillo (punto A) del pie de soporte pero resumiendo su dinámica con la influencia en este punto de una fuerza F_A con componentes (F_{Ax}, F_{Ay}, F_{Az}) y un momento M_A con componentes (M_{Ax}, M_{Ay}, M_{Az}) , el peso del pie actúa en un punto G donde se encuentra su **CoG** y también se experimenta en un punto P del pie la reacción de contacto con el piso, cuyo efecto mantiene todo el sistema en equilibrio.

La reacción del piso total tiene 3 componentes de fuerza (F_{Rx}, F_{Ry}, F_{Rz}) y 3 de momento (M_{Rx}, M_{Ry}, M_{Rz}) . Dado que la fuerza de fricción actúa en el punto de contacto del pie con el piso y el pie esta en reposo (se asume que no se desliza) entonces los componentes horizontales (F_{Rx}, F_{Ry}) pertenecen a la fuerza de fricción y están balanceando los componentes (F_{Ax}, F_{Ay}) , también hay un momento de fricción vertical M_{Rz} y balancea M_{Az} y por último el componente de reacción del piso F_{Rz} balancea la fuerza vertical F_{Az} y solo falta entonces balancear los componentes horizontales del momento. Dada la naturaleza unilateral del contacto la fuerza de reacción del piso solo se produce en un solo sentido que sería en este caso hacia arriba F_{Rz} y los componentes horizontales del momento (M_{Ax}, M_{Ay}) solo pueden compensarse cambiando la posición donde actúa F_R dentro del polígono de soporte ya que en realidad (M_{Rx}, M_{Ry}) no existen, debido a esto los componentes horizontales de M_A efectivamente tienen el efecto de desplazar el punto donde actúa la fuerza de reacción para estar balanceados, si el polígono de soporte no es lo suficientemente grande para abarcar este punto, entonces F_R actuará en uno de los bordes del pie y los momentos horizontales no balanceados provocarán una rotación del sistema con respecto a ese borde haciendo que el sistema se desequilibre. Por lo tanto se puede decir que la condición necesaria y suficiente para que el sistema mantenga su estabilidad dinámica es que para el punto P donde actúa F_R se consiga $M_{Ax} = 0, M_{Ay} = 0$ de esta condición se tomó el nombre de punto de momento cero, los elementos involucrados que se acaban de describir tienen una representación gráfica en la figura 2-5.

El siguiente paso es encontrar la posición del **ZMP** dada la dinámica del sistema, para lo cual se dan las ecuaciones siguientes:

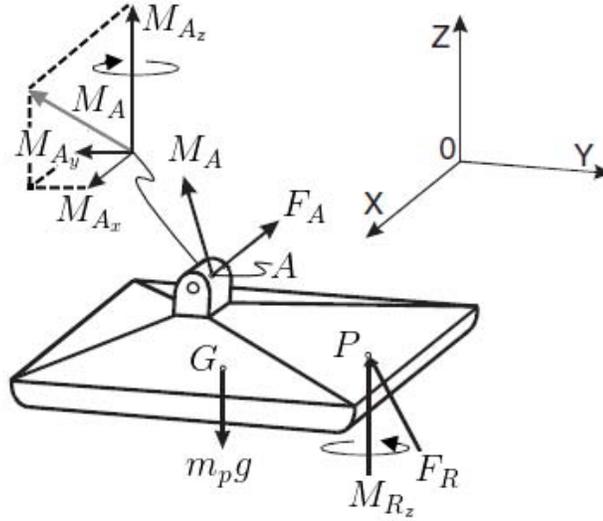


Figura 2-5: Las fuerzas y momentos que actúan en el pie durante el soporte simple, imagen adaptada de [55].

$$F_R + F_A + m_p g = 0 \quad (2-2)$$

$$\vec{O}P \times \vec{F}_R + \vec{O}G \times m_p g + M_A + M_{R_z} + \vec{O}A \times F_A = 0 \quad (2-3)$$

En donde:

m_p es la masa del pie.

g es la aceleración de la gravedad.

$\vec{O}P$, $\vec{O}G$ y $\vec{O}A$ son vectores radiales desde el origen del sistema coordenado de 3 ejes xyz a los puntos P , G y A .

Si se considera al punto P como el origen del sistema entonces la ecuación 2-3 proyectada en el eje z para el momento de fricción vertical se puede expresar como:

$$M_{R_z} = - \left(M_{A_z} + \left(\vec{O}A \times F_A \right)_z \right) \quad (2-4)$$

Y proyectando la ecuación 2-3 en el plano horizontal:

$$\left(\vec{O}P \times \vec{F}_R \right)_{xy} + \vec{O}G \times m_p g + M_{A_{xy}} + \left(\vec{O}A \times F_A \right)_{xy} = 0 \quad (2-5)$$

2.1. MÉTODOS USADOS PARA SOLUCIONAR EL PROBLEMA DE CAMINADO BÍPEDO

Con la ecuación 2-5 se tiene la base para calcular la posición de la fuerza de reacción del piso P que cumple con ser **ZMP**. Hay que notar que el **ZMP** solo existe dentro del polígono de soporte, por lo que si este P calculado queda dentro del polígono de soporte el sistema es dinámicamente estable, si no es así, la fuerza de reacción actúa en un punto dentro del polígono de soporte pero no es **ZMP** y el sistema puede caer. Una aclaración que hay que realizar es la relación entre el **centro de presión (CoP)** y **ZMP**, ya que estos solo coinciden cuando el sistema es dinámicamente estable, es decir, el punto donde actúa la fuerza resultante del contacto del pie con el suelo siempre es considerado como el **CoP**, si esta fuerza está balanceando todas las fuerzas activas en el sistema entonces este punto también es **ZMP**.

2.1.3. Caminador pasivo dinámico

Se refiere a una clase de máquinas bípedas que pueden realizar de forma natural el caminado sin la necesidad de un control activo ni un suministro de energía propio, lo cual es posible gracias a la fuerza de gravedad que actúa sobre la máquina que se encuentra encima de un plano ligeramente inclinado y el diseño de la máquina (distribución de masa apropiada, límites de articulaciones y pies curvos entre otros) que es capaz de aprovechar la dinámica de su estructura y los efectos inerciales para lograr caminar.

Los caminadores pasivos existían desde hace ya varios años en la forma de juguetes que eran capaces de caminar dando pequeños pasos sobre un plano inclinado, un ejemplo de estos juguetes son los “Wilson Walkies” que se fabricaron en los años 30’s. Sin embargo el concepto de **camina-dor pasivo dinámico (PDW)** y la dinámica detrás de este fueron trabajados por Tad McGeer a comienzo de los 90’s en [31], primeramente para bípedos con piernas totalmente rectas y posteriormente para bípedos con rodillas articuladas en [32] donde se corroboró la ocurrencia natural de movimiento cíclico.

A pesar de que los **PDW** son extremadamente eficientes en su consumo de energía y replican los movimientos de caminado con gran parecido al de los seres humanos, requieren del plano inclinado para desencadenar el movimiento y son poco robustos. El trabajo de McGeer se ha usado para entender mejor la eficiencia del caminado y la dinámica involucrada así como para el desarrollo de robots bípedos que son más eficientes en el consumo de energía en las formas de caminado híbrido cuasi-pasivo donde pocas articulaciones si utilizan motores para recuperar la energía que se disipa al contacto con el suelo y así no requerir del plano inclinado o cuasi-activo donde las articulaciones son actuadas a excepción de algunas pocas que se dejan pasivas.

2.1.4. Generador central de patrones

El estudio de la neurobiología de seres vertebrados e invertebrados determinó la presencia de circuitos neuronales que producen patrones rítmicos de actividad neuronal sin que estos reciban un estímulo rítmico y se descubrió también que estas redes de neuronas están involucradas en actividades rítmicas fundamentales como la respiración, masticación y la locomoción. Estos generadores de patrones al no necesitar retroalimentación del sistema nervioso periférico para generar ritmos, se decidió denominarlos **generadores centrales de patrones (CPG)**. El trabajo en [10] determinó que estos generadores centrales de patrones también se encuentran presentes en los seres humanos, y tienen que ver en la generación de movimientos en las extremidades inferiores correspondientes a fases del caminado, ya que al aplicarse pulsos eléctricos de diferentes frecuencias se pudieron apreciar movimientos característicos.

Algunos experimentos descubrieron que los **CPG** son redes distribuidas compuestas de múltiples centros oscilatorios acoplados y que aunque la retroalimentación sensorial no genera los ritmos, si es esencial para la formación de los patrones generados, lo cual influye en la coordinación de los movimientos y la actividad del mismo generador, otros experimentos que se realizaron aplicando estimulación en la región mesencefálica locomotora del tronco cerebral encontraron que aumentando el nivel de estímulo se puede variar la velocidad de locomoción e incluso inducir automáticamente un cambio en el modo de locomoción (por ejemplo transición del caminado al trote y del trote a la carrera en un gato descerebrado), desde el punto de vista del control esto implica que los **CPG** pueden generar complejos comportamientos locomotores e implementan algún tipo de modelo interno que sabe que señales de comando rítmicas deben producirse para lograr determinada velocidad de locomoción.

En resumen, en los vertebrados el sistema locomotor está organizado de tal forma que los **CPG** que se encuentran en la espina son los responsables de producir patrones rítmicos básicos y los centros de alto nivel como la corteza motora, cerebelo y ganglios basales son responsables de modular esos patrones en base a las condiciones del ambiente (los estímulos recibidos), la organización distribuida de este sistema presenta las siguientes características interesantes:

1. Reduce los tiempos de retraso en el ciclo de control motor, pues los ritmos están coordinados con los movimientos mecánicos usando pequeños ciclos retroalimentados a través de la médula espinal.
2. Reduce dramáticamente la dimensionalidad de las señales de control descendientes de los

2.1. MÉTODOS USADOS PARA SOLUCIONAR EL PROBLEMA DE CAMINADO BÍPEDO

centros de alto nivel, ya que estas señales no necesitan especificar la actividad muscular, solo modular la actividad del CPG.

3. Reduce también significativamente el ancho de banda requerido entre los centros de alto nivel y la médula espinal.

Todo lo mencionado con anterioridad respecto a los CPG proviene de [21].

Debido a las capacidades y propiedades de los CPG biológicos, se ha buscado desarrollar modelos artificiales de CPG que puedan emplearse en el control robótico, un ejemplo se muestra en la figura 2-6. Uno de dichos modelos artificiales para la creación de CPG son los osciladores y se basan en modelos matemáticos de elementos osciladores no lineales acoplados, Kiyotoshi Matsuoka en [30] hace una propuesta de oscilador que consiste en un par de neuronas que se inhiben mutuamente e incluyen adaptación, haciendo énfasis en que esta adaptación (también conocida como fatiga o inhibición propia) es necesaria para obtener un comportamiento que se asemeje al de neuronas reales. El comportamiento de la neurona que se utiliza en este oscilador está representado por las siguientes ecuaciones:

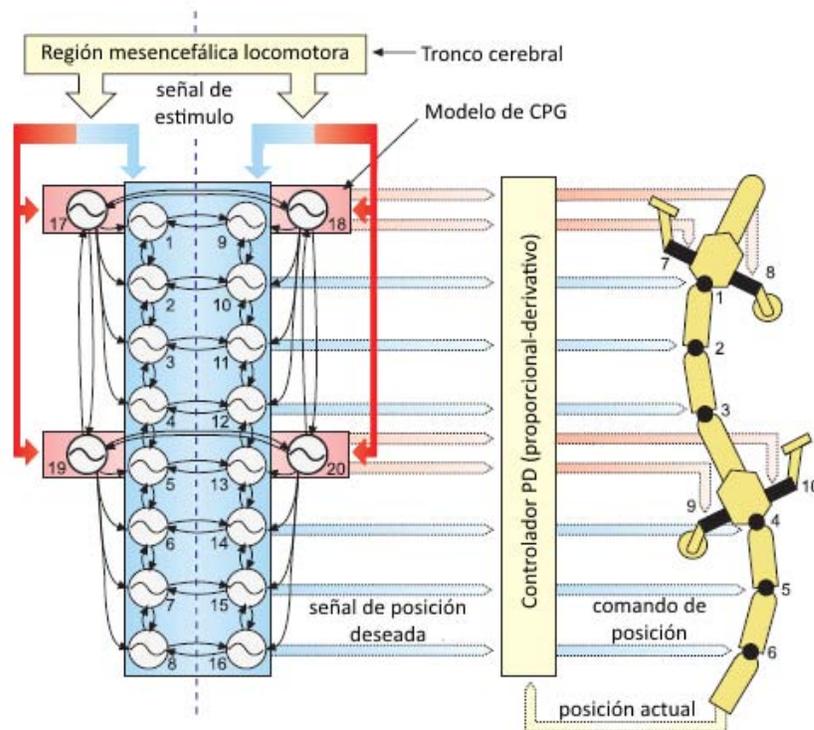


Figura 2-6: Control por CPG de una salamandra robótica, se puede observar que cada articulación está controlada por un par de elementos acoplados, imagen adaptada de [21].

$$\begin{aligned}\tau \dot{x}_i + x_i &= - \sum_{j=1}^n \alpha_{ij} s_j + c_i - \beta v_i \\ T \dot{v}_i + v_i &= s_i \\ s_i &= g(x_i - \theta) , \quad g(x_i) = \max(0, x_i)\end{aligned}\tag{2-6}$$

En donde:

α_{ij} representa la fuerza de la conexión inhibitoria entre neuronas i y j con $\alpha_{ij} \geq 0$ para $i = j$ y $\alpha_{ij} = 0$ para $i \neq j$, si la conexión proviene de otros osciladores o sensores la fuerza de conexión puede ser inhibitoria o excitatoria.

s_j representa una señal que se recibe de otra neurona, sensor u oscilador.

c_i representa una señal que proviene del exterior de la red (una entrada tónica) y se asume positiva y constante en el tiempo.

x_i representa el potencial de membrana de la neurona i .

τ es una constante de tiempo.

s_i representa la salida de la neurona i , también se le conoce a esta salida como tasa de disparo de la neurona.

v_i es una variable que representa el grado de adaptación.

T es una constante de tiempo para la adaptación.

β es una constante de la tasa de disparo para la adaptación.

θ es un valor de umbral debajo del cual la neurona no se dispara.

La figura 2-7 muestra el modelo de oscilador de Matsuoka de 2 neuronas acopladas, la inhibición mutua que se representa con γs_e o γs_f no aparece en las ecuaciones 2-6 ya que se incluye en la $\sum \alpha_{ij} s_j$.

El artículo de Matsuoka también menciona que los valores de parámetros τ , T y β se deben determinar de tal manera que la respuesta no sea una oscilación amortiguada y para ello da la siguiente condición:

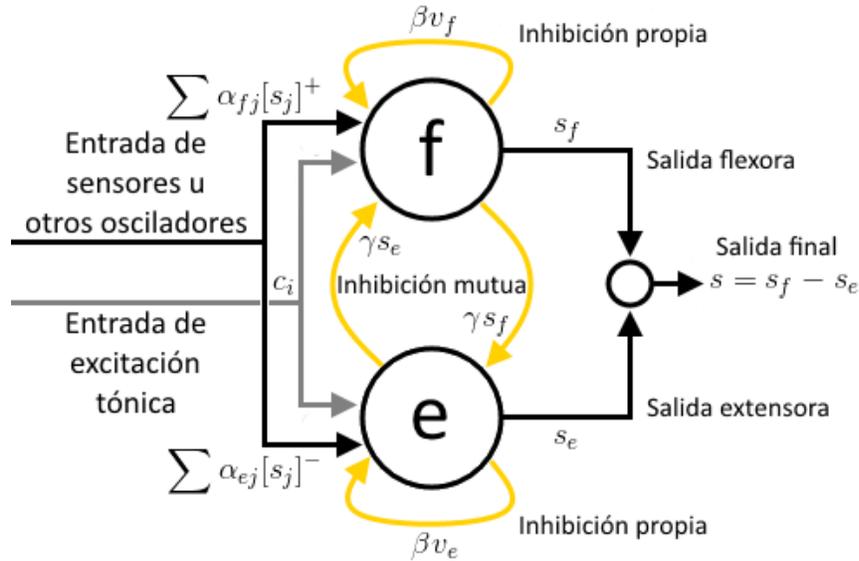


Figura 2-7: Un oscilador de Matsuoka. Las neuronas extensora y flexora son llamadas así suponiendo que el oscilador está en el contexto del control de movimiento de una articulación.

$$(T - \tau)^2 \geq 4T\tau\beta \tag{2-7}$$

2.2. Neuroevolución y robótica evolutiva

El término neuroevolución se refiere a una técnica del aprendizaje automatizado donde se utilizan algoritmos evolutivos para la construcción y/o optimización de **redes neuronales artificiales (ANN)**, el algoritmo evolutivo que se utiliza con mayor frecuencia es el **algoritmo genético (GA)**, más adelante en esta misma sección las **ANN** y los **GA** se explicarán con más detalle. En [59] se discute sobre el uso de algoritmos evolutivos para adaptación y aprendizaje en las **ANN** donde el término **redes neuronales artificiales evolutivas (EANN)** se emplea para designar a las redes que se obtienen por medio de estos métodos.

La técnica neuroevolutiva es también bioinspirada, se basa en como el proceso de evolución natural llevó al desarrollo de los sistemas nerviosos en los seres vivos permitiendo que estos expresen comportamientos inteligentes que les permiten solucionar problemas complejos, ya que con la neuroevolución se logra el aprendizaje de comportamiento inteligente en entes artificiales, como los robots y agentes virtuales, es una herramienta frecuentemente utilizada por la robótica evolutiva [34] y la vida artificial.

La interacción de los seres vivos con el entorno es muy importante para guiar el proceso de evolución natural, ya que el entorno es quien presenta los retos que deben ser superados por un organismo para lograr sobrevivir y reproducirse, se puede considerar entonces que la supervivencia y éxito reproductivo de un ser vivo representan una medida que indica que este se desempeñó apropiadamente. La robótica evolutiva también requiere de la interacción y retroalimentación del ambiente para dirigir la síntesis de controladores para entes artificiales autónomos, la neuroevolución tiene la ventaja de permitirnos lograr este objetivo incluso cuando el comportamiento deseado no está claramente definido por un modelo o datos y la interacción se efectúa en un entorno dinámico e impredecible donde lo que se puede percibir es limitado, ya que la neuroevolución puede realizar el aprendizaje de forma supervisada o de forma no supervisada.

Como se mencionó que el entorno es importante para la robótica evolutiva, hay que considerar las 2 posibilidades de entorno que se pueden usar para la evolución.

- Un entorno real, en cuyo caso se debe usar un robot real y la evolución se considera se realiza encarnada. La ventaja que ofrece es que la evolución se realiza totalmente apegada a la realidad, por lo que los comportamientos se comprueban físicamente, sin embargo el proceso de evolución solo puede realizarse en tiempo real, presenta riesgos reales para el robot, no permite la posibilidad de evolucionar el diseño físico del robot y si el experimento tiene ciertas condiciones iniciales estas se deben restablecer manualmente en cada prueba.
- Un entorno simulado, en cuyo caso se puede usar un robot simulado y la evolución se considera no encarnada. La ventaja que ofrece es que la evolución se puede realizar más rápido que en el tiempo real, sin riesgos y se pueden manipular automáticamente el entorno, el robot y las condiciones de simulación para ajustarse a diversas necesidades, sin embargo sufre las desventajas de requerir la especificación de las dinámicas que rigen la interacción física entre todos los elementos y la naturaleza aproximativa de los métodos que se utilizan para resolver los cálculos de las restricciones dinámicas junto con el hecho de la posible imprecisión cuando el diseñador especifica las propiedades de la simulación hacen que exista una discrepancia entre la simulación y la realidad. A esta discrepancia se le llama la brecha de realidad y es un problema grave ya que implica que los resultados obtenidos en la evolución con un simulador pueden no funcionar cuando se prueban en mundo real.

En [6] de Rodney Brooks, trabajo que puede discutiblemente considerarse un precursor de la robótica evolutiva, ya se hace mención del problema de la brecha de realidad y como con casi total

seguridad un resultado obtenido en simulación va a fallar cuando se trate de probar en un robot real, actualmente, con la mejora de los métodos empleados por los simuladores basados en física y la capacidades de procesamiento de las computadoras para realizar más cálculos, este problema se ha mitigado considerablemente haciendo factible lograr buenos resultados si el diseñador se preocupa de calibrar apropiadamente esta simulación.

2.2.1. Algoritmos genéticos

Son métodos para encontrar soluciones a problemas de optimización o búsqueda, son especialmente útiles en problemas donde los espacios de posibles soluciones son muy grandes y se carece de métodos directos y eficientes para hallar las soluciones óptimas, al igual que todos los algoritmos evolutivos de los cuales los GA son una subclase, se basan en abstracciones del proceso de evolución natural.

Un GA se basa en el proceso de selección natural mediante el cual los seres vivos más aptos en una población son los que se reproducen y heredan sus características a la siguiente generación donde el mismo proceso tiene lugar logrando eventualmente incrementar la aptitud de los seres vivos, los elementos y operadores que se utilizan en el GA se originan de esta idea y se describen a continuación:

- Una población de individuos que representan las soluciones al problema que se trata de resolver, cada individuo aparece en la forma de una codificación (binaria, real, etc.) que resume las características de la solución y que lleva el nombre de genoma o genotipo.
- Un método de evaluación donde cada uno de los genomas en la población se decodifica para expresar la solución con todas sus características, esta expresión de características lleva el nombre de fenoma o fenotipo, y en base a una función que determina que tan bien una solución resuelve el problema se le asigna una calificación, la función utilizada se denomina de aptitud u objetivo y por tanto la calificación se conoce como la aptitud o desempeño del individuo.
- Un operador de selección estocástico que en base a la aptitud de los individuos elige, con mayor probabilidad para los más aptos, un par de estos para que sirvan como padres de un nuevo individuo.
- Un operador de cruza que utiliza los genomas de los individuos elegidos como padres para recombinarlos y así crear los genomas de los descendientes.

- Un operador de mutación que puede, con baja probabilidad, modificar las porciones básicas del genoma que se denominan genes.

De esta forma, los operadores proveen los mecanismos para explorar el espacio de soluciones. La selección representa la competencia entre individuos, permitiendo elegir y utilizar las mejores soluciones que se han encontrado. La cruce representa la herencia de genes ventajosos, se crean nuevas soluciones usando elementos de las soluciones elegidas. La mutación representa la posible variabilidad de la descendencia, permite evitar la convergencia prematura de las soluciones en óptimos locales. Hay que destacar que la función de aptitud es la base que permite al algoritmo distinguir una solución de otra, esta función debe guiar al algoritmo en el espacio de soluciones apropiadamente para poder evitar los estancamientos a la vez que refleja el objetivo que las soluciones deben lograr.

El procedimiento general que sigue un GA es iniciar con una población de tamaño determinado, generada de manera aleatoria con el fin de empezar con variedad de posibles soluciones, se evalúa cada individuo para determinar su aptitud y después se seleccionan individuos sobre los cuales aplicar los operadores genéticos de cruce y mutación para formar una nueva generación que reemplaza a la anterior y repetir el proceso con la nueva población, una iteración de este proceso es un ciclo generacional y se ilustra en la figura 2-8.

Ya que el GA incorpora elementos estocásticos y se realiza de manera iterativa, es difícil imaginar que puedan lograr la convergencia a una solución óptima global, en [45] se realiza un análisis mediante el cual se llega a la conclusión de que en problemas de optimización estáticos, un GA logra la convergencia al óptimo global si a través del tiempo el algoritmo conserva la mejor solución encontrada (el algoritmo es elitista). Para que el algoritmo no itere indefinidamente se da una condición de terminación que puede ser que el número de ciclos generacionales completados llegue a un número máximo establecido o la aptitud de algún individuo en la población alcanza una suficiente aproximación a un nivel de aptitud deseado.

Un ejemplo de problema con un espacio de soluciones enorme, que se ha buscado resolver con GA, es el del agente viajero (traveling salesman problem), en el cual dada una lista de n ciudades y donde cada ciudad $i \in [1, 2, \dots, n]$ tiene asociado un vector $(d_{i1}, d_{i2}, \dots, d_{in})$ que indica su distancia a cada ciudad, se busca encontrar la distancia más corta, siguiendo una ruta que partiendo de una ciudad origen y visitando todas las ciudades una sola vez regresa a esa ciudad de origen. El espacio de búsqueda es de tamaño $n!$, si se desea resolver verificando todas las posibles soluciones para $n = 5$ existen 120 posibles soluciones, para $n = 10$ existen 3.6288E06 y para $n = 50$ 3.04140932E64

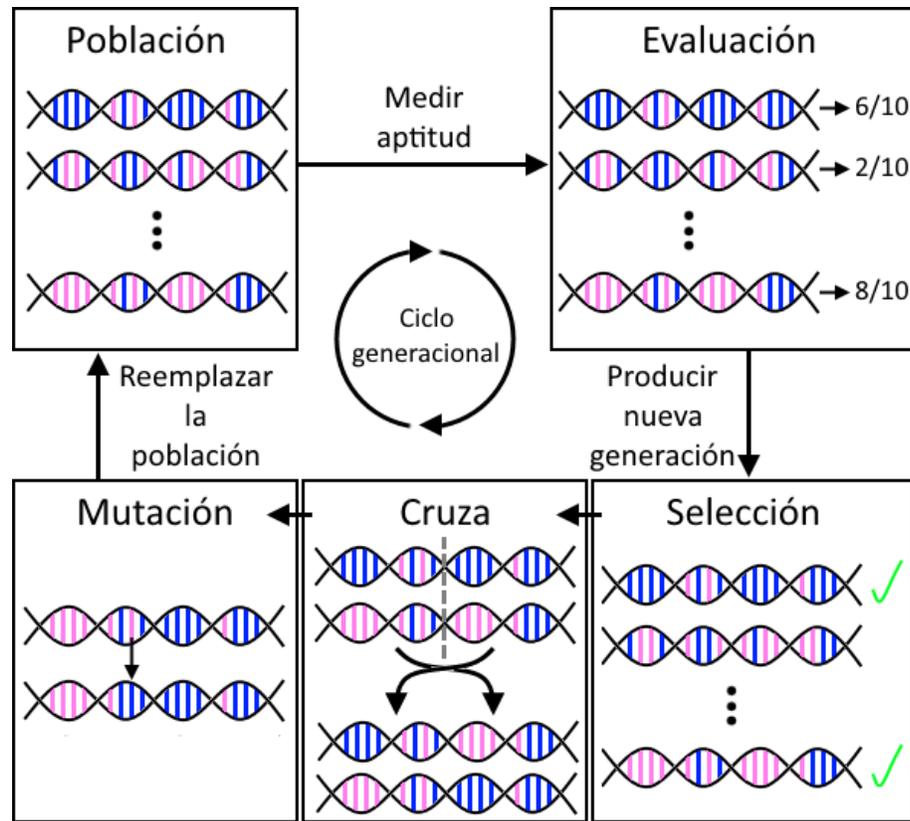


Figura 2-8: El proceso de evolución en un algoritmo genético.

soluciones.

2.2.2. Redes neuronales artificiales

Son un modelo de cómputo inspirado en la estructura y funcionamiento de los sistemas nerviosos biológicos, particularmente el cerebro. En una computadora digital, que sigue el modelo de cómputo basado en la arquitectura de Von Neumann, en forma muy general, se realiza el procesamiento de información controlando y secuenciando el flujo de señales binarias entre procesador, memoria y dispositivos de entrada/salida aplicando un conjunto de instrucciones lógicas y aritméticas de diversa complejidad para transformar las señales. En un cerebro, también en forma muy general, el procesamiento de información se da como resultado de la transformación de señales que sucede al propagarse estas de forma paralela a través de una red formada por la conexión masiva y jerárquica de células llamadas neuronas que tienen la capacidad de recibir un estímulo, reaccionar ante él y transmitir esta reacción como un estímulo a otras neuronas con las cuales se comunican mediante las conexiones llamadas sinapsis.

Los sistemas nerviosos naturales poseen una gran complejidad, son resultado de muchos millones de años de evolución y están compuestos por miles o hasta billones de neuronas de distinto tipo con un número exponencialmente mayor de sinapsis que las conectan, incluso las interacciones moleculares que se llevan a cabo en las neuronas y las sinapsis juegan un papel importante en el proceso total que realizan. Hay que recalcar que aunque la estructura del sistema nervioso en un ser vivo ya viene predeterminada por sus genes, tal vez el aspecto más interesante de las redes neuronales biológicas es como a través del tiempo las experiencias de un ser vivo provocan que la red se auto-organice, es decir se fortalecen, debilitan o generan nuevas conexiones, lo cual permite el aprendizaje y la adaptación, a esta capacidad se le denomina plasticidad y es una característica sumamente importante a considerar. El *conexionismo* es un enfoque de las ciencias cognitivas y la inteligencia artificial que postula que los fenómenos y modelos mentales son procesos emergentes de redes de elementos sencillos e interconectados. Bajo esta perspectiva se inicio la búsqueda de modelos artificiales tomando como base lo que se encuentra en la biología, se pueden abstraer las siguientes características para poder describir los elementos que forman una ANN:

1. Se forma de múltiples unidades simples, que también reciben el nombre de neuronas o nodos, las cuales realizan procesamiento básico.
2. La estructura o arquitectura de la red está dada por la organización y disposición de las neuronas así como por el patrón de conexiones que las relaciona, esto determina el flujo de información en la red y por lo tanto también afecta el cómputo que se realiza.
3. El intercambio de información entre las neuronas se da a través de las conexiones (sinápsis) y es en estas donde tiene lugar el almacenamiento del conocimiento.

Lo primero a considerar entonces es un modelo para estas neuronas artificiales, múltiples modelos han sido propuestos por los investigadores y varían su formulación dependiendo de que aspectos de la neurona biológica se desea recrear. Uno de los modelos tempranos, simples y más populares es el de la neurona tipo perceptrón de Frank Rosenblatt que se muestra en la figura 2-9 y que matemáticamente se puede formular de la siguiente manera:

$$a_j = \sum_{i=1}^n p_{ij}e_i \quad (2-8)$$

$$s_j = f(a_j + b_j) \quad (2-9)$$

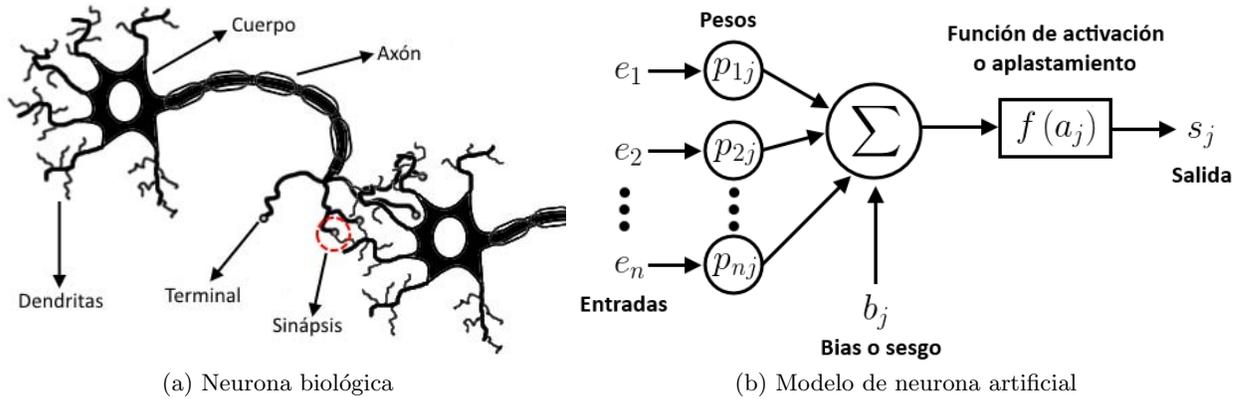


Figura 2-9: Representación de la neurona biológica y su modelo artificial.

En donde:

p_{ij} es el peso sináptico o de conexión que proviene de la neurona i y llega a la j .

e_i es la entrada (también llamado estímulo) que proviene de la neurona i .

Σ es un combinador lineal usado para reducir todas las entradas ponderadas a un solo valor .

a_j es el potencial de activación de la neurona j .

$f(\cdot)$ es la función de activación (también llamada de aplastamiento o transferencia) que puede ser lineal o no lineal, recibe un solo argumento y genera la salida de la neurona limitando su amplitud, los límites de esta salida son comúnmente $[0, 1]$ o $[-1, 1]$.

b_j es el bias (también llamado sesgo) que tiene el efecto de ayudar a excitar o inhibir la neurona.

En la función de activación esto se percibe como una transformación que desplaza los valores de salida de la función. En el plano cartesiano por ejemplo, si la función es $y = f(x + b)$, el desplazamiento es a la izquierda si el bias es positivo y a la derecha si el bias es negativo.

s_j es la salida de la neurona j .

Los pesos y entradas pueden expresarse como vectores $P = (p_{1j}, p_{2j}, \dots, p_{nj})$ y $E = (e_1, e_2, \dots, e_n)$ y el bias puede expresarse como un peso adicional y una entrada adicional constante de 1 en los vectores extendidos $P = (p_{1j}, p_{2j}, \dots, p_{nj}, b_j)$ y $E = (e_1, e_2, \dots, e_n, 1)$, de manera que la ecuación general de una neurona es:

$$s_j = f \left(\sum_{i=1}^{n+1} P_i E_i \right) \quad (2-10)$$

Lo segundo a considerar es la estructura o arquitectura de la red. Convencionalmente la forma de agrupar las neuronas en la red es mediante capas de entrada, ocultas y de salida. Neuronas que tienen alguna entrada proveniente del exterior de la red (reciben las entradas del sistema) son *neuronas de la capa de entrada*, típicamente la capa de entrada solo es un distribuidor de las entradas a las siguientes capas, es decir, las neuronas de entrada son pasivas, no realizan cómputo y su salida es igual a la única entrada que reciben. Neuronas cuya salida va al exterior de la red (generan las salidas del sistema) son *neuronas de la capa de salida*. Neuronas cuyas entradas provienen exclusivamente de otras neuronas y su salida va exclusivamente a otras neuronas son *neuronas de la capa oculta*, se puede decir que las capas ocultas se encuentran en medio de las de entrada y salida y tienen la función de transformar las entradas en algo útil para la capa de salida, puede haber varias capas ocultas en una red y cada capa permite a la red extraer de las entradas, estadísticas de más alto orden.

Con base al flujo de la información a través de estas capas se dan 2 clases principales de arquitectura:

Redes neuronales de alimentación progresiva, en las cuales el flujo de información va de una capa a la siguiente siempre en un sentido progresivo de la capa de entrada a la de salida, en este tipo de red no existe ninguna conexión que forme un ciclo dirigido.

Redes neuronales recurrentes (RNN), en las cuales el flujo de información puede regresar a una capa por la cual ya había pasado con anterioridad, en este tipo de red existe cuando menos una conexión que forma un ciclo dirigido, estas conexiones se conocen como retroalimentación.

Una red neuronal de alimentación progresiva ante un mismo patrón de entrada siempre genera el mismo patrón de salida, su comportamiento es estático ya que la salida está determinada únicamente por el patrón de conexiones y los respectivos pesos a través de las capas de la red, debido a que no hay un elemento temporal en estas redes se puede considerar que una sola activación (propagación de la señal a través de la red) es suficiente para determinar el estado de activación de la red (salida de todas las neuronas) y obtener la salida final al proporcionar una entrada, este tipo de red puede subclasificarse en *totalmente conexa* si entre capas adyacentes todas las neuronas están conectadas y *parcialmente conexa* si faltan conexiones entre capas adyacentes. Un ejemplo de red de alimentación

progresiva se ve en la figura 2-10.

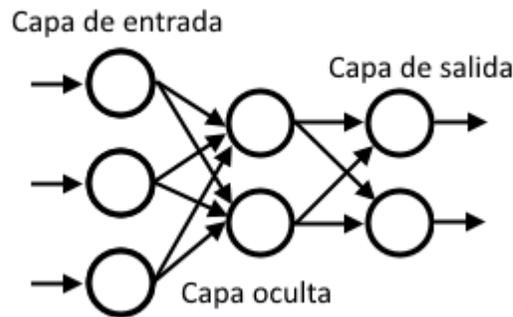


Figura 2-10: Red neuronal de alimentación progresiva de una capa oculta. Esta red es totalmente conexas

En una **RNN** las retroalimentaciones permiten reutilizar parte del cómputo generado por la red, se podría decir que además de la experiencia almacenada en los pesos de conexión esta red también tiene una memoria de corto plazo de los eventos pasados en la forma de la señal que se retroalimenta, lo cual significa que en estas redes, si existe una dimensión temporal la cual se implementa agregado unidades de retraso unitario en las retroalimentaciones, el comportamiento de estas redes es dinámico pues ahora también está determinado por el estado de la red y ante un patrón de entrada se pueden generar diferentes patrones de salida. Un aspecto importante es que estas redes pueden requerir de varias activaciones para propagar la señal a través de la red y las recurrencias hasta que se alcanza la estabilidad del estado de activación, debido a las dinámicas de las redes recurrentes esta estabilidad no está garantizada.

Las **RNN** también pueden subclasificarse en *globalmente recurrentes* si las retroalimentaciones están presentes arbitrariamente en la red e incluso de una neurona a si misma o *parcialmente recurrentes* si las retroalimentaciones ocurren solo entre neuronas específicas tal que la estructura principal está generalmente libre de retroalimentaciones. Un ejemplo de red recurrente se ve en la figura 2-11.

Por último, hay que considerar el problema del aprendizaje, es decir, ¿Cómo pueden determinarse los pesos en las conexiones para que la red aprenda a realizar una función de utilidad?. Al igual que en los sistemas biológicos donde complejos procesos bioquímicos realizan estos ajustes es deseable desarrollar métodos que permitan que el ajuste de estos pesos en las **ANN** se realice de forma automática. Hay dos formas de aprendizaje que permiten dividir estos métodos, el *aprendizaje supervisado* y el *aprendizaje no supervisado*.

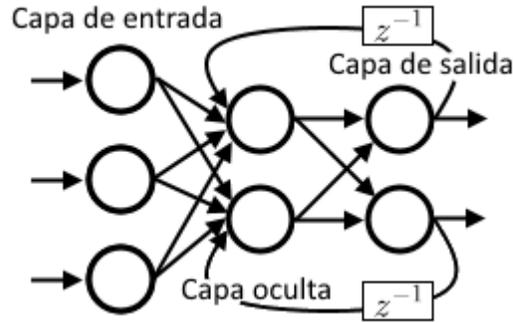


Figura 2-11: Red neuronal recurrente. Esta red muestra recurrencias parciales

En el aprendizaje supervisado, se tiene un conjunto de ejemplos que indican las entradas y la salidas que se deben producir ante estas entradas, los ejemplos permiten realizar una comparación entre la respuesta de la red y la respuesta deseada y esta se usa para determinar el ajuste necesario por realizar. Los algoritmos que se utilizan para aprendizaje supervisado llevan el nombre de algoritmos de aprendizaje o entrenamiento, el más representativo de estos es el algoritmo de retropropagación. En el algoritmo de retropropagación un subconjunto de los ejemplos se toma como el conjunto de entrenamiento, del cual se presentan las entradas a la red y mediante una función de error se comparan las salidas producidas por red y las salidas correctas, se puede calcular el gradiente de esta función de error con respecto a los pesos y propagar este error hacia atrás (de las neuronas en la capa de salida hacia las capas previas) y en base a esto realizar el ajuste de cada uno de los pesos buscando minimizar el error, el proceso se repite iterativamente para ir minimizando el error hasta que se obtiene el comportamiento deseado en la red. Una limitante de la retropropagación es que al utilizar el gradiente, la función de activación de la red debe ser diferenciable, dos funciones de activación que se utilizan ampliamente son la sigmoide, ecuación 2-11 y la tangente hiperbólica, ecuación 2-12, ambas se ilustran en la figura 2-12.

$$s_j = \frac{1}{1 + e^{pa_j}} \quad (2-11)$$

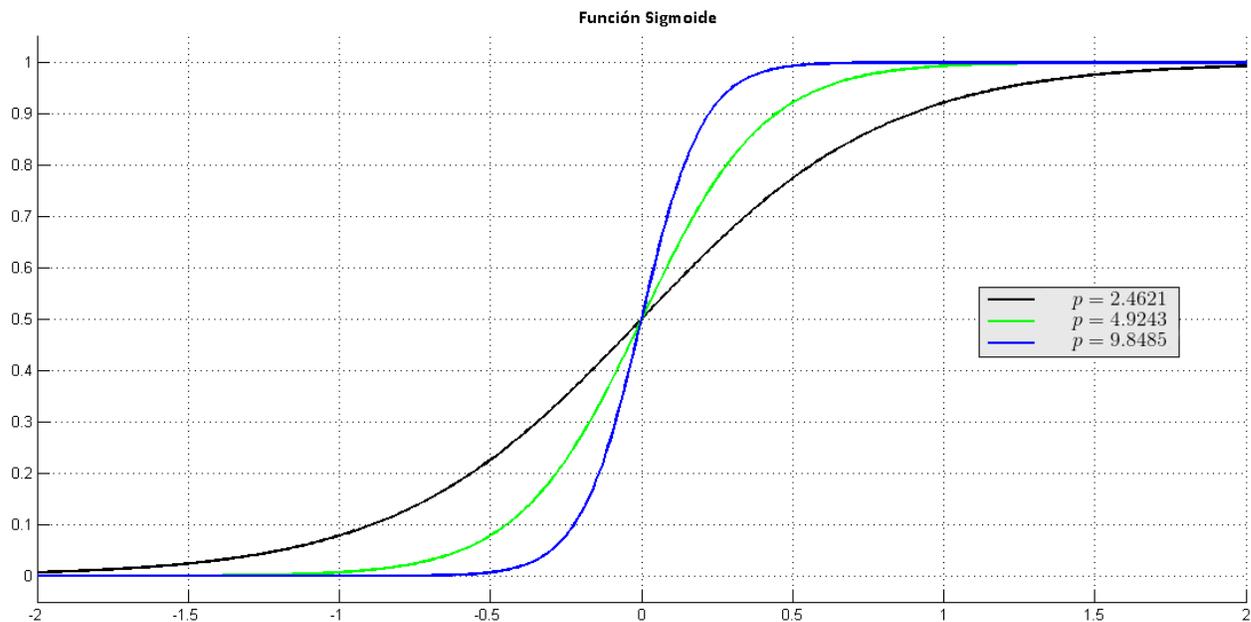
$$s_j = \frac{e^{pa_j} - 1}{e^{pa_j} + 1} \quad (2-12)$$

En donde:

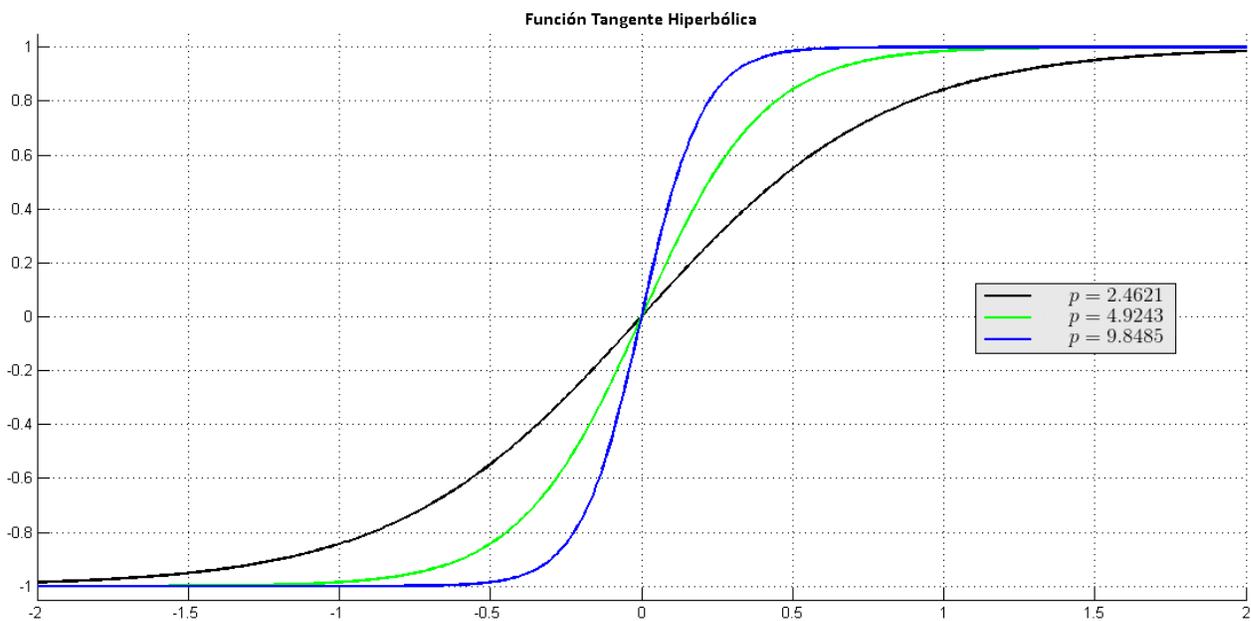
p es un valor constante que afecta la forma de la curva.

a_j es el potencial de activación de la neurona j .

s_j es la salida de la neurona j .



(a) Función sigmoide con distintos valores de p



(b) Función tangente hiperbólica con distintos valores de p

Figura 2-12: Funciones de activación.

En el aprendizaje no supervisado, no hay datos para el entrenamiento de la red, es decir no se conoce la salida que debe producirse ante una entrada, en este caso el aprendizaje no supervisado

puede usarse para agrupar los datos (clustering) al encontrar patrones en los datos a través de la competencia de las neuronas en la red. Los algoritmos evolutivos son aptos para la realización del aprendizaje supervisado y no supervisado.

Las tecnologías y metodologías de ANN actuales todavía necesitan madurar mucho antes de poder recrear artificialmente un sistema que pueda ser equiparable a los sistemas biológicos más desarrollados, no obstante, sin lugar a dudas las ANN son herramientas interesantes que poseen una arquitectura de computadora y de algoritmo muy simple (interconexión de elementos simples que realizan operaciones simples) [18], y son atractivas para un gran número de problemas ya que presentan varias capacidades:

1. No linealidad, al estar compuesta por neuronas lineales o no lineales distribuidas, la red se convierte en un procesador de información inherentemente lineal o no lineal, lo cual permite el tratamiento de problemas de la misma naturaleza.
2. Son aproximadores universales de funciones, [9] da una prueba de esto a través de la superposición de la función sigmoideal.
3. El aprendizaje y adaptación, a través del ajuste de pesos la red adquiere conocimiento y aprende, si se implementa un método para que los pesos puedan ajustarse en línea de acuerdo a la interacción con el ambiente la red también puede adaptarse a los cambios que se presentan. Debido a la capacidad de auto-organización una misma red tiene la posibilidad de permanecer válida para una amplia variedad de problemas.
4. Modelado libre de dominio, la red puede aprender a través de la interacción con el ambiente sin que el diseñador tenga que proveer un modelo del mundo o de la tarea a priori.
5. Capacidad de tratar información incompleta y robustez ante el ruido, ya que la red puede generalizar y producir salidas coherentes aun cuando recibe entradas que no formaron parte del entrenamiento.
6. Tolerancia a fallas, la naturaleza distribuida y gran redundancia de elementos les confiere esta cualidad si es que se presenta la falla de algún elemento, sin embargo, es necesario implementar en hardware la ANN para que esta cualidad se presente.

Gracias a las anteriores características las ANN se han aplicado exitosamente a problemas de: procesamiento de señales, reconocimiento de patrones, clasificación, identificación de sistema,

memorias asociativas, aproximación de funciones y control robótico por mencionar algunos. En el caso del control de robots y de agentes autónomos las ANN se han estudiado y utilizado mucho desde los años 90's, las RNN al poder adquirir representaciones de estado son especialmente útiles cuando se requiere que el robot exhiba comportamientos estrechamente ligados con el ambiente a través de ciclos retroalimentados [56], comportamientos que son dependientes del contexto [33] o que deben variar en el tiempo.

2.3. Trabajo relacionado

El trabajo de [5] utiliza neuroevolución y simulación 2-D para encontrar controladores de navegación reactiva en un robot simple, el proceso de evolución solo ajusta los parámetros de las diferentes arquitecturas de red utilizadas.

La evolución de estructura y comportamiento de creaturas virtuales es presentado en [47], emplea algoritmos genéticos con una codificación de genotipo especial basada en grafos, en estos grafos se codifican la morfología y control de la creatura y operadores de mutación pueden incrementar el tamaño de los grafos, con la metodología que se desarrolló se pudieron evolucionar creaturas para nadar, saltar o seguir una fuente de luz en un entorno 3-D simulado con física.

En [41] se logra evolucionar el caminado bípedo en 3-D con un simulador basado en física y un modelo de robot virtual de 6 DoF, se usa un algoritmo genético para optimizar los parámetros de un CPG de 10 neuronas totalmente interconectadas, el movimiento de los actuadores se modula con control PD que genera los comandos de movimiento a partir del ángulo deseado que proviene de la red. Las medidas usadas para guiar la evolución de controladores fueron la distancia máxima recorrida y que el centro de gravedad no bajara de una altura determinada. La primera implementación que no usa estímulos propioceptivos muestra que se logró el caminado estable en línea recta sobre un plano en un ambiente sin perturbaciones, en otro experimento se añade una neurona sensora y se muestra que se puede modular la actividad del CPG para que el caminado se dirija hacia la fuente de la señal sensada.

El trabajo más relacionado con esta tesis se reporta en [1] donde se utiliza también el algoritmo NEAT para evolucionar redes neuronales de elementos simples pero estructura compleja con el fin de lograr el caminado de bípedos antropomórficos en un simulador basado en física 3-D, algunas de las especificidades de este desarrollo incluyen que solo pocas articulaciones están controladas por la red mientras que el resto permanece solo bajo la influencia del un control PD, se fuerza a la

simetría bilateral al usar 2 redes idénticas para controlar cada pierna con un bias extra en cada red que permite inducir la asimetría de estado inicial y por último un arnés virtual que proporciona soporte completo al robot y disminuye gradualmente dentro del rango de 150 generaciones. Los resultados muestran que se logro realizar el caminado para distintas variaciones antropomórficas que están dentro del rango de las que puede presentar un ser humano, sin embargo los controladores no resultaron estables ya que solo se pudo caminar hasta distancias de 10 metros.

Otros trabajos relacionados con la teoría expuesta en este capítulo son [12, 17, 28, 29, 35, 39, 53, 61, 62].

2.3.1. Estado del arte

Uno de los mejores trabajos en materia de generación de caminado a través de técnicas evolutivas con un simulador basado en física 3-D es el que Thomas Geijtenbeek presenta en [16].

Los modelos que utiliza están compuestos por cuerpos rígidos y están totalmente actuados solo por músculos 3-D. Estos músculos siguen el modelo de Hill (véase la figura 2-13) que se compone de 3 partes:

- Un elemento contráctil (**CE**) que representa las fibras musculares que se contraen en base al estado de activación del músculo.
- Un elemento elástico paralelo (**PEE**) que representa el material elástico y pasivo que rodea las fibras musculares.
- Un elemento elástico serial (**SEE**) que representa los tendones que conectan el músculo a los huesos.

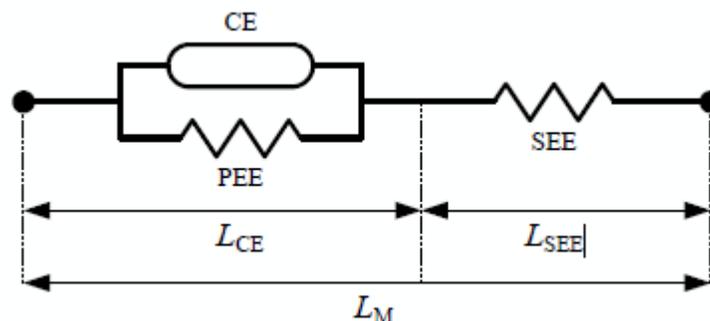


Figura 2-13: Elementos del modelo de Hill, imagen tomada de [16].

2.3. TRABAJO RELACIONADO

Las dinámicas de contracción del músculo están dadas por las relaciones entre esos 3 elementos así como sus longitudes y la velocidad de contracción que poseen. En resumen, la fuerza generada depende del estado de activación muscular y la longitud del músculo mientras que la geometría del músculo está determinada por los puntos de fijación del músculo en los cuerpos rígidos e influye en la dirección y magnitud del torque que el músculo puede producir y que se transmite a los cuerpos rígidos precisamente en los puntos de fijación. Estos puntos de fijación de cada músculo así como las propiedades fisiológicas de la contracción son parámetros que se optimizan con el fin de que los músculos y la ruta que siguen logren ser eficientes, la ubicación de los puntos de fijación solo puede darse dentro regiones específicas de cada cuerpo rígido y están determinadas por plantillas que el usuario debe proveer, un ejemplo se muestra en la figura 2-14.

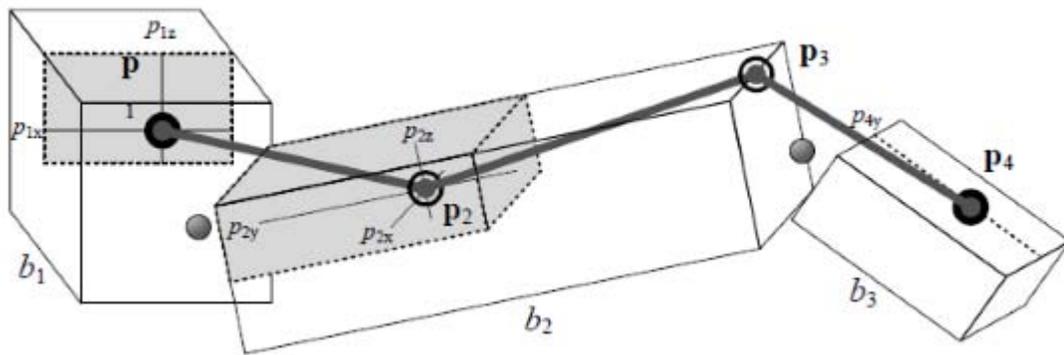


Figura 2-14: Ejemplo de puntos de fijación en los cuerpos rígidos y las regiones permitidas, la línea gris muestra la ruta del músculo, imagen tomada de [16].

El sistema de control tiene por objetivo producir las señales de excitación muscular, este sistema consiste en una máquina de estados finitos que distingue entre 4 estados; soporte de la pierna, despegue de la pierna, oscilación de la pierna y preparación para el soporte. Cada pierna tiene su máquina de estados y en cada paso de simulación esta se actualiza en base al estado de la pierna y compone un conjunto de poses objetivo para un conjunto mínimo de partes corporales (cabeza, torso y segmentos de las piernas). Dichas poses objetivo están basadas en reglas de retroalimentación diseñadas para permitir la variación de velocidad, control de dirección y balance. Las poses se encuentran a través de la optimización de parámetros que se usan para construirlas. Se utiliza una variación orientada a músculos del control Jacobiano transpuesto para calcular la señales de excitación que se deben emitir para que la fuerzas musculares generadas dirijan las partes corporales a la orientación y posición definidas por las poses objetivo, un pequeño retraso se incorpora en cada

señal de excitación para simular el retraso natural que existe en los sistemas neuronales. La figura 2-15 muestra este sistema de control.

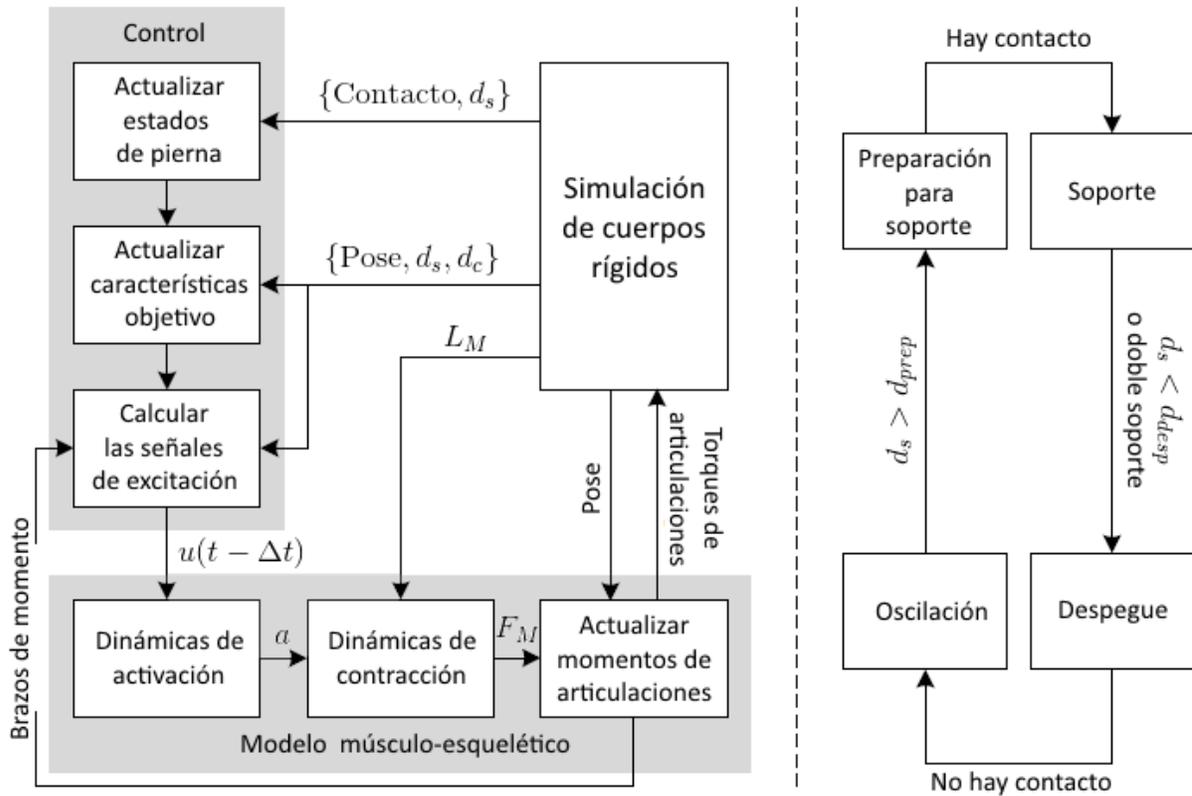


Figura 2-15: El sistema de control muscular, imagen adaptada de [16]. Donde: u es la señal de excitación producida, L_M longitud del músculo, a el nivel de activación, F_M la fuerza del músculo, d_s , d_c distancia entre el pie y centro de masa en plano sagital y coronal respectivamente, por último d_{prep} , d_{desp} son umbrales.

La optimización de los múltiples parámetros del modelo muscular y de control se hace a través de la **estrategia evolutiva de adaptación de matriz de covarianza (CMA-ES)** [19] con el objetivo de minimizar un error total conformado por términos de: velocidad objetivo, desviación de la orientación de la cabeza, desviación de la velocidad objetivo de la cabeza, deslizamientos de pies, y esfuerzo realizado. Durante una evaluación, condiciones adicionales deben cumplirse para que esta no sea terminada prematuramente, estas condiciones son; el **CoM** no debe caer debajo del 90% de su altura normal, la dirección del caminado no se desvíe con respecto a la dirección deseada por más de 45 grados y que colisiones mutuas o cruzamiento de piernas no ocurran.

Aunque el trabajo de Geijtenbeek está orientado a la animación de personajes, los resultados demuestran que el marco de trabajo desarrollado es capaz de generar caminados que son biológica

2.3. *TRABAJO RELACIONADO*

y visualmente creíbles, estables, pueden ser dirigidos hacia los lados y son robustos ante cambios de velocidad, terreno irregular y perturbaciones externas, se probaron además en diferentes morfologías humanas y otras parecidas a aves o dinosaurios.

3.1. Algoritmo NEAT

El algoritmo de **Neuroevolución de topologías incrementales (NEAT)** fue propuesto por Kenneth Stanley en [51], a diferencia de otros métodos de **EANN** donde las redes por evolucionar tienen una estructura (topología) fija, en la cual no se agregan nuevas neuronas ni conexiones entre las neuronas que están presentes, **NEAT** implementa mecanismos que permiten explorar el espacio de búsqueda efectivamente mientras se incrementa gradualmente la estructura (neuronas y conexiones), a este proceso se le denomina *complejización*. El término **redes neuronales artificiales de topología y pesos en evolución (TWEANN)** se aplica cuando la neuroevolución modifica tanto los pesos como la topología de las redes [59]. Al complejizar una red se obtiene una modificación en la función de la red y se incrementa el espacio de búsqueda, si se efectúa correctamente esto tiene la ventaja de permitir que el método determine la estructura correcta librando al diseñador de tener que determinar la estructura apropiada, lo cual en si no es una tarea sencilla. A continuación se explican los pormenores del algoritmo complementando la información del artículo original donde se describe este algoritmo con detalles que provienen de [50] y la página oficial del algoritmo **NEAT**¹.

3.1.1. El genoma y fenoma

Al igual que en un **GA** típico, en el algoritmo **NEAT** también se opera sobre una población de individuos codificados, en este caso el genoma de un individuo está dividido en 2 estructuras. La primera estructura describe las neuronas de la red y se puede denominar como los *genes de nodo* o simplemente *nodos* y la segunda estructura describe las conexiones entre neuronas y se puede

¹<https://www.cs.ucf.edu/~kstanley/neat.html>

denominar como los *genes de conexión* o simplemente *genes*. La figura 3-1 muestra esta codificación.

La información que contienen los genes de nodo:

- Identificador numérico.
- Tipo de nodo, el tipo *sensor* designa neuronas que son pasivas, se ubican en la capa de entrada de la red y solo distribuyen la entrada que reciben sin realizar cómputo, el tipo *neurona* designa neuronas activas que son las que realizan cómputo en la red.
- Etiqueta, designa la capa en la que se ubica la neurona o el rol de la neurona, estas etiquetas son; entrada, bias, escondida y salida.

La información que contienen los genes de conexión:

- Número de innovación, es un indicador del origen evolutivo del gen de conexión.
- Nodo a la entrada, el identificador numérico del nodo que está a la entrada de esta conexión, es decir, la salida de este nodo se transmite por esta conexión.
- Nodo a la salida, el identificador numérico del nodo que está a la salida de esta conexión, es decir, una entrada de este nodo se recibe por esta conexión.
- Peso, es el peso sináptico de la conexión.
- Habilitación, es una bandera o bit que indica si esta conexión se manifiesta o no en el fenotipo cuando se hace la decodificación.

3.1.2. El número de innovación

El número de innovación, que se mencionó como parte de la información que contienen los genes de conexión, es un marcador que tiene el objetivo de indicar el orden de aparición de una característica estructural y va a permitirle al algoritmo **NEAT** realizar lo que en la genética biológica se conoce como homologación de secuencias (la identificación de secuencias en el código genético que comparten un mismo origen evolutivo) y al igual que en la naturaleza esto le permite al algoritmo diferenciar los genes para alinearlos y realizar la cruce de individuos recomblando sus genes. El número de innovación es global y a lo largo de todo el proceso solo se asigna y modifica cuando a través de la mutación aparecen nuevos genes de conexión (nueva estructura de red) que se introducen al genoma de un individuo, la intuición que se sigue es que entre generaciones las circunstancias por las cuales aparece un gen son distintas, por lo tanto un gen solo es una innovación

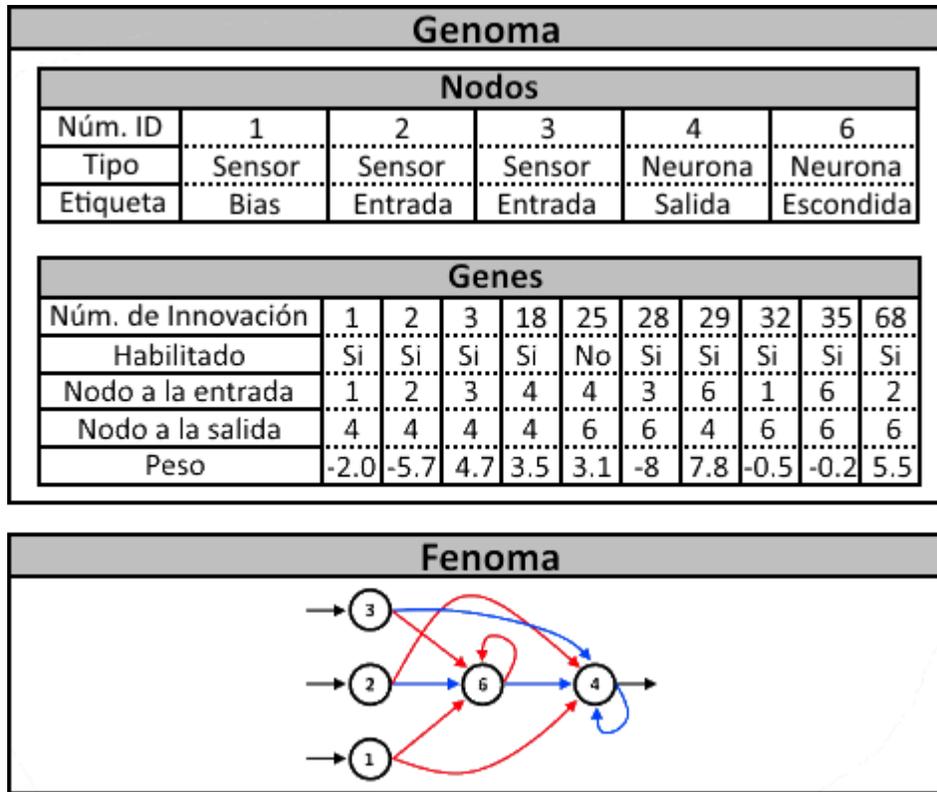


Figura 3-1: Genoma y fenoma de NEAT.

cuando aparece por primera vez en la generación que esta transcurriendo, si un gen aparece en esa misma generación para realizar la misma función entonces tiene un mismo origen. De acuerdo a esa intuición, en **NEAT** por generación se mantiene una lista de innovaciones que lleva el registro de las innovaciones ocurridas, de esta forma al surgir un nuevo gen que no aparece en esta lista se agrega al genoma con el número de innovación actual, se incrementa el número de innovación y se registra la innovación ocurrida en la lista, ahora bien, si al surgir un nuevo gen hay un registro en la lista que indica que no es innovación, al agregarse este gen al genoma lo hace con el mismo número de innovación que indica el registro. Un gen de conexión se considera que tiene la misma función que otro (mismo origen) si es que surge como resultado del mismo tipo de mutación e involucra a los mismos elementos estructurales.

3.1.3. Las mutaciones

El operador de mutación es el único que puede añadir nueva estructura a las redes, hay 3 tipos principales de mutación en **NEAT** y todas ocurren de acuerdo a sus determinadas probabilidades.

La *mutación de pesos* puede cambiar el valor del peso de cada conexión. La *mutación de añadir conexión* escoge aleatoriamente un par de nodos de la red y si estos no están conectados agrega una nueva conexión entre ellos, añadiendo un gen de conexión con peso aleatorio, una probabilidad adicional determina si se permite que la conexión por realizar sea recurrente. La *mutación de añadir nodo* elige aleatoriamente una conexión habilitada en la red y la deshabilita, esta conexión es reemplazada por un nuevo nodo y dos conexiones, la primera conexión sale del nodo que estaba a la entrada de la conexión que se deshabilitó y llega al nuevo nodo, al agregar al genoma su gen de conexión este recibe un peso de 1, la segunda conexión sale del nuevo nodo y llega al nodo que esta a la salida de la conexión que se deshabilitó, este gen de conexión recibe el mismo peso que tiene la conexión que se deshabilitó, el nuevo nodo en cuestión se agrega a los genes de nodo recibiendo el siguiente número de identificación de nodos en la población (este número también es global) si es que esta mutación es una innovación, si no lo es, entonces recibe el número de identificación que indique el registro en la lista de innovaciones. Mutaciones adicionales que pueden cambiar el estado de habilitación de una conexión al azar también se pueden efectuar en este algoritmo. La figura 3-2 ilustra la mutación.

3.1.4. La cruza

La cruza puede realizarse de forma efectiva en **NEAT** gracias a los números de innovación, considerando los genomas de 2 padres que van a cruzarse, sus genes de conexión se alinean en base a sus números de innovación, genes que coinciden en número de innovación son *genes emparejados*, para los demás genes, el número de innovación máximo entre ambos padres se compara y el más pequeño entre estos define el límite numérico superior en el que pueden encontrarse genes emparejados, en base a esto, genes cuyo número de innovación es más pequeño que el límite y no aparecen en alguno de los padres son genes disjuntos, genes cuyo número de innovación es más grande que el límite son genes de exceso respectivamente. Al elegir los genes del nuevo individuo, cada gen emparejado se hereda aleatoriamente de alguno de los padres, mientras que un gen que sea disjunto o de exceso solo se hereda del padre con mejor aptitud, si los padres empatan en aptitud el que tenga genoma más corto es mejor. Cuando se han determinado los genes de conexión del descendiente solo los nodos indicados por la información de los genes de conexión se utilizan para asignarle su genes de nodos. La figura 3-3 ilustra la cruza.

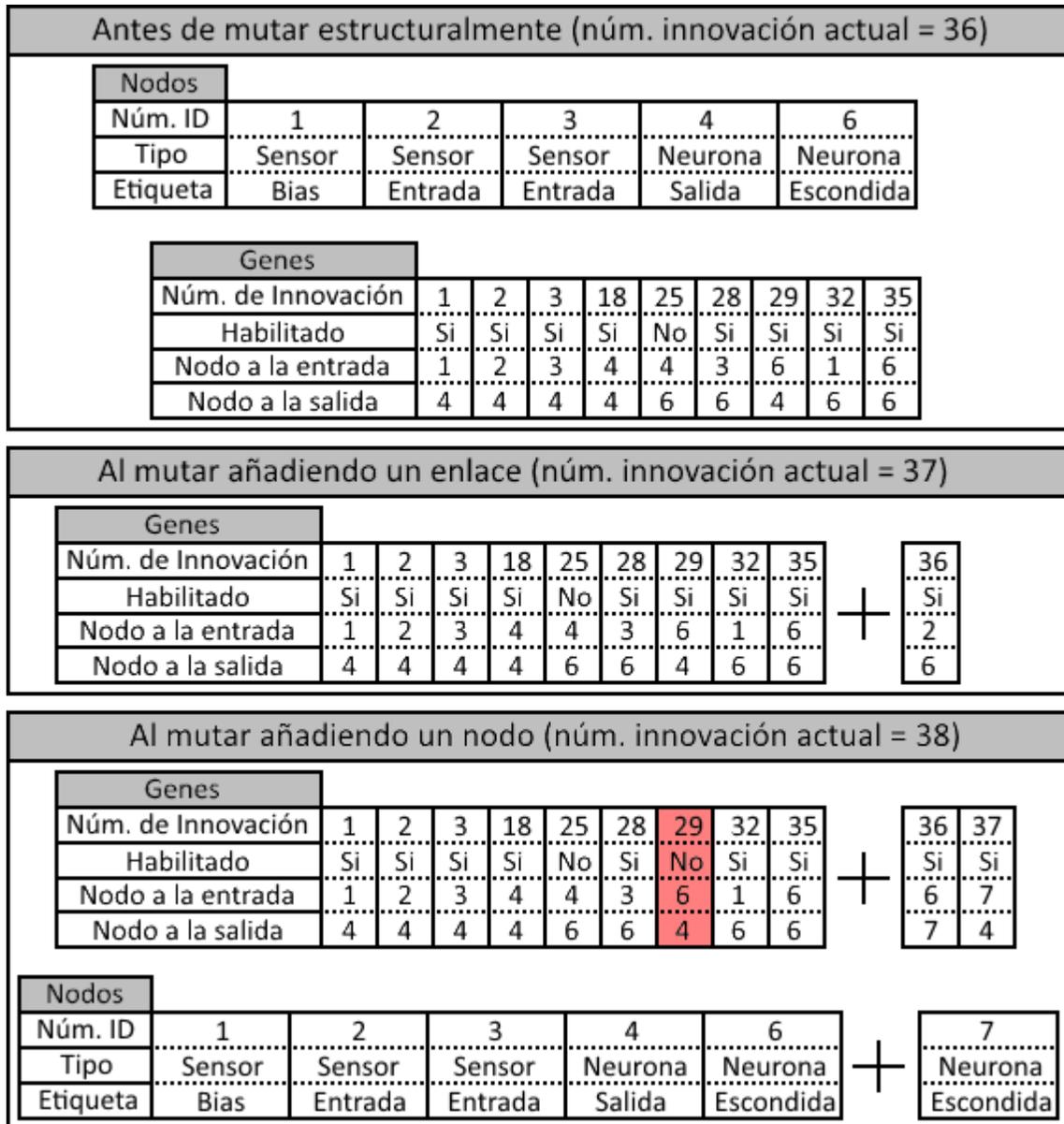


Figura 3-2: Mutación en NEAT.

3.1.5. La especiación y reproducción

Ya que al añadir estructura a una red, esta cambia su funcionalidad sin necesariamente presentar de inmediato una mejora de aptitud es posible que en la siguiente generación un individuo que acaba de modificar su estructura no sobreviva pues tiene que competir con individuos más antiguos y que se encuentran en ese momento más optimizados, los mecanismos de *especiación* y *compartición explícita de aptitud* (explicit fitness sharing) se utilizan para evitar este fenómeno.

Genes del padre 1 aptitud 3.5															
Núm. de Innovación	1	2	3	4	11	12	57	72	395				475		
Habilitado	Si	No	No	Si	Si	Si	Si	Si	Si				Si		
Nodo a la entrada	1	2	3	4	3	6	2	1	5				4		
Nodo a la salida	5	5	5	5	6	5	6	6	6				6		
Genes del padre 2 aptitud 5.3															
	Genes emparejados								Genes Disjuntos				Genes de Exceso		
Núm. de Innovación	1	2	3	4	11	12	57	72		404	451	452		526	560
Habilitado	Si	Si	Si	Si	Si	Si	Si	Si		Si	Si	Si		Si	Si
Nodo a la entrada	1	2	3	4	3	6	2	1		6	1	25		25	4
Nodo a la salida	5	5	5	5	6	5	6	6		6	25	6		5	25
Descendiente															
	↓ Si son emparejados cada padre tiene 50% de heredar su gen								↓ En otro caso solo se heredan del padre más apto				↓		
Núm. de Innovación	1	2	3	4	11	12	57	72		404	451	452		526	560
Habilitado	Si	Si	No	Si	Si	Si	Si	Si		Si	Si	Si		Si	Si
Nodo a la entrada	1	2	3	4	3	6	2	1		6	1	25		25	4
Nodo a la salida	5	5	5	5	6	5	6	6		6	25	6		5	25

Figura 3-3: Cruza en NEAT.

La especiación consiste en que individuos con características similares se agrupan y compiten principalmente entre ellos con la finalidad de darles mejores oportunidades de sobrevivir. Las diferencias estructurales pueden usarse para separar una red de otro, por tanto los genes de conexión, en sí el número de genes emparejados, disjuntos y de exceso, son apropiados para ayudar a determinar que tan parecidos son 2 individuos. Formalmente se requiere una medida que permita definir que tan alejado está un individuo de otro, la siguiente ecuación define esta medida como la *distancia de compatibilidad*:

$$\delta(i, j) = \frac{c_d D}{N^s} + \frac{c_e E}{N^s} + c_p \bar{P} \quad (3-1)$$

En donde:

i, j son los individuos que se están comparando.

3.1. ALGORITMO NEAT

c_d , c_e y c_p son coeficientes que se utilizan para ajustar la importancia que cada término tiene en la determinación de la distancia.

D y E son el número de genes disjuntos y de exceso respectivamente que hay entre i y j .

N es el número de genes de conexión que hay en el genoma más grande.

\bar{P} es el promedio de las diferencias que existen entre los pesos sinápticos de los genes emparejados.

s es un coeficiente que puede tomar el valor de 0 ó 1. Con $s = 0$ el número de genes disjuntos o de exceso no se normaliza. Con $s = 1$ el número de genes disjuntos o de exceso se normaliza con respecto al número de genes en el genoma mas grande. Para genomas grandes $s = 1$ se considera mejor ya que la distancia es determinada por la proporción de genes y no el número total.

Una lista de especies se utiliza para saber que especies existen en la población en todo momento y que organismos las componen, cuando se crea un nuevo individuo i antes de que este pueda asignarse a una especie primero hay que definir un *umbral de distancia* δ_u , entonces se prosigue a recorrer esta lista de especies de mejor a peor aptitud, se toma un individuo representativo j (normalmente el primero) de la especie y se calcula $\delta(i, j)$, el individuo i puede asignarse entonces a la primer especie donde se cumpla $\delta < \delta_u$. Si esa condición no se cumple para ninguna de las especies existentes se crea una nueva especie que recibe a i . Este método de asignar a una especie es el predilecto ya que es el más económico en tiempo, pero si es necesario pueden adoptarse variantes, por ejemplo, el individuo solo se asigna a la especie que cumpla la condición $\delta < \delta_u$ y que también minimiza δ al comparar contra todas las especies o en lugar de comparar contra un solo individuo representativo se compara con todos los individuos de la especie y δ se calcula como el promedio de esas comparaciones.

La compartición explícita del desempeño tiene por objetivo que cada individuo comparta su desempeño la especie a la que pertenece, para que al momento de reproducir la población la descendencia se asigne de forma más pareja entre especies, la siguiente fórmula indica cómo la aptitud original que un individuo recibe al ser evaluado se ajusta con respecto a su especie:

$$f'_i = \frac{f_i}{S_i} \quad (3-2)$$

En donde:

f_i y f'_i son la aptitud original de un individuo y a aptitud ajustada respectivamente.

S_i el número de individuos que forman la especie.

Una vez que la aptitud del individuo se ha ajustado se calcula el número de descendientes que se le otorgan a su especie por medio de la fórmula:

$$N'_j = \frac{\sum_{i=1}^{N_j} f'_{ij}}{\bar{f}'} \quad (3-3)$$

En donde:

N_j y N'_j son el número actual de individuos en la especie y el número de individuos esperados respectivamente.

f'_{ij} es la aptitud ajustada del individuo i perteneciente a la especie j .

\bar{f}' es el promedio de las aptitudes ajustadas de todos los individuos en la población.

Ya que se realizó el cálculo de los descendientes esperados de cada especie se procede a eliminar un porcentaje de los peores individuos en cada una, los individuos restantes se pueden seleccionar como posibles candidatos para generar la descendencia que reemplazará a la población total de la especie.

3.1.6. La minimización de la dimensionalidad

Algunos algoritmos para **TWEANN** normalmente inician con una población con estructura aleatoria con el fin proveer la diversidad de topologías desde un inicio ya que no poseen mecanismos para asegurar la supervivencia de nueva estructura que pueda surgir posteriormente, como en **NEAT** si hay mecanismos para afrontar ese problema, se vuelve más importante el garantizar que el espacio de búsqueda sea explorado efectivamente, esto se hace inicializando con una población uniforme de redes de topología mínima, el diseñador solo debe asegurarse de indicar el mínimo de nodos que se requieren (solo el número de entradas y salidas las cuales se conectan todas directamente) y el algoritmo se va a encargar de explorar ese espacio mínimo e incrementar gradualmente la estructura solo cuando este aumento lleve a mejoras en la aptitud de los individuos. Para problemas donde el número de entradas/salidas es muy grande es discutible que iniciar con una configuración de entradas y salidas totalmente conectadas sea un espacio mínimo, en estos casos una red totalmente

conexa con algunos nodos escondidos entre las capas de entrada y salida puede resultar en un espacio considerablemente más pequeño que de mejores resultados conforme se evoluciona.

3.1.7. Ventajas y desventajas del algoritmo

Los mecanismos y operadores de **NEAT** le otorgan las siguientes ventajas sobre otros métodos:

- Su codificación es muy simple y permite tanto el crecimiento del genoma y el alineamiento de los genes para la cruce.
- La cruce es un operador que tradicionalmente se trata de evitar en algoritmos que evolucionan redes neuronales. El problema de las convenciones competitivas (permutaciones de los pesos sinápticos en una estructura de red, pueden llevar a una misma funcionalidad) y el problema de longitud variable del genoma (no hay una forma totalmente a prueba de fallos de comparar y cruzar genomas que no tienen una misma longitud y representan distintas estructuras) hacen que la cruce efectuada sin un análisis estructural para evitar estos problemas produzca descendientes defectuosos, si bien es cierto que **NEAT** no realiza complejos análisis estructurales con el fin de solucionar estos problemas, las marcas históricas se han probado como un mecanismo muy eficiente para evitar incurrir en estos 2 problemas y hacen de **NEAT** un método con una relación más estrecha con los proceso de evolución natural.
- La especiación y compartición explícita de la aptitud son mecanismos que protegen la innovación estructural y evitan que especies con muy buena aptitud acaparen totalmente la descendencia, especies con muchos individuos se ven penalizadas cuando sus individuos empiezan a ser poco aptos y especies con pocos individuos se ven recompensadas si sus individuos empiezan a ser muy aptos, al explorar múltiples estructuras separadas por especie se evita la convergencia prematura y se cubre mejor el espacio de búsqueda.
- Al iniciar con estructuras mínimas permite encontrar efectivamente soluciones en los espacios dimensionales apropiados para un problema, debido a la especiación soluciones que permanecen competitivas sobreviven aun si su estructura es mínima, otro aspecto es que al no realizarse una cruce arbitraria de segmentos de los genomas se evita el crecimiento desmesurado de las soluciones.
- Se puede aplicar para el aprendizaje no supervisado ya que no requiere de datos de entrenamiento para la redes, en contraste el algoritmo de EPNet propuesto en [60] requiere de

entrenamientos parciales y las redes que evoluciona solo son progresivas.

Algunas de las desventajas que se presentan:

- Posee un gran número de parámetros que se describen en el apéndice B, encontrar una combinación que permita el funcionamiento apropiado del algoritmo requiere mucha prueba y error.
- El algoritmo no es muy bueno para explorar espacios de búsqueda donde las soluciones están separadas por brechas dimensionales.

Un aspecto importante que hay que mencionar es que la activación de una red NEAT difiere de la forma tradicional en donde la propagación de las entradas hasta la salidas se realiza en un solo paso de tiempo y si hay recurrencias presentes estas tienen un retraso unitario. Ya que la estructura puede resultar arbitraria, compleja y recurrente, en la activación de la red NEAT solo se propaga la señal de las neuronas activadas en el último paso a las siguientes neuronas que se encuentran en sus conexiones de salida, de esta forma propagar la señal desde las entradas hasta la salidas puede requerir de múltiples activaciones. En problemas donde se requiere que la red produzca una salida definitiva como resultado, clasificación por ejemplo, hay dos criterios que pueden seguirse para activar la red el número de veces apropiado, el primero consiste en determinar un número fijo de activaciones que sea suficiente para que la señal se propague completamente (la profundidad de la red, el camino más largo formado por las conexiones de la red que conecta las entradas con las salidas, puede utilizarse como el número de activaciones), el segundo criterio más apropiado es activar cuantas veces sea necesario hasta que la red se estabiliza, es decir, las neuronas ocultas y de salida no cambien sus valores de salida o no cambian más de lo establecido por un umbral, recordando que en una RNN la estabilidad no puede garantizarse esta forma puede no ser conveniente. En un problema de control las entradas a la red pueden estar continuamente variando en el tiempo por lo que el estado de la red seguramente nunca se estabilizará, en este caso la activación de la red se realiza de acuerdo a un ciclo de control temporizado donde nuevas entradas se alimentan, se activa la red y se emiten las salidas de control que en ese punto se hayan activado en la red. El tipo de activación que emplea NEAT si bien tiene la desventaja de no permitir a la red reaccionar inmediatamente le permite evolucionar propiedades temporales como patrones de oscilación y se relaciona más con el funcionamiento de las redes neuronales naturales donde una señal toma cierto tiempo en propagarse. En la figura 3-4 se muestran ejemplos de redes que NEAT puede evolucionar.

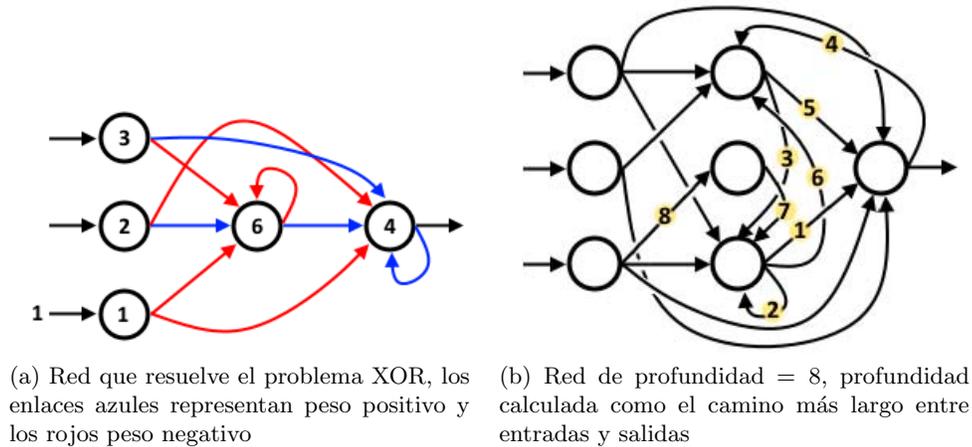


Figura 3-4: Ejemplos de redes evolucionadas con NEAT.

3.2. Simulador basado en física

Una herramienta que se ha vuelto especialmente útil para la investigación en materia de robótica son los simuladores, los puntos principales que hacen a los simuladores tan útiles son que permiten: diseñar y probar prototipos, algoritmos mucho más rápido que en implementaciones físicas y sin el riesgo de que ocurra algún daño si algo falla, recrear los ambientes donde se desean realizar las pruebas, ajustar las condiciones que rigen las interacciones físicas y recolectar datos de manera confiable.

Un simulador que solo es capaz de recrear la cinemática de los cuerpos puede ser útil si en las pruebas que se desean realizar las interacciones entre los cuerpos y partículas carecen de importancia. Ahora bien, en el mundo real hay leyes físicas que rigen las interacciones entre cuerpos y partículas, si se desea realizar una prueba donde si hay interacciones entre elementos y ambiente físicos entonces es esencial utilizar un simulador que pueda replicar los efectos que están determinados por la física (un simulador basado en física). El motor de física es el elemento de software que en un simulador basado en física se encarga de realizar los cálculos aproximativos que aseguran que la interacción de los elementos simulados se efectúen de acuerdo a la dinámica de cuerpos rígidos, es decir, de acuerdo a las propiedades de los cuerpos, cómo las fuerzas y momentos afectan el movimiento de estos a lo largo del tiempo. Un motor de física puede centrarse únicamente en la dinámica de los cuerpos rígidos, pero también existen motores de física que soportan o se enfocan exclusivamente a la dinámica de cuerpos elásticos o fluidos.

El simulador mantiene el estado del mundo y sus elementos en un tiempo t , para avanzar en el

tiempo y actualizar el estado del mundo y sus elementos se realizan los cálculos para determinar el estado siguiente en $t + \epsilon$ donde ϵ es el tamaño de un paso de simulación que se realiza, la figura 3-5 ilustra el ciclo de simulación.

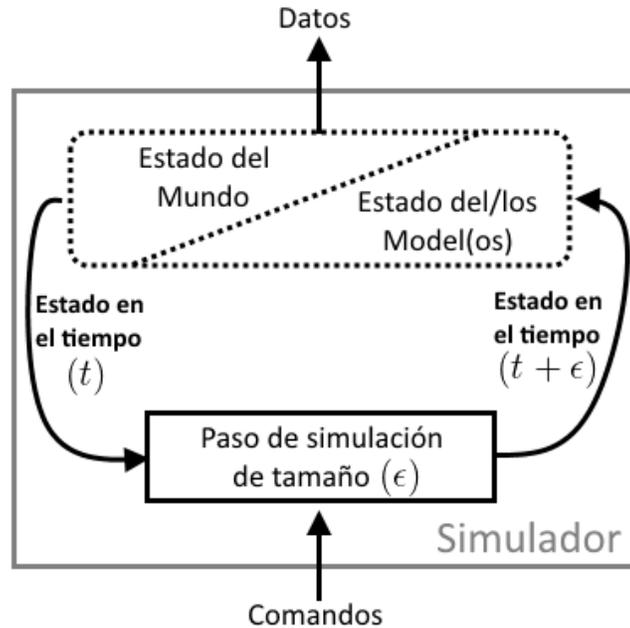


Figura 3-5: Ciclo de simulación.

3.2.1. Simulador Gazebo

Gazebo [38] es un simulador 3-D basado en física que permite simular poblaciones de robots en ambientes interiores y exteriores complejos, en los últimos años se ha vuelto muy popular y ha sido empleado con éxito para simular apropiadamente robots humanoides y robots móviles, algunas plataformas conocidas que se han simulado con este simulador se muestran en la figura 3-6.

Algunos aspectos que hacen interesante y deseable el uso de este simulador se mencionan a continuación:

- Es un proyecto de código abierto soportado por la OSRF (Open Source Robotics Foundation), es totalmente gratuito y periódicamente se realizan nuevas liberaciones del software que implementan nueva funcionalidad y revisiones del código.
- Hay una comunidad de usuarios que lo utiliza y contribuye a su desarrollo, documentación y tutoriales.



(a) Atlas de Boston Dynamics (b) PR2 de Willow Garage (c) Turtlebot de Willow Garage

Figura 3-6: Plataformas robóticas que se han simulado con Gazebo, imagenes tomadas de internet.

- Permite utilizar diferentes motores de física como (ODE, Bullet entre otros).
- El simulador se compone de un servidor que realiza el trabajo de simular el mundo y un cliente gráfico que se conecta al servidor y es el que permite visualizar los elementos con gráficos de muy buena calidad. El servidor y cliente pueden correrse por separado lo cual otorga la libertad de poder utilizar solamente el servidor.
- Se puede hacer interfaz con este simulador de diferentes maneras, a través del paso de mensajes (implementado con protobuf de Google y boost::ASIO), plugins (componente de software que la herramienta carga durante ejecución permitiendo personalizar o añadir funcionalidad) o a través del middleware que ofrece ROS.
- Los elementos y parámetros del mundo virtual así como de los robots, sensores y plugins que se desean agregar a la simulación se pueden definir fácilmente a través de archivos; un archivo del mundo indica todos los elementos y modelos de la simulación mientras que los archivos de modelo se enfocan a definir los elementos de una sola entidad (robot u objeto) y sus propiedades, ambos archivos deben seguir la convención definida por el **formato de descripción de la simulación (SDF)**.
- Múltiples sensores ya vienen integrados, pero si no son suficientes se pueden crear nuevos o alterar los existentes mediante los plugins.

La elección de Gazebo con ODE como el simulador que se emplea en este trabajo se realizó en base a los aspectos anteriores. Originalmente Bullet era el motor de física que ofrecía las cualidades más convincentes pero al probarlo con Gazebo se manifestaron problemas que lo hicieron menos viable que ODE ya que este muestra estar mejor integrado con el simulador. La encuesta que se presenta en [22] muestra que Gazebo es una de las mejores opciones entre diversos simuladores libres mientras que el estudio comparativo de motores de física en [43] se enfoca principalmente en cuestiones de la dinámica que afectan a robots que caminan y muestra que ODE es una buena opción.

Un aspecto que dificultó la adopción de este simulador se debió a detalles muy específicos (parámetros, propiedades o funciones del API) que tuvieron que descubrirse por prueba y error o investigarse en otros sitios, ya que la documentación y tutoriales de Gazebo no los explicaban a fondo.

3.2.2. Detalles importantes de ODE

ODE [48] es una librería libre con la cual se puede simular la dinámica de cuerpos rígidos y detectar colisiones, es uno de los motores de física para cuerpos rígidos más conocidos y se ha usado en varios juegos y se ha embebido en muchas herramientas.

En un cuerpo articulado, una articulación entre dos cuerpos limita las posiciones y orientaciones que puede haber entre estos, una articulación es entonces una relación que puede llamarse restricción.

Cuando se simulan cuerpos rígidos a lo largo del tiempo se hace mediante un proceso que se llama integración, para avanzar el tiempo y actualizar el estado de los cuerpos rígidos se debe efectuar un paso de integración (paso de simulación), el integrador que ODE utiliza es muy estable pero requiere un tamaño de paso pequeño para que sea preciso. Para poder integrar la solución, primero se necesita resolver un sistema de ecuaciones diferenciales dado por las restricciones para garantizar que estas se van a satisfacer, ODE tiene 2 funciones que realizan el paso de simulación y aplican diferente solución. La primera función se puede considerar que es un “paso normal”, usa un método basado en una matriz grande y toma tiempo $O(m^3)$ y memoria $O(m^2)$ donde m es el número de renglones de restricciones, es el método más preciso pero en simulaciones de sistemas con muchas restricciones puede volverse muy lento. La segunda función se puede considerar que es un “paso rápido”, usa un método iterativo y toma tiempo $O(mN)$ y memoria $O(m)$ donde N es el número de iteraciones que se desea que efectúe el método, es un método considerablemente más

3.2. *SIMULADOR BASADO EN FÍSICA*

rápido pero más impreciso, efectuar un número mayor de iteraciones con este método aumenta la precisión pero consume más tiempo.

4.1. Modelo del robot y propiedades del ambiente simulado

El primer paso para poder realizar la simulación de un robot, es definir el modelo virtual que va a representarlo y definir las propiedades físicas que lo caracterizan. En el caso del robot NimbRo-OP al ser una plataforma abierta el equipo que lo desarrolló, ya se encargó de proveer múltiples archivos que definen cada segmento del robot y un archivo adicional que también se usa para especificar la simulación y contiene la mayoría de las propiedades físicas que se necesitan, este archivo viene en un formato que no es **SDF** pero se pudo convertir a este formato usando una herramienta de Gazebo, posteriormente el archivo generado se modificó a mano para incluir los ajustes y elementos adicionales requeridos en esta simulación

La figura 4-1 muestra la cadena cinemática del robot NimbRo-OP que se desea simular, en esta simple representación se pueden apreciar los segmentos principales del robot y las articulaciones que las interconectan resultando en un total de 20 **DoF**. Con el fin de simplificar el modelo y ya que la tarea de movilidad bípeda involucra principalmente el torso y las piernas, en el modelo virtual final, al cual vamos a denominar a partir de ahora como el *caminador*, no se incluyen los brazos ni la cabeza del robot por lo que solo hay 12 **DoF** que pueden controlarse.

4.1.1. Segmentos

Cada segmento del robot es un cuerpo rígido para el cual se necesita definir; (1) la representación visual, que tiene solo el fin de aportar a la visualización del modelo cuando se ejecuta el cliente gráfico de la simulación, (2) la representación geométrica que va a emplearse para la detección de colisiones con otros elementos, (3) la posición del origen y orientación del segmento con respecto al segmento base del modelo (en este caso el segmento base es el torso), y finalmente(4) las propiedades

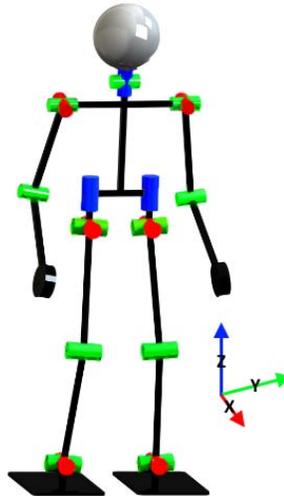


Figura 4-1: Representación del robot NimbRo-OP como cadena cinemática, los ejes coordenados mostrados ilustran el eje alrededor del cual se realizan las rotaciones de las articulaciones y coinciden con el sistema de coordenadas del simulador, imagen adaptada de [46].

físicas que definen su comportamiento dinámico (masa, posición del **CoM**, matriz de inercia) y las propiedades del contacto (**CoF** estática (μ_s), coeficientes de amortiguamiento k_d y rigidez k_p). La figura 4-2 muestra las representación visual y geométrica del caminador.

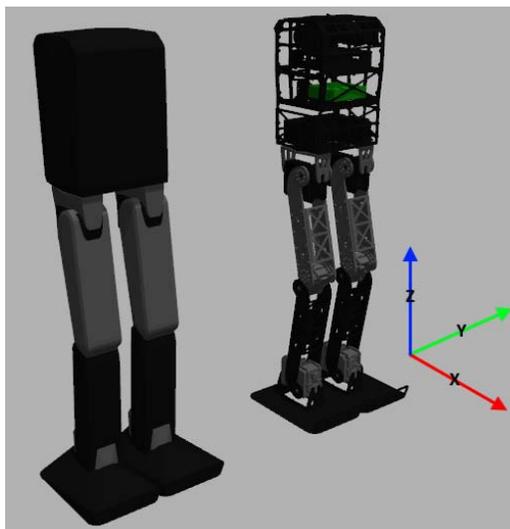


Figura 4-2: Modelos virtuales del caminador, a la izquierda se ven los segmentos representados con cerrres convexos que se usan como la geometría para colisiones y a la derecha se ven mallas de triángulos complejas que se usan solo para la visualización.

Normalmente se escogen figuras simples para la geometría de colisiones, debido a que los métodos

de detección de colisiones están optimizados para trabajar con cuboides, esferas y cilindros, en este caso se usa un cierre convexo que si bien no es una figura simple, es considerablemente más simple que una malla triangular detallada y delimita apropiadamente los segmentos, si se usa una malla de triángulos muy detallada la detección de colisiones puede consumir más ciclos de cómputo, afectando el *factor de tiempo real* $\frac{\text{segundos simulados}}{\text{segundos reales transcurridos}}$ que el simulador puede lograr. Otro aspecto respecto a las colisiones es que entre los segmentos de un mismo modelo estas se deshabilitan con el fin de ahorrar cálculos y evitar el comportamiento errático que sucede debido a colisiones en los traslapes de segmentos que ocurren cerca de las articulaciones.

Las propiedades físicas que describen cada segmento se muestran en la tabla 4-1, ya que los pies son los que deben realizar el contacto con el suelo son los únicos que tienen propiedades de contacto, el **CoF** estático se determinó experimentalmente poniendo uno de los pies del robot real sobre un plano de material plástico (similar al pasto sintético que se planea usar a partir del año 2015 en las canchas para las competencias de Robocup) y determinando el ángulo de inclinación del plano en donde ocurre el deslizamiento, los coeficientes de amortiguamiento y rigidez partieron de los establecidos en el modelo de contacto del robot Atlas y mediante prueba y error se ajustaron para asegurar que el contacto no tuviera rebote ni penetración entre superficies. La posición del **CoM** no se muestra ya que esta tiene que definirse relativa al origen de la geometría de colisión que representa al segmento, en el torso este origen se encuentra muy cerca del **CoM** mientras que en todos los demás segmentos el origen se encuentra en el punto donde debe existir una articulación (el eje de rotación de la articulación pasa por el origen) la figura 4-3 muestra un ejemplo de segmentos y la posición de su origen.

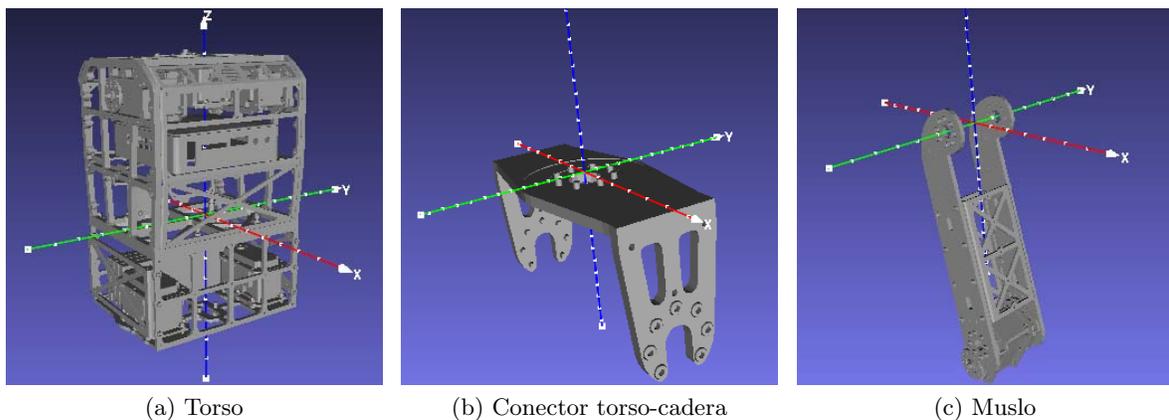


Figura 4-3: Algunos segmentos y su punto de origen.

Segmento	Dimensiones caja envolvente (largo, ancho, alto)	Masa	Tensor de Inercia		Contacto
Torso	0.136152 0.1618 0.2437	1.8051	$I_{xx} = 0.016444$ $I_{yy} = 0.013972$ $I_{zz} = 0.0064508$	$I_{xy} = 1.4574E-05$ $I_{xz} = 0.00048254$ $I_{yz} = 5.0355E-06$	-
Conector torso-cadera izquierdo	0.1012 0.05 0.054925	0.073374	$I_{xx} = 2.6774E-05$ $I_{yy} = 0.00011478$ $I_{zz} = 0.00010743$	$I_{xy} = 1.9727E-19$ $I_{xz} = -8.7961E-19$ $I_{yz} = -2.3128E-19$	-
Conector torso-cadera derecho	0.1012 0.05 0.054925	0.073374	$I_{xx} = 2.6774E-05$ $I_{yy} = 0.00011478$ $I_{zz} = 0.00010743$	$I_{xy} = 1.5536E-19$ $I_{xz} = -9.0622E-19$ $I_{yz} = -2.2736E-19$	-
Conector cadera-muslo izquierdo	0.0992 0.0592 0.0.651	0.35844	$I_{xx} = 0.00015501$ $I_{yy} = 0.00032958$ $I_{zz} = 0.00027786$	$I_{xy} = -1.7299E-06$ $I_{xz} = 8.5156E-06$ $I_{yz} = -2.4435E-07$	-
Conector cadera-muslo derecho	0.0992 0.0592 0.0.651	0.35844	$I_{xx} = 0.00015503$ $I_{yy} = 0.0003299$ $I_{zz} = 0.0002782$	$I_{xy} = 1.738E-06$ $I_{xz} = 8.5148E-06$ $I_{yz} = 2.4403E-07$	-
Muslo izquierdo	0.078741 0.0652 0.258876	0.38276	$I_{xx} = 0.0020952$ $I_{yy} = 0.0020044$ $I_{zz} = 0.00027919$	$I_{xy} = -1.9541E-07$ $I_{xz} = -0.00017822$ $I_{yz} = 5.4912E-06$	-
Muslo derecho	0.078741 0.0652 0.258876	0.38276	$I_{xx} = 0.0020951$ $I_{yy} = 0.0020043$ $I_{zz} = 0.00027919$	$I_{xy} = 1.8755E-07$ $I_{xz} = -0.00017816$ $I_{yz} = -5.4867E-06$	-
Espinilla izquierda	0.067282 0.0652 0.254456	0.20549	$I_{xx} = 0.0011378$ $I_{yy} = 0.0010393$ $I_{zz} = 0.00019656$	$I_{xy} = -2.8771E-12$ $I_{xz} = 9.7123E-05$ $I_{yz} = -1.4479E-11$	-
Espinilla derecha	0.067282 0.0652 0.254456	0.20549	$I_{xx} = 0.0011378$ $I_{yy} = 0.0010393$ $I_{zz} = 0.00019656$	$I_{xy} = -2.8771E-12$ $I_{xz} = 9.7123E-05$ $I_{yz} = -1.4479E-11$	-
Conector espinilla-pie izquierdo	0.0992 0.0592 0.0651	0.35903	$I_{xx} = 0.00015503$ $I_{yy} = 0.0003299$ $I_{zz} = 0.0002782$	$I_{xy} = -1.738E-06$ $I_{xz} = -8.5148E-06$ $I_{yz} = 2.4403E-07$	-
Conector espinilla-pie derecho	0.0992 0.0592 0.0651	0.35903	$I_{xx} = 0.00015501$ $I_{yy} = 0.00032958$ $I_{zz} = 0.00027786$	$I_{xy} = 1.7299E-06$ $I_{xz} = -8.5156E-06$ $I_{yz} = -2.4435E-07$	-
Pie izquierdo	0.208053 0.132 0.049475	0.1488	$I_{xx} = 0.0001853$ $I_{yy} = 0.00060485$ $I_{zz} = 0.00075213$	$I_{xy} = 1.1466E-05$ $I_{xz} = -8.3793E-06$ $I_{yz} = -5.9046E-06$	$\mu_s = 0.643426$ $k_p = 1000000$ $k_d = 1000$
Pie derecho	0.208 0.132 0.049125	0.1488	$I_{xx} = 0.00018527$ $I_{yy} = 0.00060485$ $I_{zz} = 0.00075217$	$I_{xy} = 1.1504E-05$ $I_{xz} = -7.2667E-06$ $I_{yz} = 5.8883E-06$	$\mu_s = 0.643426$ $k_p = 1000000$ $k_d = 1000$

Tabla 4-1: Esta tabla indica los atributos de los segmentos del robot virtual, todas las unidades de propiedades físicas son las estándares del sistema internacional.

4.1.2. Articulaciones

Cada articulación del robot indica la relación entre 2 cuerpos rígidos restringiendo cómo puede moverse uno con respecto al otro, se puede decir que al definir las articulaciones, se le indica al simulador cual es el comportamiento cinemático del modelo sin la necesidad de proporcionar un modelo cinemático. En el caso del caminador todas las articulaciones son rotacionales (solo permiten movimiento angular alrededor de 1 solo eje) y se encuentran actuadas, la figura 4-4 muestra las articulaciones. El primer aspecto importante de una articulación es su posición, la cual se define en relación al origen del segmento precedente o hijo que se conecta al segmento precedente o padre (un ejemplo de esta precedencia es la espinilla que tiene como padre al muslo) ya que en este caso todos los segmentos que pueden ser hijos tienen su origen en un punto que coincide con los ejes de rotación, no se necesita indicar una posición de la articulación. Las propiedades del esfuerzo máximo y velocidad máxima del actuador se toman de las especificaciones de los servos reales del robot, Dynamixel modelo MX-106, mientras que los límites superior e inferior se basan en una aproximación a la posición angular máxima que puede alcanzar antes de que el servomotor empiece a forzarse (estos límites ayudan a prevenir colisiones entre segmentos padre e hijo), una propiedad adicional es el amortiguamiento de velocidad que se introduce con un pequeño valor para evitar que la energía se conserve y propague por el modelo haciendo que vibre. La tabla 4-2 muestra estas propiedades.

Al componer el modelo del caminador en el archivo **SDF** indicando la posición y orientación de cada segmento, se define una postura inicial del modelo, en el presente caminador esta postura es de pie con ambas piernas alineadas y pies firmemente sobre el suelo de manera que es estable, se le denomina *postura cero* a esta configuración inicial y es en esta postura donde todas las articulaciones tienen un valor de posición de cero por lo tanto el límite superior e inferior indicado para cada articulación queda definido a partir de la postura cero.

Articulación	Esfuerzo límite	Velocidad angular límite	Límite angular inferior	Límite angular superior	Amortiguamiento
Guiñada cadera izquierda	8.4	4.71238898	-0.6981317008	1.570796327	0.1
Guiñada cadera derecha	8.4	4.71238898	-1.570796327	0.6981317008	0.1
Alabeo cadera izquierda	8.4	4.71238898	-1.570796327	1.832595715	0.1
Alabeo cadera derecha	8.4	4.71238898	-1.832595715	1.570796327	0.1
Cabeceo cadera izquierda	8.4	4.71238898	-2.35619449	0.872664626	0.1
Cabeceo cadera derecha	8.4	4.71238898	-2.35619449	0.872664626	0.1
Cabeceo rodilla izquierda	8.4	4.71238898	-0.3316125579	2.670353756	0.1
Cabeceo rodilla derecha	8.4	4.71238898	-0.3316125579	2.670353756	0.1
Cabeceo tobillo izquierdo	8.4	4.71238898	-1.535889742	1.535889742	0.1
Cabeceo tobillo derecho	8.4	4.71238898	-1.535889742	1.535889742	0.1
Alabeo tobillo izquierdo	8.4	4.71238898	-1.623156204	1.623156204	0.1
Alabeo tobillo derecho	8.4	4.71238898	-1.623156204	1.623156204	0.1

Tabla 4-2: Esta tabla indica los atributos de las articulaciones del robot virtual, todas las unidades de propiedades físicas son las estándares del sistema internacional.

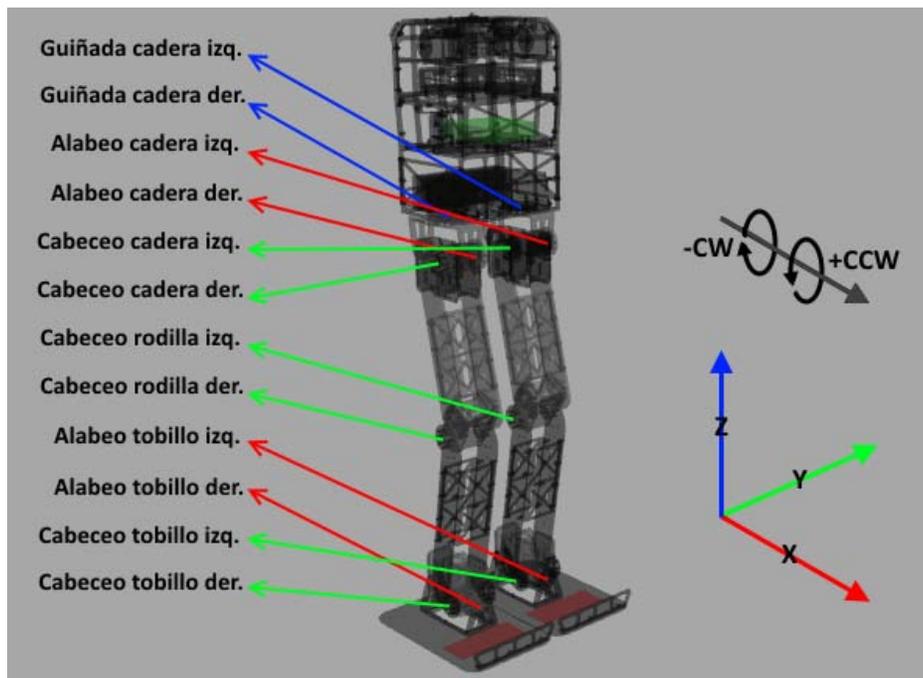


Figura 4-4: Articulaciones del caminador, el nombre de los movimientos guiñada (yaw), alabeo (roll) y cabeceo (pitch) se refieren a las maniobras de navegación de un aeroplano y se aplican también a rotaciones 3-D, también se muestra el signo con el cual una rotación se efectua en sentido o contra sentido de las manecillas del reloj.

4.1.3. Ambiente virtual

El ambiente virtual para los experimentos es muy simple y consiste en un plano totalmente horizontal de 100 x 100 metros, con fuerza normal en la dirección del eje z y restricción de número de colisiones a un máximo de 30, el **CoF** estática de este plano es igual al que se definió para los pies del caminados (al contacto entre 2 elementos el simulador emplea el **CoF** más pequeño entre estos). Los parámetros del simulador y motor de física se muestran en la tabla 4-3.

Aceleración de la gravedad	9.8067
Tamaño del paso de simulación	0.001 (valor por defecto)
Tasa de actualización en tiempo real	6000
Motor de física	ODE
Tipo de solución	paso rápido
Número de iteraciones de la solución	30
Sobre relajación sucesiva (SOR)	1.3 (valor por defecto)
Mezcla de fuerzas de restricciones (CFM)	0 (valor por defecto)
Parámetro de reducción de error (ERP)	0.2 (valor por defecto)
Velocidad máxima de corrección al contacto	100 (valor por defecto)
Capa superficial de contacto	0.001 (valor por defecto)

Tabla 4-3: Esta tabla indica los parámetros del simulador y motor de física, la combinación de estos parámetros junto con las especificaciones del modelo determinan el factor de tiempo real, si todos los cálculos pueden realizarse en el lapso de tiempo apropiado, factor de tiempo real = tamaño del paso de simulación · tasa de actualización en tiempo real.

Los parámetros de **ODE** se pueden consultar con detalle en [48], una breve explicación de cada uno se da a continuación:

- SOR, es el factor de relajación que utiliza para acelerar la convergencia del método de solución que lleva el mismo nombre y que es una variación del método de Gauss-Seidel para resolver sistemas de ecuaciones lineales.
- ERP, determina la proporción de error entre articulaciones que debe corregirse en cada paso de simulación, si este valor se deja en 0 el error que se va acumulando en las articulaciones provoca que los segmentos empiecen a separarse.
- CFM, indica que tanto se puede infringir la dureza de una restricción, haciendo que la relación entre segmentos dada por una articulación se comporte de forma suave como si los materiales fueran elásticos.

- Velocidad máxima de corrección al contacto, indica la máxima velocidad que puede generar la fuerza de corrección de contactos.
- Capa superficial de contacto, indica cuanta penetración entre objetos es permitida antes de que se aplique una fuerza de corrección de contactos.

Cuando se definen las propiedades de contacto para un segmento es posible indicar valores para los últimos 4 parámetros mencionados arriba que aplican solo al segmento en cuestión.

4.2. Sensores, métricas y el control de los actuadores

El caminador necesita, además de su modelo virtual, otros elementos que tienen la finalidad de ayudarlo a llevar a cabo su cometido, sensores para obtener información, métricas que son datos adicionales que se extraen de la simulación y se usan para evaluar la aptitud y finalmente un control que ante una salida de la red pueda comandar los actuadores a la posición correspondiente.

4.2.1. Sensores

El robot físico, en la tarjeta controladora de los servomotores CM-730 que se ubica muy cerca del **CoM** dispone de una unidad de medición inercial (**IMU**) con acelerómetro y giroscopio de 3 ejes cada uno, y cada servomotor tiene un codificador sin contacto por lo que puede reportar su posición angular con gran precisión. Adicionalmente se implementa otro tipo de sensor que no está presente en el robot físico, un par de sensores de contacto binario, estos consisten en una pequeña región táctil en la planta de cada pie. En la figura 4-5 se pueden ver la posición de estos sensores en el caminador.

El simulador Gazebo ya implementa un sensor tipo **IMU**, mientras que para los sensores de contacto y de posición angular no hay un sensor directo, para poder obtener los sensores correctos del caminador, se crearon 3 plugins de sensor, con el primer plugin se hace la modificación del complejo sensor de contacto que provee el simulador para obtener un sensor de contacto binario, con el segundo plugin se hace la modificación del sensor de torque que provee el simulador para obtener un sensor de posición angular/lineal que puede usarse en articulaciones rotacionales (con respuesta en grados o radianes) o prismáticas (con respuesta en metros), con la finalidad de poder interpretar más rápidamente las medidas angulares del sensor estas se manejan en grados, por último con el tercer plugin se realiza el escalamiento de los valores que reporta el sensor **IMU** por defecto del simulador al rango de valores que es capaz de reportar el sensor **IMU** real de la tarjeta

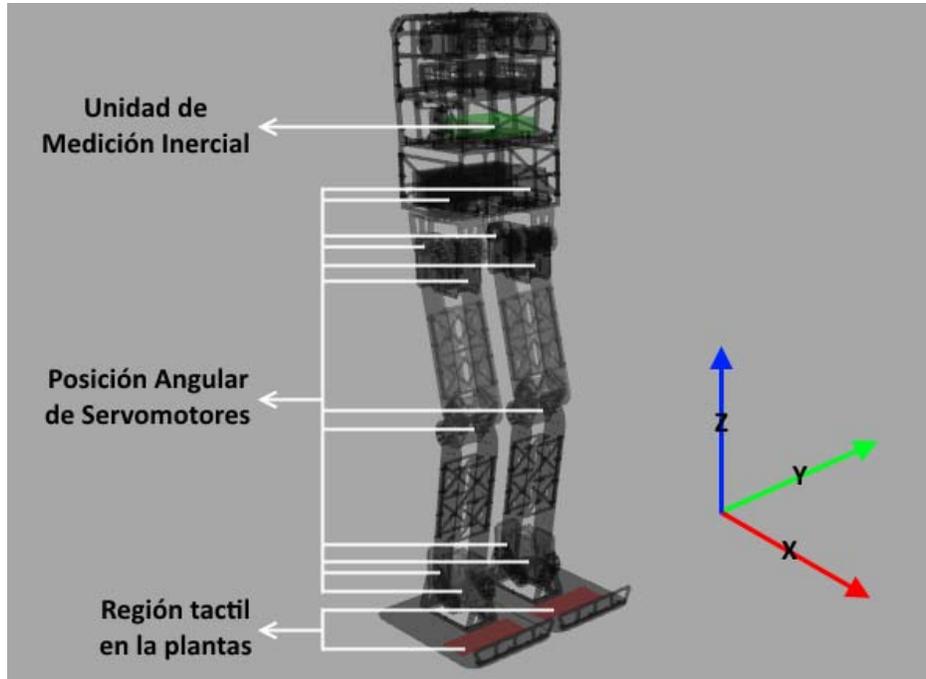


Figura 4-5: Sensores del caminador, el rectángulo verde en el torso y los planos rojos en los pies son segmentos adicionales que se definieron para que les fueran asignados los sensores de IMU y contactos respectivamente y se conectan al modelo a través de articulaciones fijas (sin movimiento ni actuación), cada articulación actuada de las piernas tiene asignado un sensor de posición angular.

controladora. En este caso el acelerómetro virtual por defecto del simulador reporta valores sin un límite preestablecido en las unidades de $\frac{m}{s^2}$ y el acelerómetro real reporta un valor de 10 bits (es decir un valor entre $[0, 1023]$) y corresponde a un valor en el rango de $[-4, 4)$ con unidades en g (aceleración de la gravedad), una lectura de 512 en el acelerómetro real corresponde entonces a $0g$. El giroscopio virtual por defecto también reporta valores sin un límite preestablecido pero en unidades de $\frac{rad}{s}$ mientras que el giroscopio real también reporta un valor de 10 bits pero en este caso corresponde al rango de $[-1600, 1600)$ con unidades en $\frac{deg}{s}$, una lectura de 512 en el giroscopio real equivale a $0\frac{deg}{s}$, las ecuaciones 4-1 muestran las fórmulas usadas para escalar los valores de la IMU virtual por defecto al valor de 10 bits que reportaría la IMU real del robot, si el valor escalado resultante queda fuera de rango de $[0, 1023]$ entonces se asigna el valor máximo o mínimo del rango según corresponda.

$$\begin{aligned} \text{acel}_{\text{escalado}} &= \text{Redondeo}(13.05230098(\text{acel}_{\text{por defecto}}) + 512) \\ \text{giro}_{\text{escalado}} &= \text{Redondeo}(18.33464944(\text{giro}_{\text{por defecto}}) + 512) \end{aligned} \quad (4-1)$$

Para que su comportamiento sea más apegado a sensores reales, a los sensores simulados se les puede añadir ruido. Al acelerómetro se le añade ruido de distribución Gaussiana con media $\mu = 0$ y desviación estándar $\sigma = 0.102863$ y al giroscopio con media $\mu = 0$ y desviación estándar $\sigma = 0.025438$, estos valores se determinaron de una muestra de 4500 mediciones tomadas del acelerómetro y giroscopio en el robot físico sin aplicarle perturbaciones, ya que tanto el acelerómetro como el giroscopio tienen 3 ejes, se toma la σ más grande entre los ejes y μ al tener valores muy pequeños se decidió dejarlo en 0. En el caso del sensor de contacto binario y de posición angular no se añade ruido. También se puede definir la frecuencia a la cual un sensor se actualiza y emite mensajes con información, en este caso todos los sensores tienen frecuencia de $60Hz$ por cada segundo simulado.

4.2.2. Control PID

El control Proporcional-Integral-Derivativo (**PID**) es un tipo de control retroalimentado, que trata de minimizar la diferencia entre una señal de estado objetivo que se desea alcanzar y una señal de estado actual que indica cómo se encuentra el sistema, en el caso del control de posición se busca minimizar la diferencia entre la posición objetivo que se ordene y la posición actual. Ya que no requiere del modelo de lo que está controlando el control **PID** se puede considerar una técnica libre de modelo y puede emplearse para diferentes aplicaciones incluyendo el control de posición de robots, un diagrama de bloques de este control se muestra figura 4-6.

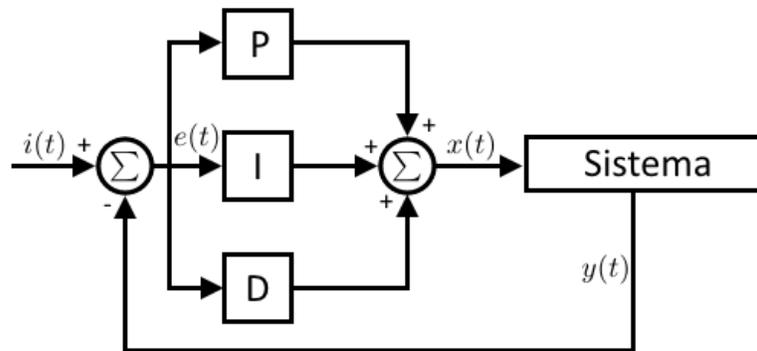


Figura 4-6: Diagrama de bloques de un controlador PID, $i(t)$ es la señal de estado objetivo, $e(t)$ es el error, $x(t)$ es la señal de control formada por la suma de los términos PID, $y(t)$ es la señal de estado actual del sistema.

Un controlador **PID** se compone de 3 términos de los cuales toma su nombre:

- Proporcional, que produce salida en proporción al error actual $x_p = K_p e(t)$, donde K_p tiene que ser un valor no negativo y se le denomina coeficiente proporcional.
- Integral, que produce salida en proporción a la magnitud de errores pasados y la duración de estos $x_i = K_i \int_0^t e(t) dt$, donde K_i tiene que ser un valor no negativo y se le denomina coeficiente integral.
- Derivativo, que produce salida en proporción a tasa de cambio del error con respecto al tiempo $x_d = K_d \frac{de(t)}{dt}$, donde K_d tiene que ser un valor no negativo y se le denomina coeficiente derivativo.

El simulador Gazebo también provee un algoritmo de control **PID** para el control de posición de las articulaciones, este algoritmo recibe la posición objetivo a la cual se desea mover una articulación y resolviendo $\tau = x(t) = x_p(t) + x_i(t) + x_d(t)$ determina el torque necesario que hay que aplicar para llegar a esa posición. Se tuvieron que determinar los valores apropiados para los coeficientes de cada término, a través de prueba y error los valores se pueden ir ajustando hasta lograr el comportamiento deseado, después de realizar varias pruebas en donde el objetivo fue lograr que el caminador conservara la postura cero y las articulaciones no oscilaran tal que el caminador no se moviera de su posición inicial en el espacio al mismo tiempo que todas las lecturas de la posición angular de los actuadores regresaran un valor cercano a cero con error permitido de $\pm[2, 4]$ grados los valores quedaron en $K_p = 100$, $K_i = 0$ y $K_d = 1$ resultando más bien en un control Proporcional-Derivativo (**PD**) de posición.

4.2.3. Métricas

Los valores de los sensores pueden proveer parte de la información que se puede necesitar para determinar la aptitud del controlador, sin embargo hay información que no nos pueden decir, por ejemplo si el caminador se ha caído, cuanta distancia ha avanzado, que posiciones en el espacio tienen los segmentos. Para obtener esta información adicional se desarrolló un plugin de modelo especial.

El plugin de modelo se creó con el fin de poder ejercer influencia en el modelo y extraer información de su estado. La influencia se da en la forma de fuerzas que se pueden aplicar al segmento del torso con la finalidad de darle soporte vertical y un pequeño empuje en la dirección del eje x (que es la dirección en la que se desea que se desplace el caminador), con un coeficiente modificable se puede determinar la magnitud del efecto de estas fuerzas, desde aplicarse en su totalidad hasta

estar completamente cancelado. La información de estado está relacionada con medidas que pueden obtenerse de posiciones y orientaciones del caminador, esta información se emite en un mensaje en cada paso de simulación transcurrido y a continuación se detalla el contenido de esa información.

Información que solo depende de posición u orientación espacial de segmentos:

- Origen en x del torso, es el componente x de la posición inicial que tiene el origen del torso.
- Origen en y del torso, es el componente y de la posición inicial que tiene el origen del torso.
- Ubicación en x del torso, es el componente x de la posición actual que tiene el origen del torso.
- Ubicación en y del torso, es el componente y de la posición actual que tiene el origen del torso.
- Altura del torso, es el componente z de la posición actual que tiene el origen del torso.
- Guiñada del torso, es el componente de guiñada de la orientación actual que tiene el torso, su unidad de medición está en grados.
- Ubicación aproximada del pie izquierdo, los componentes de posición que tiene el origen del segmento que contiene al sensor de contacto en el pie izquierdo.
- Ubicación aproximada del pie derecho, los componentes de posición que tiene el origen del segmento que contiene al sensor de contacto en el pie derecho.

Información que depende de cálculos vectoriales ente puntos y segmentos de línea definidos por posiciones de segmentos:

- Distancia entre rodillas, la distancia entre los orígenes que tienen las espinillas.
- Distancia entre pies, la distancia entre los orígenes que tienen los segmentos que contienen los sensores de contacto de los pies.
- Distancia entre muslos, la distancia entre las líneas formadas por los segmentos de muslos.
- Distancia entre espinillas, la distancia entre las líneas formadas por los segmentos de espinillas.
- Pie derecho al frente, su valor es verdadero si tomando en cuenta la orientación del torso el pie derecho está al frente del izquierdo en el plano horizontal.

- Pie izquierdo al frente, su valor es verdadero si tomando en cuenta la orientación del torso el pie izquierdo está al frente del derecho en el plano horizontal.
- Distancia del paso, la distancia que separa los pies sobre el vector de dirección horizontal que representa la orientación del torso.
- Otras distancias entre segmentos distintos, muslo-espínilla, pie-espínillas.

Con estas métricas se pueden determinar sucesos muy específicos, como una caída del caminador, distancia recorrida en x , y o lineal, desviación de la orientación del torso con respecto a la dirección deseada, colisiones entre piernas, si se realizó un paso con un determinado pie y de que tamaño fue ese paso. Estos sucesos pueden usarse para tomar decisiones acerca del estado de la evaluación o componer los términos de la función de aptitud.

4.3. Integración de NEAT e interfaz con el simulador Gazebo

Para poder realizar los experimentos deseados es necesaria una herramienta de software que permita la interfaz con el simulador, más específicamente las funciones primordiales que esta herramienta debe desempeñar son; (1) ejecutar el algoritmo **NEAT** y (2) asegurar la correcta evaluación de los individuos controlando e intercambiando información con la simulación. La herramienta se creó no solo con estas 2 funciones en mente sino también con la intención de proveer una interfaz de usuario que permita al experimentador, establecer los parámetros del algoritmo **NEAT**, visualizar información del proceso evolutivo y gráficas de la historia evolutiva de la población (cambios de la aptitud, de la estructura de la red y de las especies a través de las generaciones), funciones para el guardado y carga de, parámetros, población e historia evolutiva que resultan especialmente útiles si un experimento llega a interrumpirse ya que puede reanudarse desde un estado reciente, una interfaz especial para poder interactuar manualmente con el simulador y otra que permita elegir un individuo de la población y visualizar el comportamiento que exhibe en el simulador. La figura 4-7 muestra un diagrama de flujo entre los elementos de la herramienta y del simulador.

Para cumplir con la primera función primordial de la herramienta, se necesita hacer el desarrollo del algoritmo u obtener una implementación que pueda utilizarse. En este caso, el código del algoritmo de **NEAT** que se usa, se obtuvo de la página oficial del algoritmo ¹ y corresponde a la versión 1.2.1 liberada el 20/08/2011, el código tal cual se obtuvo de esta página facilita todos los elementos necesarios para hacer uso del algoritmo pero presenta algunos errores severos por lo que no

¹<http://nn.cs.utexas.edu/?neat-c>

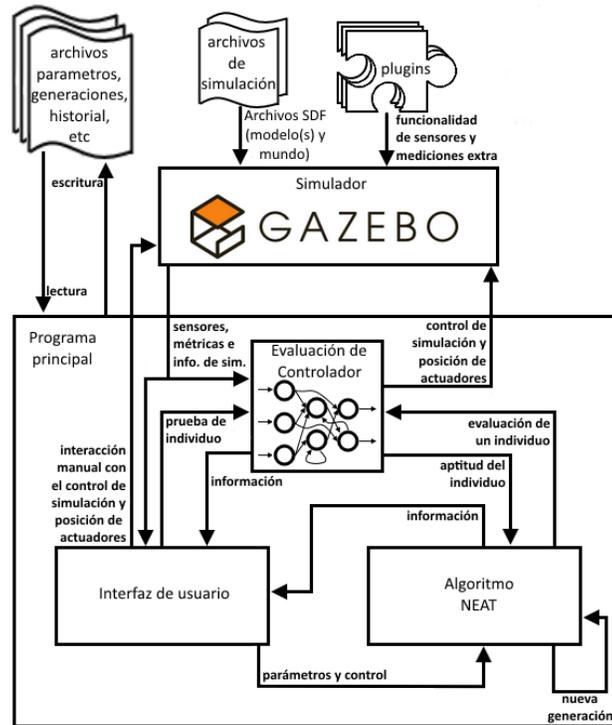


Figura 4-7: Diagrama de flujo del sistema de aprendizaje.

funciona apropiadamente, al momento de integrarlo a la herramienta se revisó exhaustivamente para corregir los errores y se añadió algo de funcionalidad extra para que se pueda usar y parametrizar apropiadamente con la herramienta. Los parámetros y diagrama de clases principales de **NEAT** se muestran en el apéndice **B**.

Para cumplir con la segunda función primordial de la herramienta, la evaluación emplea un ciclo de control temporizado que se coordina con el simulador y debe cumplir con ser suficientemente robusto para asegurar que cada evaluación se realiza correctamente, los siguientes mecanismos se introducen en el algoritmo de evaluación para satisfacer esa necesidad:

- Verificar que el servidor del simulador se este ejecutando y que las conexiones de todos los mensajes que tienen que recibirse y mandarse se establezcan correctamente, en caso de que el proceso del servidor se detenga, paralice o alguna conexión falle, se puede detectar y se envía un comando al sistema con argumentos que indican los archivos de simulación que deben cargarse al reiniciar el servidor para que se tenga el mundo y los modelos virtuales propios del experimento que se está realizando y una vez que se comprueba que el proceso del servidor se ha iniciado exitosamente se restablecen las conexiones.

- El simulador por cada paso de simulación que se efectúa emite un mensaje que contiene el tiempo de simulación que ha transcurrido, este tiempo de simulación se utiliza ampliamente en la evaluación para:
 - Retrasar el inicio del ciclo de control, esto es útil para dejar que transcurra el tiempo equivalente a algunas iteraciones de control para que se alcance una condición inicial de la cual debe partir la evaluación (por ejemplo que el caminador adopte su postura cero) o para que después de un reset de simulación los sensores tengan un tiempo para estabilizarse y empezar a emitir información correcta.
 - Coordinar el ciclo de control, un chequeo especial al final de una iteración espera a que el tiempo de simulación correcto para iniciar la siguiente iteración sea alcanzado antes de continuar el ciclo.
 - Determinar si el simulador se ha paralizado, si el tiempo simulado deja de transcurrir, es un indicativo de que algo está mal con el simulador y debe reiniciarse la evaluación.
 - Determinar si el tiempo de simulación alcanza un umbral (tiempo de simulación permitido) para salir del ciclo de control y proceder al cálculo de la aptitud.
 - Verificar que después de un comando de reset de simulación, la simulación regresa a las condiciones iniciales, si no llegó a las condiciones iniciales la evaluación debe repetirse.
 - Determinar el tiempo promedio de las iteraciones del ciclo de control, si este tiempo promedio excede el valor del tiempo determinado mas un umbral entonces se repite la evaluación.
- Por evaluación se contabilizan los mensajes de comando enviados al simulador y los mensajes de comando reconocidos por la herramienta para identificar pérdidas de mensajes, si al concluirse una evaluación menos del 90% de los comandos enviados llegó a su destino entonces se repite la evaluación.

En la herramienta creada, los mecanismos de coordinación empleados en la evaluación permiten que el simulador corra a velocidades superiores al tiempo real y que el ciclo de control se adapte a esa velocidad de manera robusta. Incluso si el factor de tiempo real fluctúa levemente, el ciclo de control se ejecutará con una frecuencia cercana a la deseada. Comúnmente, cuando se utilizan simuladores, los pasos de simulación que deben efectuarse se comandan en algún punto del mismo ciclo de control, en resumen, esto significa que el simulador espera la indicación para realizar uno

4.3. INTEGRACIÓN DE NEAT E INTERFAZ CON EL SIMULADOR GAZEBO

o más pasos de simulación, en un sistema real el tiempo y estado cambian independientemente sin esperar a que una iteración del ciclo de control termine, en este trabajo el paso del tiempo en el simulador permanece independiente del ciclo de control por lo que se puede considerar que es más cercano a la realidad.

Las figuras 4-8, 4-9 y 4-10, muestran ejemplos de la interfaz de usuario de la herramienta desarrollada.

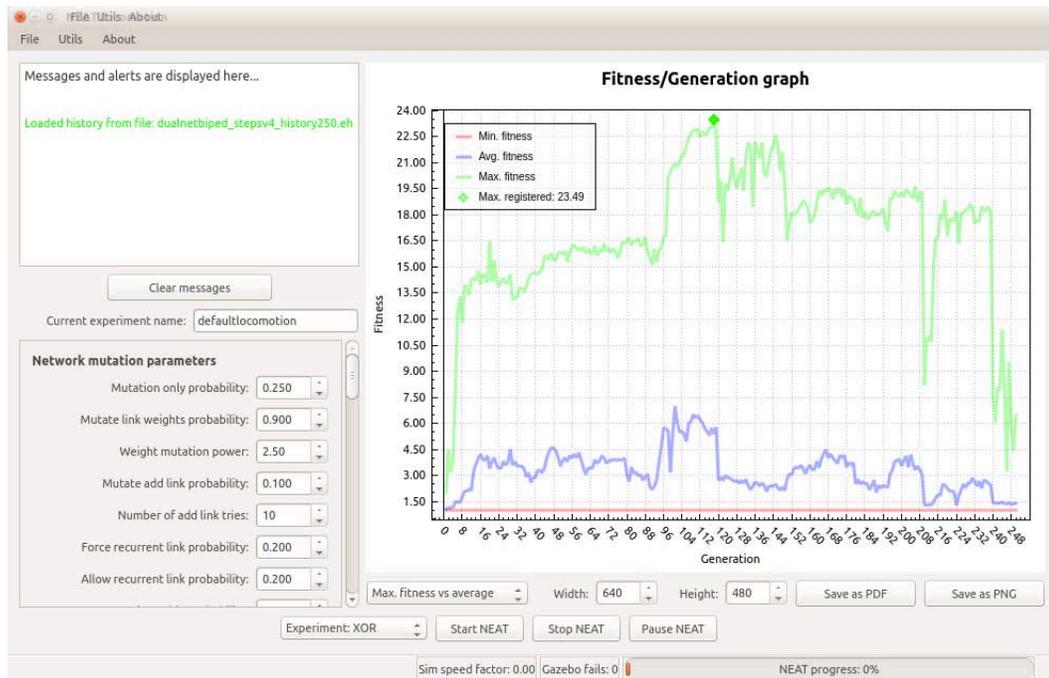


Figura 4-8: Ventana principal para el control de NEAT.

4.3. INTEGRACIÓN DE NEAT E INTERFAZ CON EL SIMULADOR GAZEBO



Figura 4-9: Ventana para la interacción manual con Gazebo.



Figura 4-10: Ventana para la evaluación del comportamiento de individuos.

4.4. Funciones de aptitud y su importancia

Cuando se realiza un sistema de aprendizaje, uno de los objetivos que se persiguen es que este sea lo más automático posible para que, cuando tenga que utilizarse, la intervención de un ser humano en cómo el sistema aprende a realizar una tarea sea mínima.

En el caso de sistemas de aprendizaje a través de la evolución, cómo se dirige el proceso evolutivo a través del tiempo afecta indudablemente el tipo de soluciones que se van a obtener, ya que la función de aptitud es lo que va a indicar si las soluciones se están acercando o no al objetivo u objetivos planteados, se convierte en el principal encaminador del proceso evolutivo y es de extrema importancia que esta función pueda resumir correctamente todos los aspectos que se desean lograr para que el proceso de evolución pueda resultar exitoso.

En [37] se realiza un estudio sobre las funciones de aptitud usadas en la robótica evolutiva explicando que la función de aptitud incorpora conocimiento a priori que especifica el objetivo que se desea lograr y que la cantidad de conocimiento que se incorpora sesga los espacios de búsqueda repercutiendo en la habilidad del sistema de aprendizaje para encontrar soluciones diversas y novedosas. En base a la cantidad de conocimiento a priori que incorporan las funciones de aptitud se pueden clasificar como muestra la figura 4-11, y a continuación se da una breve descripción.

En orden de mayor a menor conocimiento incorporado:

- Funciones de datos de entrenamiento, son las que más conocimiento a priori incorporan, se necesita de un conjunto de datos suficiente con los cuales entrenar la red y de un método que minimiza el error.
- Funciones de conducta, incorporan un conocimiento a priori alto, son funciones formuladas a mano que miden varios aspectos relacionados con que tarea se está realizando y cómo se está realizando, una función de comportamiento generalmente se compone de términos que se combinan a través de una suma o multiplicación ponderada y los hay de 2 tipos, los denominados *términos de comportamiento* que reflejan la forma en que se está realizando la tarea y miden aspectos simples como mapeos de sensores-actuadores, características o eventos que se presume deben manifestarse en la solución, y los denominados *términos agregados* que reflejan el grado de cumplimiento de la tarea sin importar cómo.
- Funciones de funcionalidad incremental, la función cambia a lo largo del proceso evolutivo, iniciando normalmente para que se cumpla un objetivo simple sobre el cual las

4.4. FUNCIONES DE APTITUD Y SU IMPORTANCIA

modificaciones a la función de aptitud puedan ir incorporando objetivos más complejos hasta lograr llegar al objetivo deseado.

- Funciones a medida, incorporan un conocimiento a priori moderado, en estas funciones se consideran términos de comportamiento y agregados, pero los términos agregados están completamente separados de cualquier comportamiento.
- Funciones de ambiente incremental, en lugar de aumentar la dificultad de la función, se aumenta gradualmente la dificultad del ambiente.
- Funciones competitivas, la competencia se realiza directamente entre soluciones en el ambiente por lo que la influencia de una solución sobre otra afecta directamente su aptitud.
- Funciones agregadas, incorporan muy poco conocimiento a priori, se componen únicamente de términos agregados así que solo miden el grado de cumplimiento de la tarea sin importar ningún aspecto de cómo esta se llegue a cumplir.

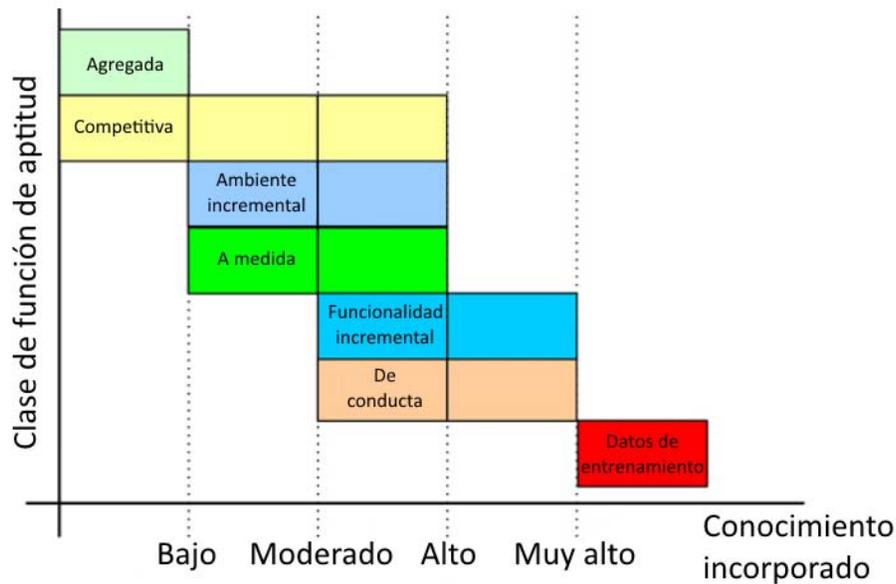


Figura 4-11: Clases de funciones de aptitud / Grado de conocimiento incorporado, imagen adaptada de [37].

Las métricas y sensores explicados con anterioridad en este capítulo van a emplearse en términos de comportamiento y agregados para componer funciones de aptitud para los diferentes experimentos que se detallan en el siguiente capítulo.

4.5. Pruebas de concepto

El enfoque central de este trabajo está en el aprendizaje de movimientos para un robot bípedo virtual, sin embargo dada que esta tarea es sumamente compleja se realizan dos experimentos considerablemente más sencillos para probar la capacidad del sistema solucionando problemas de control no lineales, estas 2 pruebas de concepto son el péndulo invertido simple y péndulo invertido doble.

4.5.1. Especificaciones del modelo virtual del péndulo invertido simple

El modelo virtual se compone solamente de figuras simples y consiste en un riel, fijo al mundo, sobre el cual se encuentra un deslizador conectado al riel por medio de una articulación prismática que le permite moverse a lo largo del riel, una vara se conecta al deslizador por medio de una articulación rotacional que permite que gire en el eje horizontal perpendicular al riel, al otro extremo de la vara se conecta de forma fija una masa. La figura 4-12 ilustra este modelo, mientras que las tablas 4-4 y 4-5 listan las propiedades de sus segmentos y articulaciones.

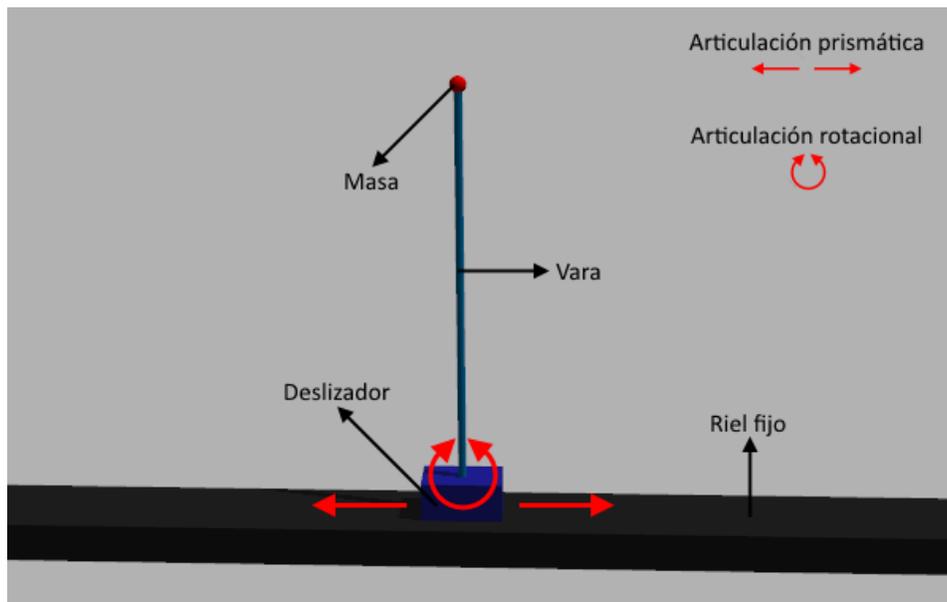


Figura 4-12: Modelo virtual del péndulo simple.

Segmento	Dimensiones caja envolvente (largo, ancho, alto)	Masa	Tensor de Inercia		Contacto
Riel	10 0.2 0.1	5	$I_{xx} = 2.0833E-02$ $I_{yy} = 41.6708$ $I_{zz} = 41.6708$	$I_{xy} = 0$ $I_{xz} = 0$ $I_{yz} = 0$	-
Deslizador	0.2 0.1 0.1	0.1	$I_{xx} = 1.67E-04$ $I_{yy} = 4.17E-04$ $I_{zz} = 4.17E-04$	$I_{xy} = 0$ $I_{xz} = 0$ $I_{yz} = 0$	-
Vara	0.01 0.01 1	0.1	$I_{xx} = 8.336E-03$ $I_{yy} = 8.336E-03$ $I_{zz} = 5E-06$	$I_{xy} = 0$ $I_{xz} = 0$ $I_{yz} = 0$	-
Masa	0.02 0.02 0.02	0.1	$I_{xx} = 4.8E-05$ $I_{yy} = 4.8E-05$ $I_{zz} = 4.8E-05$	$I_{xy} = 0$ $I_{xz} = 0$ $I_{yz} = 0$	-

Tabla 4-4: Esta tabla indica los atributos de los segmentos del péndulo invertido simple, todas las unidades de propiedades físicas son las estándares del sistema internacional.

Articulación	Esfuerzo límite	Velocidad angular/lineal límite	Límite angular/lineal inferior	Límite angular/lineal superior	Amortiguamiento
Riel al mundo	-	-	0	0	-
Deslizador al riel	8.4	10	-4.9	4.9	-
Vara al deslizador	-	-	-1.570796	1.570796	-
Masa a vara	-	-	0	0	-

Tabla 4-5: Esta tabla indica los atributos de las articulaciones del péndulo invertido simple, todas las unidades de propiedades físicas son las estándares del sistema internacional.

4.5.2. Especificaciones del modelo virtual del péndulo invertido doble

El modelo virtual es prácticamente idéntico al del péndulo simple con la excepción de que tiene un péndulo adicional formado por una segunda vara y masa conectada a la primera masa por una articulación rotacional. La figura 4-13 ilustra este modelo, mientras que las tablas 4-6 y 4-7 listan las propiedades de sus segmentos y articulaciones.

Los sensores que utilizan ambos péndulos son los de posición angular/lineal y se aplican a cada articulación que no es fija, estos sensores reportan su valor con una frecuencia de $100Hz$ por cada segundo simulado. El control de posición PD para la articulación prismática de los péndulos utiliza $K_p = 10$, $K_i = 0$ y $K_d = 3$.

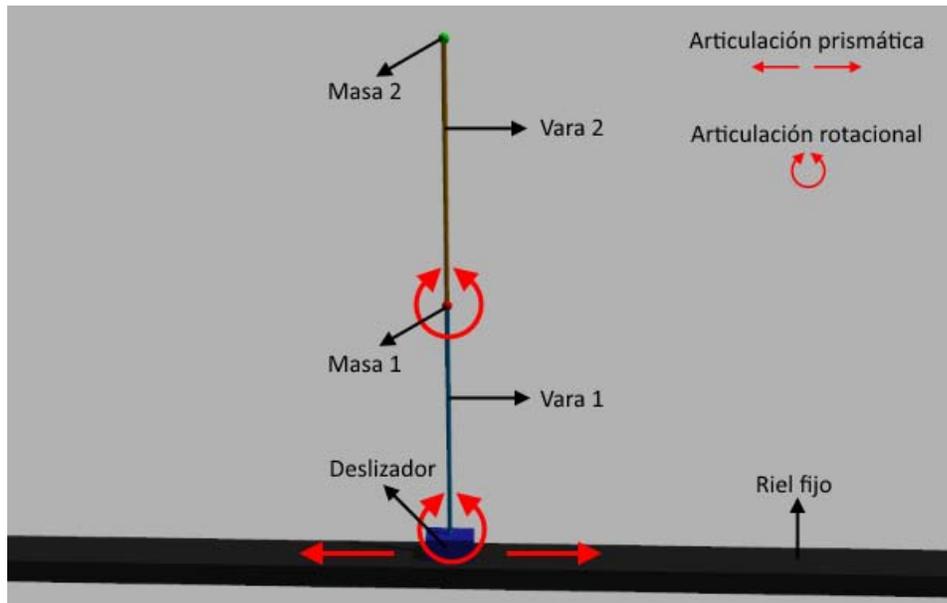


Figura 4-13: Modelo virtual del péndulo doble.

Segmento	Dimensiones caja envolvente (largo, ancho, alto)	Masa	Tensor de Inercia	Contacto
Riel	10 0.2 0.1	5	$I_{xx} = 2.0833E-02$ $I_{xy} = 0$ $I_{yy} = 41.6708$ $I_{xz} = 0$ $I_{zz} = 41.6708$ $I_{yz} = 0$	-
Deslizador	0.2 0.1 0.1	0.1	$I_{xx} = 1.67E-04$ $I_{xy} = 0$ $I_{yy} = 4.17E-04$ $I_{xz} = 0$ $I_{zz} = 4.17E-04$ $I_{yz} = 0$	-
Vara 1	0.01 0.01 1	0.1	$I_{xx} = 8.336E-03$ $I_{xy} = 0$ $I_{yy} = 8.336E-03$ $I_{xz} = 0$ $I_{zz} = 5E-06$ $I_{yz} = 0$	-
Masa 1	0.02 0.02 0.02	0.1	$I_{xx} = 4.8E-05$ $I_{xy} = 0$ $I_{yy} = 4.8E-05$ $I_{xz} = 0$ $I_{zz} = 4.8E-05$ $I_{yz} = 0$	-
Vara 2	0.01 0.01 1	0.1	$I_{xx} = 8.336E-03$ $I_{xy} = 0$ $I_{yy} = 8.336E-03$ $I_{xz} = 0$ $I_{zz} = 5E-06$ $I_{yz} = 0$	-
Masa 2	0.02 0.02 0.02	0.1	$I_{xx} = 4.8E-05$ $I_{xy} = 0$ $I_{yy} = 4.8E-05$ $I_{xz} = 0$ $I_{zz} = 4.8E-05$ $I_{yz} = 0$	-

Tabla 4-6: Esta tabla indica los atributos de los segmentos del péndulo invertido doble, todas las unidades de propiedades físicas son las estándares del sistema internacional.

Articulación	Esfuerzo límite	Velocidad angular/lineal límite	Límite angular/lineal inferior	Límite angular/lineal superior	Amortiguamiento
Riel al mundo	-	-	0	0	-
Deslizador al riel	8.4	10	-4.9	4.9	-
Vara 1 al deslizador	-	-	-1.570796	1.570796	-
Masa 1 a vara 1	-	-	0	0	-
Vara 2 a masa 1	-	-	-3.14159	3.14159	-
Masa 2 a vara 2	-	-	0	0	-

Tabla 4-7: Esta tabla indica los atributos de las articulaciones del péndulo invertido doble, todas las unidades de propiedades físicas son las estándares del sistema internacional.

Experimentos y resultados

Antes de detallar las características específicas y resultados que se obtuvieron en cada experimento, hay aspectos que se mantienen igual para todo experimento, (1) la frecuencia del ciclo de control es de $62.5Hz$ lo cual resulta en un tiempo de $16ms$ simulados entre cada iteración del ciclo, (2) la función de activación es la tangente hiperbólica $s_j = \frac{e^{pa_j} - 1}{e^{pa_j} + 1}$ con $p = 4.924273$, se eligió esta función ya que dentro del contexto del control de posición, la salida de esta función es simétrica en valores positivos y negativos como se aprecia en la figura 2-12 y nos permite especificar la posición en ambos sentidos de un actuador, (3) el tiempo de simulación por el que se puede correr el ciclo de control antes de que se de por terminada la evaluación de un individuo es de $20s$ simulados y lo denominaremos como *tiempo de prueba máximo*.

5.1. Péndulo invertido simple

La tarea que se desea que las redes por evolucionar logren resolver en estos experimentos es controlar la posición del deslizador con el fin de mantener balanceado el péndulo invertido simple, antes de iniciar el ciclo de control se parte de la posición cero a la cual se le aplica una perturbación que puede venir del lado izquierdo o derecho para que el modelo no inicie en un estado balanceado. En cada iteración del ciclo de control se realiza una lectura de sensores para usarlos después en los términos de la función de aptitud.

Los parámetros específicos de NEAT para estos experimentos los muestra la tabla 5-1.

Los parámetros de número de neuronas y tipo de estructura determinan la estructura de los individuos en la población inicial, aunque no sea indicado en estos parámetros por defecto siempre se añade una neurona de bias.

Las 2 neuronas de entrada reciben escalados en al rango de $[-1, 1]$, el valor del sensor de posición

Parámetro	Valor
Probabilidad de solo mutar	25 %
Probabilidad de mutar el peso de una conexión	75 %
Potencia de la mutación de peso	2.5
Probabilidad de añadir una conexión	10 %
Número de intentos para añadir conexión	10
Probabilidad de forzar una conexión recurrente	20 %
Probabilidad de alternar el estado de habilitación de una conexión	0 %
Probabilidad de rehabilitar una conexión	0 %
Probabilidad de añadir nodo	0.3 %
Probabilidad de solo cruzar	20 %
Probabilidad de cruza multipunto	60 %
Probabilidad de cruza multipunto promediada	40 %
Probabilidad de cruza en un solo punto	0 %
Tasa de cruza entre especies	5 %
Umbral de supervivencia	20 %
Significancia de la edad	1
Edad de descarte	1000
Descendencia robada	10
Umbral de compatibilidad	2
Coefficiente de genes disjuntos	1
Coefficiente de genes de exceso	1
Coefficiente de la diferencia de mutaciones	0.8
Tamaño de la población	100
Número de generaciones	1000
Número de neuronas de entrada	2
Número de neuronas de salida	1
Número de neuronas escondidas	0
Tipo de estructura inicial	0
Límite de los pesos	8, [-8, 8]
Compatibilidad normalizada	<i>Falso</i>

Tabla 5-1: Parámetros de NEAT en el experimento de péndulo invertido simple.

lineal, que indica la posición del deslizador sobre el riel, y el valor del sensor de posición angular, que indica la inclinación de la vara con respecto a la vertical. La neurona de salida produce, en el rango de $[-1, 1]$, la posición objetivo a la cual el deslizador debe moverse, el valor se escala a los límites de la articulación antes de enviarse como comando al control **PD** que genera el torque apropiado. La neurona de bias siempre recibe una entrada constante de 1.

5.1.1. Experimento sin perturbaciones

Uno de los experimentos se realiza sin que se apliquen perturbaciones, adicionales a la perturbación inicial, durante la evaluación, en este experimento la función de aptitud es la siguiente:

$$f_i = C \cdot (A + P) \quad (5-1)$$

En donde:

- f_i es la aptitud del individuo i .
- C es el término de caída, sea en este caso t_s el tiempo simulado que un individuo ha sobrevivido cuando ocurre una situación de alto a la evaluación y t_p el tiempo de prueba máximo que se establece en 20, $C = \frac{t_s}{t_p}$.
- A es el término de ángulo, sea en este caso a_k el valor de posición angular de la vara, normalizada al rango $[-1, 1]$, que se recibe en la iteración k y N el número de iteraciones del ciclo de control que se realizaron (ya que la medición de sensores y métricas se hace dentro del ciclo, N equivale al número de mediciones realizadas), $A = 100 \cdot \left(1 - \frac{\sum_{k=1}^N |a_k|}{N}\right)$.
- P es el término de posición lineal, sea en este caso p_k el valor de posición lineal del deslizador, normalizada al rango $[-1, 1]$, que se recibe en la iteración k , $P = 20 \cdot \left(1 - \frac{\sum_{k=1}^N |p_k|}{N}\right)$.

Las situaciones de alto a la evaluación se establecen con el fin de que el proceso sea finito y que se puedan ahorrar recursos si es que la simulación llega a un estado que no es deseable, para los péndulos el alto a la evaluación se da cuando el tiempo por el que se ha simulado un individuo llega al tiempo de prueba máximo o si se registra que el péndulo ha caído, mediante la condición $a_k > 80$ antes de que a_k se haya normalizado.

Las gráficas en las figuras 5-1, 5-2 y 5-3 muestran el cambio de aptitud, nodos, conexiones a través de las generaciones, mientras que la gráfica de la figura 5-4 muestra las especies existentes en la última generación.

La gráfica en la figura 5-5 muestra el comportamiento del péndulo a lo largo del tiempo.

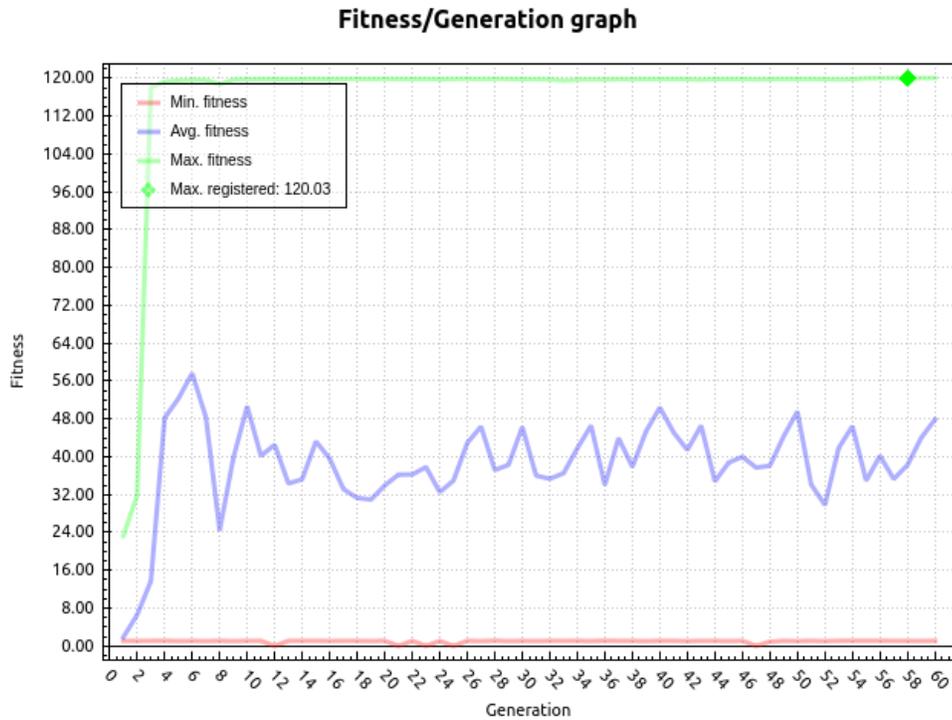


Figura 5-1: Aptitud por generación del péndulo invertido simple sin perturbaciones.

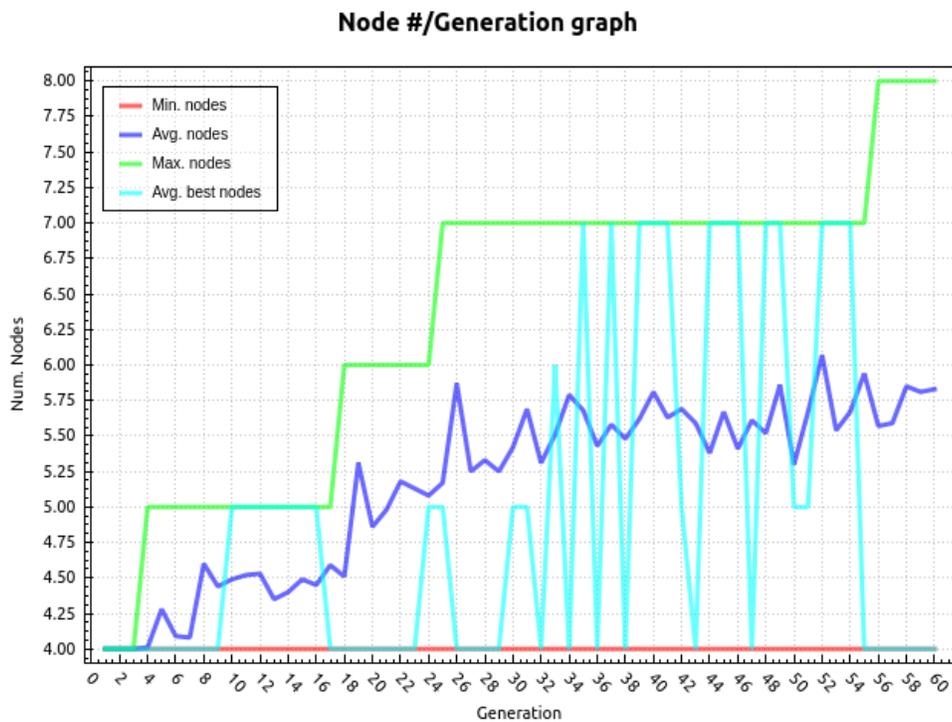


Figura 5-2: Nodos por generación del péndulo invertido simple sin perturbaciones.

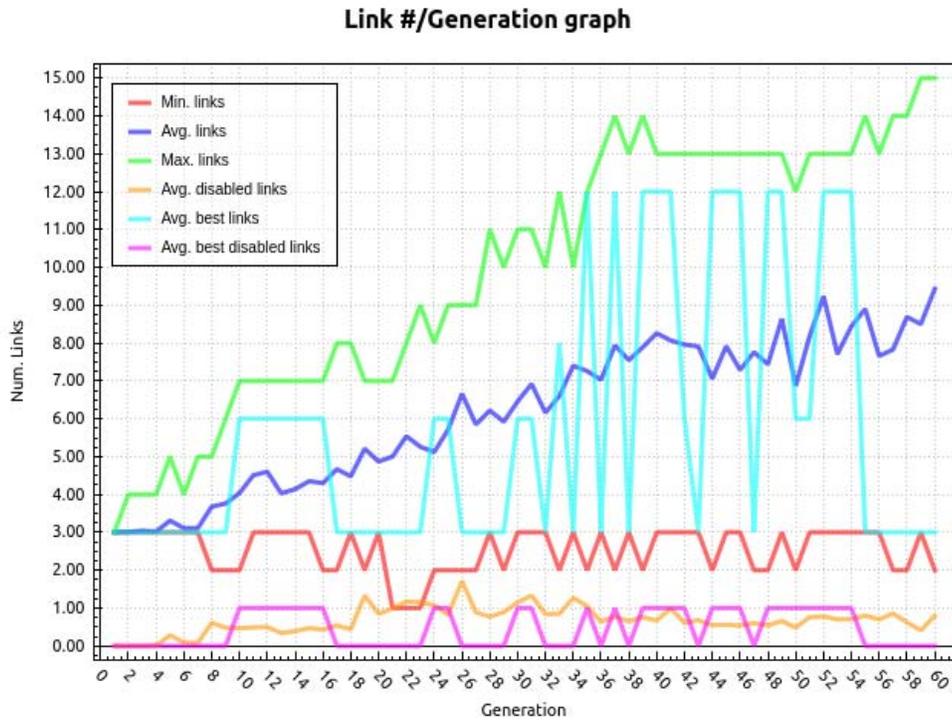


Figura 5-3: Conexiones por generación del péndulo invertido simple sin perturbaciones.

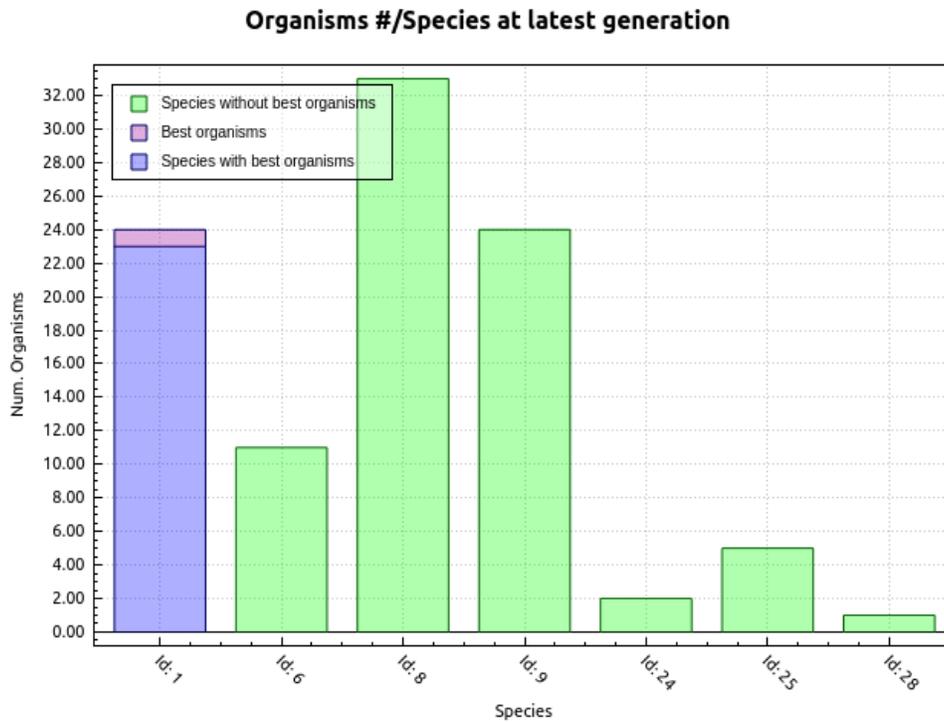


Figura 5-4: Especies de la última generación del péndulo invertido simple sin perturbaciones.

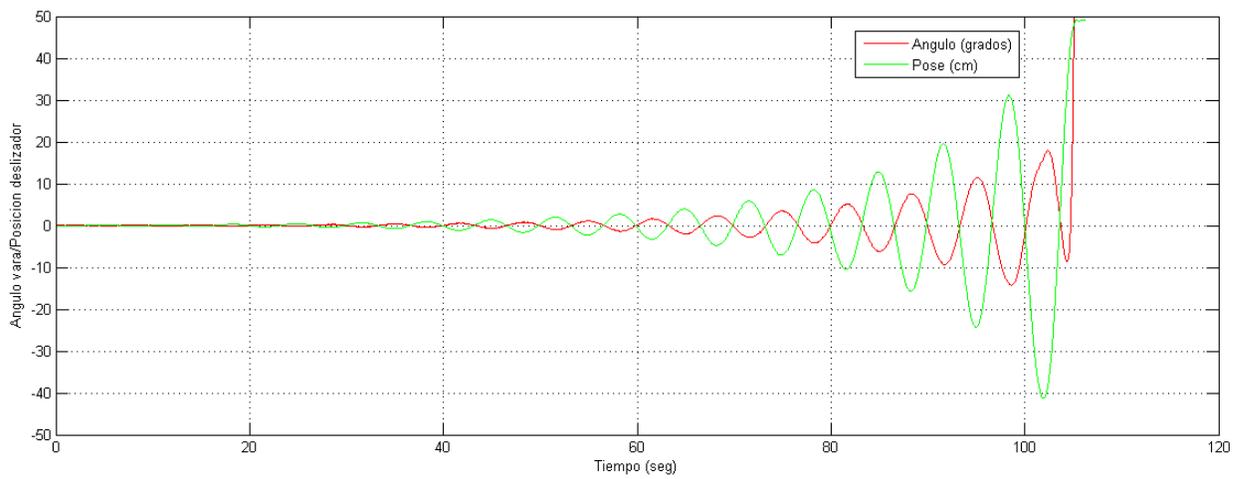


Figura 5-5: Comportamiento del péndulo invertido simple bajo el control del mejor individuo de la última generación (60).

5.1.2. Experimento con perturbaciones

En el otro experimento, existe una muy pequeña probabilidad de que se aplique una perturbación de dirección aleatoria a la vara durante una iteración, en este experimento la función de aptitud no cambia mucho y es la siguiente:

$$f_i = C \cdot (A + P + E) \quad (5-2)$$

E es simplemente el número de perturbaciones que se registraron.

Las gráficas en las figuras 5-6, 5-7 y 5-8 muestran el cambio de aptitud, nodos, conexiones a través de las generaciones, mientras que la gráfica de la figura 5-9 muestra las especies existentes en la última generación.

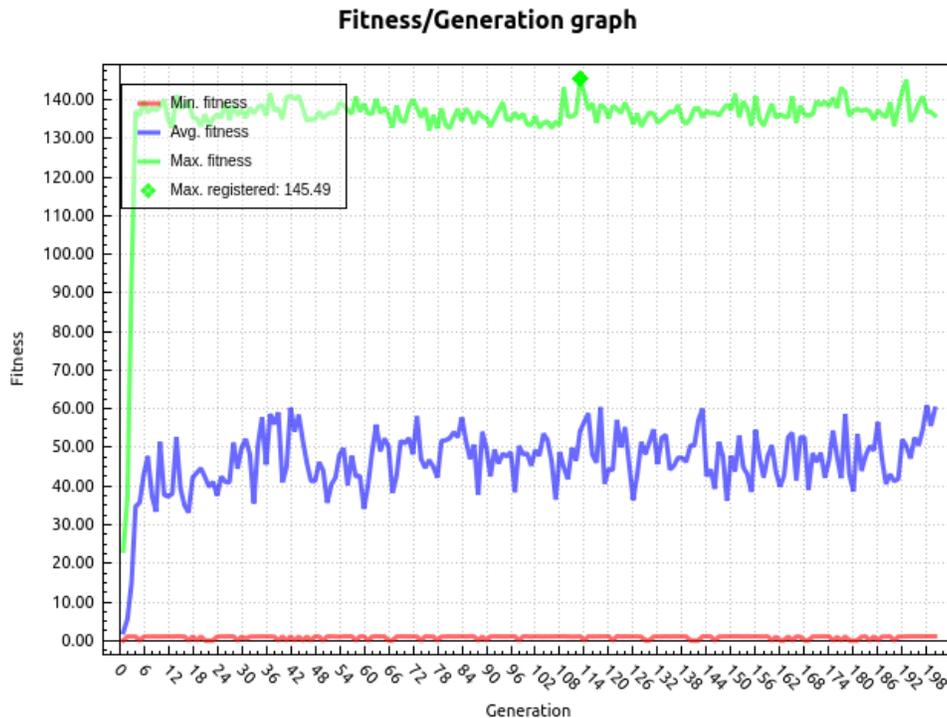


Figura 5-6: Aptitud por generación del péndulo invertido simple con perturbaciones.

La gráfica en la figura 5-10 muestra el comportamiento del péndulo a lo largo del tiempo.

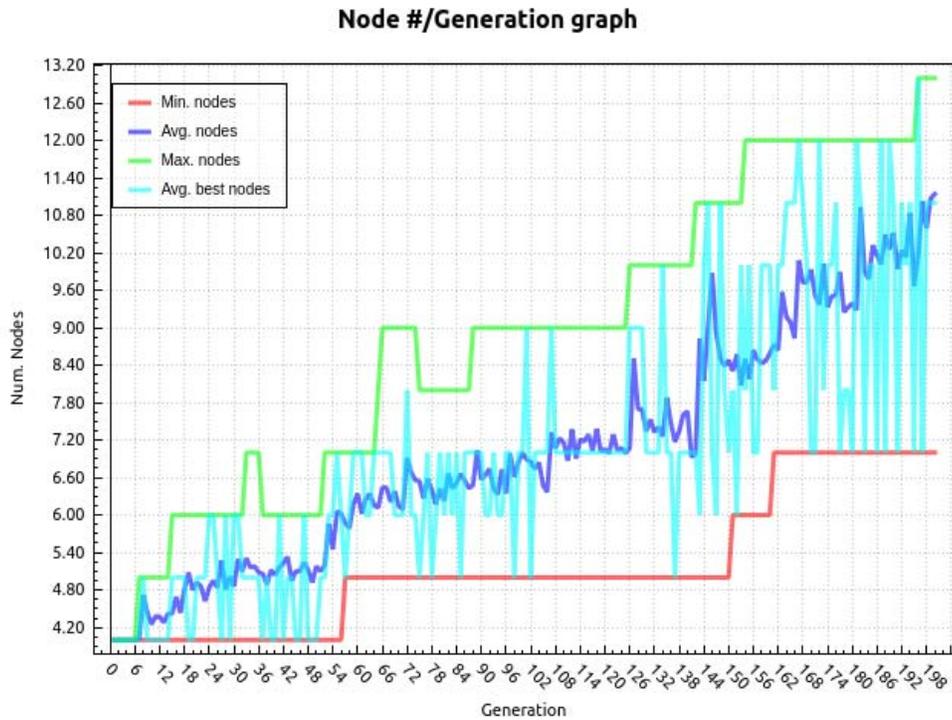


Figura 5-7: Nodos por generación del péndulo invertido simple con perturbaciones.

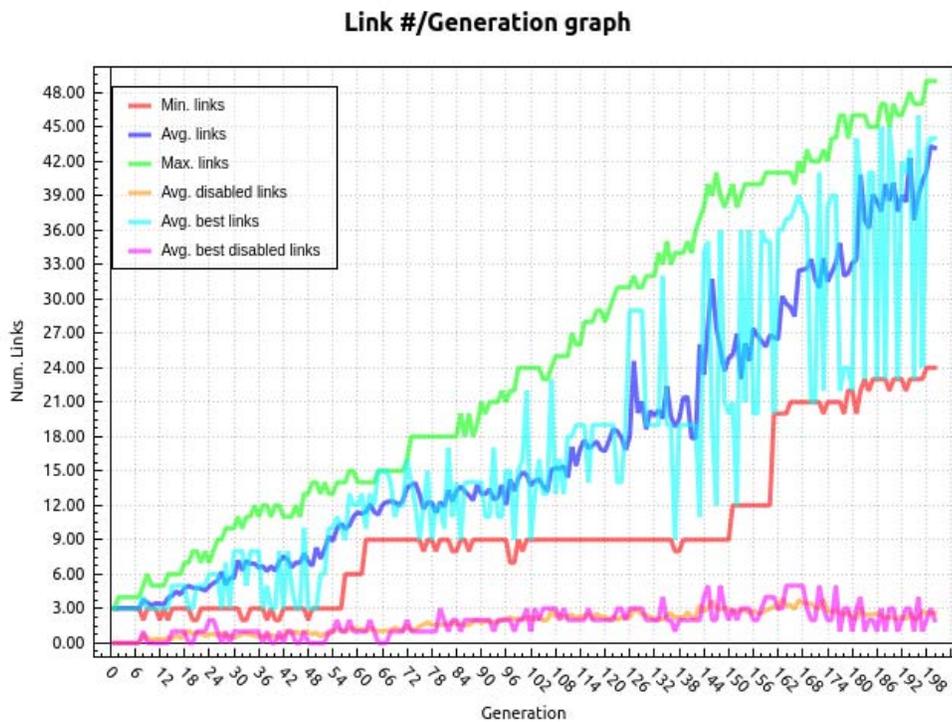


Figura 5-8: Conexiones por generación del péndulo invertido simple con perturbaciones.

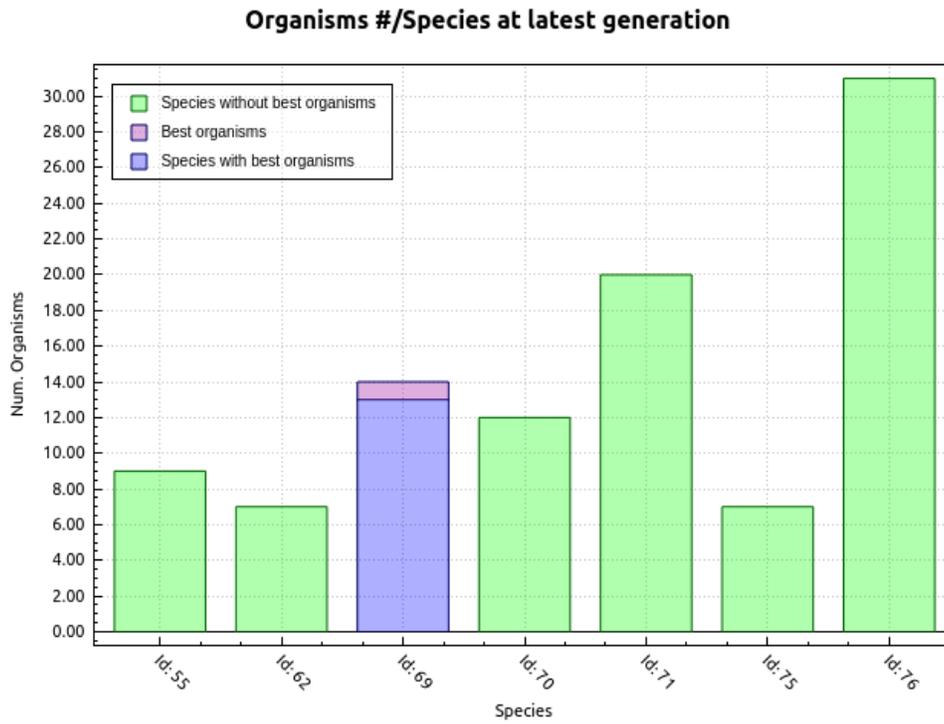


Figura 5-9: Especies de la última generación del péndulo invertido simple con perturbaciones.

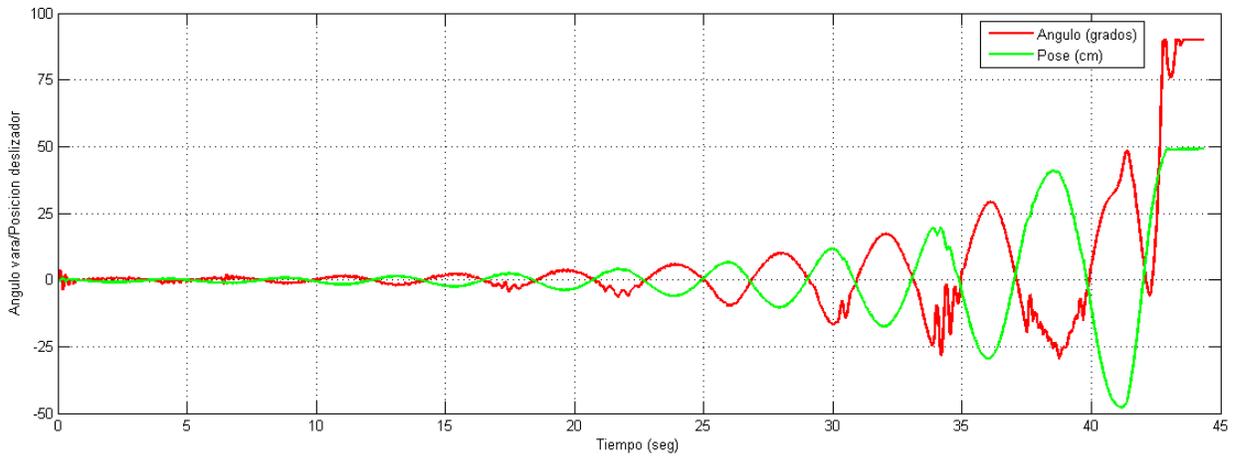


Figura 5-10: Comportamiento del péndulo invertido simple bajo el control del mejor individuo de la última generación (200).

5.2. Péndulo invertido doble

En esencia tiene el mismo objetivo que el péndulo invertido simple, solo que aquí ambos péndulos deben mantenerse balanceados.

Los parámetros específicos de NEAT para estos experimentos los muestra la tabla 5-2.

Parámetro	Valor
Probabilidad de solo mutar	25 %
Probabilidad de mutar el peso de una conexión	75 %
Potencia de la mutación de peso	2.5
Probabilidad de añadir una conexión	10 %
Número de intentos para añadir conexión	10
Probabilidad de forzar una conexión recurrente	20 %
Probabilidad de alternar el estado de habilitación de una conexión	10 %
Probabilidad de rehabilitar una conexión	10 %
Probabilidad de añadir nodo	0.3 %
Probabilidad de solo cruzar	20 %
Probabilidad de cruza multipunto	60 %
Probabilidad de cruza multipunto promediada	40 %
Probabilidad de cruza en un solo punto	0 %
Tasa de cruza entre especies	5 %
Umbral de supervivencia	20 %
Significancia de la edad	1
Edad de descarte	1000
Descendencia robada	10
Umbral de compatibilidad	2
Coficiente de genes disjuntos	1
Coficiente de genes de exceso	1
Coficiente de la diferencia de mutaciones	0.8
Tamaño de la población	100
Número de generaciones	1000
Número de neuronas de entrada	3
Número de neuronas de salida	1
Número de neuronas escondidas	0
Tipo de estructura inicial	0
Límite de los pesos	8, [-8, 8]
Compatibilidad normalizada	<i>Falso</i>

Tabla 5-2: Parámetros de NEAT en el experimento de péndulo invertido simple.

En este caso los individuos de la población inicial se crean con una neurona de entrada más ya que ahora hay 2 sensores de posición angular, uno indica la inclinación de la vara 1 con respecto a la vertical y el otro indica la inclinación de la vara 2 con respecto a la vara 1, obtener la inclinación de la vara 2 con respecto a la vertical es muy sencillo ya que solo hay que sumar ambos sensores, en este experimento se realiza este cálculo anterior para obtener el valor del sensor de posición angular 2 y que reporte el ángulo de la vara 2 con respecto a la vertical.

5.2.1. Experimento sin perturbaciones

La función de aptitud para este experimento es la siguiente:

$$f_i = C \cdot (A_1 + A_2 + P) \quad (5-3)$$

Las diferencias con respecto a la función de aptitud del péndulo invertido simple son:

- $A_1 = 50 \cdot \left(1 - \frac{\sum_{k=1}^N |a_{1k}|}{N}\right)$, a_{1k} se encuentra normalizado al rango $[-1, 1]$.
- $A_2 = 50 \cdot \left(1 - \frac{\sum_{k=1}^N |a_{2k}|}{N}\right)$, a_{2k} se encuentra normalizado al rango $[-1, 1]$.

La condición de alto a la evaluación se modifica para considerar ahora la caída de cualquiera de los péndulos durante la evaluación, si $a_{1k} > 80$ ó $a_{2k} > 80$ entonces la evaluación termina en la presente iteración, en este caso a_1 y a_2 se utilizan en la condición anterior sin haberse normalizado.

Las gráficas en las figuras 5-11, 5-12 y 5-13 muestran el cambio de aptitud, nodos, conexiones a través de las generaciones, mientras que la gráfica de la figura 5-14 muestra las especies existentes en la última generación.

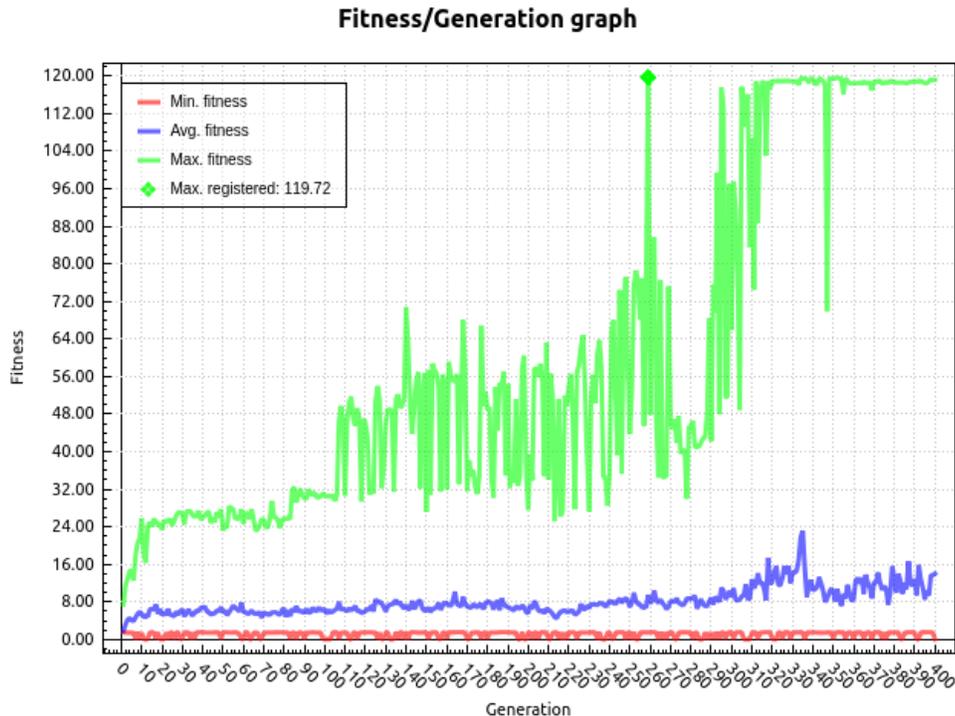


Figura 5-11: Aptitud por generación del péndulo invertido doble sin perturbaciones.

La gráfica en la figura 5-15 muestra el comportamiento del péndulo a lo largo del tiempo.

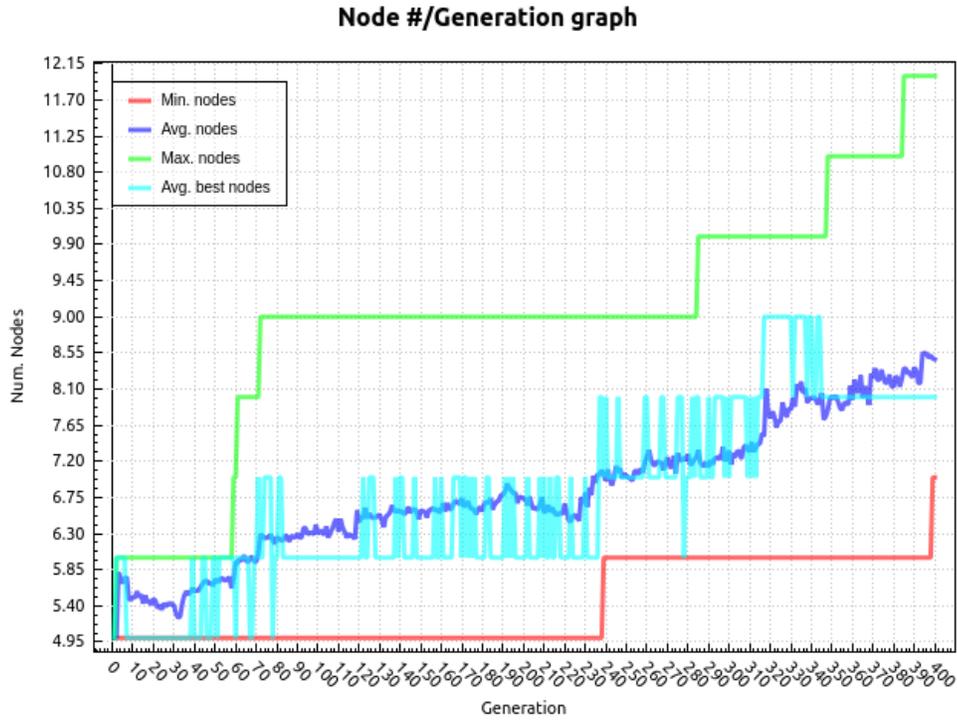


Figura 5-12: Nodos por generación del péndulo invertido doble sin perturbaciones.

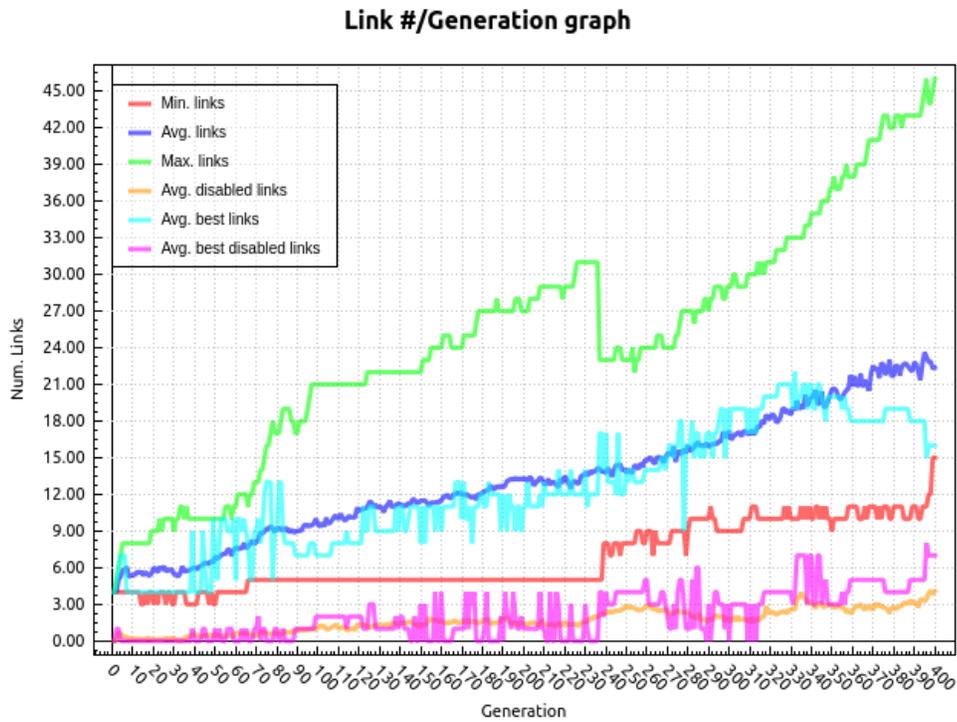


Figura 5-13: Conexiones por generación del péndulo invertido doble sin perturbaciones.

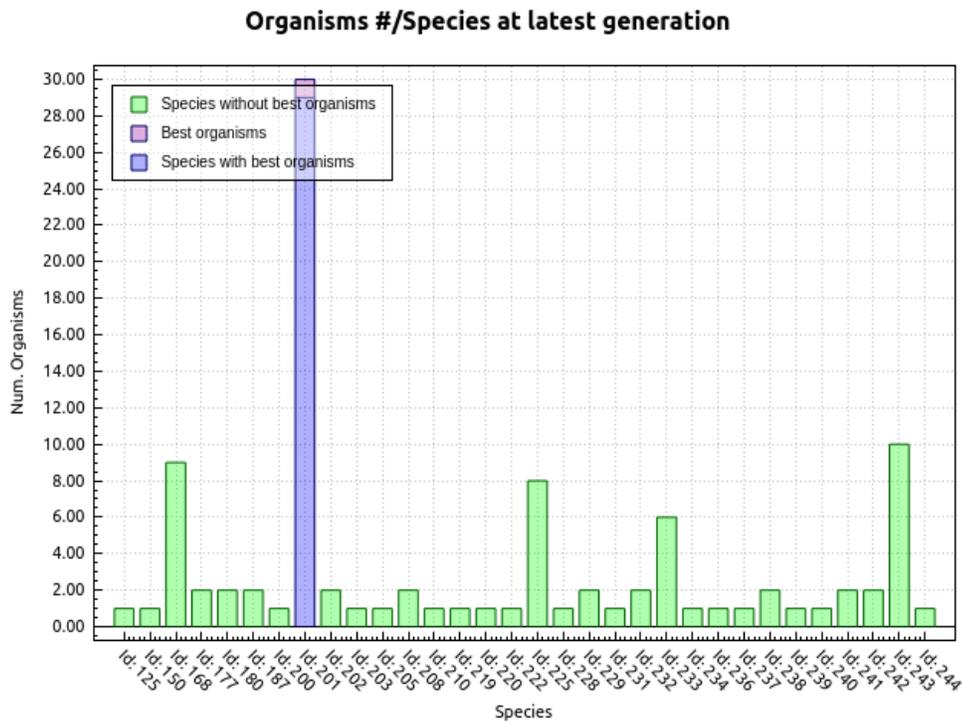


Figura 5-14: Especies de la última generación del péndulo invertido doble sin perturbaciones.

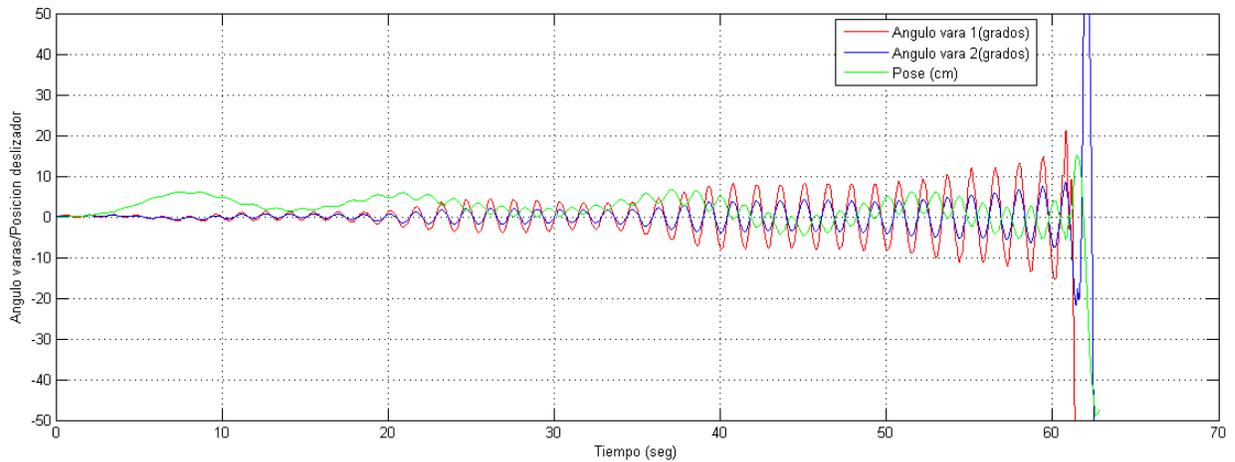


Figura 5-15: Comportamiento del péndulo invertido doble bajo el control del mejor individuo de la última generación (400).

5.2.2. Experimento con perturbaciones

La función de aptitud para este experimento es la siguiente:

$$f_i = C \cdot (A_1 + A_2 + P + E) \quad (5-4)$$

Las gráficas en las figuras 5-16, 5-17 y 5-18 muestran el cambio de aptitud, nodos, conexiones a través de las generaciones, mientras que la gráfica de la figura 5-19 muestra las especies existentes en la última generación.

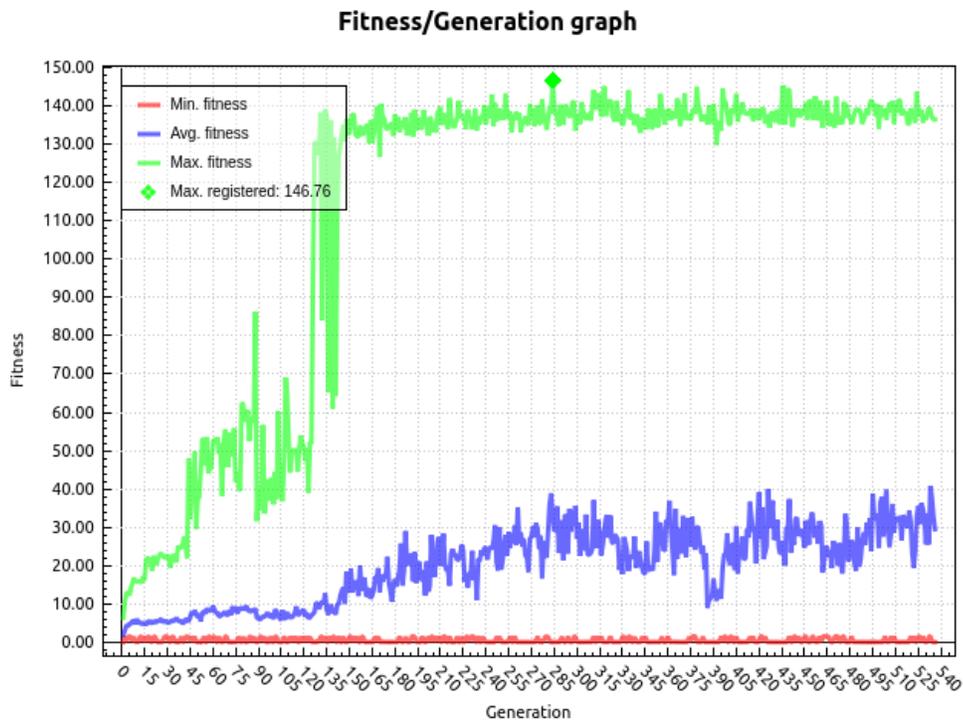


Figura 5-16: Aptitud por generación del péndulo invertido doble con perturbaciones.

La gráfica en la figura 5-20 muestra el comportamiento del péndulo doble a lo largo del tiempo.

La figura 5-21 ilustra la estructura de red del mejor individuo en la última generación.

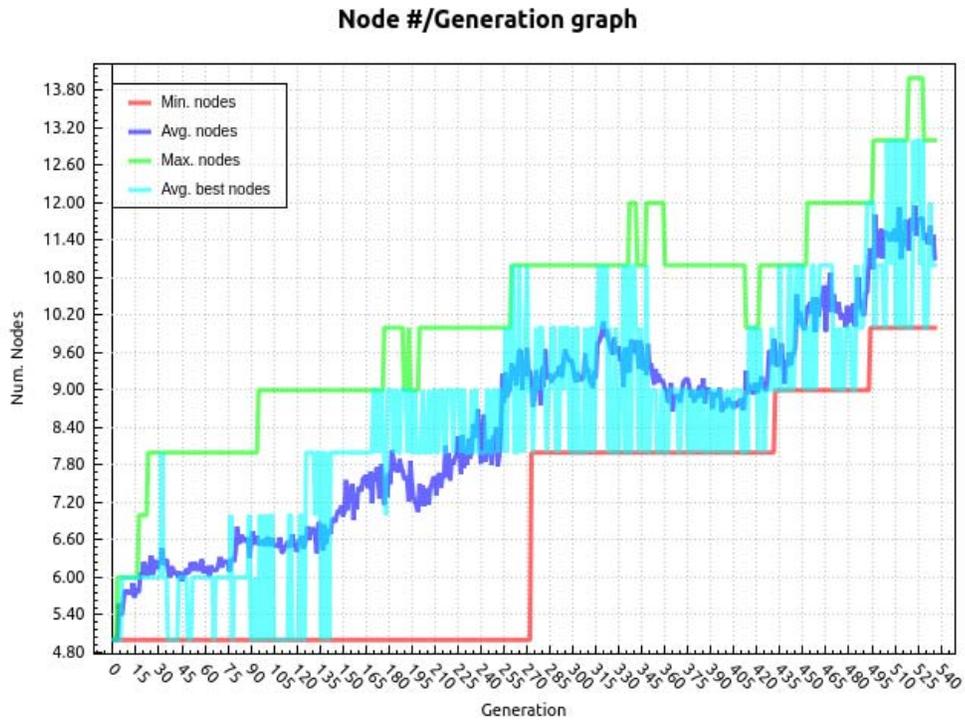


Figura 5-17: Nodos por generación del péndulo invertido doble con perturbaciones.

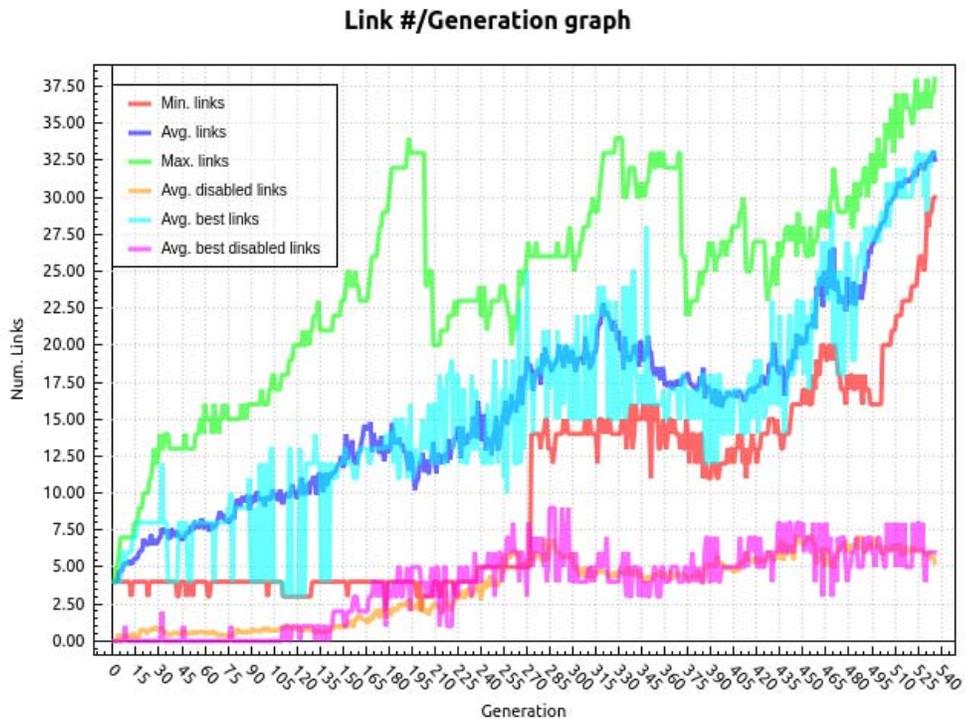


Figura 5-18: Conexiones por generación del péndulo invertido doble con perturbaciones.

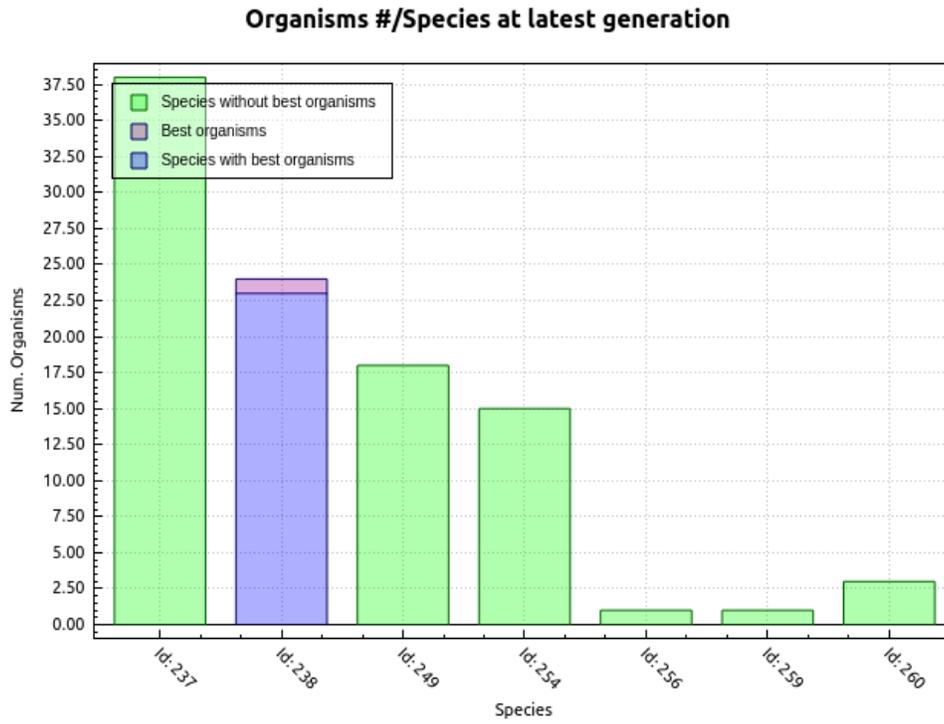


Figura 5-19: Especies de la última generación del péndulo invertido doble con perturbaciones.

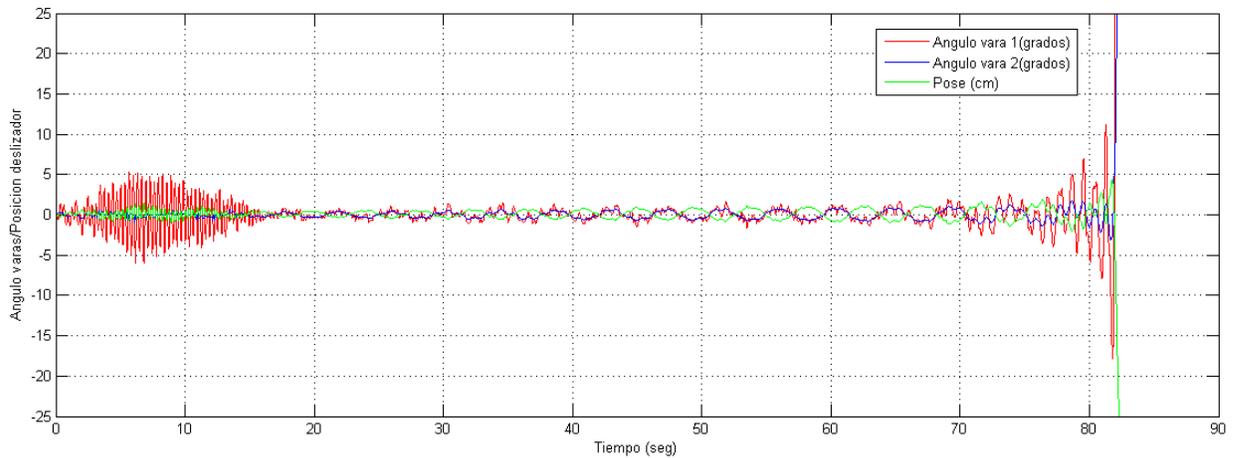


Figura 5-20: Comportamiento del péndulo invertido doble bajo el control del mejor individuo de la última generación (536).

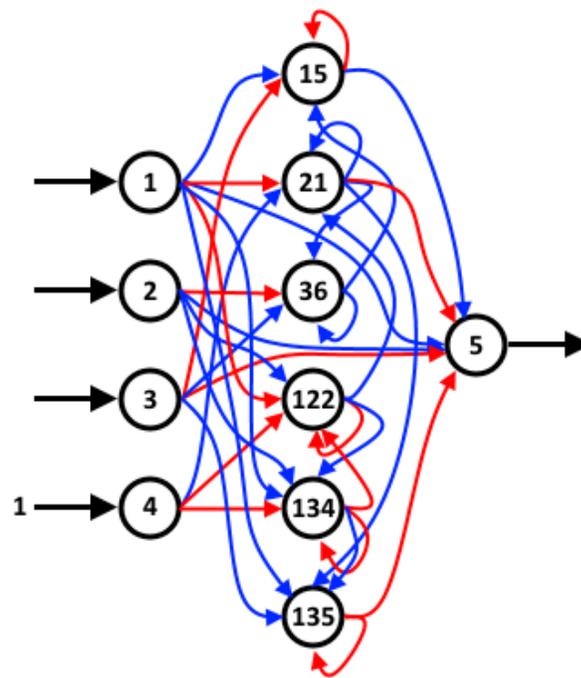


Figura 5-21: Estructura de red del mejor individuo de la última generación (536).

5.3. Robot bípedo

Los primeros experimentos con el caminador utilizaron una estructura de red inicial de 20 entradas, 1 por cada sensor en el modelo, 1 bias y 12 salidas (una por cada actuador, 6 en cada pierna), este espacio inicial junto con el hecho de que el escalamiento de las posiciones angulares de las articulaciones se hace respecto a los límites definidos en el modelo virtual del caminador constituyen un espacio demasiado grande de búsqueda, resultando en frecuentes estancamientos de la red y mínimas ganancias de aptitud que rápidamente se perdían, aun cuando la función de aptitud era tan simple como la distancia máxima en el eje de las x .

Para disminuir considerablemente el tamaño del espacio de búsqueda e implementar mecanismos adicionales para ayudar a que las redes puedan encontrar soluciones al problema, se adoptaron varias ideas que en [1] dieron resultados limitados. Estas ideas son:

Utilizar una red más pequeña y simple que al momento de realizar la evaluación sea duplicada y así tener una red que controla la pierna izquierda y otra la pierna derecha, esto tiene también la intención de que se logren alcanzar comportamientos simétricos más rápidamente. Las red a evolucionar ahora es de 14 entradas, 2 bias y 6 salidas, 6 de las entrada provienen de los sensores de posición angular de las articulaciones en la pierna que se va a controlar, las siguientes 6 provienen de la IMU y las últimas 2 de los sensores de contacto, los cuales entran a la red como el contacto del pie en la pierna propia y como el contacto del pie en la pierna contraria, por lo que se intercambia el orden de estas entradas en la red de la pierna izquierda y derecha. Las 6 salidas corresponden a la posición objetivo para las articulaciones de la pierna que se está controlando y el bias adicional en la red tiene la función de introducir asimetría desde la primera activación de la red con el fin de que el estado de la red izquierda y derecha inmediatamente pueda diferir y generar distinto comportamiento, este segundo bias va a recibir una entrada constante de 1 en la red izquierda y de -1 en la derecha.

Retomando la idea de que el caminado es un movimiento que se realiza principalmente en el plano sagital donde las articulaciones en este plano son las que más influyen, se limita pues el rango de escalamiento de las posiciones angulares en las articulaciones de guiñada y alabeo de manera que estas queden muy restringidas, y que las de cabeceo solo estén moderadamente restringidas. Un plus añadido al limitar los rangos de escalamiento es que se evita parcial o totalmente que la red genere movimientos de extremidades que resultarían en una colisión entre segmentos. Al igual que todos los sensores que entran a la red el escalamiento de las

posiciones articulares a los límites ya sean los establecidos en el modelo en el archivo **SDF** o los límites restringidos que se definan siempre se hace abarcando completamente el rango de $[-1, 1]$.

Se utilizará el plugin de métricas, el cual se programó con la capacidad de aplicar fuerzas sobre el torso del caminador, específicamente aplicar una fuerza horizontal de $4.7N$ con el propósito de proporcionar un impulso en la dirección del eje x sobre la cual se desea que avance el caminador y una vertical de $33.36N$ que contrarresta el 70% del peso del caminador para disminuir el riesgo de caídas, con esto se intenta que individuos que podrían estar haciendo los movimientos adecuados pero que no han aprendido a mantener el equilibrio tengan mejores oportunidades de optimizarse, la proporción del efecto que las fuerzas aplicadas por este plugin tienen sobre el modelo se va disminuyendo conforme transcurren las generaciones, por lo tanto esto tiene un impacto directo en la aptitud ya que al disminuirse la ayuda que se le da al caminador su aptitud puede disminuir.

Con el uso de una red para cada pierna se introducen 2 parámetros de control que pueden usarse para afectar el comportamiento. Por un lado la *inversión de las salidas para articulaciones de guiñada y alabeo en la pierna derecha* puede permitir que el sentido de estos movimientos sea opuesto entre las piernas y puede ayudar a la simetría, si se observan los límites de la tabla 4-2 para estas articulaciones se notará que se intercambian entre piernas. Por el otro lado el *cruce de las entradas de posición angular entre pierna izquierda y derecha* puede permitir que la pose de la pierna derecha afecte el resultado de la red izquierda y viceversa.

El las siguientes secciones se explican los detalles de cada experimento y los resultados de cada uno.

5.3.1. Experimento 1, movimiento por saltos

En este experimento los parámetros del algoritmo **NEAT** los muestra la tabla 5-3 y los parámetros de control/red adicionales los muestra la tabla 5-4.

La función de aptitud para el experimento 1 es:

$$f_i = 1 + (C \cdot D) \cdot (100 \cdot S \cdot P + 10 \cdot M \cdot G) \quad (5-5)$$

En donde:

Parámetro	Valor
Probabilidad de solo mutar	25 %
Probabilidad de mutar el peso de una conexión	50 %
Potencia de la mutación de peso	0.5
Probabilidad de añadir una conexión	10 %
Número de intentos para añadir conexión	20
Probabilidad de forzar una conexión recurrente	20 %
Probabilidad de alternar el estado de habilitación de una conexión	3 %
Probabilidad de rehabilitar una conexión	0 %
Probabilidad de añadir nodo	1 %
Probabilidad de solo cruzar	20 %
Probabilidad de cruce multipunto	60 %
Probabilidad de cruce multipunto promediada	40 %
Probabilidad de cruce en un solo punto	0 %
Tasa de cruce entre especies	5 %
Umbral de supervivencia	50 %
Significancia de la edad	1
Edad de descarte	100
Descendencia robada	50
Umbral de compatibilidad	2
Coefficiente de genes disjuntos	10
Coefficiente de genes de exceso	10
Coefficiente de la diferencia de mutaciones	1.5
Tamaño de la población	500
Número de generaciones	1000
Número de neuronas de entrada	14
Número de neuronas de salida	6
Número de neuronas escondidas	0
Tipo de estructura inicial	0
Límite de los pesos	2, [-2, 2]
Compatibilidad normalizada	Verdadero

Tabla 5-3: Parámetros de NEAT en el experimento 1 del caminador.

- D , es el término de desviación del torso, sea en este caso d_k la desviación normalizada del torso que se recibe en la iteración k , $D = 1 - \frac{\sum_{k=1}^N |d_k|}{N}$.
- S , es el término de soporte de los pies, sea s_{1k} el valor del contacto del pie izquierdo y s_{2k} el del derecho en la iteración k , $S = 0$ inicialmente, luego al momento de calcular la aptitud

$$S = \begin{cases} 0.5 & \text{si } 0.5 \leq \frac{\sum_{k=1}^N s_{1k}}{N} < 0.75 \\ 0.2 \frac{\sum_{k=1}^N s_{1k}}{N} & \text{caso contrario} \end{cases} + \begin{cases} 0.5 & \text{si } 0.5 \leq \frac{\sum_{k=1}^N s_{2k}}{N} < 0.75 \\ 0.2 \frac{\sum_{k=1}^N s_{2k}}{N} & \text{caso contrario} \end{cases}$$

- P , es el término de distancia en el eje x , sea p_{ax} el componente x de la posición actual del torso medida en la iteración más reciente y p_{ox} el componente x de la posición inicial del torso, $P = p_{ax} - p_{ox}$.

Parámetro	Valor
Red simétrica	Verdadero
Cruce de entradas	Falso
Inversión de guiñada y alabeo en la pierna derecha	Verdadero
Límites de escalamiento Guiñada cadera izquierda	[-5,5]
Límites de escalamiento Guiñada cadera derecha	[-5,5]
Límites de escalamiento Alabeo cadera izquierda	[-5,5]
Límites de escalamiento Alabeo cadera derecha	[-5,5]
Límites de escalamiento Cabeceo cadera izquierda	[-30,30]
Límites de escalamiento Cabeceo cadera derecha	[-30,30]
Límites de escalamiento Cabeceo rodilla izquierda	[-30,30]
Límites de escalamiento Cabeceo rodilla derecha	[-30,30]
Límites de escalamiento Alabeo tobillo izquierdo	[-5,5]
Límites de escalamiento Alabeo tobillo derecho	[-5,5]
Límites de escalamiento Cabeceo tobillo izquierdo	[-15,15]
Límites de escalamiento Cabeceo tobillo derecho	[-15,15]
Fuerzas de soporte	De la generación 1 a la 300 disminución del efecto 10 % cada 30 generaciones

Tabla 5-4: Parámetros extra del experimento 1 del caminador.

- M , es el término de paso mas largo que se haya logrado, sea m_{max} esta distancia y m_a la distancia de paso medida en la iteración más reciente, inicialmente $m_{max} = 0$ y luego al momento de calcular la aptitud $M = m_{max}$ y en cada iteración

$$m_{max} = \begin{cases} m_a & \text{si } m_a > m_{max} \\ m_{max} & \text{caso contrario} \end{cases}$$

- G , es el término de número de pasos realizados, sea g_d y g_i banderas que indican si el siguiente paso debe ser derecho o izquierdo para ser válido, n_p el contador de pasos y g_l un indicador de que pie está al frente (si vale 1 es el derecho, si vale 0 es el izquierdo). Entonces inicialmente $g_d = g_i = 1$ y $n_p = 0$, luego al momento de calcular la aptitud $G = n_p$

$$\text{y en cada iteración } n_p = \begin{cases} n_p + 1, g_d = 0, g_i = 1 & \text{si } g_d = 1 \wedge g_i = 1 \wedge (s_{1k} = 1 \vee s_{2k} = 1) \wedge m_a > 0.02 \\ n_p + 1, g_d = 1, g_i = 0 & \text{si } g_i = 1 \wedge g_l = 0 \wedge (s_{1k} = 1 \vee s_{2k} = 1) \wedge m_a > 0.02 \\ n_p & \text{caso contrario} \end{cases}$$

Estos complejos términos se diseñaron con el propósito de recompensar los esfuerzos del controlador por intentar realizar muchos pasos, con contacto plano del pie en el suelo.

Las gráficas en las figuras 5-22, 5-23 y 5-24 muestran el cambio de aptitud, nodos y conexiones a través de las generaciones, no se muestran especies ya que en este caso el proceso de evolución no llegó a generar diferencias suficientes para que se crearan nuevas especies.

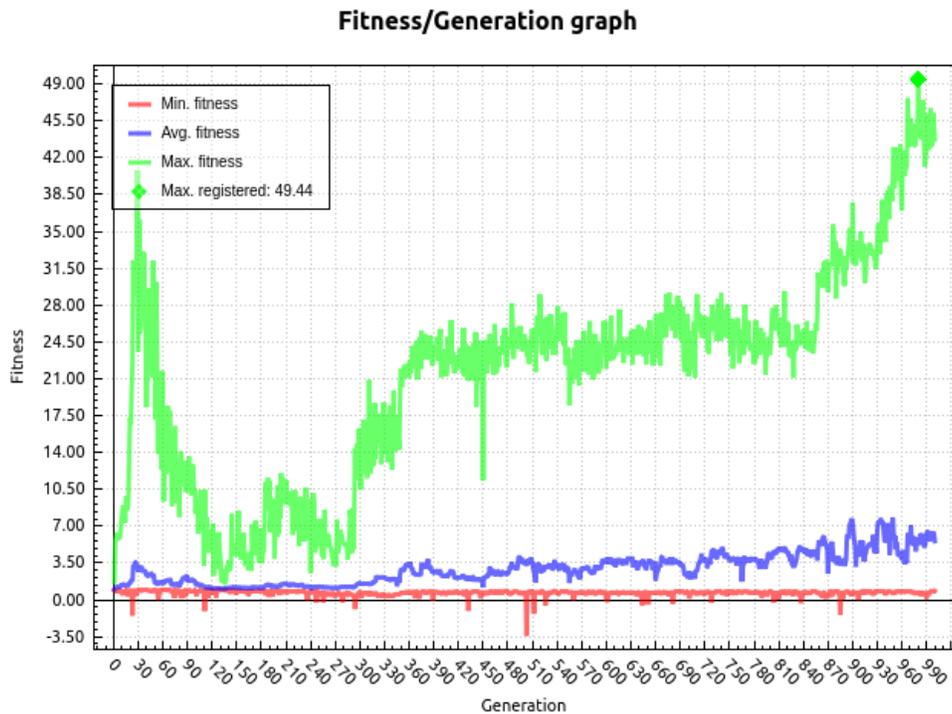


Figura 5-22: Aptitud por generación del experimento 1.

La siguiente tabla 5-5 compila el resultado de probar al mejor individuo de la generación 980 unas 20 veces para verificar que tan consistente es su comportamiento, para juntar estas 20 muestras se tuvo que evaluar 31 veces ya que en 11 ocasiones el modelo cayó antes de transcurridos 5 segundos de simulación.

Las siguientes gráficas en las figuras 5-25, 5-26 y 5-27 muestran la ruta promedio y respuesta promedio del sensor de IMU del caminador en las 5 pruebas en donde el modelo no cayó antes de llegar a los 60 segundos simulados. Las gráficas en las figuras 5-28, 5-29 y 5-30 muestran esa misma información pero para la mejor prueba que alcanzó la mayor distancia en x . Una secuencia visual

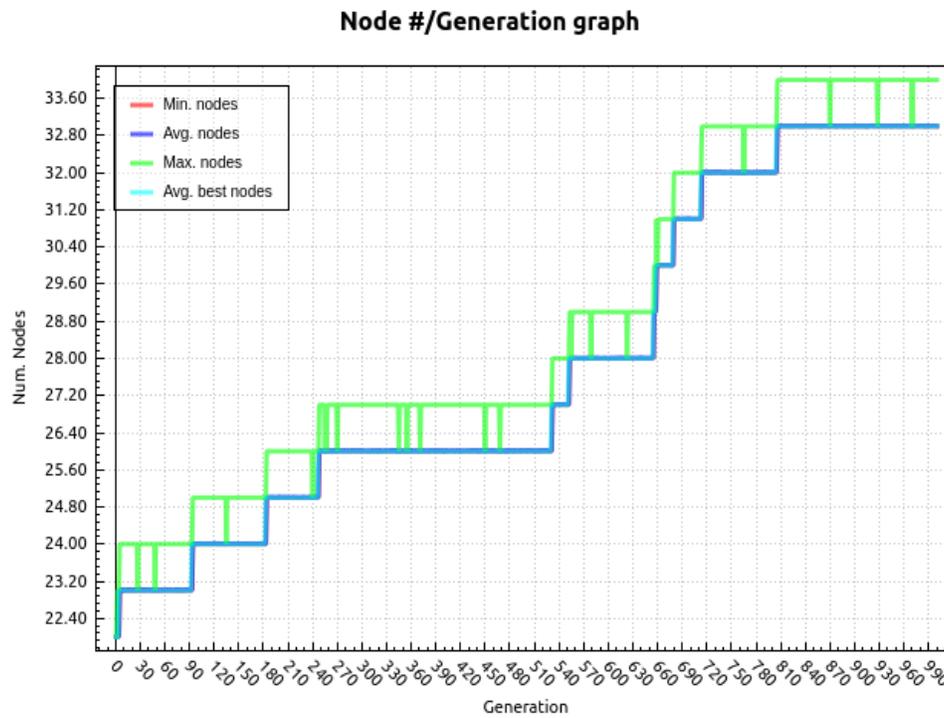


Figura 5-23: Nodos por generación del experimento 1.

del comportamiento se aprecia en la figura 5-31.

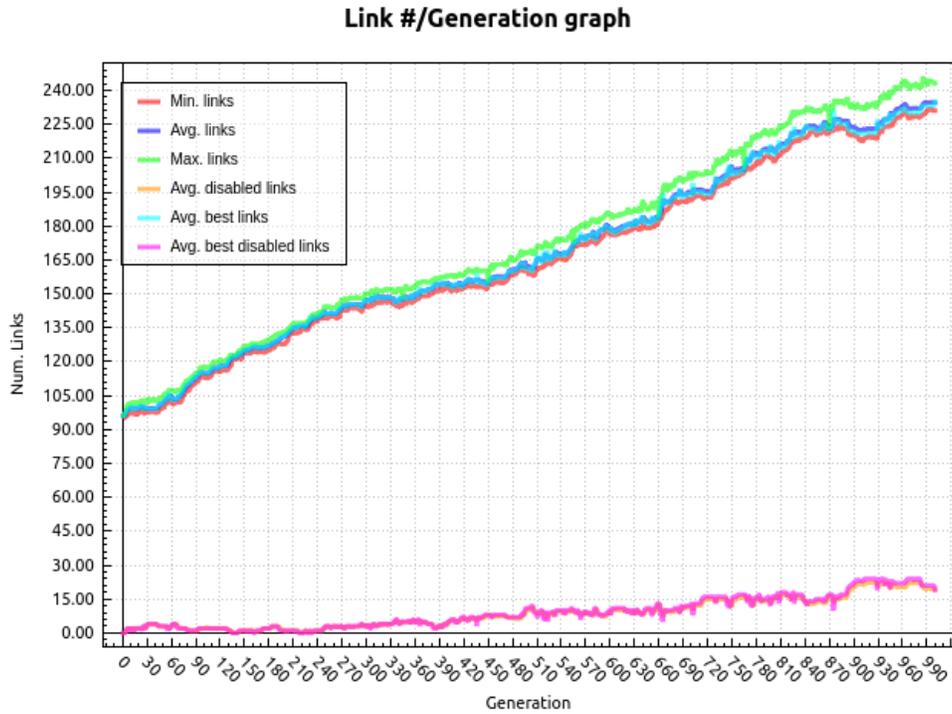


Figura 5-24: Conexiones por generación del experimento 1.

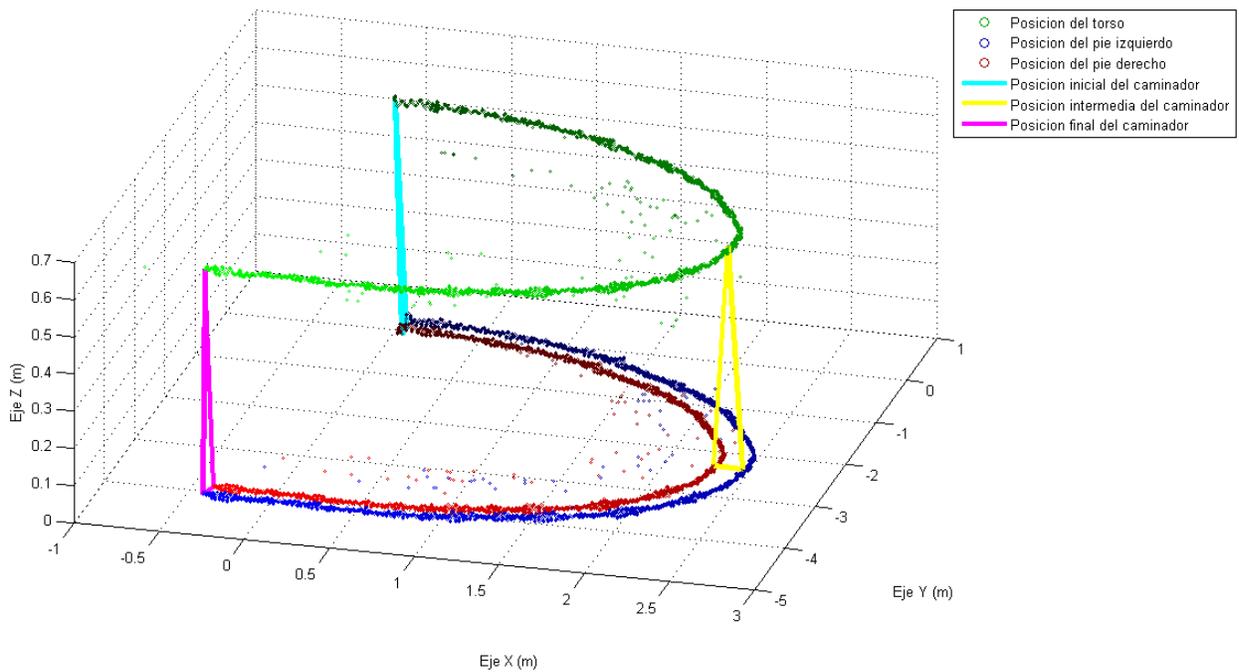


Figura 5-25: Ruta promedio seguida por el caminador del experimento 1.

Prueba	Distancia máxima recorrida en X	Tiempo transcurrido antes de caer ó terminado a los ~60 segundos	Tiempo promedio de las iteraciones del ciclo de control
1	2.5522m	50.139s	0.016017s
2	3.9917m	42.402s	0.016018s
3	2.2775m	25.192s	0.016025s
4	2.1220m	19.881s	0.016033s
5	2.5458m	59.987s	0.016009s
6	1.8860m	25.722s	0.016026s
7	1.5082m	59.975s	0.016006s
8	4.0687m	57.411s	0.016005s
9	1.6771m	60.000s	0.016012s
10	3.3644m	38.783s	0.016012s
11	2.9349m	59.997s	0.016012s
12	4.8306m	59.984s	0.016008s
13	1.2772m	47.640s	0.016013s
14	2.4648m	32.378s	0.016021s
15	1.9281m	11.742s	0.016041s
16	3.7101m	35.048s	0.016018s
17	1.2599m	10.047s	0.016049s
18	3.4225m	51.335s	0.016007s
19	3.1215m	57.624s	0.016006s
20	1.9941m	11.265s	0.016047s

Tabla 5-5: Resultados de 20 pruebas de comportamiento del mejor individuo obtenido en el experimento 1.

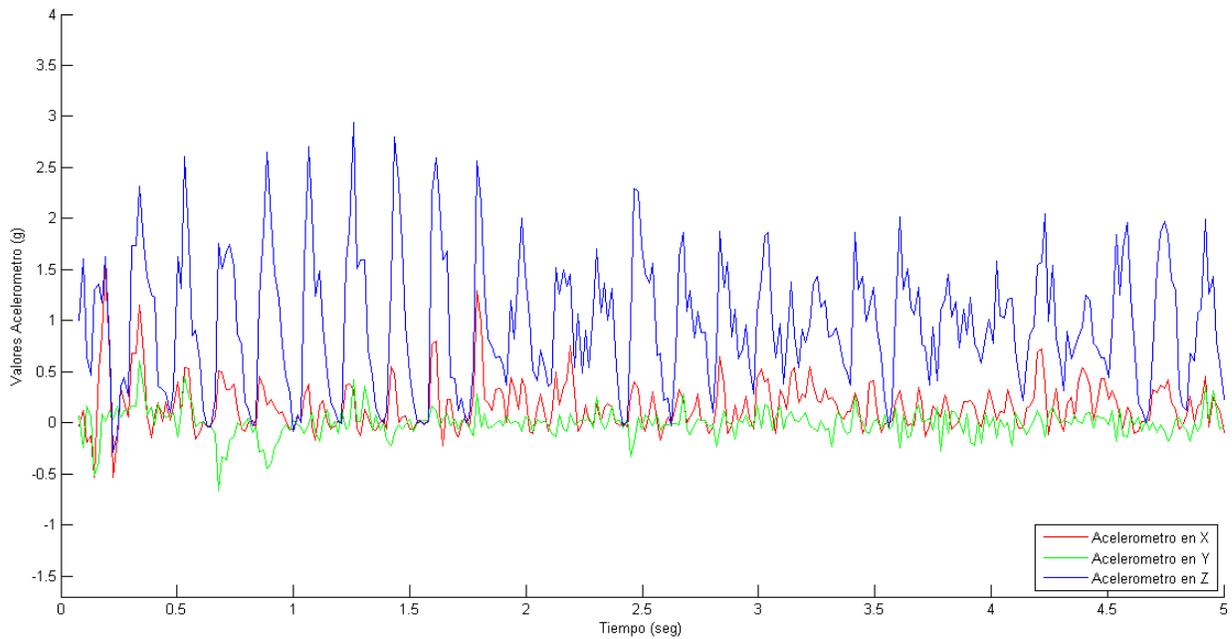


Figura 5-26: Respuesta promedio del acelerómetro en el caminador del experimento 1.

5.3. ROBOT BÍPEDO

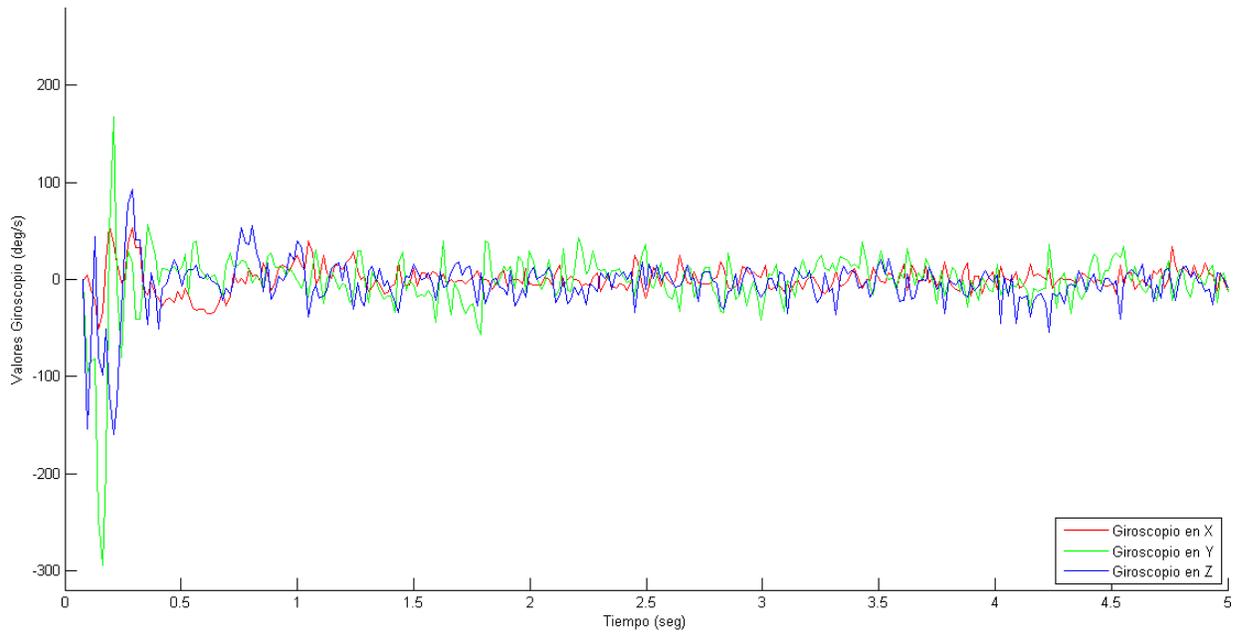


Figura 5-27: Respuesta promedio del giroscopio en el caminador del experimento 1.

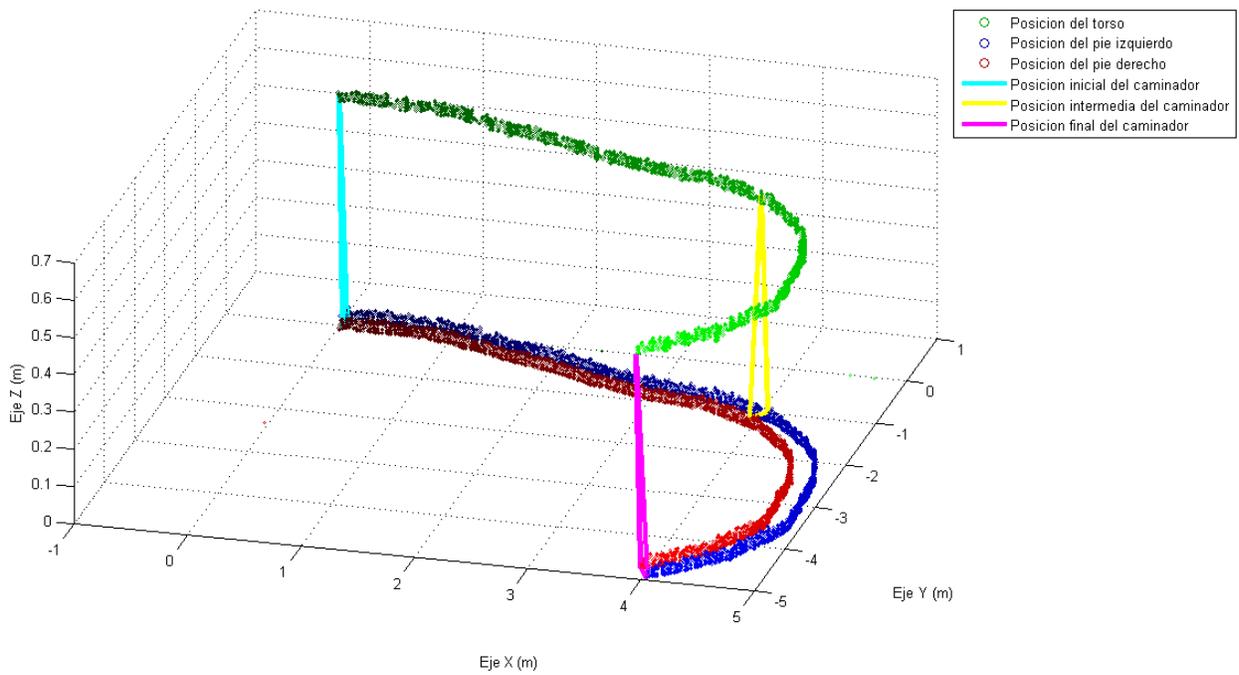


Figura 5-28: Ruta seguida por el caminador en la mejor prueba del experimento 1.

5.3. ROBOT BÍPEDO

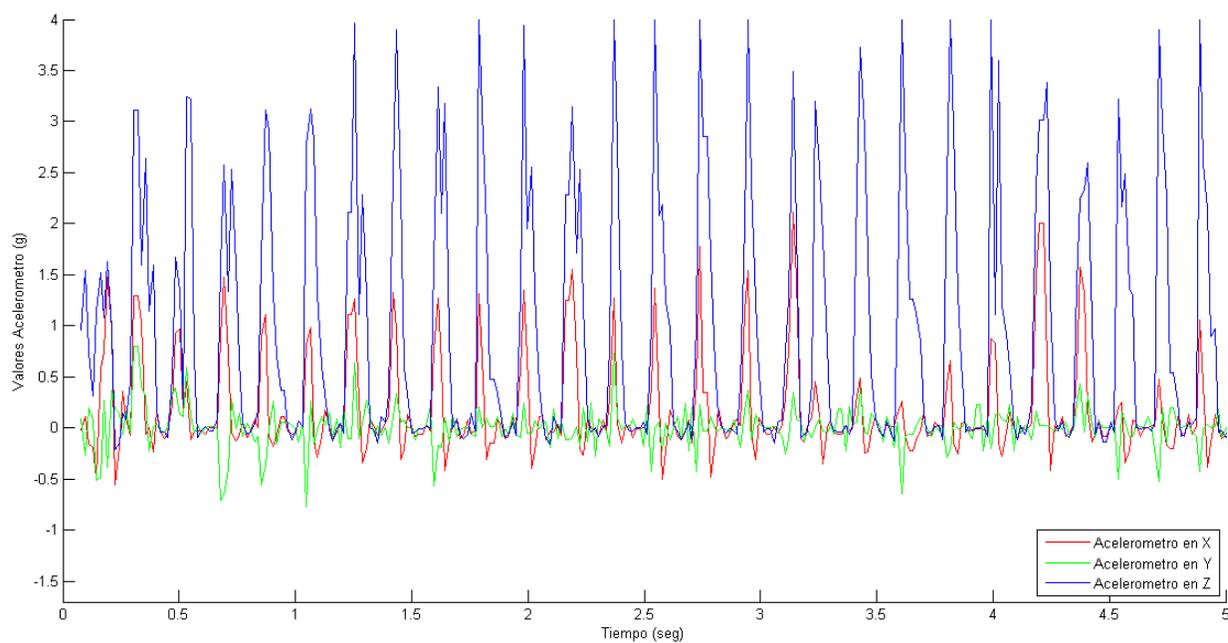


Figura 5-29: Respuesta del acelerómetro en la mejor prueba del caminador del experimento 1.

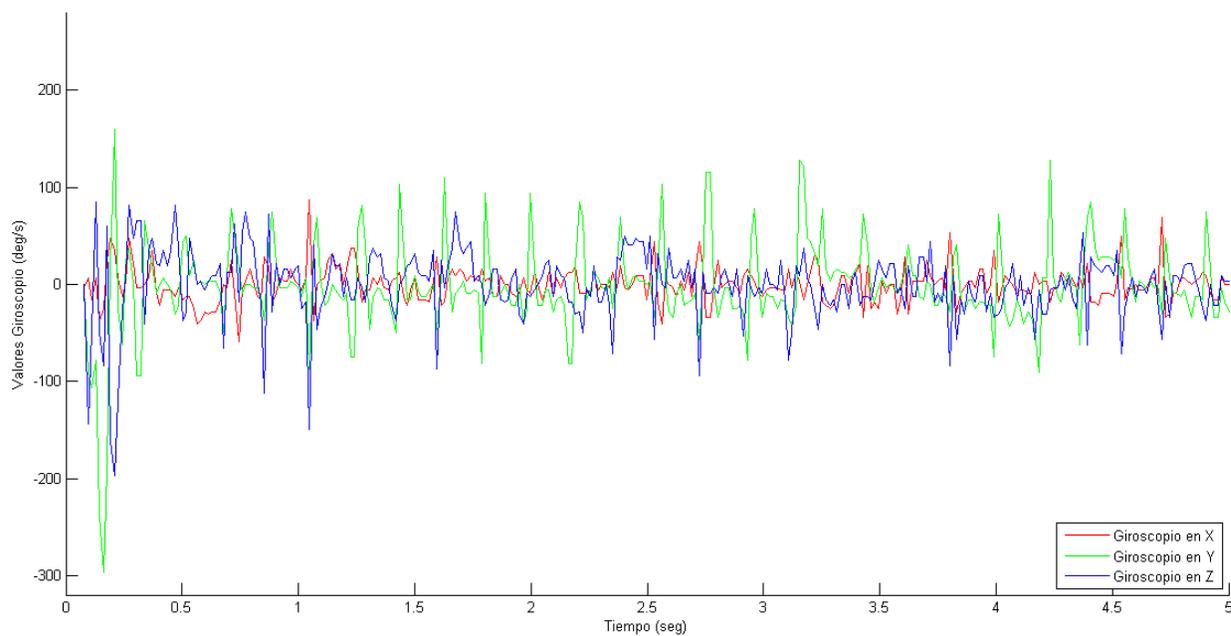


Figura 5-30: Respuesta del giroscopio en la mejor prueba del caminador del experimento 1.

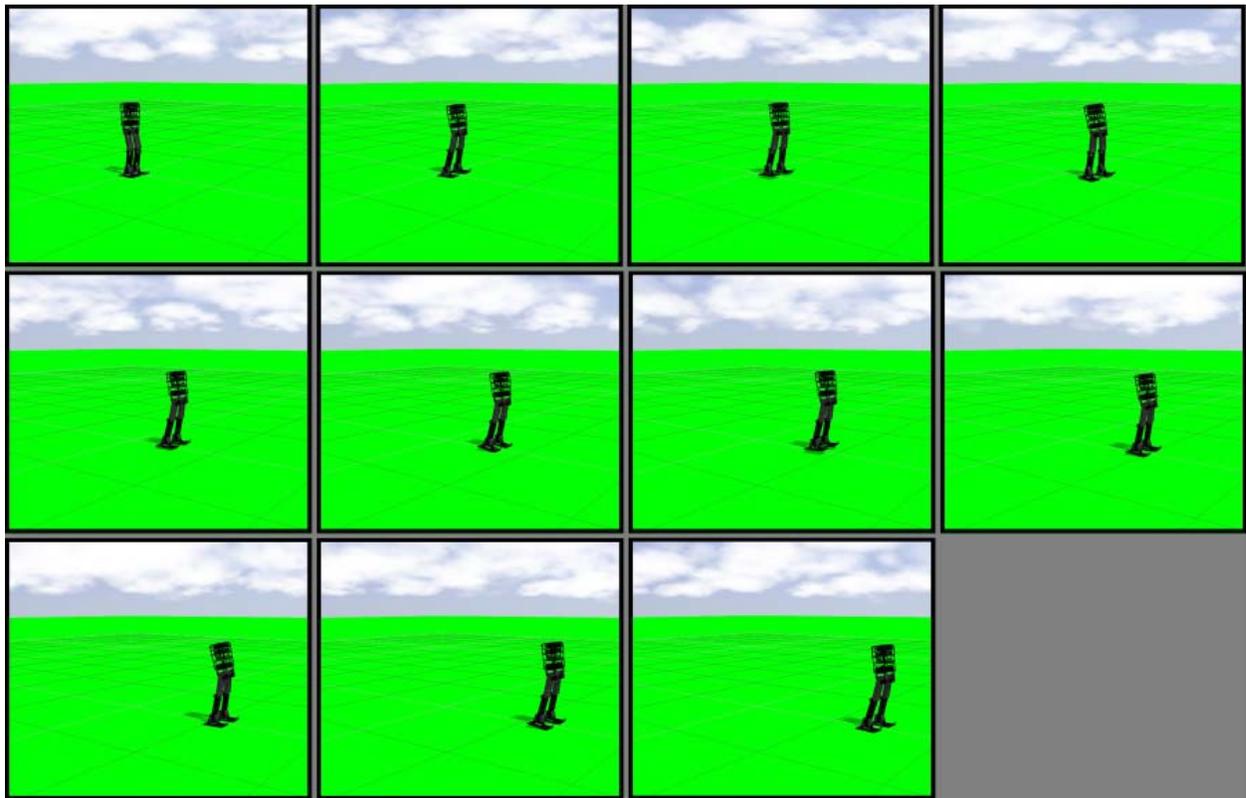


Figura 5-31: Secuencia visual del comportamiento del experimento 1.

5.3.2. Experimento 2, movimiento por deslizamiento

En este experimento los parámetros del algoritmo **NEAT** los muestra la tabla 5-6 y los parámetros de control/red adicionales los muestra la tabla 5-7.

Parámetro	Valor
Probabilidad de solo mutar	25 %
Probabilidad de mutar el peso de una conexión	30 %
Potencia de la mutación de peso	2.5
Probabilidad de añadir una conexión	10 %
Número de intentos para añadir conexión	20
Probabilidad de forzar una conexión recurrente	20 %
Probabilidad de alternar el estado de habilitación de una conexión	2 %
Probabilidad de rehabilitar una conexión	0 %
Probabilidad de añadir nodo	5 %
Probabilidad de solo cruzar	20 %
Probabilidad de cruce multipunto	90 %
Probabilidad de cruce multipunto promediada	10 %
Probabilidad de cruce en un solo punto	0 %
Tasa de cruce entre especies	5 %
Umbral de supervivencia	30 %
Significancia de la edad	1
Edad de descarte	100
Descendencia robada	50
Umbral de compatibilidad	1
Coefficiente de genes disjuntos	10
Coefficiente de genes de exceso	10
Coefficiente de la diferencia de mutaciones	1
Tamaño de la población	500
Número de generaciones	1000
Número de neuronas de entrada	14
Número de neuronas de salida	6
Número de neuronas escondidas	6
Tipo de estructura inicial	2
Límite de los pesos	2, [-2, 2]
Compatibilidad normalizada	<i>Verdadero</i>

Tabla 5-6: Parámetros de NEAT en el experimento 2 del caminador.

La función de aptitud en este experimento cambia a lo largo del tiempo, en las primeras 300 generaciones la función es:

$$f_i = 1 + (C \cdot D) \cdot (10 \cdot M \cdot G) \quad (5-6)$$

Y a partir de la generación 301 hasta la 1000 es:

$$f_i = 1 + (C \cdot D) \cdot (100 \cdot S \cdot G) \quad (5-7)$$

Las función de aptitud se intercambia en el punto donde el soporte de fuerzas en el torso

Parámetro	Valor
Red simétrica	Verdadero
Cruce de entradas	Verdadero
Inversión de guiñada y alabeo en la pierna derecha	Falso
Límites de escalamiento Guiñada cadera izquierda	[-5,5]
Límites de escalamiento Guiñada cadera derecha	[-5,5]
Límites de escalamiento Alabeo cadera izquierda	[-5,5]
Límites de escalamiento Alabeo cadera derecha	[-5,5]
Límites de escalamiento Cabeceo cadera izquierda	[-30,30]
Límites de escalamiento Cabeceo cadera derecha	[-30,30]
Límites de escalamiento Cabeceo rodilla izquierda	[-30,30]
Límites de escalamiento Cabeceo rodilla derecha	[-30,30]
Límites de escalamiento Alabeo tobillo izquierdo	[-5,5]
Límites de escalamiento Alabeo tobillo derecho	[-5,5]
Límites de escalamiento Cabeceo tobillo izquierdo	[-30,30]
Límites de escalamiento Cabeceo tobillo derecho	[-30,30]
Fuerzas de soporte	De la generación 1 a la 300 disminución del efecto 10 % cada 30 generaciones

Tabla 5-7: Parámetros extra del experimento 2 del caminador.

desaparece, la intención fue que durante las primeras 300 generaciones el controlador aprendiera a realizar pasos tratando de no caerse ni modificando su orientación pero sin importar que avanzara o no, para que en el resto de las generaciones usara lo aprendido para ahora buscar moverse en línea recta maximizando el soporte que le proveen las piernas y la distancia que recorre.

Las gráficas en las figuras 5-32, 5-33 y 5-34 muestran el cambio de aptitud, nodos y conexiones a través de las generaciones, no se muestran especies ya que en este caso el proceso de evolución tampoco llegó a generar diferencias suficientes para que se crearan nuevas especies.

La siguiente tabla 5-8 compila el resultado de probar al mejor individuo de la generación 985 unas 20 veces para verificar que tan consistente es su comportamiento, como se observará en este caso el comportamiento es bastante consistente.

Las siguientes gráficas en las figuras 5-35, 5-36 y 5-37 muestran la ruta promedio y respuesta

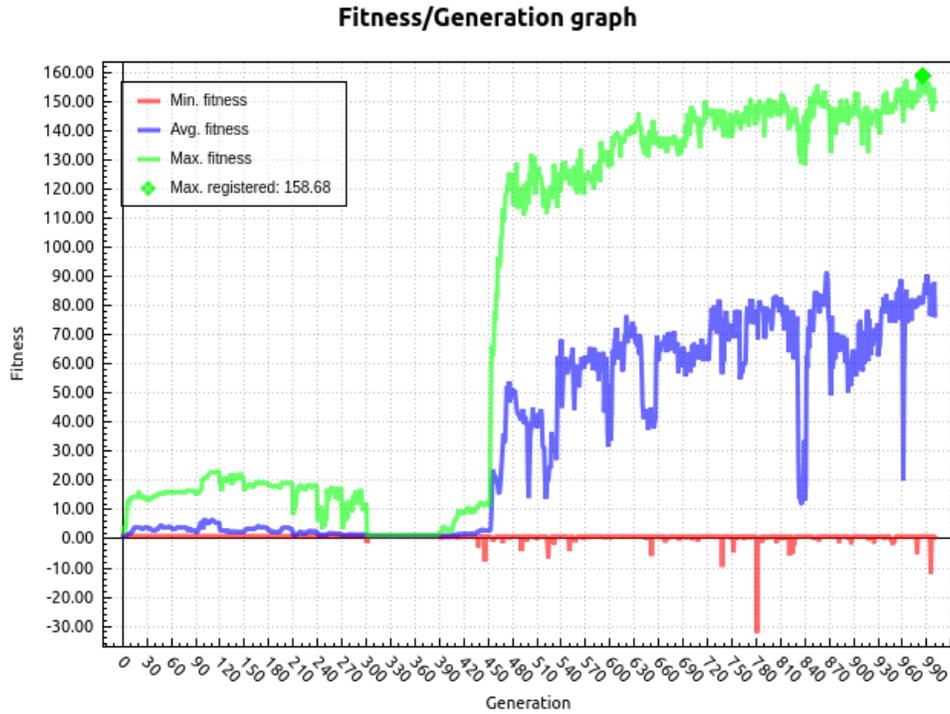


Figura 5-32: Aptitud por generación del experimento 2.

promedio del sensor de **IMU** del caminador en las 20 pruebas en donde el modelo no cayó antes de llegar a los 60 segundos simulados. Las gráficas en las figuras [5-38](#), [5-39](#) y [5-40](#) muestran esa misma información pero para la mejor prueba que alcanzó la mayor distancia en x . Una secuencia visual del comportamiento se aprecia en la figura [5-41](#).

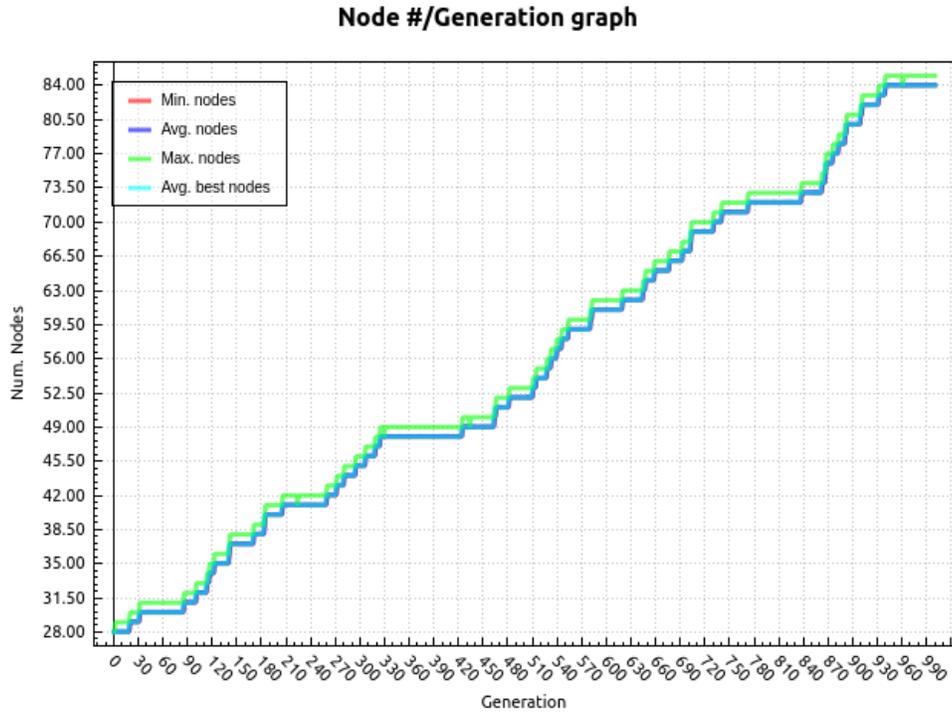


Figura 5-33: Nodos por generación del experimento 2.

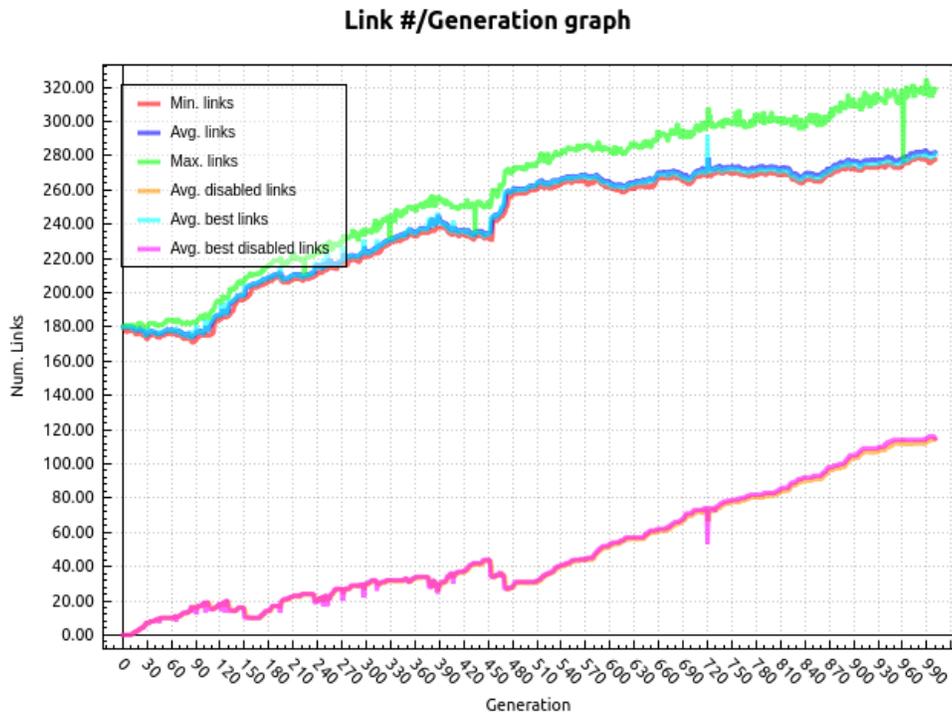


Figura 5-34: Conexiones por generación del experimento 2.

5.3. ROBOT BÍPEDO

Prueba	Distancia máxima recorrida en X	Tiempo transcurrido antes de caer ó terminado a los ~60 segundos	Tiempo promedio de las iteraciones del ciclo de control
1	9.6456m	59.991s	0.016010s
2	9.5058m	59.975s	0.016006s
3	9.5096m	59.988s	0.016009s
4	8.9056m	59.994s	0.016011s
5	8.3898m	59.994s	0.016011s
6	9.3148m	59.976s	0.016006s
7	9.6882m	60.006s	0.016014s
8	9.6411m	59.968s	0.016004s
9	9.6341m	59.975s	0.016006s
10	9.8097m	59.991s	0.016010s
11	9.9165m	59.978s	0.016006s
12	9.8712m	59.994s	0.016011s
13	9.6156m	60.011s	0.016015s
14	9.8435m	60.002s	0.016013s
15	10.1763m	59.975s	0.016006s
16	9.9523m	59.988s	0.016009s
17	9.6872m	60.000s	0.016012s
18	9.4575m	60.009s	0.016015s
19	9.7564m	59.991s	0.016010s
20	9.7235m	59.991s	0.016010s

Tabla 5-8: Resultados de 20 pruebas de comportamiento del mejor individuo obtenido en el experimento 2.

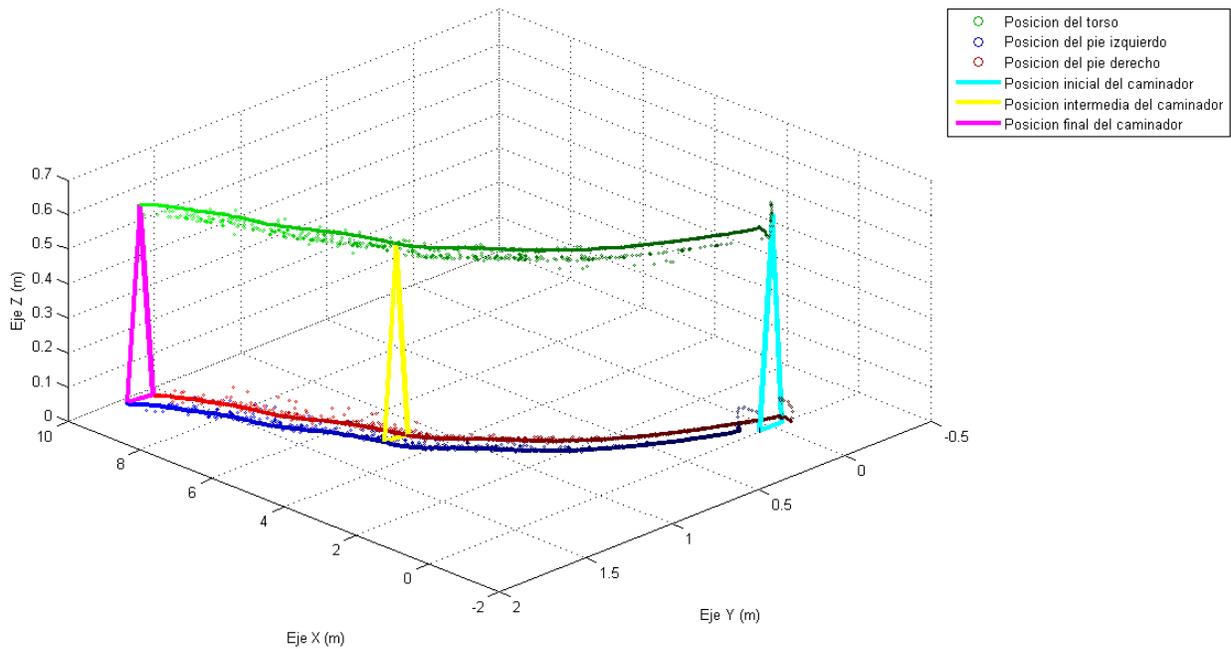


Figura 5-35: Ruta promedio seguida por el caminador del experimento 2.

5.3. ROBOT BÍPEDO

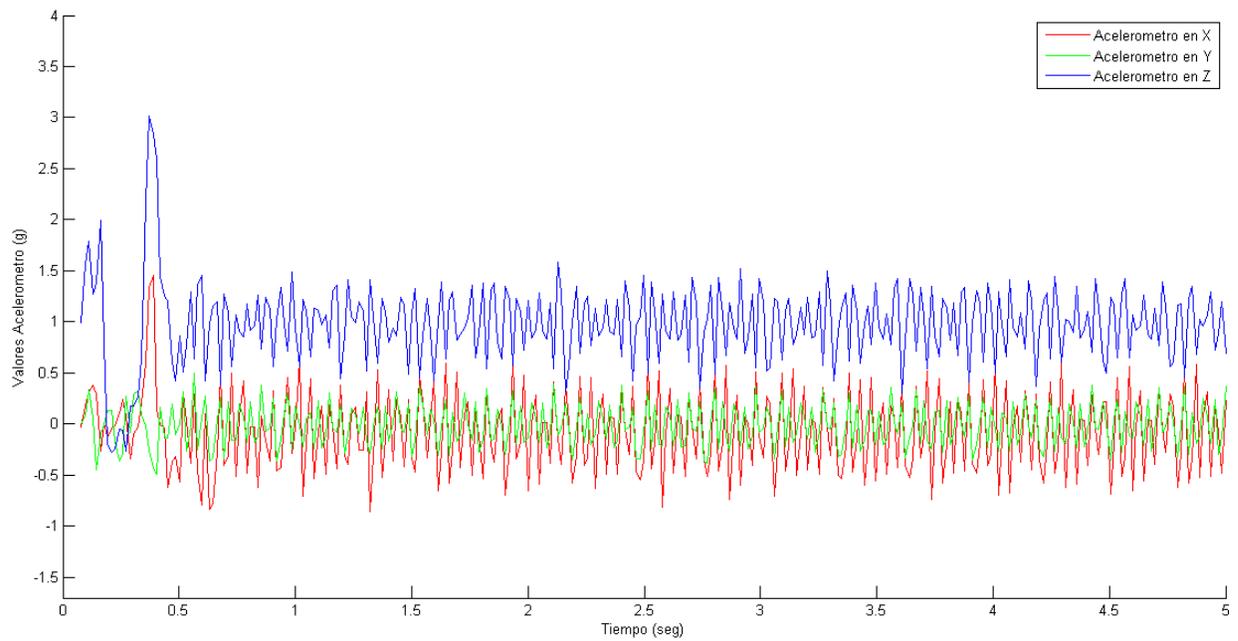


Figura 5-36: Respuesta promedio del acelerómetro en el caminador del experimento 2.

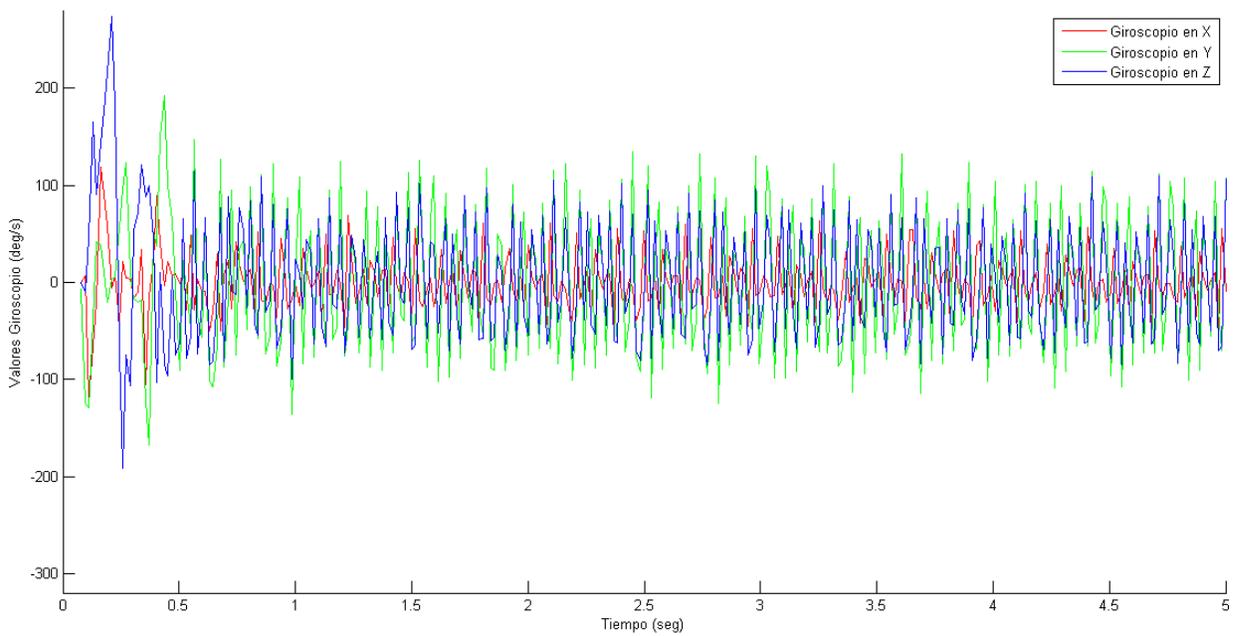


Figura 5-37: Respuesta promedio del giroscopio en el caminador del experimento 2.

5.3. ROBOT BÍPEDO

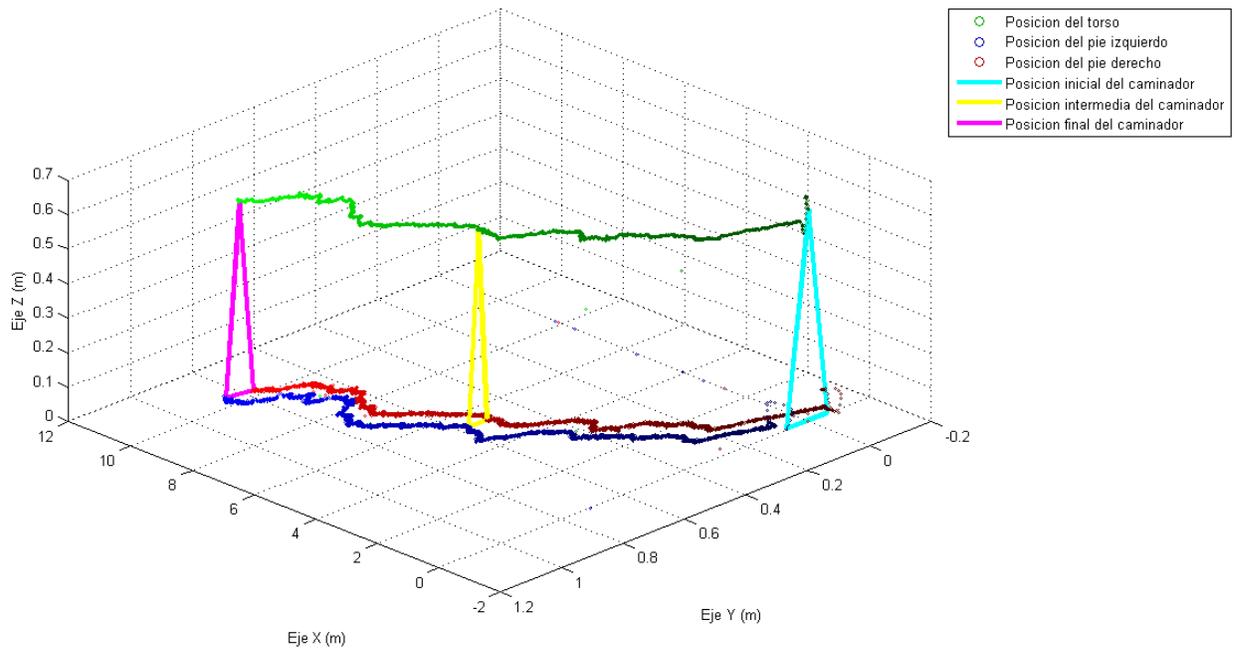


Figura 5-38: Ruta seguida por el caminador en la mejor prueba del experimento 2.

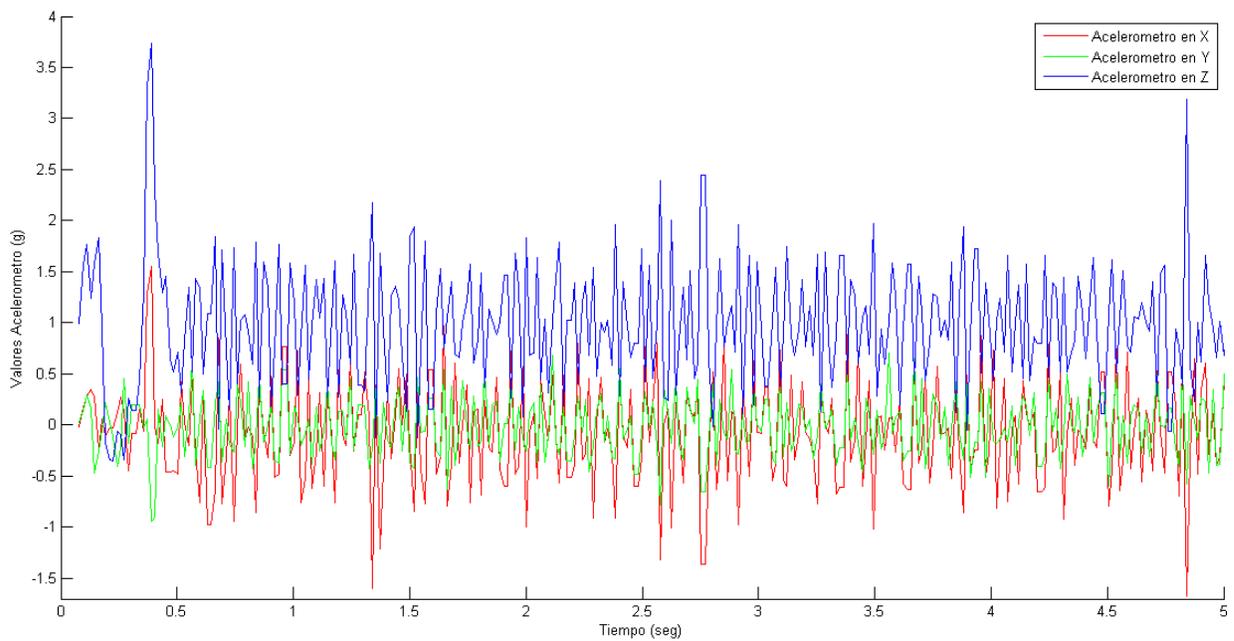


Figura 5-39: Respuesta del acelerómetro en la mejor prueba del caminador del experimento 2.

5.3. ROBOT BÍPEDO

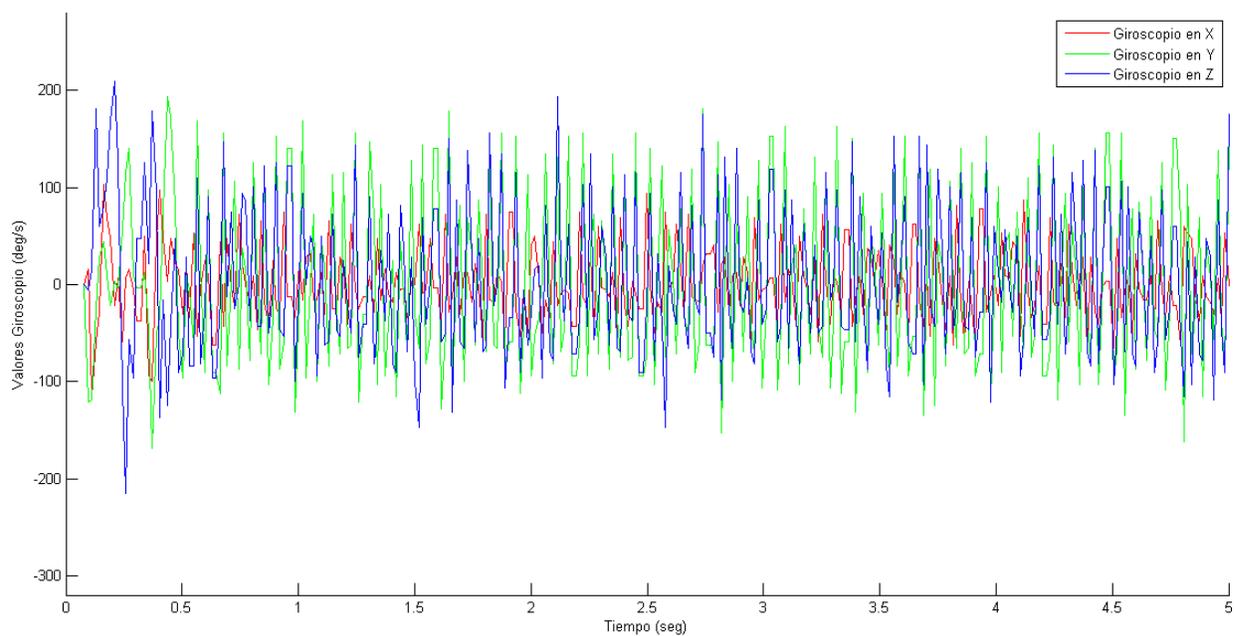


Figura 5-40: Respuesta del giroscopio en la mejor prueba del caminador del experimento 2.

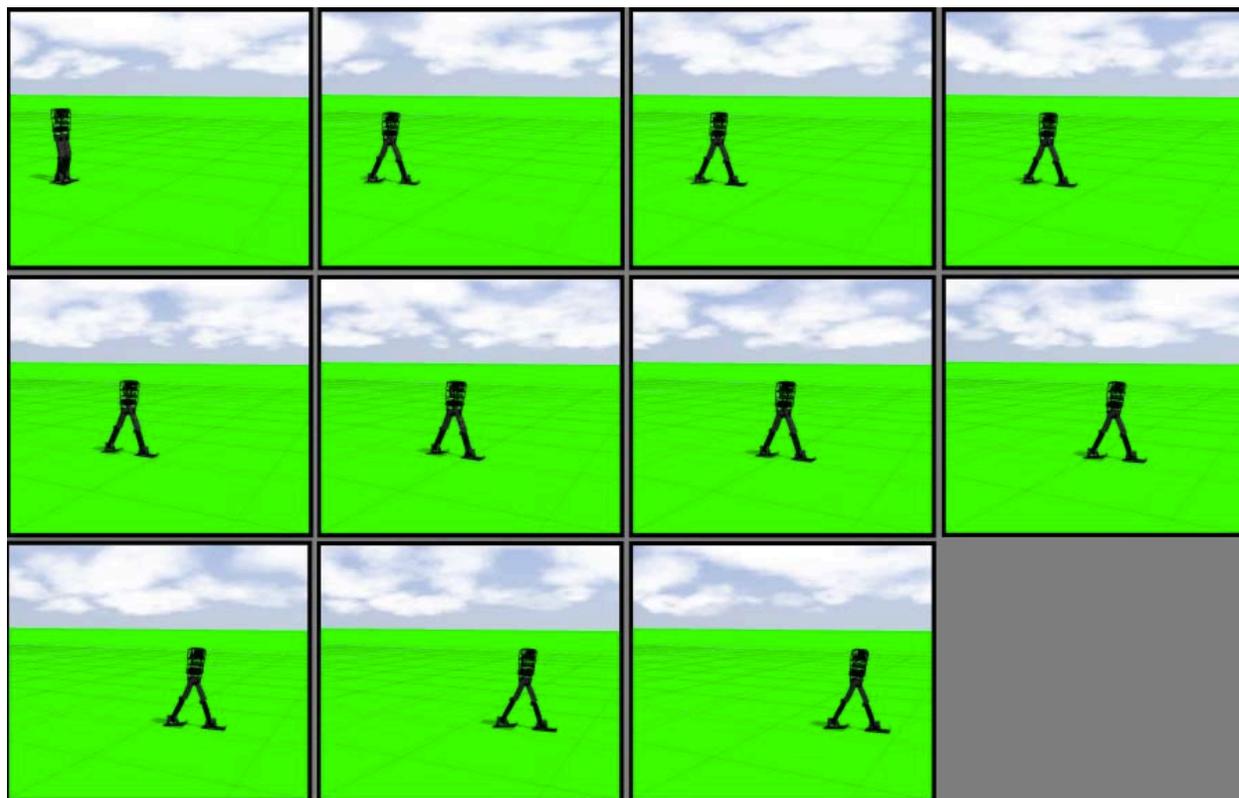


Figura 5-41: Secuencia visual del comportamiento del experimento 2.

6.1. Discusión

El presente trabajo mostró como la integración de varios componentes para crear un sistema de aprendizaje no es una tarea trivial, la elección de estos componentes depende en gran parte del tipo de problemas a los que se les quiere dar solución y los detalles específicos de cada componente, por un lado el algoritmo **NEAT** se puede personalizar muy fácilmente y una vez corregidos los errores del código inicial fue muy sencillo integrarlo, por otro lado el simulador Gazebo demostró ser una herramienta adecuada para simular robots humanoides y móviles en ambientes complejos, no obstante fue sumamente laborioso llegar al nivel de conocimiento adecuado para poderlo integrar tan robustamente como se hizo en la herramienta.

Las pruebas de concepto fueron exitosas y pudieron consistentemente balancear los péndulos durante 20 segundos, mostrando que **NEAT** es apto para evolucionar controladores para sistemas no lineales. En los 2 experimentos con el caminador que se ilustran en este trabajo se encontraron soluciones poco convencionales al problema de movimiento bípedo que explotan las características del ambiente y del modelo, en el experimento 1 dando cortos y rápidos saltos y en el experimento 2 aumentando la **BoS** y moviendo los pies de tal forma que los contactos y la fricción crean un impulso hacia adelante, en ambos casos, para estos experimentos parece que el tiempo de $16ms$ en el ciclo de control fue el principal habilitador que le permitió a las redes encontrar controladores que generan movimientos mínimos y rápidos que desencadenaran el movimiento. Sin duda **NEAT** fue capaz de aumentar la estructura y ajustar los pesos para llegar a las **ANN/RNN** que sirvieron como controladores que producen estos curiosos comportamientos que satisfacen algunas de las complicadas restricciones impuestas en la función de aptitud, lo cual es un claro indicativo de que las técnicas neuroevolutivas son alternativas viables y flexibles para el aprendizaje de controladores,

que a diferencia de métodos tradicionales de control, donde se busca modelar precisamente el sistema para cancelar la dinámica, un método de aprendizaje puede encontrar una forma de aprovechar la dinámica y producir resultados novedosos.

Aunque el sistema de aprendizaje tuvo éxito no está exento de complicaciones. La primera complicación que presenta el utilizar la herramienta desarrollada es que se tiene una gran cantidad de parámetros por definir, tanto para el algoritmo **NEAT** como para otros elementos involucrados en el control y evaluación del modelo, encontrar aquellos que permitan dirigir el proceso de evolución a las soluciones deseadas constituye en si otro problema de optimización.

Una segunda complicación es que tal como se encuentran los controladores resultantes, dudosamente van a poder utilizarse en el robot real y que sucedan los mismos resultados, en este trabajo si se hizo énfasis por replicar adecuadamente, los sensores, actuadores y las propiedades físicas del modelo y del ambiente pero para que las simulaciones pudieran realizarse en tiempos aceptables se aplicaron restricciones y se utiliza el método de solución rápida de **ODE**, el cual es impreciso. Esto no significa que los resultados sean inservibles pero para poder tener algo de certeza en que pueden usarse los resultados, deben probarse primero en otra simulación donde se emplean soluciones mucho más precisas y de ser necesario para realizar el ajuste de los controladores se puede repetir el proceso evolutivo por unas cuantas generaciones usando el simulador de alta precisión y como población inicial los individuos del mejor resultado.

A pesar de que solo se plasmaron 2 experimentos del caminador en realidad se realizaron varios que no llevaron a soluciones, el principal factor que afectó que no se realizaran más experimentos es el tiempo y es la tercera complicación que presenta esta herramienta. El simulador no tuvo problemas en correr más rápido que el tiempo real en ninguno de los experimentos sin embargo mientras más complejo es el modelo más cálculos debe realizar y por lo tanto menos rápido resultará la simulación. En el caso de los péndulos, el simulador pudo correr con un factor de tiempo real de 6 mientras que para el caminador solo logró alcanzar 4, con poblaciones grandes y si los controladores aprenden rápidamente a evitar las condiciones de alto a la evaluación, las generaciones tardan más tiempo en completarse pues las simulaciones corren por un tiempo determinado, en el caso de los experimentos del caminador estos pueden tardar varios días en completarse, por ejemplo, el experimento 2 del caminador tardó en completarse 15 días aproximadamente. En ninguno de los experimentos que se realizaron se logró encontrar un controlador que generara los movimientos característicos del caminado bípedo lo cual no es de extrañarse pues es un problema sumamente complicado y con espacio de búsqueda inmenso, aún no puede descartarse que con ajustes en los parámetros y/o

características de las neuronas empleadas en las redes, la herramienta pueda ser capaz de lograr un caminado bípedo.

6.2. Trabajo futuro

- Para reducir el número de sensores empleados puede recurrirse a técnicas de fusión sensorial, no solo esto ayudaría a disminuir el espacio de búsqueda sino que la combinación y procesamiento de diferentes sensores para generar información más precisa que los sensores separados pueden proveer sobre el estado del mundo es sumamente conveniente.
- La herramienta desarrollada se puede utilizar para probar otros comportamientos que desean realizarse en robots que actúan en ambientes tridimensionales, solo se requiere crear el modelo virtual, y definir la función de aptitud apropiada. Algunos ejemplos de estas tareas son la navegación en ambientes dinámicos, manipulación de objetos, interacción competitiva o cooperativa de múltiples robots por mencionar algunos.
- La funcionalidad de la herramienta probó ser muy efectiva para facilitar la realización de experimentos, pruebas de individuos en la población y recolección de datos de **NEAT** y Gazebo para generar gráficas, aun así hay muchas cosas que por cuestiones de tiempo o que no se visualizaron antes de realizar el código no se desarrollaron y en una siguiente versión de la herramienta podrían ser útiles:
 1. Separar la obtención de métricas para evaluación del ciclo evolutivo, actualmente la toma de mediciones de sensores y otras métricas que se reciban del simulador se realiza dentro del ciclo evolutivo por lo que el número de mediciones que se realizan está limitado el número de iteraciones que el ciclo logre realizar. Si se independiza la obtención de métricas estas pueden obtenerse con una frecuencia que sea más adecuada a las necesidades de análisis que quieran realizarse sobre los datos.
 2. Reducir el número de parámetros que se usan en el algoritmo, en este caso hay demasiados y esto dificulta el uso del algoritmo **NEAT**.
 3. Agregar la visualización de estructura de redes, sensores y métricas que se reciben cuando se realiza la prueba de un individuo son tareas que se deseaban completar en la primera versión de la herramienta pero que no pudieron ser.

- Así como se emplearon muchos mecanismos poco convencionales que dieron resultados, como usar una red dual, fuerzas de soporte, entre otras, se puede recurrir a la integración de alguna técnica adicional a **NEAT** para ver si los resultados que se obtienen mejoran. Los **CPG** han logrado excelentes resultados en tareas de control motor para robots, una modificación que puede realizarse en el algoritmo **NEAT** es desarrollar los genes y operadores necesarios para que se pueda incluir un modelo de **CPG** como un tipo de neurona cuyos parámetros específicos también son evolucionados por el proceso.
- Si se desea que los procesos evolutivos se realicen con más rapidez, una forma posible sería paralelizar la evaluación de los individuos de la población y que cada uno se realice en un procesador o computadora distinta. En este trabajo todos los experimentos se realizaron en una computadora laptop de alto desempeño pero es totalmente posible que computadoras más poderosas pudieran acelerar significativamente la realización de experimentos.
- Se puede tratar de integrar otras técnicas neuroevolutivas en la herramienta de manera que se pueda elegir entre técnicas para buscar la evolución de las soluciones desde diferentes enfoques o incluso dejar que las soluciones de diferentes técnicas compitan, haciendo que un factor importante para dirigir los procesos evolutivos sea la competencia.

Apéndice

- Leyes de movimiento de Newton.¹
 - 1^{era} ley: Todo cuerpo permanecerá en un estado de reposo o un estado de movimiento uniforme amenos que una fuerza externa actúe sobre el obligándolo a cambiar de estado.
 - 2^{nda} ley: El cambio de movimiento (aceleración) de un cuerpo es proporcional a la fuerza neta impuesta sobre el e inversamente proporcional a la masa del cuerpo $F = ma$.
 - 3^{era} ley: Si un cuerpo ejerce una fuerza sobre otro cuerpo, entonces este segundo cuerpo ejerce a su vez una fuerza sobre el primero de igual magnitud pero opuesta en dirección.

- Masa: Es la cantidad de materia que posee un cuerpo relacionada a la densidad y el volumen. Masa gravitacional es la propiedad de un cuerpo que determina la fuerza de atracción que ejerce este sobre otro cuerpo y es proporcional al producto de las masas de ambos cuerpos y la constante de gravitación universal e inversamente proporcional al cuadrado de la distancia entre los cuerpos. Masa inercial es la propiedad de un cuerpo de resistir la aceleración cuando se le aplica una fuerza. La masa gravitacional e inercial son equivalentes.

La unidad de medición internacional de la masa es el kilogramo *kg*.

- Cuerpo rígido: es la idealización de un cuerpo sólido que se considera posee geometría definida y que no se deforma ante la aplicación de fuerzas. Las propiedades que posee este cuerpo, como su masa por ejemplo, determinan como este reacciona ante la aplicación de fuerza en un punto específico.

¹Estas leyes, formuladas por Isaac Newton y publicadas en su obra *Philosophiæ naturalis principia mathematica* en el año de 1687, constituyen la base del estudio de la estática y dinámica de cuerpos rígidos.

- Posición: ubicación de un cuerpo en el espacio con respecto a un marco de referencia, normalmente un sistema de coordenadas con un determinado origen. En el caso de un cuerpo rígido esta posición tiene componentes lineales, simbolizados con x , y angulares, simbolizados por θ .

Las unidades de medición internacional para una ubicación (distancia al origen) son los metros y para la orientación (rotación) son los radianes.

- Grado de libertad (DoF): es el número de tipos de movimiento que puede tener un objeto, el máximo es 6, 3 traslaciones y 3 rotaciones.
- Velocidad: es la razón de cambio de posición con respecto al tiempo, cuando el cambio es con respecto a un desplazamiento angular entonces se llama velocidad angular.

La unidad de medición internacional para la velocidad v son los metros sobre segundo $\frac{m}{s}$ y para la velocidad angular ω son los radianes sobre segundo $\frac{rad}{s}$.

- Aceleración: es la razón de cambio de velocidad con respecto al tiempo, cuando el cambio es de la velocidad angular entonces se llama aceleración angular.

La unidad de medición internacional para la aceleración a son los metros sobre segundo al cuadrado $\frac{m}{s^2}$ y para la aceleración angular α son los radianes sobre segundo al cuadrado $\frac{rad}{s^2}$.

- Ecuaciones de movimiento.

Lineal	Angular
$x = \bar{v}t$	$\theta = \bar{\omega}t$
$v = v_0 + at$	$\omega = \omega_0 + \alpha t$
$x = v_0 t + \frac{1}{2}at^2$	$\theta = \omega_0 t + \frac{1}{2}\alpha t^2$

- Fuerza: Es la acción de un cuerpo sobre otro. Una fuerza es resultante o neta cuando es la suma vectorial de todas las fuerzas que actúan sobre un objeto. Una fuerza superficial es aquella que actúa sobre la superficie de un cuerpo al hacer contacto con otro.

La unidad de medición internacional para la fuerza son los newtons $N = \frac{kg \cdot m}{s^2}$.

- Fuerza normal: es el componente de la fuerza de reacción que ocurre al contacto entre dos superficies y que es perpendicular a la superficie de contacto $F_{normal} = mg \cos(\phi)$ donde ϕ es el ángulo de inclinación de la superficie que soporta el cuerpo y g es la aceleración provocada

por la gravedad, se puede pensar en la fuerza normal como la fuerza que evita que un cuerpo traspase la superficie sobre la que se apoya.

- Torque: también llamado momento de una fuerza, es el efecto de rotación de un objeto alrededor de un eje provocado por la aplicación de una fuerza en algún punto de este objeto, el torque es igual a la magnitud del componente de la fuerza que es perpendicular al centro de rotación multiplicada por la distancia perpendicular que separa la línea de aplicación de la fuerza sobre el objeto del centro de rotación $\tau = F d_{perpendicular} \cos(\phi)$.

La unidad de medición internacional para el torque son los newtons-metro Nm .

- Contacto unilateral: restricción mecánica que previene la penetración entre dos cuerpos al entrar estos en contacto.
- Centro de gravedad (**CoG**): punto en un cuerpo donde se considera actúa la resultante de las fuerzas gravitacionales totales.
- Centro de masa (**CoM**): punto en un cuerpo donde se considera concentrada la masa total y es igual al primer momento de la masa dividido sobre la masa total $r_{centro.masa} = \frac{1}{m_{total}} \int r \, dm$ donde r representa la posición del elemento de masa con respecto a un punto de referencia (origen). El centro de masa se utiliza como el punto de referencia que describe la posición lineal de un cuerpo rígido y como descriptor de la respuesta del cuerpo a fuerzas y torques externos. El centro de masa y centro de gravedad coinciden cuando el campo gravitacional que afecta al cuerpo es uniforme.
- Centro de presión (**CoP**): punto donde la fuerza resultante de una presión distribuida sobre una superficie actúa sobre la misma.
- Peso: La fuerza que ejerce la atracción gravitacional de un planeta sobre un cuerpo, es proporcional a la masa del objeto.
- Fricción: fuerza que surge al contacto de 2 superficies y se opone al deslizamiento entre estas. Se denomina fricción seca a aquella que ocurre al contacto entre 2 superficies sólidas y se opone al deslizamiento lateral, está se subdivide por un lado en fricción estática que es la que actúa mientras las superficies no están en movimiento relativo una de la otra, es decir solo actúa antes de producirse el deslizamiento, y por el otro lado en fricción dinámica o cinética

que actúa mientras las superficies están en movimiento relativo una de la otra, es decir cuando hay deslizamiento.

- Coeficiente de fricción (CoF) estática: se simboliza este coeficiente con μ_s y denota la razón de la fuerza de fricción máxima a la fuerza normal en el punto de deslizamiento inminente, $F_{fs,max} = \mu_s F_{normal}$.
- Coeficiente de fricción (CoF) dinámica: se simboliza este coeficiente con μ_k y denota la razón de la fuerza de fricción a la fuerza normal durante el deslizamiento entre superficies, $F_{fk} = \mu_k F_{normal}$. Este coeficiente es de un 20 a 25 % menor con respecto al de fricción estática.
- Momento de inercia: también conocido como inercia rotacional o masa angular, representa el torque necesario para que la rotación de un cuerpo alrededor de un eje de rotación alcance una aceleración angular determinada, el momento de inercia depende de la geometría del cuerpo y la distribución de la masa con respecto al eje de rotación en virtud del cual se calcula el momento. El momento de inercia I multiplicado por la aceleración angular indica el torque $\tau = I\alpha$ y es el equivalente rotacional de la 2^{nda} ley de Newton
- Matriz de inercia: también conocida como tensor de inercia, esta matriz simétrica se compone de los momentos de inercia alrededor de los ejes principales y los productos de momentos de inercia, la matriz describe la resistencia a la aceleración rotacional de un cuerpo.

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}$$

Donde $I_{xx} = \int (y^2 + z^2) dm$, $I_{yy} = \int (x^2 + z^2) dm$ y $I_{zz} = \int (x^2 + y^2) dm$ son los momentos de inercia principales alrededor de los ejes x , y y z respectivamente y $I_{xy} = I_{yx} = - \int xy dm$, $I_{xz} = I_{zx} = - \int xz dm$ y $I_{yz} = I_{zy} = - \int yz dm$ son los productos de los momentos de inercia.

Para cuerpos con formas comunes y distribución uniforme de masa ya se poseen fórmulas directas para los momentos de inercia, como ejemplo se ilustran los cuerpos rígidos más comunes que se utilizan para modelar robots virtuales:

Esfera sólida de radio r	$I = \begin{bmatrix} \frac{2}{5}mr^2 & 0 & 0 \\ 0 & \frac{2}{5}mr^2 & 0 \\ 0 & 0 & \frac{2}{5}mr^2 \end{bmatrix}$
Cuboide sólido de largo d_x , ancho d_y y alto d_z	$I = \begin{bmatrix} \frac{1}{12}m(d_y^2 + d_z^2) & 0 & 0 \\ 0 & \frac{1}{12}m(d_x^2 + d_z^2) & 0 \\ 0 & 0 & \frac{1}{12}m(d_x^2 + d_y^2) \end{bmatrix}$
Cilindro sólido de radio r y altura d_z	$I = \begin{bmatrix} \frac{1}{12}m(3r^2 + d_z^2) & 0 & 0 \\ 0 & \frac{1}{12}m(3r^2 + d_z^2) & 0 \\ 0 & 0 & \frac{1}{2}mr^2 \end{bmatrix}$

- Efecto de Coriolis: es la desviación aparente de un cuerpo en movimiento cuando este movimiento se describe relativo a un marco de referencia en rotación y se aprecia como que el cuerpo se mueve lateralmente en la dirección opuesta a la rotación del marco de referencia.
- Fuerza de Coriolis: es la fuerza ficticia que representa el efecto de Coriolis en un marco de referencia no inercial, $F_{coriolis} = 2m\omega v$, donde ω es la velocidad angular del marco de referencia y v es la velocidad del cuerpo con respecto al marco de referencia.
- Efecto centrífugo: es la desviación aparente de un cuerpo en movimiento cuando este movimiento se describe relativo a un marco de referencia en rotación y se aprecia como que el cuerpo se mueve alejándose de forma perpendicular al eje de rotación del marco de referencia.
- Fuerza centrífuga: es la fuerza ficticia que representa el efecto centrífugo en un marco de referencia no inercial, $F_{centrifuga} = m\omega^2 r$, donde ω es la velocidad angular del marco de referencia y r la distancia del cuerpo al eje de rotación del marco de referencia.
- Cinemática: rama de la mecánica que estudia el movimiento de partículas y cuerpos rígidos sin considerar las causas detrás de este. En la robótica hay 2 problemas que la cinemática ayuda a resolver. El cálculo de la posición de un robot dados parámetros específicos para las articulaciones, lo cual se conoce como cinemática directa y el cálculo de los parámetros específicos para las articulaciones dada una posición específica, lo cual se conoce como cinemática inversa.
- Dinámica: rama de la mecánica que estudia el movimiento de partículas y cuerpos rígidos considerando las causas que lo provocan y las leyes que lo rigen. En la robótica hay 2 problemas

A. *CONCEPTOS DE FÍSICA*

que la dinámica ayuda a resolver. El cálculo del movimiento de un robot dadas las fuerzas y momentos que actúan sobre el, lo cual se conoce como dinámica directa y el cálculo de las fuerzas y momentos requeridos para que se genere un movimiento deseado, lo cual se conoce como dinámica inversa.

Parámetros y diagrama de clases en NEAT

B.1. Parámetros

Los siguientes parámetros permiten ajustar el comportamiento del algoritmo **NEAT**, hay parámetros adicionales que permiten controlar la mutación de rasgos en los genes, estos no se mencionan ya que en el presente trabajo no se utilizan y en el código descargado la implementación de los rasgos no está completa.

Parámetros que afectan la mutación:

1. Probabilidad de solo mutar, define la probabilidad de que un descendiente se obtenga solo aplicando mutación al padre seleccionado.
2. Probabilidad de mutar el peso de una conexión, con que probabilidad se muta el valor del peso de una conexión.
3. Potencia de la mutación de peso, define que tanto se puede modificar un peso a través de la mutación.
4. Probabilidad de añadir una conexión, con que probabilidad puede ocurrir la mutación que añade conexión.
5. Número de intentos para añadir conexión, cuantos intentos se realizan para hallar un par de nodos no conectados entre los cuales se pueda crear una conexión.
6. Probabilidad de forzar una conexión recurrente, probabilidad de que la conexión por añadir se busque sea recurrente.
7. Probabilidad de alternar el estado de habilitación de una conexión, probabilidad de que a una conexión elegida al azar se le alterne su estado de habilitación.

8. Probabilidad de rehabilitar una conexión, probabilidad de que a la primera conexión deshabilitada en la red se la vuelva a habilitar.
9. Probabilidad de añadir nodo, con que probabilidad puede ocurrir la mutación de añade un nodo.

Parámetros que afectan la cruce:

1. Probabilidad de solo cruzar, define la probabilidad de que un descendiente se obtenga solo aplicando la cruce entre los padres seleccionados.
2. Probabilidad de cruce multipunto, con que probabilidad la cruce entre genes se hace multipunto normal.
3. Probabilidad de cruce multipunto promediada, si la cruce multipunto normal no ocurre con que probabilidad se hace una cruce multipunto donde en lugar de heredar directamente genes emparejados se promedian sus pesos.
4. Probabilidad de cruce en un solo punto, esta probabilidad se resta a la probabilidad de cruce multipunto promediada, ya que si la cruce multipunto no ocurre, la cruce se hace eligiendo un punto en el genoma más corto a partir del cual se intercambian genes.
5. Tasa de cruce entre especies, probabilidad con la cual uno de los padres puede elegirse de una especie ajena a la que se está reproduciendo en ese momento.

Parámetros de selección y supervivencia:

1. Umbral de supervivencia, porcentaje de individuos más aptos de cada especie a los cuales se les va a permitir reproducirse.
2. Significancia de la edad, coeficiente que pondera la aptitud de individuos en una especie recién surgida y que no ha alcanzado una determinada edad, otorga ventaja a especies nuevas para que puedan sobrevivir.
3. Edad de descarte, indica el número de generaciones que una especie debe pasar estancada (sin mejorar la aptitud máxima alcanzada por sus individuos) antes de que se le marque para que sea destruida. La destrucción de una especie se realiza penalizando severamente la aptitud de sus individuos antes de que se haga la asignación de descendencia lo cual lleva a la disminución del tamaño de la especie hasta que simplemente se extingue.

4. Descendencia robada, indica un número de descendientes que se va a robar de las peores especies y se va a asignar a las mejores especies, este parámetro hace más elitista al algoritmo ya que permite que algunos de los descendientes sean copias de los mejores individuos en las especies dominantes. Es conveniente nunca asignarle un valor equivalente a un porcentaje significativo de la población ya que sesga severamente la evolución. Aún cuando no exista descendencia robada, durante la reproducción si la especie tiene un tamaño suficiente se realiza una copia del mejor individuo de esa especie.

Parámetros de compatibilidad entre especies

1. Umbral de compatibilidad, el umbral de la distancia entre especies que debe satisfacerse para considerar que 2 individuos son de una misma especie.
2. Coeficiente de genes disjuntos, es el coeficiente que ajusta la importancia del término de genes disjuntos.
3. Coeficiente de genes de exceso, es el coeficiente que ajusta la importancia del término de genes de exceso.
4. Coeficiente de la diferencia de mutaciones, es el coeficiente que ajusta la importancia del término de diferencia promedio de pesos entre genes emparejados.

Parámetros de la ejecución

1. Tamaño de la población, cuanto individuos hay en la población.
2. Número de generaciones, cuantas generaciones van a evolucionarse.
3. Imprimir archivos cada n generaciones, cada cuantas generaciones se hace un guardado de archivos que describen cada individuo de la población y la historia evolutiva.

Otros parámetros:

1. Número de neuronas de entrada, con cuantas neuronas de entrada deben crearse los individuos que forman la población inicial.
2. Número de neuronas de salida, con cuantas neuronas de salida deben crearse los individuos que forman la población inicial.
3. Número de neuronas escondidas, con cuantas neuronas escondidas deben crearse los individuos que forman la población inicial, neuronas escondidas solo se crean en el tipo 2 de estructura inicial.

- [1] B. Allen and P. Faloutsos. Complex networks of simple neurons for bipedal locomotion. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS'09*, pages 4457–4462. IEEE Press, 2009. ISBN 9781424438037.
- [2] N. Ando, S. Balakirsky, T. Hemker, M. Reggiani, and O. von Stryk. *Simulation, Modeling, and Programming for Autonomous Robots*, volume 6472 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2010.
- [3] S. Behnke. Online trajectory generation for omnidirectional biped walking. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 1597–1603, May 2006. doi: 10.1109/ROBOT.2006.1641935.
- [4] S. Behnke. Humanoid robots - from fiction to reality? *KI*, 22(4):5–9, 2008.
- [5] A. Bermúdez. Robótica evolutiva: Búsqueda de comportamientos de navegación reactiva para robots móviles. Master thesis, Universidad Nacional Autónoma de México, Aug 2014.
- [6] R. Brooks. Artificial life and real robots. In *Proceedings of the 1st European Conference on Artificial Life*, pages 3–10. MIT Press, 1992.
- [7] C. Chevallereau, G. Bessonnet, G. Abba, and Y. Aoustin. *Bipedal Robots: Modeling Design and Walking Synthesis*. ISTE. Wiley, 2009. ISBN 9781848210769.
- [8] A. H. Cohen. Control principles for locomotion - looking toward biology. In H. Kimura, K. Tsuchiya, A. Ishiguro, and H. Witte, editors, *Adaptive Motion of Animals and Machines*, pages 41–51. Springer Tokio, 2006.

- [9] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.
- [10] M. Dimitrijevic, Y. Gerasimenko, and M. M. Pinter. Evidence for a spinal central pattern generator in humans. *Annals of the New York Academy of Sciences*, 860(1):360–376, 1998. doi: 10.1111/j.1749-6632.1998.tb09062.x.
- [11] G. Dudek and M. Jenkin. *Computational Principles of Mobile Robotics*. Computational Principles of Mobile Robotics. Cambridge University Press, 2010. ISBN 9780521692120.
- [12] G. Endo, J. Morimoto, J. Nakanishi, and G. Cheng. An empirical exploration of a neural oscillator for biped locomotion control. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, volume 3, pages 3036–3042, April 2004. doi: 10.1109/ROBOT.2004.1307523.
- [13] P. Faloutsos, M. van de Panne, and D. Terzopoulos. Composable controllers for physics-based character animation. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 251–260. ACM, 2001. ISBN 158113374X.
- [14] H. Fujimoto, M. O. Tokhi, H. Mochiyama, and G. S. Virk. *Emerging Trends in Mobile Robotics: Proceedings of the 13th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*. World Scientific, 2010. ISBN 9789814327978.
- [15] T. Fukuda, Y. Hasegawa, K. Sekiyama, and T. Aoyama. *Multi-Locomotion Robotic Systems: New Concepts of Bio-inspired Robotics*. Springer Tracts in Advanced Robotics. Springer-Verlag Berlin Heidelberg, 2012. ISBN 9783642301353.
- [16] T. Geijtenbeek. *Animating Virtual Characters using Physics-Based Simulation*. PhD Thesis, Utrecht University, 2013.
- [17] T. Geng, B. Porr, and F. Wörgötter. Fast biped walking with a sensor-driven neuronal controller and real-time online learning. *The International Journal of Robotics Research*, 25(3): 243–259, Mar 2006. doi: 10.1177/0278364906063822.
- [18] D. Graupe. *Principles of Artificial Neural Networks*. World Scientific Publishing Co., Inc., 2007. ISBN 9812706240.

- [19] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, Jun 2001. doi: 10.1162/106365601750190398.
- [20] K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka. The development of honda humanoid robot. In *Robotics and Automation, 1998. Proceeding. 1998 IEEE International Conference on*, volume 2, pages 1321–1326, May 1998. doi: 10.1109/ROBOT.1998.677288.
- [21] A. J. Ijspeert. Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21(4):642–653, 2008. doi: 10.1016/j.neunet.2008.03.014.
- [22] S. Ivaldi, J. Peters, V. Padois, and F. Nori. Tools for simulating humanoid robot dynamics: A survey based on user feedback. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 842–849, Nov 2014.
- [23] S. Kajita, F. Kanehiro, K. Kando, K. Yokoi, and H. Hirukawa. The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 1, pages 239–246, 2001.
- [24] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa. Biped walking pattern generation by using preview control of zero-moment point. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference On*, volume 2, pages 1620–1626, Sep 2003. doi: 10.1109/ROBOT.2003.1241826.
- [25] A. D. Kuo, J. M. Donelan, and A. Ruina. Energetic consequences of walking like an inverted pendulum: step-to-step transitions. *Exercise and sport sciences reviews*, 33(2):88–97, Apr 2005.
- [26] A. F. Kuri-Morales. The best neural network architecture. In A. Gelbukh, F. C. Espinoza, and S. Galicia-Haro, editors, *Nature-Inspired Computation and Machine Learning*, volume 8857, pages 72–84. Springer International Publishing, 2014.
- [27] J.-C. P. Lamarche, O. Bruneau, and J.-G. Fontaine. *The current humanoid robots with seven degrees of freedom per leg are not sufficient to accurately reproduce human walking kinematics*, chapter 58, pages 467–474. In Fujimoto et al. [14], 2010. ISBN 9789814327978. doi: 10.1142/9789814329927_0058.

- [28] J. Lee and J. H. Oh. Biped walking pattern generation using reinforcement learning. In *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*, pages 416–421, Nov 2007. doi: 10.1109/ICHR.2007.4813903.
- [29] M. A. Lewis, A. H. Fagg, and G. A. Bekey. *Genetic Algorithms for Gait Synthesis in a Hexapod Robot*, chapter 11, pages 317–331. In *World Scientific series in robotics and intelligent systems* Zheng [64], 1994.
- [30] K. Matsuoka. Sustained oscillations generated by mutually inhibiting neurons with adaptation. *Biological Cybernetics*, 52(6):367–376, 1985.
- [31] T. McGeer. Passive dynamic walking. *The International Journal of Robotics Research*, 9(2): 62–82, 1990. doi: 10.1177/027836499000900206.
- [32] T. McGeer. Passive walking with knees. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 1640–1645, May 1990. doi: 10.1109/ROBOT.1990.126245.
- [33] L. Medsker and L. C. Jain. *Recurrent Neural Networks: Design and Applications*. International Series on Computational Intelligence. CRC Press, 1999. ISBN 9781420049176.
- [34] J.-A. Meyer, P. Husbands, and I. Harvey. Evolutionary robotics: A survey of applications and problems. In *Proceedings of the 1st European Workshop on Evolutionary Robotics*, pages 1–21. Springer-Verlang, 1998.
- [35] W. T. Miller III. Real-time neural network control of a biped walking robot. *Control Systems, IEEE*, 14(1):41–48, Feb 1994. doi: 10.1109/37.257893.
- [36] M. Missura and S. Behnke. Balanced walking with capture steps. In *Proceedings of the 18th Robocup International Symposium*, July 2014.
- [37] A. L. Nelson, G. J. Barlow, and L. Doitsidis. Fitness functions in evolutionary robotics: A survey and analysis. *Robotics and Autonomous Systems*, 57(4):345–370, 2009. doi: 10.1016/j.robot.2008.09.009.
- [38] Open Source Robotics Foundation. Gazebo, 2015. URL www.gazebosim.org.

- [39] J. Pratt, P. Dilworth, and G. Pratt. Virtual model control of a bipedal walking robot. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 1, pages 193–198, Apr 1997. doi: 10.1109/ROBOT.1997.620037.
- [40] R. D. Quinn. *Animals as models for robot mobility and autonomy: crawling, walking, running, climbing and flying*, chapter 2, pages 4–8. In Fujimoto et al. [14], 2010. ISBN 9789814327978. doi: 10.1142/9789814329927_0002.
- [41] T. Reil and P. Husbands. Evolution of central pattern generators for bipedal walking in a real-time physics environment. *Evolutionary Computation, IEEE Transactions on*, 6(2):159–168, Apr 2002. doi: 10.1109/4232.996015.
- [42] C. D. Remy, O. Baur, M. Latta, A. Lauber, M. Hutter, M. A. Hoeplinger, C. Pradalier, and R. Siegwart. Walking and crawling with ALoF: a robot for autonomous locomotion on four legs. *Industrial Robot: An International Journal*, 38(3):264–268, 2011. doi: 10.1108/01439911111122761.
- [43] A. Roennau, F. Sutter, G. Heppner, J. Oberlaender, and R. Dillmann. Evaluation of physics engines for robotic simulations with special focus on the dynamic walking of robots. In *Advanced Robotics (ICAR), 2013 16th International Conference on*, pages 1–7, Nov 2013. doi: 10.1109/ICAR.2013.6766527.
- [44] R. Rojas. *Neural Networks: A Systematic Introduction*. Springer-Verlag New York, Inc., 1996. ISBN 3540605053.
- [45] G. Rudolph. Convergence analysis of canonical genetic algorithms. *Neural Networks, IEEE Transactions on*, 5(1):96–101, Jan 1994. doi: 10.1109/72.265964.
- [46] M. Schwarz, M. Schreiber, S. Schueller, M. Missura, and S. Behnke. NimbRo-OP humanoid teensize open platform. In *Proceedings of 7th Workshop on Humanoid Soccer Robots, IEEE-RAS International Conference on Humanoid Robots, Osaka*, Nov 2012.
- [47] K. Sims. Evolving virtual creatures. In *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '94*, pages 15–22. ACM, 1994. ISBN 0897916670.
- [48] R. Smith. Open Dynamics Engine™, 2007. URL www.ode.org.

- [49] C. Soler-Gracia, R. Lafuente, P. Vera, J. H. Fuentes, E. Viosca, M. Comín, and A. Cortés. *Biomecánica de la marcha humana normal y patológica*. Instituto de Biomecánica de Valencia, 2006. ISBN 9788492397464.
- [50] K. O. Stanley. *Efficient Evolution of Neural Networks*. PhD Thesis, University of Texas at Austin, 2004.
- [51] K. O. Stanley and R. Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, Jun 2002. doi: 10.1162/106365602320169811.
- [52] R. Tedrake, T. W. Zhang, and H. S. Seung. Learning to walk in 20 minutes. In *14th Yale workshop on Adaptive and Learning Systems*, 2005.
- [53] M. van de Panne and E. Fiume. Sensor-actuator networks. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '93*, pages 335–342. ACM, 1993. ISBN 0897916018.
- [54] P. van Zutven, D. Kostić, and H. Nijmeijer. *On the Stability of Bipedal Walking*, pages 521–532. Volume 6472 of *Lecture Notes in Computer Science* Ando et al. [2], 2010. doi: 10.1007/978-3-642-17319-6_47.
- [55] M. Vukobratović and B. Borovac. Zero-moment point - thirty five years of its life. *International Journal of Humanoid Robotics*, 1(1):157–173, 2004. doi: 10.1142/S0219843604000083.
- [56] L. Wang, K. C. Tan, and C. M. Chew. *Evolutionary Robotics: From Algorithms to Implementations*. World Scientific series in robotics and intelligent systems. World Scientific Publishing, 2006. ISBN 9789812568700.
- [57] X. Wang and E. K. Blum. Discrete-time versus continuous-time models of neural networks. *Journal of Computer and System Sciences*, 45(1):1–19, 1992.
- [58] J. Weng, W. S. Hwang, Y. Zhang, C. Yang, and R. J. Smith. Developmental humanoids: Humanoids that develop skills automatically. In *Proceedings the 1st IEEE-RAS International Conference on Humanoid Robots*. IEEE Press, 2000.
- [59] X. Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, Sep 1997. doi: 10.1109/5.784219.

- [60] X. Yao and Y. Liu. A new evolutionary system for evolving artificial neural networks. *Neural Networks, IEEE Transactions on*, 8(3):694–713, May 1997. doi: 10.1109/72.572107.
- [61] E. Yazdi and A. T. Haghghat. Evolution of biped walking using neural oscillators controller and harmony search algorithm optimizer. *CoRR*, abs/1006.4553, 2010.
- [62] K. Yin, K. Loken, and M. van de Panne. SIMBICON: Simple biped locomotion control. *ACM Transactions on Graphics*, 26(3), Jul 2007. doi: 10.1145/1276377.1276509.
- [63] A. Zalzalá and A. Morris. *Neural Networks For Robotic Control*. APearson print on demand edition. Ellis Horwood, 1996. ISBN 9780131198920.
- [64] Y. F. Zheng. *Recent Trends in Mobile Robots*. World Scientific series in robotics and intelligent systems. World Publishing, 1994.