Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Unidad Zacatenco

Departamento de Computación

Generación de caminata robótica bípeda mediante análisis de datos de la marcha humana

Tesis que presenta

Williams Antonio Pantoja Laces

para obtener el Grado de

Maestro en Ciencias en Computación

Directores de tesis

Dra. Xiaoou Li Zhang

Dr. Wen Yu Liu

México, Distrito Federal — Diciembre, 2015

Resumen

Un alto porcentaje de los robots humanoides que se implementan hoy en día intentan imitar nuestro sistema motriz para desplazarse e interactuar con el medio que les rodea. La ventaja principal de la utilización de robots humanoides reside en que este tipo de robots puede trabajar directamente en los mismos ambientes que los humanos sin que se deban realizar modificaciones sobre dicho ambiente; al contrario de lo que ocurre actualmente con los robots móviles con ruedas. Actualmente, se buscan novedosas formas que ayuden al perfeccionamiento y optimización de la marcha bípeda en robots. Una propuesta de mejora es el uso de datos que describan la marcha humana, debido a que la locomoción humana tiene características flexibles, estables y naturales útiles en los robots bípedos. En este trabajo se genera marcha robótica bípeda estable mediante el análisis de datos con técnicas de *data mining*. Primero se realizan controladores clásicos como el PD y PID, los cuales no utilizan datos humanos y son punto de referencia para implementar técnicas que combinan el control de robot bípedos y el procesamiento de datos humanos utilizando *data minig*. Las técnicas que se utilizaron son de agrupación los cuales son regression clustering (RC), k-means y fuzzy c-means (FCM). Para poder comprobar su funcionamiento y garantizar su estabilidad de la marcha (sin caídas), se diseñaron experimentos los cuales están basados en la macha suave, el comportamiento de los métodos ante el aumento de velocidad y cambio de inclinación del suelo. Los resultados obtenidos con las implementaciones de las técnicas que utilizan datos humanos son mejores ante el aumento velocidad y la robustez presentada es mejor que las técnicas de control clásico PD y PID. Ademas si se realiza una comparación entre las tres técnicas de agrupación se observa que el más rápido es el k-means seguido de FCM y por ultimo RC. Sin embargo el movimiento bípedo mas naturalmente parecido al ser humano es el brindado por el controlador RC seguido de FCM y por ultimo k-means. Todo el análisis se puede observar mediante simulaciones realizados en un simulador de marcha robótica bípeda, este simulador fue implementado y contiene los cinco métodos anteriormente mencionados. Por último se comenta que esta tesis aborda una integración de tres tópicos importantes: información humana, minería de datos y robots bípedos. Por lo que se piensa que este trabajo contribuye en el estudio de estas tres temas. De manera que los resultados obtenidos brindan una alternativa para controlar la marcha de robots bípedos.

Abstract

A high percentage of humanoid robots that are deployed today try to imitate our drive system to move and interact with the environment around them. The main advantage of the use of humanoid robots is that such robots can work directly in the same environments that humans without modifications to be carried out on such an environment; contrary to what is currently happening with mobile robots with wheels. Currently, new ways to help the development and optimization of bipedal gait in robots are sought. A proposed improvement is the use of data describing human motion, because human locomotion has flexible, stable and natural features useful in bipedal robots. In this paper stable bipedal robotic motion is generated by data analysis techniques data mining. First classic drivers like PD and PID are made, which no human data are used benchmark for implementing techniques that combine the control of biped robot and human data processing using data minig . The techniques used are grouping which are clustering regression (CR), k-means and fuzzy c-means (FCM). To check its operation and ensure stability of motion (falls), experiments which are based on the soft macha, the behavior of the methods to increase speed and change of inclination of the floor were designed. The results obtained with the implementation of the techniques they use human data are better at increasing speed and robustness presented is better than classical control techniques PD and PID. Also if a comparison between the three clustering techniques performed shows that the faster the k-means followed by FCM and finally RC. But naturally the most human-like bipedalism movement is provided by the RC controller followed by FCM and finally k-means. The entire analysis can be observed by simulations performed on a simulator biped robot motion, this simulator was implemented and contains the aforementioned five methods. Finally it is said that this thesis deals with the integration of three major topics: human data, data mining and bipedal robots. It is thought that this work contributes to the study of these three themes. So the results provide an alternative to control of bipedal walking robots.

Agradecimientos

A Dios.
Por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos, además de su infinita bondad y amor.

A mis padres.
Juan Pantoja Barbosa y Antonia Laces Hernández, por mostrarme con paciencia, amor y comprensión el camino hacia la superación por medio del conocimiento.

A mis hermanos.
Juan Pantoja Laces y Briseida Pantoja Laces, por estar conmigo en mi formación académica.

A la Dra. Xiaoou Li y al Dr. Wen Yu a quienes agradezco enormemente su disposición al resolver mis dudas, su paciencia para explicar magistralmente los temas de su área de estudio, y en general, por todo el conocimiento y experiencia adquirido. A mis sinodales la Dra. Sonia Mendoza y Dr. Sergio Salazar, a quienes agradezco por el tiempo que se tomaron para revisar esta tesis.

Agradezco también al Departamento de computación del CINVESTAV por permitirme realizar estudios de maestría y brindarme una educación con excelentes profesores. Asimismo agradezco al CONACyT por el apoyo económico brindado para poder realizar dichos estudios.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Un robot humanoide se asemeja morfológicamente a un ser humano y puede realizar casi las mismas funciones que él. Un alto porcentaje de los robots humanoides que se implementan hoy en día intentan imitar nuestro sistema motriz inferior para desplazarse e interactuar con el medio que les rodea. Dentro de los robots humanoides se encuentra el robot bípedo, este robot es capaz de moverse por cualquier tipo de entorno escarpado, es decir, tiene una gran versatilidad que lo hace apropiado para aplicaciones en las que el entorno no es lo suficientemente conocidos o es cambiante, por ejemplo, la exploración de las superficies de otros planetas, exploración en entornos hostiles u operaciones de búsqueda y rescate.

El movimiento de piernas humanas comúnmente es tomado como referencia de la locomoción bípeda, ya que los seres humanos son bípedos que poseen cierta destreza comparados con seres de mayor número de miembros inferiores, esto se debe a su habilidad de caminar sobre sus pies. Por lo tanto, los robots bípedos se diseñan principalmente intentando emular las piernas humanas, cuyas características son estables, flexibles y naturales [3] [4] [5].

Debido a las características anteriormente mencionadas, se ha incrementado el uso de datos humanos para aplicarlas en robots, sin embargo, en la extracción de parámetros que describen la marcha humana se genera una gran cantidad de datos, lo cual dificulta la clasificación y el establecimiento de relaciones entre los datos. Por tal motivo, la aplicación de técnicas de minería de datos ayudaría a extraer información útil de un conjunto de datos.

Resumiendo, lo que se pretende en este trabajo es procesar los datos del movimiento de la marcha humana utilizando minería de datos, y con los resultados obtenidos generar una marcha robótica bípeda similar al caminado humano.

1.1. Motivación

A pesar de que el tema de robótica es un tema muy estudiado en la literatura, pocos autores han realizado trabajos utilizando datos de locomoción humana aplicados a robot bípedos. Además, de esos trabajos la mayoría no aplican técnicas de minería de

datos. Por tal motivo, el número de trabajos que integren minería de datos, robots bípedos e información humana es muy limitado. De manera que el estudio del tema planteado en esta tesis brindaría una contribución al campo de estudio poco abordado por otros autores.

1.2. Planteamiento del problema

Los métodos tradicionales de control como el PD o PID han sido ampliamente utilizados para controlar robots bípedos, sin embargo estas técnicas en ocasiones resultan poco practicas ya que realizan un control con poca tolerancia ante perturbaciones y datos cambiantes. Por lo que actualmente se requiere que los robots bípedos tengan características de marcha muy similares a los humanos como la estabilidad y cadencia. Por tal motivo el imitar estos movimientos resulta todo un reto de diseño e implementación.

Una opción emergente es la utilización de minería de datos para procesar datos del movimiento de la marcha humana. Planteando un posible replicado de la marcha humana con todas sus características motrices sobre un robot bípedo, siendo la estabilidad al caminar una característica elemental deseable en un robot bípedo.

Por tal motivo, se plantea el utilizar técnicas de minería de datos para brindar de inteligencia a los métodos de control para que el robot aprenda a caminar muy similar a los humanos y tenga estabilidad. Se plantea analizar la marcha normal del robot bípedo, el cambio de velocidad y la robustez ante un posible cambio de suelo en forma de rampa.

1.3. Contribuciones

Las contribuciones que se realizaron al desarrollar esta tesis se enlistan a continuación:

- Se realiza una integración de los temas de robots bípedos, datos de marcha humana y minería de datos. Esta combinación resulta beneficiosa ya que actualmente la bibliografía es poca extensa del uso de estos tres temas. Se encuentran trabajos que utilizan robots bípedos - datos de marcha humana o robots bípedos - minería de datos, sin embargo existe muy pocos trabajos que combinan los tres tópicos robots bípedos - datos de marcha humana - minería de datos. Por lo tanto con la elaboración de esta tesis se aporta conocimiento en la literatura y este puede ser utilizado por diseñadores, ingenieros o personas dedicadas al estudio de los robots bípedos

- Se implementan controladores clásicos basados en PD y PID , los cuales se utilizan como puntos de comparación con las técnicas inteligentes que utilizan minería de datos.

- Se implementan técnicas de minería de datos los cuales procesan datos de la marcha humana para implementarlos dentro de un robot bípedo.

- Se realizó un simulador de marcha de robots bípedos, esta aplicación de escritorio esta desarrollado en MatLab utilizando herramientas de agrupamiento, simulink y el creador de interfaces gráficas. Dentro de este programa se pueden realizar simulaciones de los controladores PD y PID, igualmente en los controladores basados en regression clustering, K-means y Fuzzy c-means. En las opciones de simulación se pueden configurar el ángulo de inclinacion y la velocidad.

1.4. Estructura de la tesis

De igual modo, el resto del documento de la tesis está organizado en capítulos como se describe a continuación:

El capítulo 2 muestra el trabajo relacionado acerca de los distintos modelos de marcha y algunos trabajos mas representativos de control clásico y de control usando técnicas inteligentes.

El capítulo 3 muestra la utilización de dos controladores clásicos PD y PID para controlar un modelo de robot bípedo. También se presenta los resultados en marcha normal, aumento de velocidad y ante suelo tipo rampa.

En el capítulo 4 se presenta una solución de control utilizando el algoritmo *regression clustering*, se presenta los conceptos teóricos, después se muestra el diseño y la implementación orientado al modelo robótico y por ultimo reportan los resultados en base a una marcha normal, cambio de velocidad y comportamiento ante una pendiente.

En el capítulo 5 se aborda el uso de dos algoritmos: Kmeans y Fuzzy cmeans. Este capitulo se divide en dos partes, cada parte tiene la descripción del algoritmo, el diseño e implementación de los algoritmos orientándolos al control del robot bípedo y por último se presentan los resultados como se realiza en el capitulo 4.

En el capítulo 6 se describe un software en el cual se puede simular la marcha bípeda de los robots y se comenta las opciones que tiene para configurar la marcha.

Por último en el capítulo 7 se comentan las conclusiones acerca de la tesis y el trabajo futuro.

Capítulo 2

Marcha de robots bípedos

En este capítulo se revisará trabajos relacionados con el tema de la tesis, una vez revisados se establecerá los conocimientos teóricos necesarios para el desarrollo de las técnicas implementadas. En este capítulo se describen trabajos representativos a cada modelo y se complementa con algunos trabajos que utilizan datos humanos para el funcionamiento de sus modelos; asimismo se mencionan trabajos que utilizan técnicas de minería de datos dentro de junto con robots bípedos.

2.1. Modelos de robot bípedos

Existe un gran variedad de modelos robóticos de marcha estable, según la complejidad de sus mecanismos. Entre los modelos más mencionados en la literatura son los que se basan en la rueda sin montura, modelo de compás y basado con rodillas.

En 1990, McGeer introdujó el concepto de caminado dinámico pasivo *(Passive Dynamic Walking, PDW)* [8]. La idea básica detrás de este tipo de caminado dinámico pasivo es que la pérdida de energía en las colisiones inelásticas se equilibra exactamente por la energía cinética adquirida bajando por una rampa. McGeer demostró estas ideas mediante la construcción de un robot real que podía caminar de forma estable por una rampa utilizando peldaños para evitar el rayado pie. También en 1990 McGeer describe una dinámica andador pasiva con las rodillas [9], esta dinamica es aplicada al modelo mostrado en la figura 2.1.

2.1.1. Modelo de marcha con rueda sin montura

El modelo matemático más simple del caminado pasivo es la rueda sin montura. La rueda sin montura es rígida, tiene radios sin masa y un único punto en la cadera. Las colisiones de los pies son modelados para ser perfectamente inelásticas, mientras conservan el momento angular. Además, se supone que no hay deslizamiento en el pie de apoyo y que esta se conmuta instantáneamente. El sistema exhibe un funcionamiento basado en movimientos periódicos en diferentes velocidades, además brinda una determinada longitud y ángulo de la pendiente de las piernas.

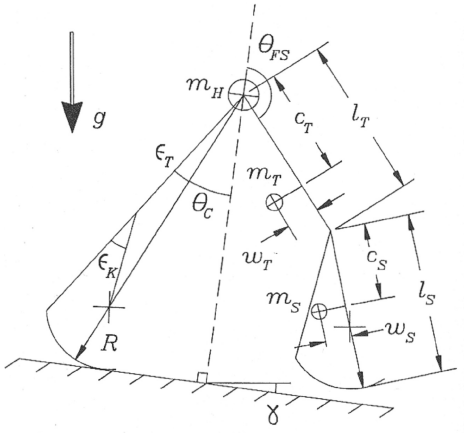

Figura 2.1: Modelo con rodillas propuesto por McGeer.

Con las hipótesis formuladas se calculan analíticamente los valores propios del ciclo límite, y se prueba la estabilidad del sistema. La dinámica y la estabilidad no lineal de la rueda sin montura fueron estudiadas ampliamente en Coleman1997 [10] y Coleman2010 [11] .

Actualmente, se encuentran trabajos en donde se introducen mejoras al diseño de la rueda sin montura. Asano2013 [12] y Tanaka2012 [13] estudian el efecto de oscilación de masa para mejorar la eficiencia de este tipo de marcha. En la Figura 2.2 se muestran los modelos que se utilizan en cada trabajo.

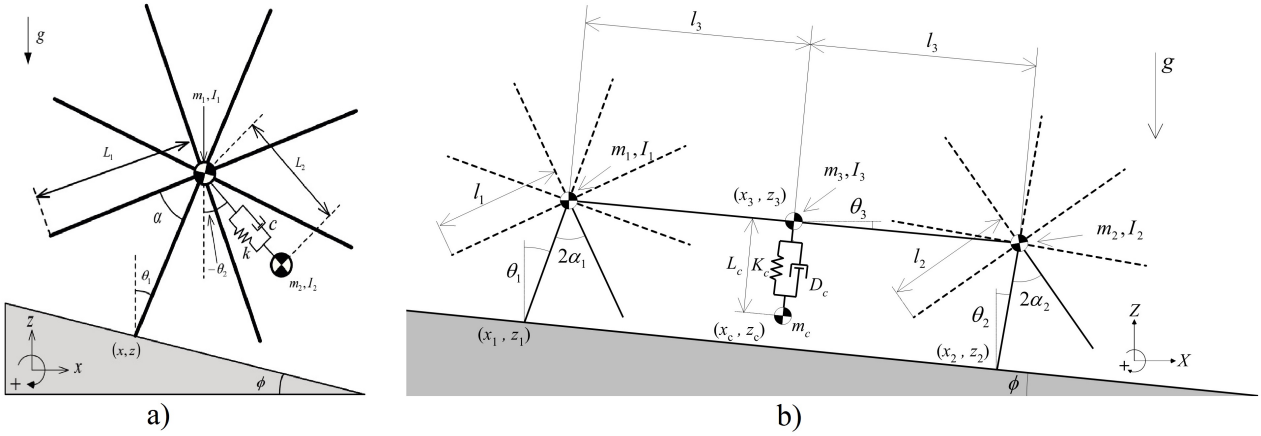

Figura 2.2: a) Rueda con amortiguador de Asano2013 y b)Doble rueda con amortiguamiento propuesto por Tanaka2012

Además se buscan nuevas formas para el mejoramiento de las ruedas sin monturas, por lo que se han realizado distintos trabajos que combinan distintos modelos y distintas técnicas; en Harata2012 [1] se presenta un trabajo el cual propone que en

cada una de los radios de la rueda se anexa una rodilla (ver Figura 2.3); en Asano2012 [2] se propone un diseño híbrido utilizando visco-elasticidad en cada radio de la rueda (ver Figura 2.4) y se reporta una mejora de la estabilidad y la eficiencia de la marcha.

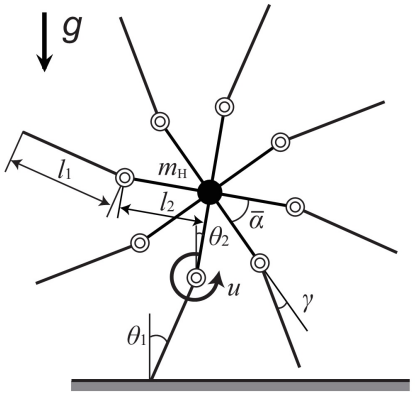

Figura 2.3: Modelo de rueda sin montura y rodillas utilizado en [1].

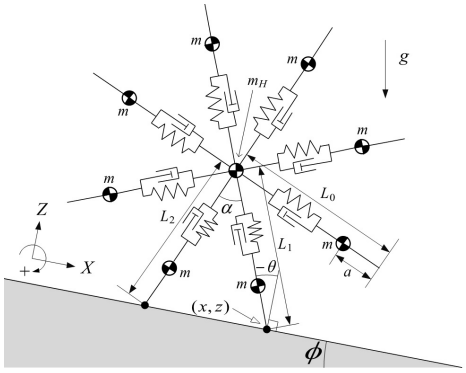

Figura 2.4: Modelo de rueda sin montura con visco-elasticidad utilizada en [2].

2.1.2. Modelo de marcha en compás

Si se toma sólo dos piernas (radios) de la rueda sin montura y se coloca una junta de pasador entre ellos, permitiendo a las piernas oscilar libremente, se obtiene el modelo de marcha en forma de compás. Las piernas se modelan como masas puntuales en posición del centro de masa; además estas masas son responsables de mover la pierna hacia delante. Los talones se modelan como colisiones inelásticas con conservación del momento angular. La postura y la oscilación de las piernas cambian instantáneamente durante la colisión generada en el siguiente paso.

Este modelo, a pesar de que se ha estudiado ampliamente Goswami1996 [14], tiene una complejidad grande al resolverlo analíticamente. Se realizan simulaciones que muestran un ciclo estable de marcha, aunque la predicción de su final es difícil

establecer. Otra característica interesante es la de mantener el ciclo de marcha indefinidamente, si las condiciones de estabilidad son idóneas.

Un caso especial de la marcha en forma de compás es el caminante más simple (*simplest walker*). Este modelo especial coloca masas en las piernas y las hace infinitamente pequeñas Garcia1998 [15],Schwab2001 [16] y Kuo2002 [17]. Estas simplificaciones adicionales permiten caracterizar al modelo con un solo parámetro, la pendiente de la rampa. El caminante simple posee una dinámica controlada por un actuador en la segunda articulación (cadera).

Recientemente se busca adicionar mejoras a este modelo de marcha. Por lo cual se proponen novedosas formas como la que propone Visser2012 [18], cuyo modelo de resorte-masa con rigidez variable en la pierna pretende imitar la capacidad humana para adaptarse continuamente para superar las perturbaciones (ver Figura 2.5).

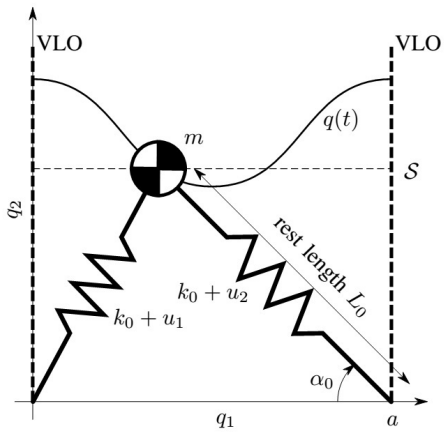

Figura 2.5: Modelo bípedo. El modelo tiene piernas con rigidez controlable. Cuando la trayectoria de la cadera $q(t)$ cruza S el sistema conmuta entre el apoyo individual y doble.

Asimismo, en Huang2012 [19] propone un modelo basado en resortes en las articulaciones del tobillo y dedos (ver Figura 2.6) para lograr una marcha estable en una pendiente impulsado por la gravedad.

2.1.3. Modelo de marcha con rodillas

A medida que se aumenta la complejidad del modo de caminar, se evoluciona del modelo de marcha en forma de compás al modelo con rodillas. La adición de rodillas tiene algunos beneficios como el logro de despeje del pie y un modo de caminar con más similitud al caminado humano. Por otra parte, las rodillas podrían evitar el arrastre del pie y ayudar a que los robots bípedos sean capaces de desplazarse en terreno complicados.

En Mochon1980 [20] y [21] muestran el diseño de andador balístico (ver Figura 2.7) el cual es un modelo matemático de balanceo de fase de marcha. La pierna de apoyo se representa como un péndulo fijo en su origen; también se supone que la rodilla no

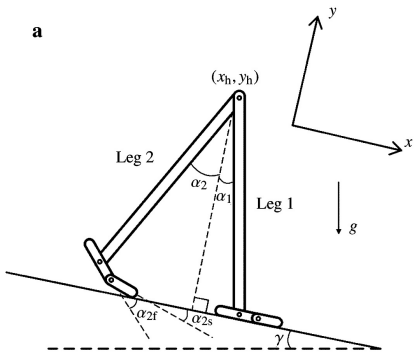

Figura 2.6: El modelo de dos dimensiones específica que consta de dos patas rígidas interconectadas individualmente a través de una bisagra. Cada pata contiene un pie segmentado.

se dobla como en la otra pierna y se balancea hacia adelante. Este movimiento se produce completamente de forma pasiva. Dada una configuración inicial apropiada, el movimiento hacia adelante y la flexión de la pierna oscilante se debe a la gravedad. Este modelo estudia el comportamiento en un solo paso de y no modela los impactos de la rodilla o el pie. Por lo tanto, el caminante balístico no viola el tema de la estabilidad en absoluto. Sin embargo, se caracteriza por un ciclo de un solo paso.

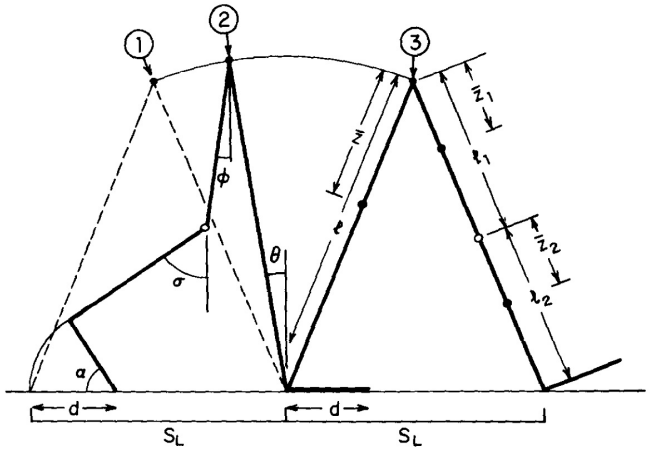

Figura 2.7: Representación esquemática del modelo.Los números (1), (2) y (3) representan la posición del modelo en el golpe de talón, posicionamiento del talón hasta que los dedos llegan, y tras el golpe de talón.

Actualmente se realizan trabajos de investigación, algunos de ellos toman en cuenta las formas de caminata, longitudes de paso, velocidad, eficiencia y estabilidad como el trabajo presentado en Huang2010 [22] y en este trabajo se utiliza un modelo robótico representado en la Figura 2.8.

También, los robots bípedos son estudiados con base en su comportamiento en distintas superficies, planas, subidas y bajadas de pendiente y escaleras, entre otras

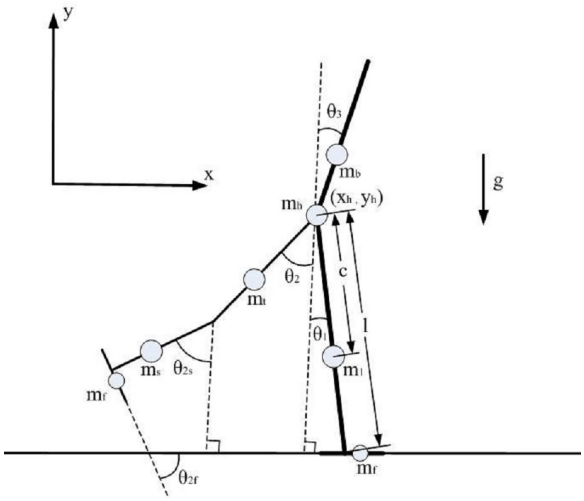

Figura 2.8: Modelo del andador dinámico con los pies planos y tobillos conformes.

variaciones de superficie irregular. Algunos trabajos realizan un diseño y análisis de los robots bípedos teniendo en cuenta el desarrollo sobre terreno plano como los presentados en Kooij2003 [3],Azevedo2003 [4],David2011 [5](ver Figura 2.9).

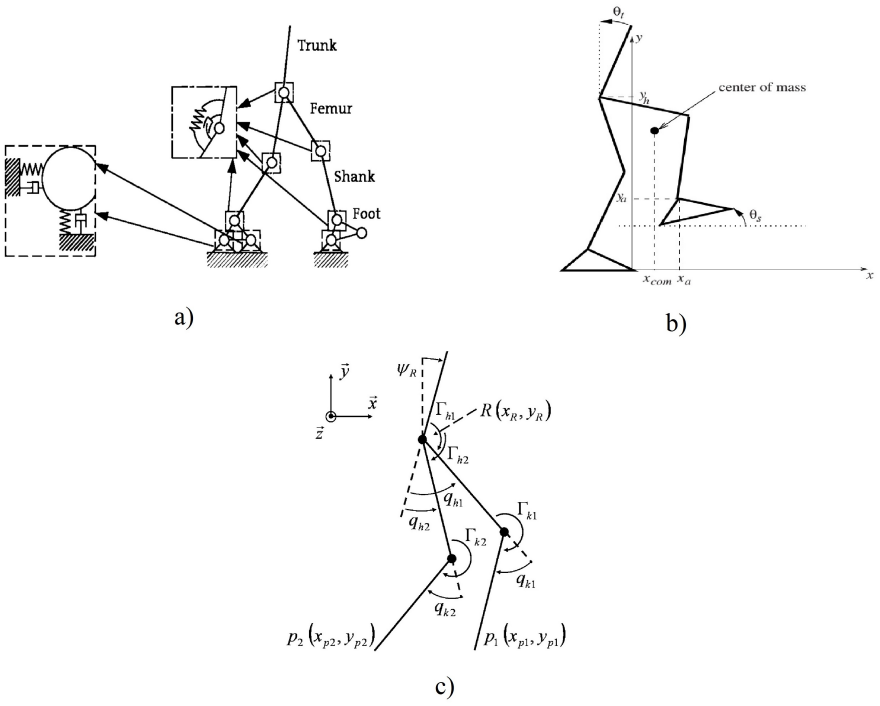

Figura 2.9: Modelos de robot bípedo sobre terreno plano a)[3] b)[4] c)[5]

2.2. Marcha estable en robots bípedos

Ante los beneficios de estabilidad y flexibilidad de la naturaleza de la marcha bípeda humana se han propuesto diversos trabajos con el fin de replicar el movimiento de marcha. Se revisaron trabajos que exponen metodologías que utilizan información de movimiento de piernas humanas en el contexto de la marcha bípeda.

En Ames2012 [23] se presenta los primeros pasos para ir a partir de datos humanos al diseño formal de un controlador experimental en el contexto de robots bípedos mediante el estudio de los datos de marcha humana y que de los cuales se extraen funciones de cinemática, éstas parecen ser canónicas a la marcha y se caracterizan por una sola función del tiempo denominadas como funciones de la marcha humana. Usando estas funciones se diseñó un controlador humano inspirado al comportamiento de la marcha humana. El resultado principal del trabajo es un problema de optimización que determina los parámetros de este controlador a fin de garantizar un comportamiento lo mas " cerca .ª la humana.

Posteriormente en Ames2014a [24] se presenta un enfoque del control bípedo robótico inspirado en la marcha humana con la utilización de los datos humanos. Se utilizan funciones de salida que parecen ser intrínsecas a la marcha humana con el fin de diseñar formalmente controladores que proporcionan un comportamiento de una marcha robótica estable. Con base a las funciones, los parámetros de este controlador puede ser determinada a través de un problema de optimización humano-inspirado que proporciona el mejor ajuste de los datos.

Similarmente en Jiang2012 [25] se presenta un método para determinar las salidas asociadas a los datos de la marcha humana que se pueden utilizar para diseñar controladores que logran caminar robótica similar a la humana. Se presenta un problema de optimización para determinar los parámetros de este controlador que produce el mejor ajuste a los datos humanos que produce simultáneamente una marcha robótica estable. El valor optimo de la función de costo se utiliza como una métrica para determinar la similaridad robótica respecto a la humana. Por ultimo tomando el enfoque anterior de tener funciones canónicas de marcha, en Powell2012 [26] se presenta un enfoque para el control de robot bípedo para múltiples comportamientos de locomoción. El comportamiento fundamental de la locomoción humana se obtiene a través del examen de datos humanos experimental para caminar sobre terreno plano, en escalera ascendente y descendente. Se demuestra que ciertos movimientos de la marcha humana independiente de terreno locomoción, se pueden caracterizar por una sola función, denominada función canónica extendida.

Las similitudes que comparten los trabajos anteriormente referenciados son el uso de técnicas de optimización que determina los parámetros de controladores bioinspirados a fin de tener un comportamiento similar a la marcha humana. Los datos humanos sirven como referencia para el mejor ajuste de dichos controladores.

Las diferencias entre los trabajos antes mencionados es en primera instancia el terreno en donde los robots bípedos caminan, siendo Ames2012 [23],Jiang2012 [25],Ames2014a [24] los que estudian el comportamiento robótico sobre terreno plano.

Por su parte Powell2012 [26] presenta un enfoque para el control de robot bípedo para múltiples comportamientos de locomoción. El comportamiento fundamental de la locomoción humana se obtiene a través del examen experimental de datos humanos para caminar sobre terreno plano, en escalera ascendente y descendente. Se demuestra que ciertos movimientos de la marcha humana independiente de terreno locomoción, se pueden caracterizar por una sola función, denominada función canónica extendida (ver figura 2.10).

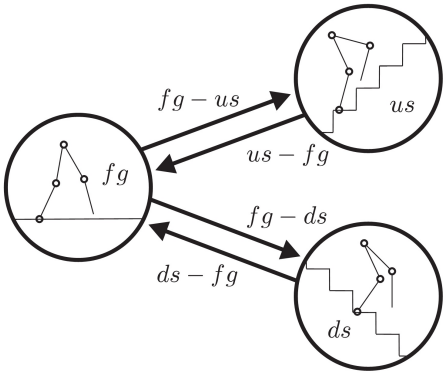

Figura 2.10: Modelo que combina trayectorias planas, subidas y bajadas.

Otra diferencia notoria de Powell2012 [26] es el uso de una segunda optimización para permitir que exista un comportamiento evolutivo entre la transición de cada modo de locomoción (terreno plano, subida y bajada).

Estos trabajos reportan novedosas formas de mejorar la estabilidad de la marcha bípeda, en Aoi2011 [27] presenta un control de un sistema de locomoción para un robot bípedo usando osciladores no lineales y se verifica el rendimiento del sistema con el fin de establecer pasos de adaptación a través de las interacciones entre la dinámica del robot, la dinámica del oscilador y el medio ambiente. Se llevan a cabo simulaciones numéricas y un análisis de estabilidad, donde analíticamente se obtienen soluciones periódicas aproximados y se examina la estabilidad local utilizando un mapa de Poincaré, en la Figura 2.11 se muestra el modelo utilizado en este trabajo.

Otra técnica interesante es la llamada CMAC Neural Networks utilizada. Esta técnica es utilizada en Sabourin2005 [28], donde la red neuronal permite aumentar la robustez de la estrategia de control. En la primera etapa del proyecto, un conjunto de trayectorias articulares de la pierna en movimiento son aprendidos por la técnica CMAC. Y en la segunda etapa se utilizan estas redes neuronales para generar las trayectorias. En la figura 2.12 se muestra el diagrama a bloques del sistema con controladores PD (*Proporcional-Derivativo*). Por último, Bououden2009 [29] propone el uso del control de par computarizado *(Computed Torque Control, CTC)* debido a que puede asegurar la estabilidad global asintótica del caminado del robot bípedo. También se utiliza un torque computarizado para lograr una alta velocidad y precisión de seguimiento, mientras que un controlador difuso sirve para corregir cualquier desvia-

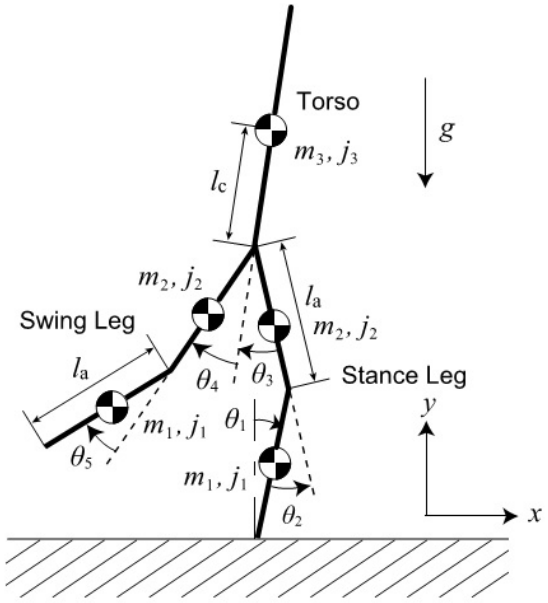

Figura 2.11: Diagrama esquemático de un robot de 5-link

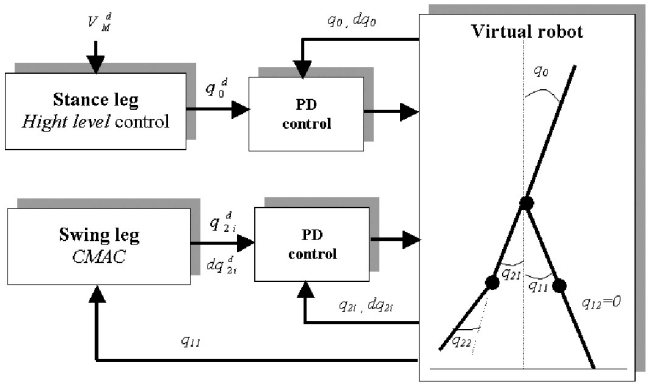

Figura 2.12: Principio de control usando redes neuronales CMAC

ción y perturbación desconocida, en la Figura 2.13 muestra la estructura mecánica del robot bípedo utilizado en este trabajo.

2.2.1. Minería de datos aplicada a robots bípedos

Ante el incremento de trabajos que utilizan datos humanos, resulta conveniente hacer uso de la minería de datos debido a que, en los experimentos realizados para extraer información de locomoción humana, se puede llegar a obtener una gran cantidad de datos.

El movimiento humano no sólo contiene una gran cantidad de información sobre las acciones e intenciones, sino también acerca de la identidad y los atributos personales

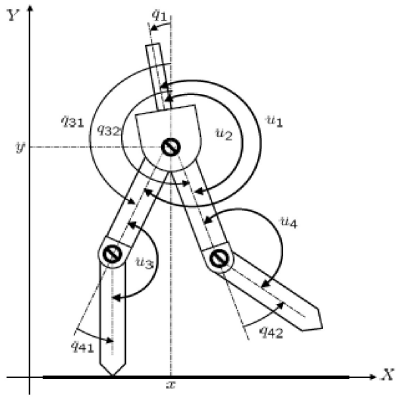

Figura 2.13: Estructura mecánica del bípedo

de la persona que se mueve. El enfoque se basa en un análisis del movimiento biológico de marcha humana. Se busca transformar estos movimientos en una serie de datos para su posterior procesamiento con técnicas de minería de datos.

Al trascurrir el tiempo se han realizado diversos trabajos que utilizan minería de datos en robótica. Uno de los trabajos más representativos en el campo de los robots móviles es Gunderson2003 [30], debido a que es uno de los primeros en donde se utiliza minería de datos para descubrir las preferencias del humano en su entorno. El método tiene aplicabilidad en una amplia gama de tareas robóticas y permite al ser humano hacer que el robot responda a las necesidades y preferencias personales. Las técnicas de la minería de datos provienen de la inteligencia artificial y de la estadística. Dichas técnicas no son más que algoritmos, más o menos sofisticados que se aplican sobre un conjunto de datos para obtener resultados, de modo que se abordarán tres enfoques de aprendizaje muy recurrentes en minería de datos: Supervisado, Refuerzo y No supervisado.

2.2.1.1. Enfoques de aprendizaje supervisado (Supervised Learning)

En *Supervised Learning (SL)* los métodos aprenden a realizar una tarea con la ayuda de un maestro, el cual proporciona la información de entrada-salida para entrenar a un sistema de control. Un agente SL actualiza el control de los parámetros para minimizar la diferencia entre los resultados deseados y reales de un sistema. Se discuten los enfoques de aprendizaje en el control de caminar bípeda a continuación.

- **Método basado en redes neuronales *back-propagation***
  En Wang1992 [31] se realiza un entrenamiento de un perceptrón multicapa *(Multi Layer Perceptron, MLP)* para aprender el comportamiento de un controlador prediseñado para un robot bípedo de 3-*link* a través de una red *backpropagation* estándar (BP). Este controlador neuronal proporciona un rendimiento superior contra perturbaciones, debido a la generalización de la red neuronal *(Neural Network,NN)*.

También en Juang1996 [32] se aplicó un MLP de tres capas para controlar un robot bípedo simulado de 5-*link*. Se empleó el algoritmo *backpropagation* a través del tiempo para entrenar un controlador neuronal, con el objetivo de hacer que el robot bípedo siga un conjunto de trayectorias de referencia en la cadera y el o el *swing* columpiado de la pierna. Después del entrenamiento, el robot bípedo fue capaz de caminar de forma estable sobre una superficie plana.

- **Método de aprendizaje ponderado *(Locally Weighted Learning)***

  Los métodos Locally Weighted Learning (LWL) ofrecen una estructura más comprensible para aprender políticas de control no lineal complejas. Los enfoques LWL han logrado un impresionante éxito en control en tiempo real robot humanoide. Algunos trabajos que utilizan este metodo de paredizaje son:

  Nakanishi2004 [33] aplica LWL para entrenar a un robot bípedo 5-*link*, con el objetivo de imitar las trayectorias de caminata humana. Las trayectorias del robot están representados por un aproximador de una función no lineal, utilizando modelos locales lineales. A través de la sintonización de los parámetros de los modelos locales, el método LWL permitió al robot bípedo caminar de forma estable sobre una superficie plana.

  Por otro lado, Loken2006 [34] aplica LWPR a dos robots bípedos con 3-*link* y 5-*link*, respectivamente. Se utilizó LWPR para representar una función eficiente que construye una regresión lineal local de un control no adaptativo. Las políticas locales de control estructuradas permitieron a los robots bípedos seguir la referencia a los movimientos humanos para caminar sobre una superficie plana rápidamente.

- **Método de máquinas de soporte vectorial *(Support Vector Machine)***
  Las técnicas de *Support Vector Machine (SVM)* proporcionan potentes herramientas para el aprendizaje de clasificación y modelos de regresión en problemas de alta dimensión. Un sistema bípedo de control a menudo tiene alta dimensión en retroalimentación de las señales sensoriales.Las SVM pueden ser aplicados a para clasificar las señales de retroalimentación y a la categorización de señales de entrada de referencia.

  Un trabajo que utiliza SVM es Kim2008 [35]. Éste aplica SVM para detectar la caída de un robot bípedo a partir de datos de un acelerómetro y un sensor de fuerza. Por su parte Ferreira2013 [36] propuso una estrategia de control basada en *Zero Moment Point (ZMP)* del balanceo del caminado con soporte de regresión vectorial *Regression Vector Support (RVS)*. El controlador basado en ZMP fue diseñado basado en un modelo de robot simulado. Cuando se implementa en el robot bípedo real. SVR calculó la corrección del torso del robot basándose en las posiciones reales ZMP y sus variaciones a las posiciones ZMP deseadas.

- **Método de árbol de decisión**

  Métodos de árboles decisión tienen también han propuesto para hacer frente a los problemas de control bípedo adaptativo bajo diferentes condiciones ambientales. En algunos trabajos como [37] se ha diseñado un sistema de control basado en árbol de decisión. La estrategia de control adaptativo basado en árboles se basa en activar un robot bípedo para hacer frente a varias superficies para caminar con diferente elasticidad y coeficientes de fricción viscosa. Una vez que se ha obtenido un árbol de decisión, el robot será capaz de seleccionar las acciones de control apropiadas cuando caminó sobre diferentes tipos de los terrenos.

2.2.1.2. Enfoques de Aprendizaje por refuerzo (Reinforcement Learning, RL)

El aprendizaje por refuerzo *(Reinforcement Learning, RL)* es una opción poderosa para implementar un control de robot bípedo, ya que un agente de aprendizaje no se le dice qué acción debe tomar; sino qué debe descubrir a través de interacciones con el sistema y su entorno, lo cual produce la acción con más alta recompensa. En lo que sigue, los métodos de RL más populares para el control del robot bípedo se presentan a continuación:

- **Perceptrón multicapa**
  RL ha sido ampliamente utilizado para entrenar un MLP para robot bípedo andante. En Salatian1997 [38] se aplicó RL para entrenar a un controlador de MLP para un robot bípedo de 8 grados de libertad. El sistema de control fue diseñado para mantener el COP *(Center of Pressure)* del robot dentro de la zona de apoyo del pie durante la marcha. Las señales de la fuerza del pie se utilizaron para calcular la posición de la COP. Un MLP fue entrenado por RL para mapear la relación entre las fuerzas del pie y el ajuste de posiciones de la articulación del tobillo.

- **Oscilador Neuronal**
  Los osciladores neuronales se han convertido en un foco de interés en el control de la caminata bípeda en los últimos años. El método más popular se llama *Central Pattern Generators (CPG)* . Los osciladores neuronales con un peso adecuado son capaces de generar diferentes tipos de patrones estables para caminar. Algunos trabajos que muestran la utilización de osciladores neuronales son por ejemplo: Mori2004 [39] presenta un actor crítico basado en un controlador CPG. Una combinación de dos neuronas primarias y dos neuronas suplementarias conformando un oscilador neuronal. Cada oscilador neuronal es responsable de controlar una articulación del robot. Otro trabajo interesante para comentar es Endo2008 [40], éste reporta una combinación de un controlador basado en CPG con una máquina de estados. Esta máquina de estados controla las articulaciones de la rodilla de acuerdo con las cuatro transicines de

estados definidas por los ángulos de las articulaciones de la cadera y la colocación de los pies. El controlador de aprendizaje basado en el CPG pudo para adquirir una política de control apropiada después de unos pocos cientos de ensayos simulados.

2.2.1.3. Aprendizaje No supervisado (Unsupervised Learning,UL)

El aprendizaje no supervisado *Unsupervised Learning (UL)* no necesita un "maestro.º cualquier información evaluativa para adquirir una política de control. En su lugar, construye estructuras de subrayado o redes asociativas para la entrada datos. Para el control del robot bípedo, hay dos enfoques principales de UL en la literatura: métodos de agrupamiento y el aprendizaje *Hebbian*.

- **La agrupación *(Clustering)***
  La agrupación es un campo de investigación muy activa. Por lo general, no se utiliza para aprender directamente las políticas de control; en lugar de ello juega un papel en el análisis y la reducción de datos en bruto. Por ejemplo, hemos mencionado que los controladores neuronales basadas en CMAC tienen cómputo rápido pero requieren grandes cantidades de memoria.

  Un ejemplo del uso de agrupamiento es el presentado en hu1999 [41], este trabajo aplica una técnica de agrupación en un sistema de caminar bípeda para reducir los requisitos de memoria de un controlador de aprendizaje basado en el CMAC.

- **Diferencial de aprendizaje de Hebb**
  El aprendizaje de Hebbian sin supervisión ha sido estudiado para el control de robot bípedo mostrado en Porr2003A [42]. Ellos han desarrollado una versión modificada de aprendizaje de Hebb clásica llamada aprendizaje de Hebb diferencial que es aplicable a sistemas con retroalimentación.

  Una aplicación de aprendizaje de Hebb diferencial con propiedades de control del robot bípedo fue realizado por manoonpong2007 [43] y manoonpong2009 [44]. Estos trabajos reportan el diseño de un control neuronal adaptativo sistema para un verdadero robot bípedo llamado RunBot, este robot tiene cuatro articulaciones activas de la pierna (izquierda / derecha, caderas y rodillas) y un componente de límite superior del cuerpo que se puede mover hacia atrás.

2.3. Conclusiones

En este capitulo se abordó los trabajos relacionados al tema de tesis. Se comenzó describiendo los tipos de modelos robóticos bípedos: rueda , compás y rodillas, se presentaron algunos trabajos donde utilizan estos modelos y con base al conocimiento de los modelos, se decidió en este trabajo de tesis utilizar el modelo robótico bípedo con rodillas. Posteriormente se abordaron algunos trabajos que utilizan el modelo robótico con rodillas utilizando control basado en CMAC junto controladores

PD, mapas de poincaré, entre otras, las cuales son técnicas que utilizan gran análisis matemático. Posteriormente se presentan trabajos que utilizan cambio de suelo en escaleras y rampas. Finalmente se presentan trabajos que implementan métodos de minería de datos para controlar robots bípedos.

Capítulo 3

Control de marcha estable en robots bípedos

En este capitulo se abordará la implementación de controladores clásicos como el controlador PD y PID. Estos controladores son de los más utilizados en aplicaciones de control debido a su rápida implementación aunque se necesite la sintonización optima de las ganancias, lo cual podría aumentar la complejidad en gran medida.

Se han realizado experimentos para conocer el funcionamiento de estos controladores, para reportar resultados se tiene en cuenta la marcha suave, aumento de velocidad y robustez ante cambio de inclinacion del suelo, a continuación se describirá en que consiste cada uno de estos tópicos.

3.1. Tópicos de evaluación para resultados

Para realizar una evaluación de los experimentos se propone evaluar la marcha suave, el aumento de velocidad y robustez (cambio de inclinación de suelo). **Marcha suave**
La marcha suave o normal es un modo de locomoción con actividad alternada de los miembros inferiores, la cual se caracteriza por una sucesión de doble apoyo y de apoyo en un solo pie, es decir que durante la marcha el apoyo no deja nunca el suelo, mientras que en la carrera, como en el salto, existen fases aéreas, en las que el cuerpo queda suspendido durante un instante. Para un robot bípedo la marcha normal es el funcionamiento primordial, ya que con este movimiento se demuestra la estabilidad del robot al no caerse en algún paso de la marcha. También es la fase básica antes de empezar a aumentar la velocidad y cambiar la forma/pendiente del suelo.

Velocidad
Una característica para analizar en un robot bípedo es la velocidad con la que se desplaza en un terreno determinado. Para que un robot adquiera una velocidad constante se requiere que tenga estabilidad, de manera que el robot tendrá más rapidez en un terreno plano que en una pendiente ascendente o descendente. La estabilidad del robot bípedo es importante ya que a mayor velocidad el tiempo que el robot esta

en el aire es mayor. Por lo que el reto es obtener una mayor velocidad posible sin perder estabilidad, lo cual podría causar que el robot sufra una caída. Un ejemplo es MABEL, este robot ha alcanzado una velocidad de hasta 11 kilómetros por hora. Este robot comenzó se construyo en 2008 en la Universidad de Michigan. Este robot intenta reproducir movimientos humanos en los robots es algo muy complicado, y el hecho de hacer correr a un robot ha sido un gran avance. MABEL distribuye su peso similar a la forma en la que lo hace un humano, cuenta con un torso pesado y unas piernas livianas y ágiles. Por lo cual MABEL tiene sus pies 40 % del tiempo en el aire.

Robustez

Para que un robot sea robusto tiene que reaccionar apropiadamente ante situaciones de cambio de pendiente del terreno.

Existen trabajos los cuales reportan caminata a través de planos, escaleras o terrenos irregulares en pendientes ascendentes o descendentes. Por otro lado, algunos enfocan su estudio hacia la marcha en planos inclinados con determinada pendiente como [6] y [7](ver Figura 3.1), sin embargo, estos no reportan adaptabilidad dinámica ante el cambio de terreno.

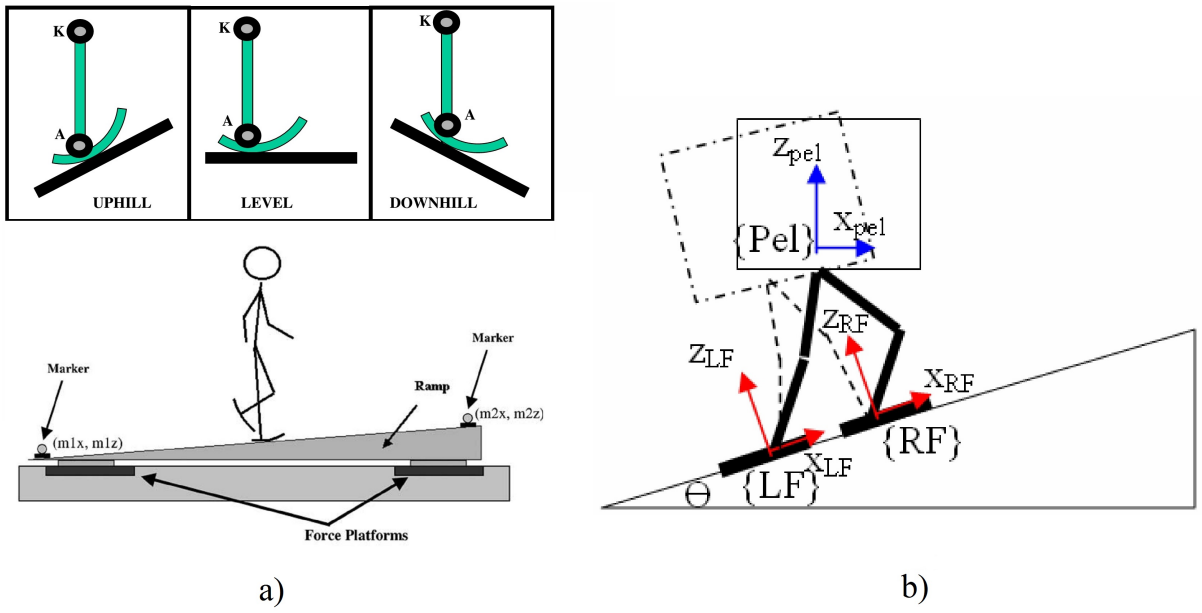

Figura 3.1: Modelos utilizados en a) Postura de tobillo y marcha en plano inclinado [6] b) Marcha sobre plano inclinado [7].

Por su parte, Seven2012 [45] considera un cambio de terreno y presenta un conjunto de técnicas difusas, las cuales se demuestra el rendimiento del algoritmo de control "Momento en Punto Cero " en diversas condiciones de pendiente. En la figura 3.2 se muestra el modelo utilizado.

Como se expone anteriormente, el terreno en donde se desenvuelve un robot bípedo es un factor que afecta el comportamiento de su marcha. Por lo que la estabilidad

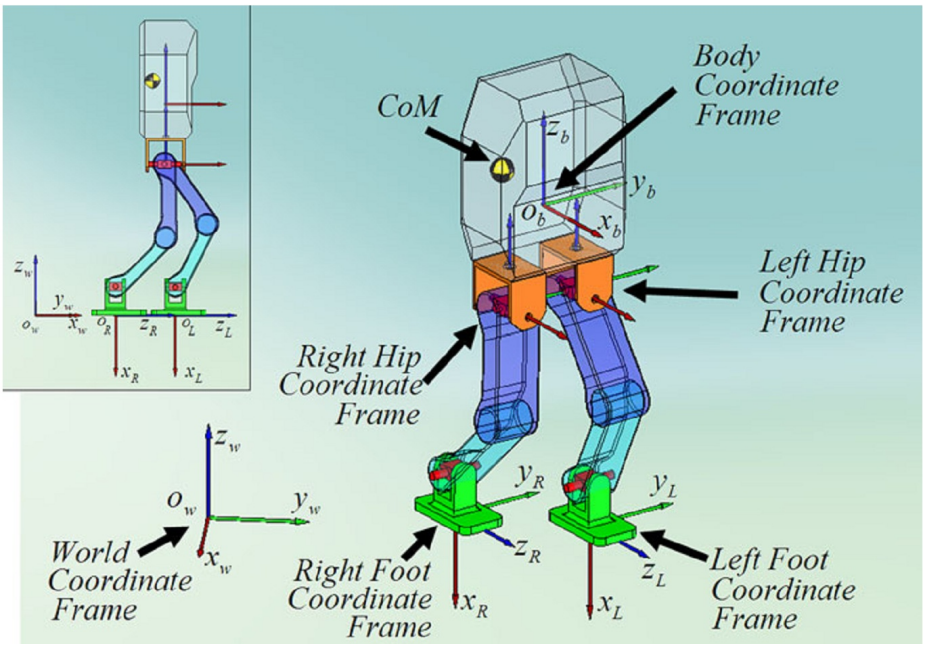

Figura 3.2: Sistemas de coordenadas asociados con la generación de referencia al caminar. *ow* y *ob* representan el origen del mundo y el origen en el marco del cuerpo,respectivamente.

del robot es una característica que se pretende mejorar y es objetivo de diversos trabajos. En esta tesis se analizará la robustez en terreno con pendiente ascendente y descendente, dependerá en gran medida si los controladores implementados soportan estos tipos de variaciones. Una vez establecidos los tópicos anteriores procede a describir el modelo robótico utilizado en esta tesis.

3.2. Modelo del robot bípedo

El modelo bípedo consiste en cinco enlaces que están conectadas sin fricción y es usado en Bigdeli2008 [46], Onn2015160 [47],Haavisto2005 [48]. El sistema bípedo se mueve libremente en el plano X-Y en cinco grados de libertad. Las piernas son idénticas con rodillas y un cuerpo rígido en forma de torso, ver figura 3.3 .

Las coordenadas seleccionadas son:

$$q = [x_0, y_0, \alpha, \beta_L, \beta_R, \gamma_L, \gamma_R] \tag{3.1}$$

Las coordenadas (X_0 y Y_0) representa la posición del centro de masa del torso y el resto de las coordenadas describe los ángulos de las articulaciones. La longitud de cada segmento son denotados (l_0, l_1, l_2) y las masas son denotadas por (m_0, m_1, m_2). Los centros de masa están localizados en las distancias (r_1, r_2, r_3) de las correspondientes articulaciones. Por otra parte, las fuerzas generalizadas correspondientes a las coordenadas están descritas como:

$$F_q = [F_{x0}, F_{y0}, F_\alpha, F_{\beta L}, F_{\beta R}, F_{\gamma L}, F_{\gamma R}]^T \tag{3.2}$$

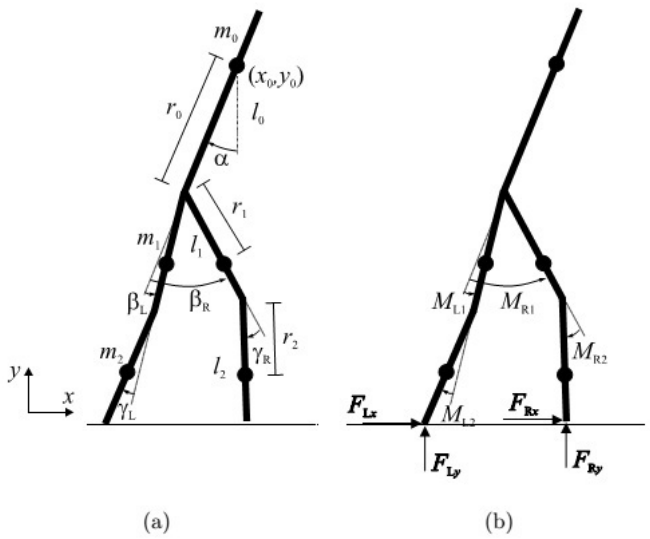

Figura 3.3: Modelo robótico. Coordenadas usadas (a) y fuerzas externas (b)

Las posiciones del centro de masa de los muslos en coordenadas cartesianas se denomina: x_{L1}, y_{L1} y x_{R1}, y_{R1}. Las posiciones del centro de masa de la espinilla se denomina: x_{L2}, y_{L2} y x_{R2}, y_{R2}. Las posiciones de punta de la pierna en coordenadas cartesianas:x_{LG}, y_{LG} y x_{RG}, y_{RG}. Los parámetros anteriores se pueden expresarse usando las coordenadas generalizadas (correspondiente a la pierna derecha):

$$x_{L1} = x_0 - r_0 \sin\alpha - r_1 \sin(\alpha - \beta_L) \tag{3.3}$$

$$y_{L1} = x_0 - r_0 \cos\alpha - r_1 \cos(\alpha - \beta_L) \tag{3.4}$$

$$x_{L2} = x_0 - r_0 \sin\alpha - l_1 \sin(\alpha - \beta_L) - r_2 \sin(\alpha - \beta_L + \gamma_L) \tag{3.5}$$

$$y_{L2} = x_0 - r_0 \cos\alpha - l_1 \cos(\alpha - \beta_L) - r_2 \cos(\alpha - \beta_L + \gamma_L) \tag{3.6}$$

$$x_{LG} = x_0 - r_0 \sin\alpha - l_1 \sin(\alpha - \beta_L) - l_2 \sin(\alpha - \beta_L + \gamma_L) \tag{3.7}$$

$$y_{LG} = x_0 - r_0 \cos\alpha - l_1 \cos(\alpha - \beta_L) - l_2 \cos(\alpha - \beta_L + \gamma_L) \tag{3.8}$$

Por otra parte, la energía traslacional del sistema robótico bípedo se denota como

$$T_t = \frac{1}{2}(m_0(\dot{x}_0^2 + \dot{y}_0^2) + m_1(\dot{x}_{L1}^2 + \dot{y}_{L1}^2 + \dot{x}_{R1}^2 + \dot{y}_{R1}^2) + m_2(\dot{x}_{L2}^2 + \dot{y}_{L2}^2 + \dot{x}_{R2}^2 + \dot{y}_{R2}^2)) \tag{3.9}$$

y la energía de rotación del sistema se define como

$$T_r = \frac{1}{2}[J_0\dot{\alpha} + J_1((\dot{\alpha}-\dot{\beta}_L)^2 + (\dot{\alpha}-\dot{\beta}_R)^2) + J_2((\dot{\alpha}-\dot{\beta}_L+\dot{\gamma}_L)^2 + (\dot{\alpha}-\dot{\beta}_R+\dot{\gamma}_R)^2)] \quad (3.10)$$

Donde J_0,J_1 y J_2 son la inercia del torso, muslo y pantorrilla. La energía total cinética se calcula:

$$T = T_t + T_r \quad (3.11)$$

Formulas de las fuerzas generalizadas

$$F_{x0} = F_{Lx} + F_{Rx} \quad (3.12)$$

$$F_{y0} = -(m_0 + 2m_1 + 2m_2)g + F_{Ly} + F_{Ry} \quad (3.13)$$

$$F_\alpha = -(\frac{\partial y_{L1}}{\partial \alpha}m_1 + \frac{\partial y_{L2}}{\partial \alpha}m_2 + \frac{\partial y_{R1}}{\partial \alpha}m_1 + \frac{\partial y_{R2}}{\partial \alpha}m_2)g + \frac{\partial y_{LG}}{\partial \alpha}F_{Ly} + \frac{\partial y_{RG}}{\partial \alpha}F_{Ry} + \frac{\partial x_{LG}}{\partial \alpha}F_{Lx} + \frac{\partial x_{RG}}{\partial \alpha}F_{Rx} \quad (3.14)$$

El modelo dinámico puede expresarse en forma de una ecuación de Langrange:

$$\frac{d}{dt}(\frac{\partial T}{\partial \dot{q}_r}) - \frac{\partial T}{\partial \dot{q}_r} = F_{qr} \quad (3.15)$$

El modelo dinámico tiene la forma :

$$A(q)\ddot{q} = b(1, \dot{q}, M, F) \quad (3.16)$$

Este modelo es actuado con 4 momentos

$$M = [M_{L1}, M_{R1}, M_{L2}, M_{R2}]^T \quad (3.17)$$

Dos de ellos actúan entre el torso y ambos muslos y dos en las articulaciones. La superficie peatonal se modela utilizando fuerzas externas que afectan a ambas piernas.

$$F = [F_{Lx}, F_{Ly}, F_{Rx}, F_{Ry}]^T \quad (3.18)$$

Cuando la pierna debe tocar el suelo, las fuerzas correspondientes se encienden para apoyar la pierna. A medida que la pierna se levanta, las fuerzas se ponen a cero.

3.3. Control robusto para robots bípedos

Una vez establecido el modelo dinámico del robot bípedo, se procede a describir las técnicas de control PID y PD. Estos controladores comparan el valor de una variable medida (señal de entrada) y el valor deseado *(set point)* para producir una señal de salida que mantenga el valor deseado de la variable y usa la diferencia entre valor deseado y el valor de la salida para manipular una variable controlada.

3.3.1. Control PD y PID

El controlador PID es una técnicas más usadas para el control de procesos en la industria Jaiswal2014 [49] . Su simplicidad y robustez proporciona una amplia gama de condiciones de operación de sistemas de control automático. Sin embargo cuando las especificaciones del proceso varía en sus condiciones de funcionamiento, el controlador PID no brinda resultados muy satisfactorios Alassar2010 [50].

El controlador PID tiene tres elementos los cuales son un término proporcional K_p, uno derivativo K_d y uno integral K_i, ver Figura 3.4.

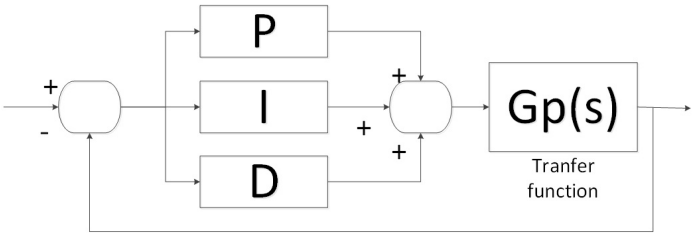

Figura 3.4: Diagrama esquemático de controlador PID

La ecuación que describe el controlador PID es la que se presenta a continuación

$$u = K_p + K_i \int_0^t e dt + K_d \frac{de}{dt} \tag{3.19}$$

Donde u es la señal de salida del controlador PID. e es el error entre la entrada y salida. K_p, K_d y K_i son las ganancias proporcional, derivativo e integral respectivamente.

Por su parte, el controlador PD genera una señal de salida mediante la combinación de la acción proporcional y la acción derivativa, se define como:

$$k \cdot [e(t) + T_d \frac{de(t)}{dt}] \tag{3.20}$$

O bien, representado con la transformada de Laplace

$$\frac{M(S)}{E(S)} = k \cdot (1 + Td \cdot S) \tag{3.21}$$

El controlador PD realiza un control con alta sensibilidad a las variaciones de la entrada y su desventaja radica en que amplifica señales de ruido no obstante tiene la ventaja de usar una acción de control derivativa la cual responde a la velocidad del cambio del error y produce una corrección significativa antes de que la magnitud del error se vuelva demasiado grande. Por tanto, el control derivativo prevé el error, inicia una acción oportuna y tiende a aumentar la estabilidad del sistema.

El controlador PID se considera la mejora del PD, por lo que al implementar PID se incluye el controlador PD, cuyas diferencia son el termino integral y el valor de las ganancias de los términos proporcional y derivativo.

A continuación se describirá el funcionamiento de cada término dentro del controlador PID, cabe aclarar que estos son utilizados similarmente en el controlador PD.

Proporcional

La parte proporcional consiste en el producto dado entre la señal de error y la constante proporcional para lograr que el error en estado estacionario se aproxime a cero.

Existe también un valor límite en la constante proporcional a partir del cual, en algunos casos, el sistema alcanza valores superiores a los deseados. Este fenómeno se llama sobreoscilación y por razones de seguridad, no debe sobrepasar el 30 % , aunque es conveniente que la parte proporcional ni siquiera produzca sobreoscilación.

La parte proporcional no considera el tiempo, por lo tanto, la mejor manera de solucionar el error permanente y hacer que el sistema contenga alguna componente que tenga en cuenta la variación respecto al tiempo, es incluyendo y configurando las acciones integral y derivativa.

La fórmula del proporcional esta descrita po la siguiente ecuación

$$P_{sal} = K_p\, e(t) \tag{3.22}$$

Integral

El modo de control Integral tiene como propósito disminuir y eliminar el error en estado estacionario provocado por el modo proporcional. El control integral actúa cuando hay una desviación entre la variable y el punto de consigna, integrando esta desviación en el tiempo y sumándola a la acción proporcional. El error es integrado, lo cual tiene la función de promediarlo o sumarlo por un período determinado y luego es multiplicado por una constante Ki.

Posteriormente, la respuesta integral es adicionada al modo Proporcional para formar el control P + I con el propósito de obtener una respuesta estable del sistema sin error estacionario.

La fórmula del integral está dada por:

$$I_{sal} = K_i \int e(t)dt \tag{3.23}$$

en donde K_i es la constante integral.

Derivativo

La acción derivativa se manifiesta cuando hay un cambio en el valor absoluto del error, de manera que si el error es constante solamente actúan los modos proporcional e integral.

El error es la desviación existente entre el punto de medida y "Set Point". La función de la acción derivativa es mantener el error al mínimo corrigiéndolo proporcionalmente con la misma velocidad que se produce; de esta manera evita que el error se incremente.

La fórmula del derivativo está dada por:

$$D_{sal} = K_d\, \frac{e(t)}{dt} \tag{3.24}$$

Cuando el tiempo de acción derivada es grande hay inestabilidad en el proceso. Cuando el tiempo de acción derivada es pequeño la variable oscila demasiado con relación al punto de consigna. Suele ser poco utilizada debido a la sensibilidad al ruido que manifiesta y a las complicaciones que ello conlleva. El tiempo óptimo de acción derivativa es el que retorna la variable al punto de consigna con las mínimas oscilaciones.

Ajuste de parámetros del PID

El objetivo de los ajustes de los parámetros PD y PID es lograr que el bucle de control corrija eficazmente y en el mínimo tiempo los efectos de las perturbaciones y se tiene que lograr la mínima integral de error. Si los parámetros del controlador PD o PID (la ganancia del proporcional, derivativo o integral) se eligen incorrectamente el proceso a controlar puede ser inestable.

Ajustar un lazo de control significa ajustar los parámetros del sistema de control a los valores óptimos para la respuesta del sistema de control deseada.

El comportamiento óptimo ante un cambio del proceso o cambio del "setpoint" varía dependiendo de la aplicación. Generalmente, se requiere estabilidad ante la respuesta dada por el controlador y este no debe oscilar ante ninguna combinación de las condiciones del proceso y cambio de "setpoints".

Ajuste manual

Un método de ajuste consiste en establecer primero los valores de I y D a cero. Posteriormente, incremente P hasta que la salida del lazo oscile. Luego establezca P a aproximadamente la mitad del valor configurado previamente. Después incremente I hasta que el proceso se ajuste en el tiempo requerido (aunque subir mucho I puede causar inestabilidad). Finalmente, incremente D, si se necesita, hasta que el lazo sea lo suficientemente rápido para alcanzar su referencia tras una variación brusca de la carga.

Por su parte, el controlador PD genera una señal de salida mediante la combinación de la acción proporcional y la acción derivativa, se define como:

$$k\cdot [e(t) + T_d \frac{de(t)}{dt}] \tag{3.25}$$

O bien, representado con la transformada de Laplace

$$\frac{M(S)}{E(S)} = k\cdot (1 + Td\cdot S) \tag{3.26}$$

El controlador PD realiza un control con alta sensibilidad a las variaciones de la entrada y su desventaja radica en que amplifica señales de ruido no obstante tiene la ventaja de usar una acción de control derivativa la cual responde a la velocidad del cambio del error y produce una corrección significativa antes de que la magnitud del error se vuelva demasiado grande. Por tanto, el control derivativo prevé el error, inicia una acción oportuna y tiende a aumentar la estabilidad del sistema.

El controlador PID se considera la mejora del PD, por lo que al implementar PID se incluye el controlador PD. Por lo que se presenta la implementación del controlador PID para controlar un sistema robótico bípedo.

3.3.2. Control PID robusto para robots bípedos

Se describirá el funcionamiento de los controladores PD y PID para un sistema robótico bípedo. Ambos controladores son casi similares cuya diferencia esta en el término integral del PID, por lo cual se detallará el PID para no ser repetitivo teniendo en cuenta que éste engloba al controlador PD.

El diagrama representativo del controlador PID se muestra en la Figura 3.5. Este sistema retroalimentado cuenta dos entradas donde una representa las señales de referencia externa y la segunda entrada representa la variable "state"generado por el sistema bípedo.

El sistema bípedo tiene como entrada el vector de momentos generado por el controlador PID y genera variables de estado en su salida.

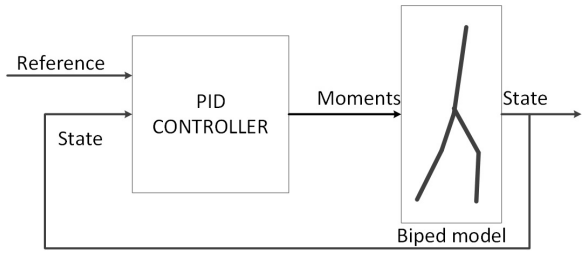

Figura 3.5: Control retroalimentado PID

La composición del bloque de control PID se compone de los sub-bloques presentados en la Figura 3.6.

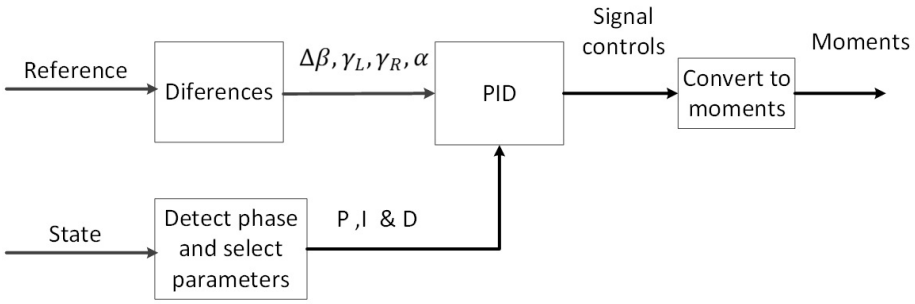

Figura 3.6: Bloque de control contenedor del controlador PID

En donde

- *Diferences* realiza el procesamiento de señales de referencia externas, este cálculo es el error dado entre una señal de referencia externa y la salida del PID, ver figura 3.7.

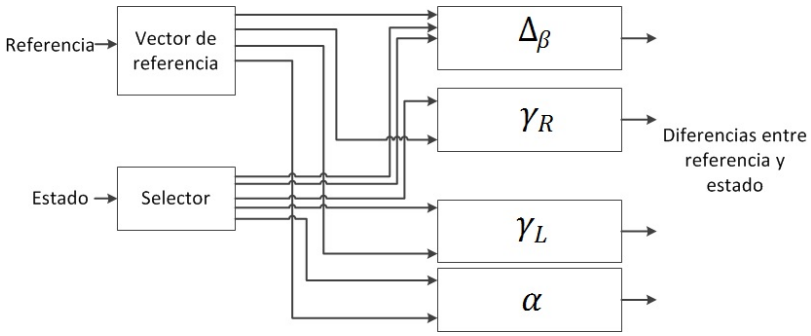

Figura 3.7: Diferencias entre los vectores de referencia y estados

Esta figura muestra las diferencias entre los vectores referencia y estados. El vector de referencia es tomado del trabajo presentado en [48]. El vector de estado contiene 16 elementos por lo que el bloque selector elige los parámetros α,γ_R,γ_R, β_L y β_R. Estos últimos para calcular el parámetro $\Delta\beta$.

- *Detect phase-select parameters* Detecta la fase de la marcha humana, las posibles posiciones de las piernas son ambas piernas en el suelo, sólo la izquierda en el suelo, sólo derecha en el suelo y ninguna en el suelo. De éstas las únicas validas son las primeras tres ya que la última no se observa en la marcha bípeda.

  La detección de fase toma en cuanta los parámetros τ_1 y τ_2. Este bloque cuenta con 4 compuertas AND y dos NOT, el funcionamiento es similar a un decodificador, el cual toma en cuenta 4 constantes para realizar un producto y suma para obtener el numero de fase, ver figura 3.8.

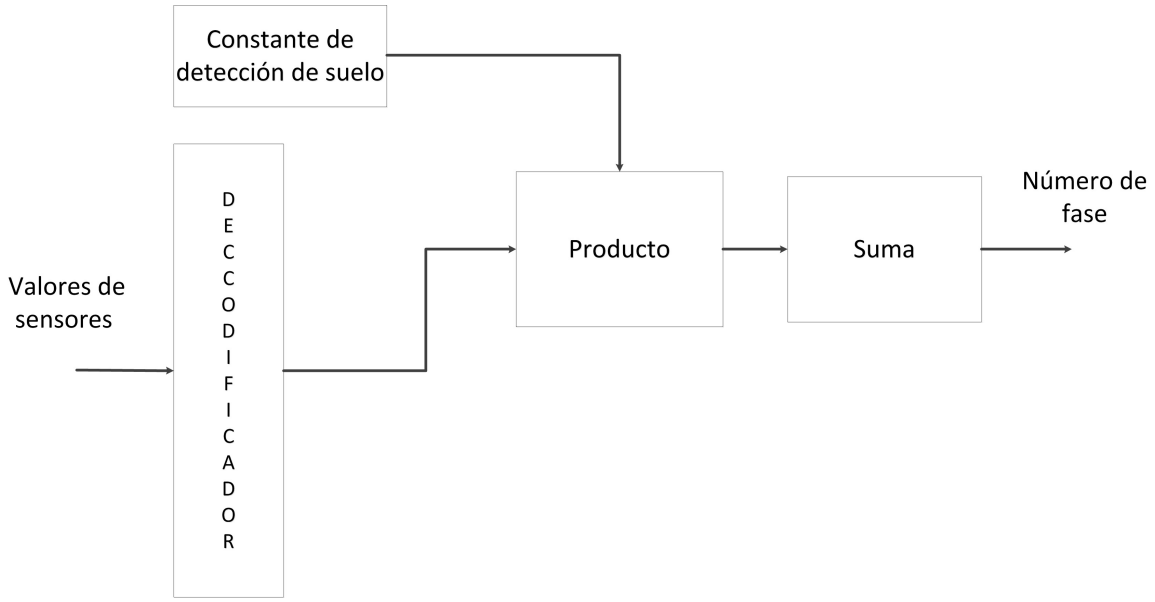

Figura 3.8: Detección de fase

Asimismo realiza una selección de parámetros de ganancias que se utilizan en el controlador PID. Con base a la fase detectada se eligen parámetros de ganancia de ya que durante el balanceo de una pierna determinada las ganancias del controlador serán más débiles que en la fase de apoyo de la otra pierna.

- PID este bloque contiene una arreglo de controladores PID.

  La entrada de este arreglo de controladores son cuatro señales de referencia internas. Por lo que se toman en cuenta señales externas de parámetros y señales que dependen del estado del sistema bípedo. El cálculo de las señales de referencia interna para cada bloque de control PID se utiliza las siguientes formulas:

$$\alpha_{R(ref)} = \alpha_{R(ext)} - \alpha_{R(state)} \tag{3.27}$$

$$\Delta\beta_{ref} = \Delta\beta_{ext} - (\beta_{L(state)} - \beta_{R(state)}) \tag{3.28}$$

$$\gamma_{L(ref)} = \gamma_{L(ext)} - \gamma_{L(state)} \tag{3.29}$$

$$\gamma_{R(ref)} = \gamma_{R(ext)} - \gamma_{R(state)} \tag{3.30}$$

  Los bloques del arreglo de controladores PID son casi similares cuya diferencia radica en la entrada. En la figura 3.9 se muestra la composición de un bloque individual. Se toman en cuenta las señales de referencia $\alpha_{ref}, \Delta\beta_{ref}, \gamma_{L(ref)}, \gamma_{R(ref)}$ como entrada de cada uno de los bloques PID.

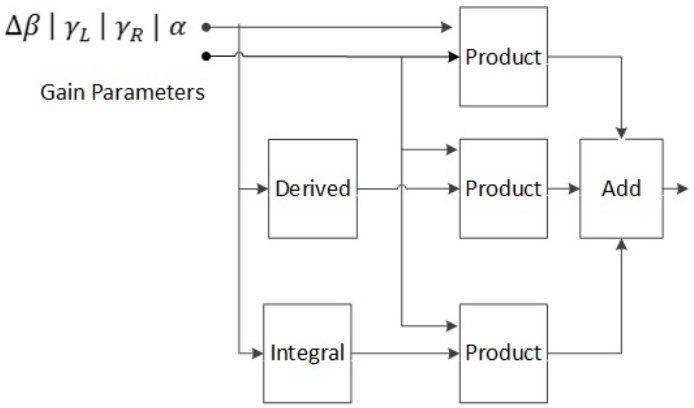

Figura 3.9: Composición del controlador PID

  Después se calcula la derivada y integral de la referencia (error), posteriormente se multiplican con los parámetros de ganancia calculados en la fase de detección. Por lo tanto se tienen 4 señales de control $\alpha_{control}$, $\Delta\beta_{control}$,$\gamma_{L(cpntrol)}$ y $\gamma_{R(control)}$.

- *Convert moments* Una vez que se tengan las señales de control, el siguiente paso es calcular el valor de los momentos, estos valores representan la salida del controlador y la entrada hacia el sistema del robot bípedo, las formulas utilizadas para calcular los momentos son

  - $M_1 = \Delta\beta_{control} + \frac{1}{2}\alpha_{control}$
  - $M_2 = \Delta\beta_{control} - \frac{1}{2}\alpha_{control}$
  - $M_3 = \gamma_{L(control)}$
  - $M_4 = \gamma_{R(control)}$

  En la figura 3.10 se muestra el diagrama a bloques, se observa que los parámetros γ_R y γ_L se transfieren directo a las salidas M_3 y M_4 respectivamente.

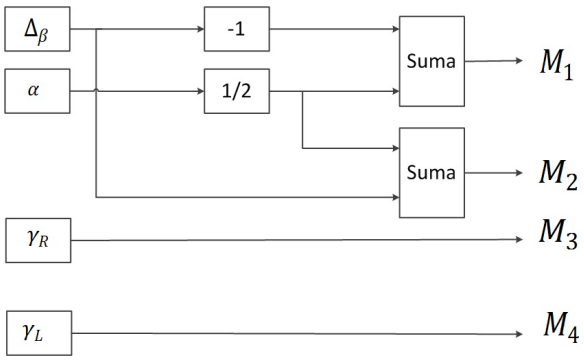

Figura 3.10: Cálculo de los Momentos

3.4. Resultados

En esta sección se describirán los resultados obtenidos de los controladores PID y PD conforme a la marcha normal sobre terreno plano; también al aumento de velocidad para conocer el valor máximo que puede ser exigido el robot sin tener una caída; además se realizarán experimentos de robustez ante una inclinacion del suelo, este análisis se realiza para conocer el angulo de inclinacion máximo y saber si el robot bípedo puede o no caminar en terreno que no sea plano.

3.4.1. Simulación de marcha suave

En la marcha normal se miden seis parámetros los cuales describen la forma de andar del robot bípedo. El primer parámetro a observar es α, el cual representa a la oscilación del torso y es mostrado en la figura 3.11.

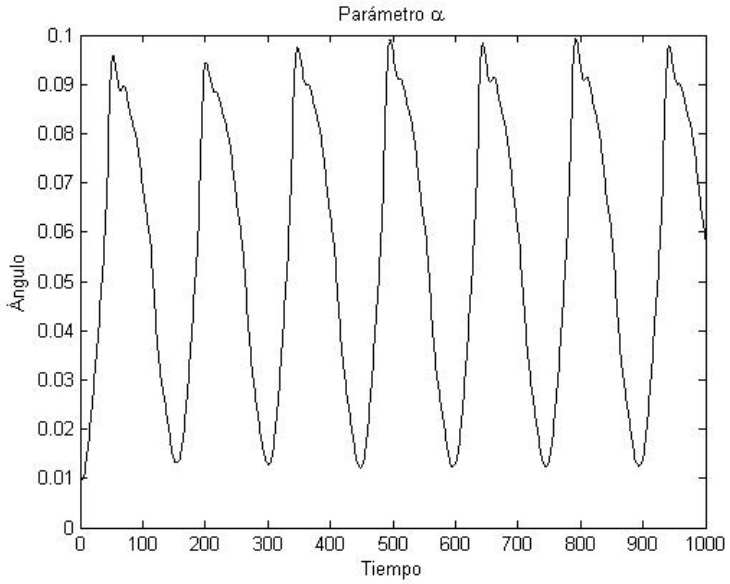

Figura 3.11: Parámetro α del controlador PID

Se observa una señal periódica la cual representa la oscilación del torso.

En la figura 3.12 y 3.13 muestran los parámetros β_R y β_L las cuales son pertenecientes al ángulo formado entre la cadera y la rodillas derecha e izquierda.

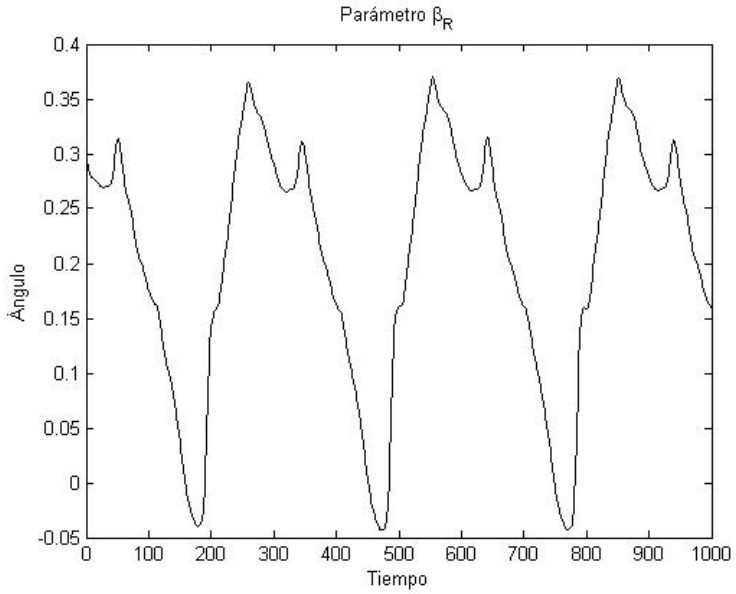

Figura 3.12: Parámetro β_R del controlador PID

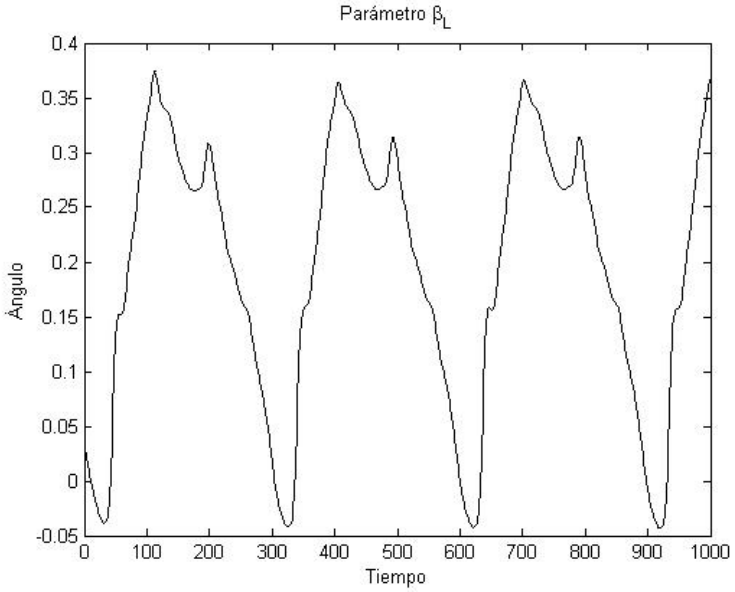

Figura 3.13: Parámetro β_L del controlador PID

También en las figuras 3.14 y 3.15 muestran el comportamiento en el tiempo de las variables γ_R y γ_L, se observa que son cíclicas y muy parecidas, la diferencia esta en que una esta adelantada en el tiempo.

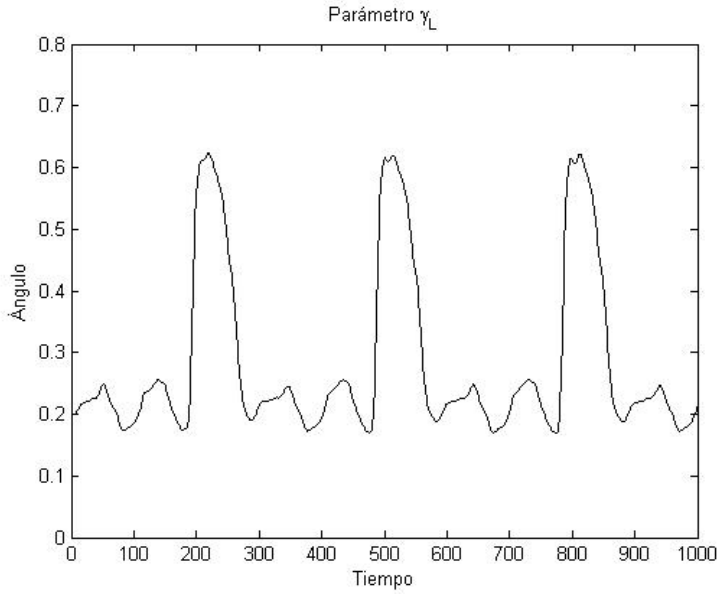

Figura 3.14: Parámetro γ_L del controlador PID

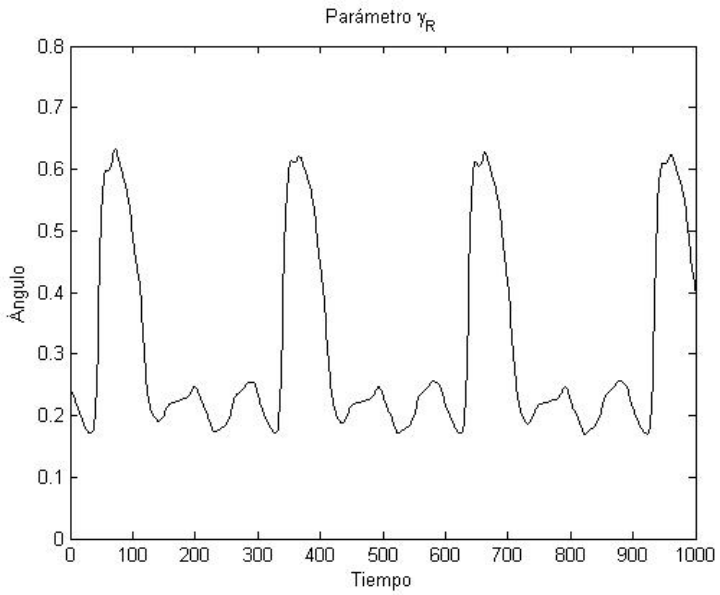

Figura 3.15: Parámetro γ_L del controlador PID

Por otra parte, para comprender las diferencias entre el comportamiento del controlador PID y PD es necesario calcular algunos valores como el promedio, mediana y desviación estándar, ya que con las gráficas no se puede comparar numéricamente.

En la tabla 3.1se muestra la obtención de parámetros estadísticos de la señales de marcha del controlador PD.

	Media	Mediana	Desviación Estandar	Mínimo	Máximo
α	0.0559	0.0590	0.0290	0.0099	0.0992
β_R	0.1917	0.2135	0.1233	-0.0428	0.3744
β_L	0.2041	0.2367	0.1143	-0.0434	0.3701
Δ_β	-0.0124	-0.0245	0.2157	-0.3104	0.3111
γ_R	0.3071	0.2300	0.1555	0.1702	0.6324
γ_L	0.2839	0.2246	0.1408	0.1696	0.6235

Tabla 3.1: Valores estadísticos de las distintas variables del controlador PID

De la tabla del controlador PD se muestran un rango de desviación estándar entre 0.0290 a 0.2157. El parámetro con el menor valor mínimo es β_L y el parámetros con el mayor valor máximo es γ_L. Esto representa que la señal β_L tiene picos más grandes y por el contrario γ_L tiene picos con valores menores.

Similarmente se procede a calcular los parámetros estadísticos de las señales del controlador PID.

	Media	Mediana	Desviación Estándar	Mínimo	Máximo
α	0.0409	0.0156	0.1006	-0.0919	0.2207
β_R	0.1885	0.2345	0.2024	-0.1896	0.6164
β_L	0.1908	0.2397	0.1980	-0.1881	0.5992
Δ_β	-0.0023	-0.0071	0.3019	-0.4201	0.4196
γ_R	0.2608	0.1863	0.2363	-0.0037	0.9192
γ_L	0.2596	0.1836	0.2369	-0.0040	0.9228

Tabla 3.2: Valores estadísticos de las distintas variables del controlador PD

De la tabla del controlador PID se muestran un rango de desviación estándar entre 0.1006 a 0.3019. El parámetro con el menor valor mínimo es γ_L y el parámetros con el mayor valor máximo es γ_L.

3.4.2. Velocidad

Se reportan los resultados de los experimentos realizados con el aumento de velocidad del robot bípedo. Los experimentos de velocidad se realizan modificando el parámetro dentro de la configuración del modelo robótico. Se comenzó con el controlador PD en una simulación en terreno plano y a velocidad normal estableciendo 0.13 $\frac{m}{s}$ con una inclinación del suelo de 0 grados. Como segundo paso de simulación se modificó el ángulo del terreno de forma ascendente y descendente a una velocidad normal. Los resultados obtenidos fueron un ángulo de inclinación ascendente con un valor 1.13 grados y un ángulo de inclinación descendente con un valor de 0.9770 grados. Resumiendo los resultados obtenidos de los experimentos realizados se muestran las tablas 3.3, 3.4 y 3.5.

	Terreno Plano 0 grados
Velocidad Normal	0.1303 $\frac{m}{s}$
Velocidad Máxima	0.2612 $\frac{m}{s}$

Tabla 3.3: Velocidad normal y máxima en 0 grados de inclinación controlador PD

	Terreno Ascendente $\theta_{Ascmax} = 2.2473$ °
Velocidad Normal	0.0977 $\frac{m}{s}$
Velocidad Máxima	0.2437 $\frac{m}{s}$

Tabla 3.4: Velocidad normal y máxima en 2.2473 ° de inclinación controlador PD

	Terreno Descendente $\theta_{Descmax} = 2.1162$ grados
Velocidad Normal	0.0611 $\frac{m}{s}$
Velocidad Máxima	0.1945 $\frac{m}{s}$

Tabla 3.5: Velocidad normal y máxima en 2.1162 ° de inclinación controlador PD

De estos resultados se observa que a velocidad normal en los experimentos realizados sobre terreno plano brinda un mejor comportamiento seguido de terreno en pendiente ascendente y por último en una pendiente descendente. Este comportamiento se ve reflejado similarmente cuando se realizan los experimentos con velocidad máxima, estableciendo mayor velocidad en terreno plano seguido de pendiente ascendente y por último pendiente descendente.

Por otra parte, para el controlador PID se realizaron simulaciones iguales que el PD. Se configuró con distintas velocidades y ángulos de inclinación del suelo, estos resultados son mostrados en las tablas 3.6,3.7 y 3.8.

	Terreno Plano 0 °
Velocidad Normal	0.1303 $\frac{m}{s}$
Velocidad Máxima	0.2843 $\frac{m}{s}$

Tabla 3.6: Velocidad normal y máxima en 0 grados de inclinación controlador PID

	Terreno Ascendente $\theta_{Ascmax} = 2.4605$ °
Velocidad Normal	0.1153 $\frac{m}{s}$
Velocidad Máxima	0.2547 $\frac{m}{s}$

Tabla 3.7: Velocidad normal y máxima en 2.4605 ° de inclinación controlador PID

	Terreno Descendente $\theta_{Descmax} = 2.218$ °
Velocidad Normal	0.1042 $\frac{m}{s}$
Velocidad Máxima	0.2156 $\frac{m}{s}$

Tabla 3.8: Velocidad normal y máxima en 2.218 ° de inclinación controlador PID

En terreno plano y a velocidad normal estableciendo 0.13 $\frac{m}{s}$ con una inclinación del suelo de 0 grados. Se modificó el ángulo del terreno de forma ascendente y descendente a una velocidad normal. Los resultados son similares en al mostrado en el controlador PD, reportando que la velocidad mayor se reporta en terreno plano, seguido de terreno ascendente y después descendente.

Revisando ambos controladores se nota que el grado de inclinación en planos ascendentes es mayor que en planos descendentes. El controlador PID ofrece mayores

y mejores valores de angulo descendente y ascendente en comparación a los que ofrece el controlador PD.

3.4.3. Robustez

Para analizar la robustez se tiene en cuenta el ángulo de inclinación del plano inclinado. Los experimentos de robustez se realizan modificando un parámetro de configuración del modelo robótico. Los experimentos de cambio de suelo se configura el parámetro "groundbp.ªntes de simular.En la figura 3.16 se muestra la posición del robot en pendiente ascendente, descendente y cero.

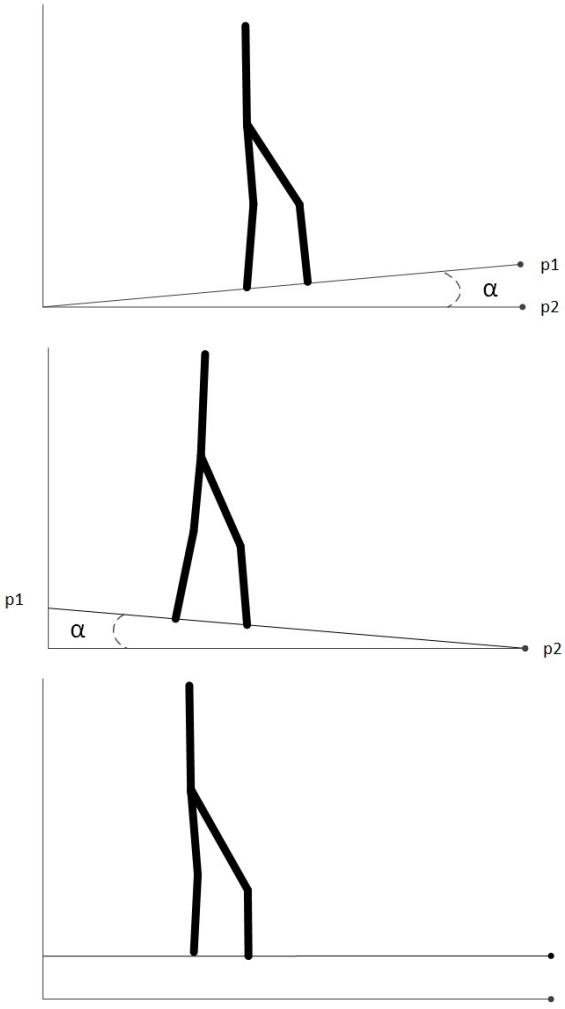

Figura 3.16: Robot en plano ascendente, descendente y terreno perpendicular

En las tablas 3.4.3 y 3.4.3 muestra los resultados obtenidos del controlador PD y PID respectivamente. En estas tablas se muestran la velocidad normal y los ángulos resultantes a esta velocidad tomada como base que se alcanza cuando en ángulos de inclinación máxima en terreno ascendente y descendente.

Ángulo de inclinación	Velocidad normal
0 °	0.1303 $\frac{m}{s}$
2.2473 °	0.0977$\frac{m}{s}$
2.1162 °	0.0611$\frac{m}{s}$

Tabla 3.9: Ángulos de inclinación a velocidad normal del controlador PD

Ángulo de inclinación	Velocidad máxima
0 °	0.2612 $\frac{m}{s}$
2.2473 °	0.2437 $\frac{m}{s}$
2.1162 °	0.1945 $\frac{m}{s}$

Tabla 3.10: Ángulos de inclinación a velocidad normal del controlador PD

En la tablas 3.11 y 3.12 se muestra las velocidades máximas ante los ángulos de inclinación de los controladores PD y PID, tomando en cuenta el modo ascendente y descendente.

Ahora bien se presenta los resultados del controlador PID, los experimentos fueron similares y se reportan diferentes ángulos a distintas velocidades.

Ángulo de inclinación	Velocidad normal
0 °	0.1303 $\frac{m}{s}$
2.4605 °	0.1153 $\frac{m}{s}$
1.216 °	0.1042 $\frac{m}{s}$

Tabla 3.11: Ángulos de inclinación a velocidad normal del controlador PID

Ángulo de inclinación	Velocidad máxima
0 °	0.2843 $\frac{m}{s}$
2.4605 °	0.2547 $\frac{m}{s}$
2.216 °	0.2156 $\frac{m}{s}$

Tabla 3.12: Ángulos de inclinación a velocidad máxima del controlador PID

Se observa que los ángulos obtenidos a velocidad máxima son menores en comparación a los ángulos obtenidos a velocidad normal. Esto es debido a que en la simulación de modifica al mismo tiempo el suelo y la velocidad, lo que afecta a estabilidad de la marcha en el robot bípedo.

3.5. Conclusiones

Se ha presentado la implementación de los controladores PD y su mejora el PID. La estructura de los controladores es similar cuya única diferencia es el termino de

integración del controlador PID.

Si se toma en cuenta el aumento de velocidad y la robustez ante pendientes, se observa que el comportamiento del controlador PID es mejor que el PD. Ambos controladores presentan un poca robustez ante el cambio de pendiente ascendente y descendente, dando un mejor angulo en un terreno ascendente. Además el aumento de velocidad es mayor en el controlador PID en comparación al PD. Por ultimo, se puede decir que los controladores PD y PID realiza un control estable del robot bípedo, sin embargo ambos controladores son susceptibles a pequeñas variaciones en sus parámetros lo cual puede provocar una caída del robot.

Capítulo 4

Generación de caminata robótica bípeda mediante análisis de datos de la marcha humana

En este capítulo se abordará a detalle el concepto teórico y el desarrollo de un controlador basado en datos humanos, se utilizan las técnicas de *regression clustering*, $k-means$ y $fuzzyc-means$ como métodos de minería de datos los cuales ayudan a procesar los datos para replicar la marcha humana en un robot bípedo.

Estos métodos son elegidos ya que el procesado de los datos se realiza antes de ejecutarse dentro de un sistema de control y además por la generación de parámetros estadísticos que representan el comportamiento de los datos.

4.1. Marcha humana

Como se ha explicado anteriormente, la marcha humana es un proceso de locomoción del cuerpo humano en posición erecta moviendo alternando el apoyo en las piernas. Por tal motivo el contacto es permanente al menos en uno de sus pies entre la persona y el suelo. Es conveniente conocer las fases de marcha humana para poder establecerla después en un robot bípedo.

Fases de la marcha humana

Sucesión de acciones comprendidas entre dos choques de talón consecutivos del mismo pie. Suele tomarse como principio del ciclo el instante en que uno de los pies toma contacto con el suelo, habitualmente a través del talón.

El ciclo de la marcha presenta dos fases:

Fase de apoyo (representa el 60 % del ciclo)

Comienza con el contacto inicial del talón en el suelo y termina con el despegue del talón.

Fase de balanceo u oscilación (representa el 40 % del tiempo)

Va desde el instante del despegue del antepié, avanzando el pie en el aire como preparación del siguiente apoyo, hasta el contacto en el suelo. La duración de cada una de

las fases podremos darnos cuenta, que sumando los ciclos que se están produciendo de manera simultánea en ambos miembros inferiores en algún momento ambos pies se encuentran en contacto con el suelo. A este nuevo periodo se le denomina fase de doble apoyo.

Parámetros de la marcha

Los siguientes parámetros de la marcha pueden ser modificados de persona en persona por factores como la talla, la edad, patologías o trastornos locomotores entre otros.

Largo de paso: Es la distancia entre el talón de un pie y la punta subsecuente del otro pie. Por ejemplo, cuando ambos pies están en contacto con el suelo, el largo de paso derecho es la distancia entre el talón del pie izquierdo y el talón del pie derecho.

- **Zancada o largo de ciclo:** Es la distancia entre el contacto inicial de un pie hasta el próximo contacto inicial del mismo pie.

- **Velocidad:** Es la velocidad promedio del cuerpo a lo largo del plano de progresión medido sobre una o más zancadas. Se mide en centímetros por segundo.

- **Cadencia:** Es el número de pasos en una unidad de tiempo (generalmente el minuto).

Una vez establecido el conocimiento general de la marcha humana se procede a describir la base de datos utilizada en la tesis y de como se preproceso antes de utilizar los métodos de minería de datos.

4.1.1. Preprocesado de datos humanos

Para este estudio se basa en los resultados del análisis de la marcha de un estudio longitudinal de 5 años de 16 niños entre las edades de 7 y 12 años, reportados previamente por Stansfield [51]. La extracción de los datos se basa en la medición de los sensores adecuados a los cuerpos de los niños,ver figura 4.1.

Figura 4.1: Extracción de datos en niños

Para realizar un análisis se toman en cuenta los datos presentados por el articulo [52]. Estos datos ya están normalizados usando las ecuaciones que se presentan en la siguiente lista.

- $Tiempo\ normalizado = tiempo \times \frac{1}{\sqrt{H/g}}$
- $Longitud\ normalizado = longitud \times \frac{1}{H}$
- $Peso\ normalizado = peso \times \frac{1}{M \times g}$
- $Velocidad\ normalizada = velocidad \times \frac{1}{\sqrt{H \times g}}$
- $Longitud\ de\ paso\ normalizado = Longitud\ de\ paso \times \frac{1}{H}$
- $Fuerza\ normalizada = fuerza \times \frac{1}{M \times g}$
- $Momento\ normalizado = moment \times \frac{1}{M \times g \times H}$

- $Potencia\ normalizada = potencia \times \frac{1}{M \times g^{\frac{3}{2}} \times H^{\frac{1}{2}}}$

En el trabajo realizado de Stansfield también se describe algunos algunos parámetros como el promedio de la edad, altura y peso, los cuales son mostrados en la tabla 4.1.

Parámetro	Valor	Tolerancia
Edad	9.5 años	1.41
Altura	1.38 m.	0.11
Longitud de piernas	0.66 m.	0.06
Masa	32.64 Kg.	8.07

Tabla 4.1: Parámetros físicos de la extracción de datos en niños.

Estos parámetros nos brindan un conocimiento acerca de las personas utilizadas para el experimento de extracción de datos. De todo el conjunto de datos obtenido se escogen diversos parámetros para utilizarlos dentro del controlador RC. En la figura 4.2 se muestra el ángulo de la cadera, rodillas y tobillos, estos correspondientes a α, β_L,β_R,γ_R y γL.

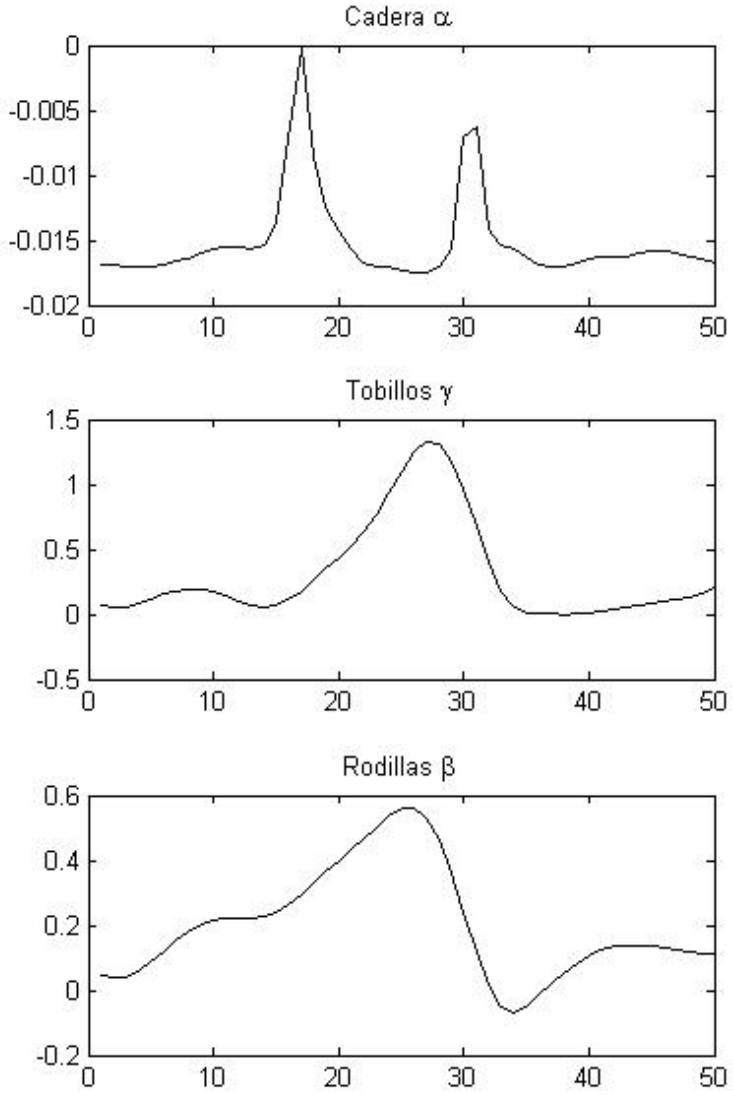

Figura 4.2: Ángulo de la cadera, rodillas y tobillos

Estas gráficas muestran los valores normalizados de un solo paso, sin embargo la entrada de datos del controlador RC necesita un formato especifico para poder trabajar. Este formato consta de una serie de pasos con determinado número. De manera que se reprodujo una cantidad de pasos tomando como base el paso descrito por lo datos de Stansfield.

4.1.1.1. Muestreo

La cantidad de datos de un paso de la muestra humana es de 50 y se requiere la cantidad de 150 datos para formar un ciclo utilizable por el controlador RC. Ante este problema se escoge el sobre muestreo de las señales. Para evitar las caídas abruptas se utiliza la técnica conocida como sobremuestreo (oversampling), que permite reconstruir, tras la conversión D/A, una señal de pendiente suave.

Se utiliza la función de MatLab *upsample(x,N)*, esta función aumenta la tasa de muestreo de x mediante la inserción de $N-1$ ceros entre las muestras. x puede ser un vector o una matriz. Si x es una matriz, cada columna se considera una secuencia separada. y tiene $x \times n$ muestras. Por ejemplo: Incrementar la tasa de muestreo en una secuencia en 3:

```
x = [1 2 3 4];
y = upsample(x,3); %  x,y
```

```
x =
1  2  3  4

y =
1  0  0  2  0  0  3  0  0  4  0  0
```

Se observa que se le agregan ceros a la señal, de modo que se necesitan rellenar esos ceros para obtener una señal suave. El método utilizado es la interpolación, la cual promedia los dos datos en los extremos inferior y superior. Esta técnica es útil para calcular el valor que remplace los ceros tal como se muestra en la figura 4.3.

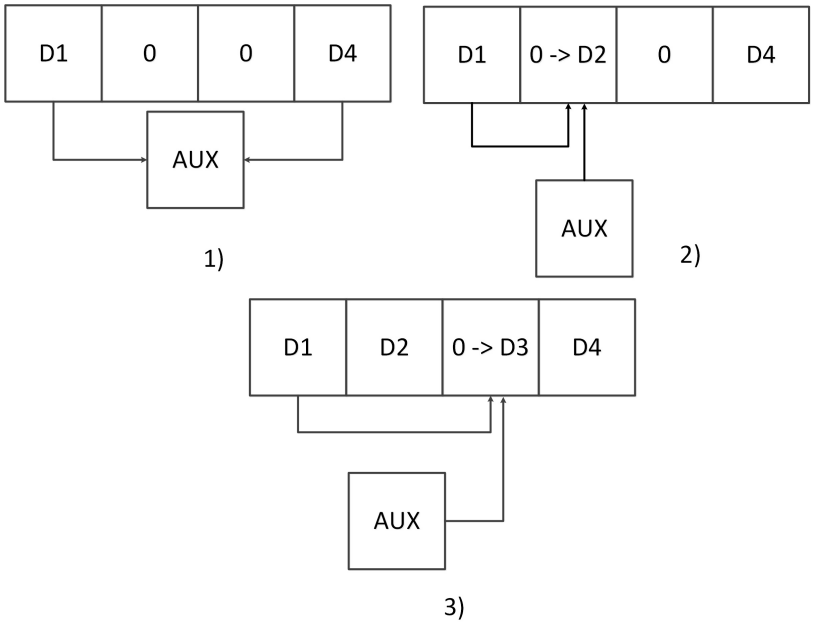

Figura 4.3: Cálculo de la interpolación en los datos faltantes de las señales.

Se genera una variable auxiliar para utilizarla como punto medio y es obtenida por la formula : $aux = \frac{D1+D4}{2}$, una vez obtenido el valor de aux se calcula el valor de $D2$ mediante $D2 = \frac{aux+D1}{2}$, también se calcula $D3$ usando $D3 = \frac{aux+D4}{2}$. De todo el calculo anterior se conforma una señal con 150 datos, el cual representa un paso. Por ultimo se procede a copiar la señal par formar una serie de pasos que representen la marcha humana, en la figura se muestra los parámetros generados α,β_R,β_L,γ_R,γ_L y Δ_β.

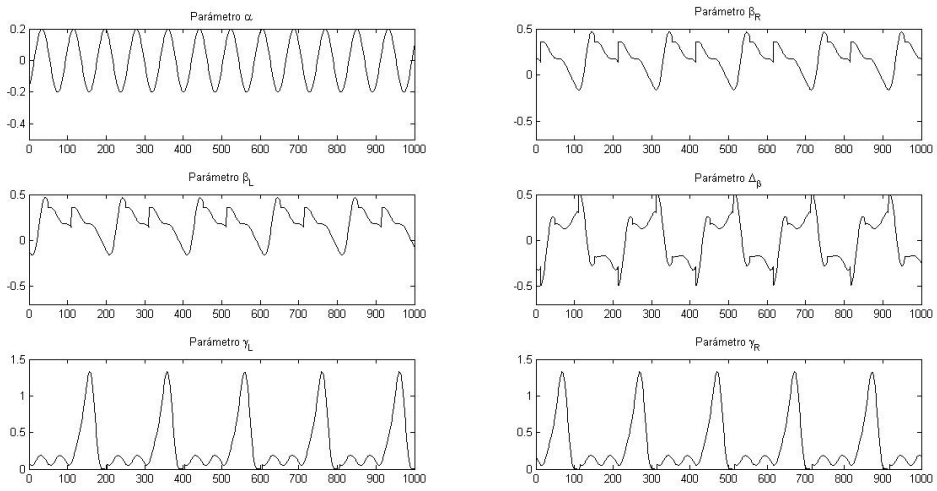

Figura 4.4: Señales de parámetros generadas para generar marcha bípeda.

Cabe notar que algunos parámetros como β_R-β_L y γ_R-γ_L son parecidos, las únicas diferencias se observan en el desfasamiento en la señal que existe entre ellos.

4.1.2. Aprendizaje local ponderado

El Aprendizaje local ponderado (Locally Weighted Learning, LWL) representa otra clase de métodos que se ajustan a las funciones no lineales complejas por funciones locales del núcleo.

Hay dos tipos principales de LWL: basado en memoria LWL que simplemente almacena todos los datos de entrenamiento de la memoria y usa técnicas de interpolación eficientes para hacer predicciones de la nueva entradas aha1997 [53]; LWL no basado en memoria construye compactas representaciones de datos de entrenamiento mediante técnicas recursivas así como para evitar el almacenamiento de grandes cantidades de datos en la memoria vijayakumar2005 [54].

La parte clave de todos los algoritmos de LWL es determinar la región de validez en el que un modelo local se puede confiar. LWL logra baja complejidad computacional y eficiente aprendizaje en espacios de dimensionales altas. Otra característica atractiva de LWL es que los modelos locales pueden asignarse según sea necesario, y el proceso de modelado se puede controlar fácilmente mediante el ajuste de los parámetros de los modelos locales. LWL han sido utilizado con bastante éxito para aprender la dinámica inversa o cinemática asignaciones en sistemas de control de robot atkeson1997 [55].

El método de aprendizaje local ofrece un mapeo general, el cual se forma utilizando varios modelos locales que tienen una estructura interna sencilla pero son válidas sólo en pequeñas regiones del espacio de entrada-salida.

Típicamente, los modelos locales utilizados son lineales, lo que garantiza la escalabilidad de la estructura del modelo. Dadas estas características pueden ser modelados

sistemas simples y complejos, únicamente variando el número de modelos locales.

Para evaluar la hipótesis de los modelos locales simples, se crea una estructura de retroalimentación basado en modelos locales, regresión agrupada (Regression Clustering-RC).

Los modelos locales se basan en el análisis de componentes principales de los datos locales. Esto significa que el comportamiento de caminar se almacena en la estructura del modelo. El funcionamiento dado el estado actual del sistema, la estimación de salida del modelo se utiliza directamente como el siguiente valor de señal de control y no se necesitan soluciones de control o señales adicionales de referencia.

4.2. Regression clustering para análisis de datos de marcha humana

En esta sección se aborda el control de un robot bípedo utilizando la técnica de minería de datos llamada Regression Clustering (*RC*). Los datos utilizados representan la marcha humana si se toma en cuenta las medidas angulares las diferentes articulaciones de las piernas.

4.2.1. Descripción del algoritmo

Se utilizó el método llamado regresión agrupada (Regression Clustering) para modelar el mapeo de las salidas-entradas del modelo bípedo. La estructura del modelo es una combinación de la regresión de componentes principales locales, (PCR).Estos PCRs están distribuidos a lo largo de la trayectoria de funcionamiento del sistema. La estimación de control del modelo de regresión en cada paso de tiempo se calcula como una combinación de todas las salidas locales de los PCRs.

Para formar el mapeo de la regresión agrupada se necesitan datos de muestreo de entrada-salida generado por la trayectoria de caminado. Los datos son divididos en clusters, cada cluster tiene un punto de operación y un modelo local, los cuales son calculados con los datos de cada cluster. La estimación total del modelo de regresión se calcula como un promedio ponderado de los locales, las estimaciones de los modelos locales más cercanos se les da la mayor peso, mientras que los otros mas alejados se les asigna menor peso.

El movimiento de marcha del caminante bípedo se divide en dos fases respecto al número de piernas que se encuentran en contacto con la tierra. Fase de doble soporte *(Double Support Phase -DSP)* se denomina cuando ambas piernas tienen contacto con el suelo. La fase de única pierna de apoyo *(Single Support Phase)*, esta fase describe una pierna en contacto con el suelo y la otra balanceándose hacia adelante. Toda la marcha se puede describir con un DSP y un SSP. Esto debido a que en SSP se utiliza una conmutación para modelar el balanceo de la pierna izquierda y derecha.

Estructura del modelo local

Cada modelo local calcula un componente principal de regresión lineal (PCR).

Los componentes principales deben de mostrar direcciones ortogonales a las mas altas variaciones de los datos de entrada. Se asume que varianza está llevando información y los mas importantes componentes principales relevantes en los datos, mientras que se omiten como ruido los menos relevantes.

La idea de la PCR es primero para mapear los datos de entrada a la menor dimensional sub-espacio de componentes principales y luego usar regresión lineal multivariable para obtener la estimación de salida. Se denota como $u^\rho(k)$ a la señal de entrada para un modelo local ρ en un instante de tiempo kh y se denota como $y^\rho(k)$. Cuando los datos de entrada son mapeados a los componentes principales, se calcula la variable latente de la señal denotada por $x^\rho(k)$. La toda la estructura del modelo puede ser almacenado en las siguientes estadísticas calculadas usando los datos del cluster ρ.

- C_u^ρ valor esperado del vector de entrada u^ρ.
- C_y^ρ valor esperado del vector de salida y^ρ.
- P_{xx}^ρ Inversa de la matriz de covarianza de la variable latente x^ρ.
- $R_{x\tilde{u}}^\rho$ Covarianza cruzada de la variable latente y los datos de entrada.
- R_{yx}^ρ Covarianza cruzada de la variable latente y los datos de salida.
- R_{uu}^ρ Covarianza de los datos de entrada.

La salida estimada de el modelo ρ dado un vector de entrada arbitrario $\tilde{u}(k)$.

$$\hat{y}^p(k) = R_{yx}^\rho (P_{xx}^\rho)^2 R_{x\tilde{u}}^\rho (\tilde{u}(k) - C_{\tilde{u}}^\rho + C_y^\rho) \tag{4.1}$$

$C_{\tilde{u}}^\rho$ es el valor esperado del estado del vector reducido.

El valor de costo de la estimación realizada en la unidad ρ debe ser evaluados para medir el error del resultado. El costo depende de la distancia entre el centro del cluster y el punto de estimación.

$$J^\rho(u(k)) = \frac{1}{2}(u(k) - C_u^\rho)^T H_1^\rho (u(k) - C_u^\rho) \tag{4.2}$$

Donde H_1^ρ es una matriz constante de pesos.

Combinación de modelos locales

La estimación global de la estructura de regresión agrupada se calcula como un promedio ponderado de todos los modelos locales. Dado un número de los puntos de funcionamiento es Nop se tiene:

$$\hat{y}(k) = \frac{\sum_{\rho=1}^{Nop} K^\rho(k)\hat{y}^\rho(k)}{\sum_{\rho=1}^{Nop} K^\rho(k)} \tag{4.3}$$

Los pesos de forma natural deben ser tales que los actualmente mejores modelos locales que coinciden afectan los resultados finales de la mayoría, mientras que los otros están prácticamente abandonados. Veamos ahora elegimos.

$$H_1^\rho = \frac{1}{\sigma} {R_{uu}^\rho}^{-1} \tag{4.4}$$

La matriz de ponderación para cada modelo local se calcula con la inversa de la matriz de covarianza, σ es el factor de escalado. Para estimar el valor de $\hat{y}(k)$

$$K^\rho(k) = \frac{1}{\sqrt{(2\pi)^n \mid H_1^{\rho-1} \mid}} \exp(-J^\rho(u(k))) \tag{4.5}$$

y se establece el valor de $\sigma = 1$.

4.2.2. RC aplicado a la marcha de robot bípedo

El desarrollo del algoritmo regression clustering puede se representado por medio del diagrama de bloques mostrado en la Figura 4.5. El diagrama de bloques esta

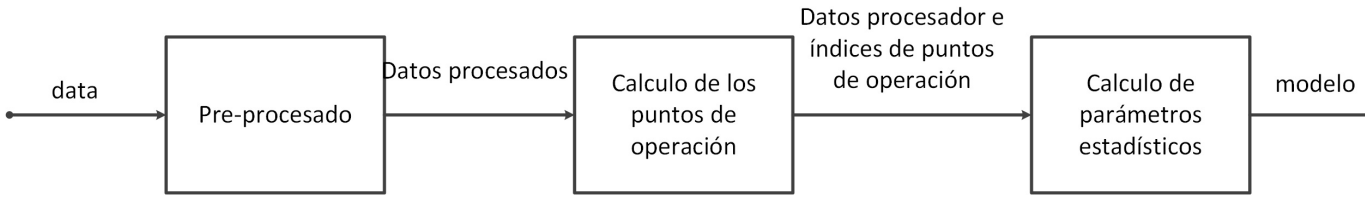

Figura 4.5: Etapas de aprendizaje del algoritmo RC

divido en 3 etapas principales, las cuales se enlistan a continuación:

- Preprocesado.
- Cálculo de los puntos de operación.
- Cálculo de los parámetros estadísticos.

4.2.2.1. Preprocesado

El bloque de preprocesado se encarga de adecuar los datos de entrada, calcular algunas propiedades y remover los datos extras. Este bloque se compone de tres módulos como se muestra en la figura 4.6.

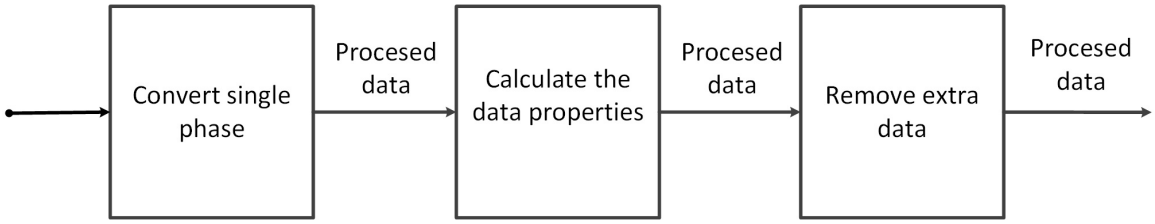

Figura 4.6: Módulos de la fase de preprocesado

Se describirá el funcionamiento de cada modulo para conocer a detalle su comportamiento.

- **Convert single phase**

  El primer modulo se toma como entrada los estados que representan una trayectoria del sistema bípedo, existe una clasifican los datos de la trayectoria en dos tipos: Double Support Phase (DSP) y Single Support Phase (SSP). Por lo tanto elimina la secuencia $Single\ Support(right) - Double\ Support\ Phase - Single\ Support(left) - Double\ Support\ Phase$ y la acota a la secuencia de paso $Single\ Support(right) - Double\ Support\ Phase$, ver figura 4.7.

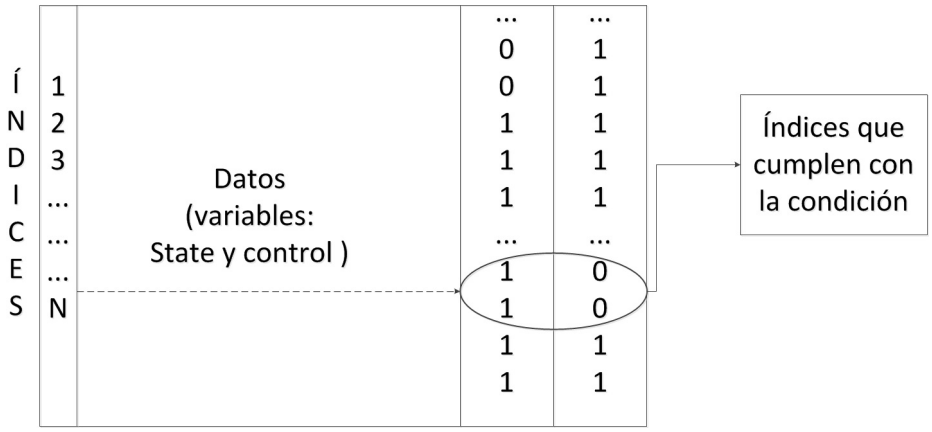

Figura 4.7: Acotación a *Single Support Phase*

- **Cálculo de propiedades de los datos**

  En el segundo módulo se calcula la media y varianza de la base de datos, esta base de datos esta conformada por dos partes principales: estados *(state)* y *control*. La parte de estados esta conformado por los parámetros X_0 y Y_0 α,β_R,β_L, γ_R y γ_L y sus respectivas derivadas (velocidades). Por otro lado, la parte de control se basa en el valor de los cuatro momentos M_1,M_2,M_3 y M_4. EL cálculo de la media y la varianza será útil para la reconstrucción de los modelos locales.

- **Remover los datos extras**

  Por ultimo, se eliminan datos no útiles para las fases posteriores, como los datos que identifica las fases DSP y SSP.

  Se eliminan una gran cantidad de datos para que se tenga lo necesario para describir una marcha bípeda. Además se resta la media calculada anteriormente con base al parámetro al cual pertenecen:

$$datos_{estados} = datos_{estados} - vectorEstados_{medias} \tag{4.6}$$

$$datos_{control} = datos_{control} - vectorControl_{medias} \tag{4.7}$$

  También se normalizan los datos a la unidad de toda la varianza mediante el uso de las siguientes ecuaciones:

$$datos_{estados} = \frac{datos_{estados}}{vector_{desviacionEstandar}} \tag{4.8}$$

$$datos_{control} = \frac{datos_{control}}{vector_{desviacionEstandar}} \tag{4.9}$$

4.2.2.2. Cálculo de puntos de operación

En este bloque los datos son divididos en Nop puntos de operación usando el algoritmo (Self Organize Map) SOM.

Por lo que se crea un modelo de esos datos, que posteriormente puede servir para agruparlos por criterios de similitud.

Los parámetros de entrada el algoritmo son:

- El training set (conjunto de entrenamiento).
- El número de clusters o nodos.
- El número de epochs (épocas).

El algoritmo SOM paso a paso es como se describe a continuación:

- Los pesos de cada nodo son inicializados, esta inicializacion se recomienda que sea aleatoria.
- Un ejemplo (o vector) es elegido aleatoriamente del conjunto de datos y presentado al *grid* de nodos.
- Cada nodo es examinado para calcular cuál tiene el peso "más cercano " al vector de entrada. El peso del nodo ganador es conocido como el Best Matching Unit (BMU).

- El radio de la vecindad del BMU es calculado. Este valor se inicializa alto, normalmente igual al "radio"del grid, pero disminuye a cada paso. Todos los nodos en el radio calculado son marcados como "dentro de la vecindad del BMU".
- El peso de cada nodo de la vecindad es ajustado en función a la tasa de aprendizaje para " acercarlo .ªl vector de entrada. (El sombrero mejicano).
- Repetir los pasos del 2 en adelante para N iteraciones.

Las limitaciones

- Poder de cómputo, en algunas ejecuciones con una gran cantidad de datos pude que consuma gran cantidad de tiempo.
- Sensitividad a la inicialización, como es aleatoria en ocasiones se requerirán mas iteraciones.

Una vez descrito el algoritmo SOM se procede a describir las etapas que conforman el calculo de puntos de operación, estas engloban la separación inicial de los datos en clusters y después el calculo de los puntos de operación realizando SOM, estas etapas se muestran la figura 4.8.

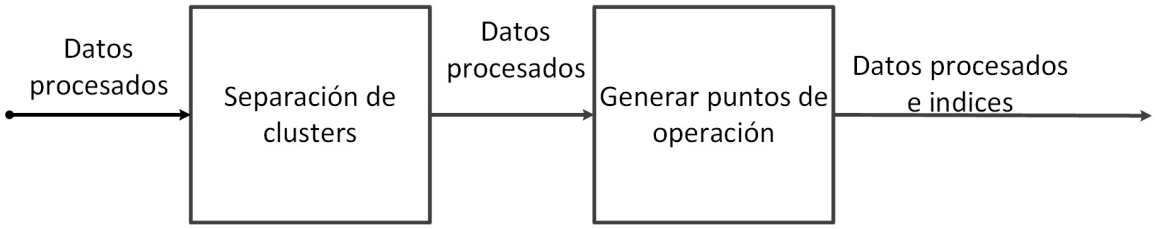

Figura 4.8: Etapas cálculo de puntos de operación.

Separación de clusters

La primera etapa consiste en separar los datos en clusters y con esto inicializar cada nodo. En la figura 4.9 se muestra el diagrama a bloques que realiza esta operación.

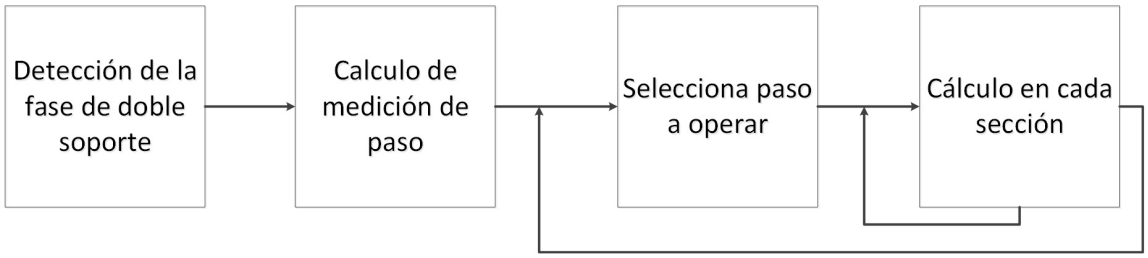

Figura 4.9: Separación de clusters

A continuación se describirán los bloques que conforman la fase de separación de cluster:

- **Detección de doble fase**

  Se tiene guardado un conjunto de datos en forma matricial, la matriz esta conformada por tres secciones: Estados, control y estado de soporte. La parte de *Estados* esta conformado por las variables X_0,Y_0,α,β_R,β_L,γ_R y γ_R, por otra parte *control* esta formado por los momentos M_1,M_2,M_3 y M_4, por ultimo la parte del estado del soporte esta conformado por una codificación binaria la cual muestra la fase de soporte de las piernas contra el suelo. El objetivo de detección de doble fase consiste en encontrar los indices que representan el inicio de un ciclo de marcha y guardar estos indices en un arreglo de números.

  En la figura 4.10 se ejemplifica las detección de doble fase obteniendo los indices de las filas de la matriz conformada por las tres secciones.

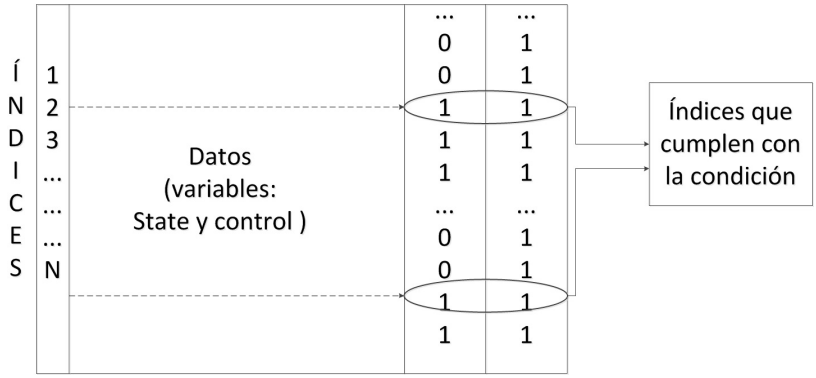

Figura 4.10: Detección de doble fase

- **Cálculo de medición de paso**

  La medición de paso consiste en obtener la distancia numérica entre dos indices guardados consecutivamente, ver figura 4.11. Las longitudes obtenidas son guardadas en un vector de valores para ser utilizadas posteriormente.

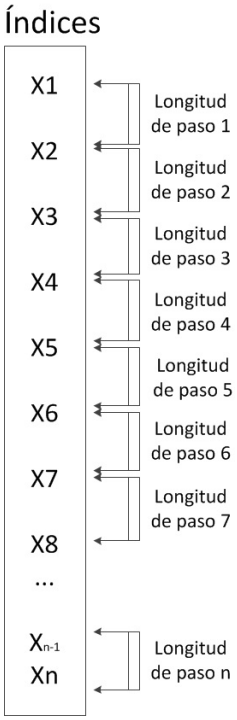

Figura 4.11: Cálculo de la longitud de paso

- **Selecciona el paso**

  Se recorre el vector de longitudes tomando cuenta la cantidad de registros existentes. Se toma un elemento y se calcula la una división entre el valor guardado y el número de clusters especificado. El cociente y el residuo serán utilizados para realizar el calculo en en la separación de cada sección.

- **Calculo y llenado de cada sección**

  En este paso se realiza una construcción de clusters en base a los indices guardados. Se calcula una cantidad de elementos que se van a añadir al cluster en base a la longitud de paso. Los elementos son descritos como números consecutivos a partir de los indices obtenidos en la detección de fase, ver figura 4.12.

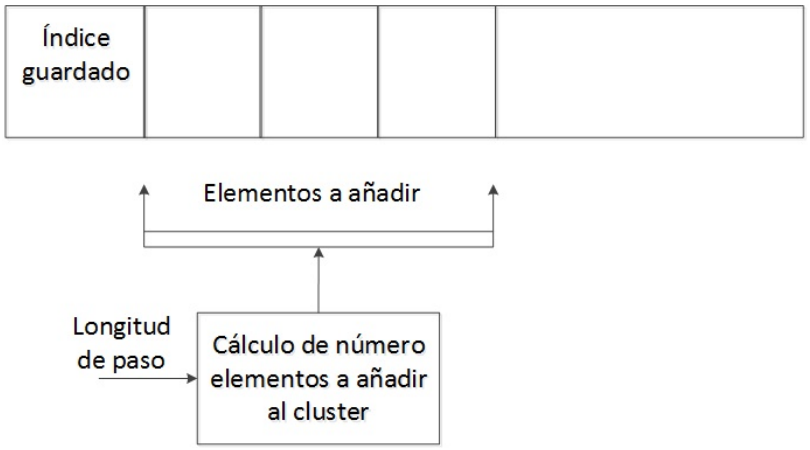

Figura 4.12: Cálculo de elementos a añadir

En la figura 4.13 se muestra el llenado de clusters en base a la cantidad total de las longitudes de paso y el número de clusters n que se establezca.

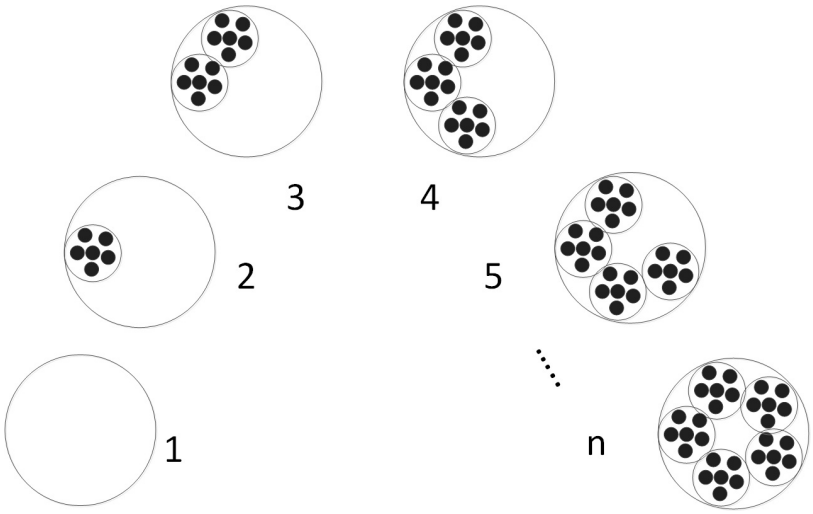

Figura 4.13: Llenado de inicial de clusters

Generación de puntos de operación

En la segunda etapa se procesa cada cluster y se calcula los puntos de operación. El diagrama a bloques de esta etapa se muestra en la figura 4.14.

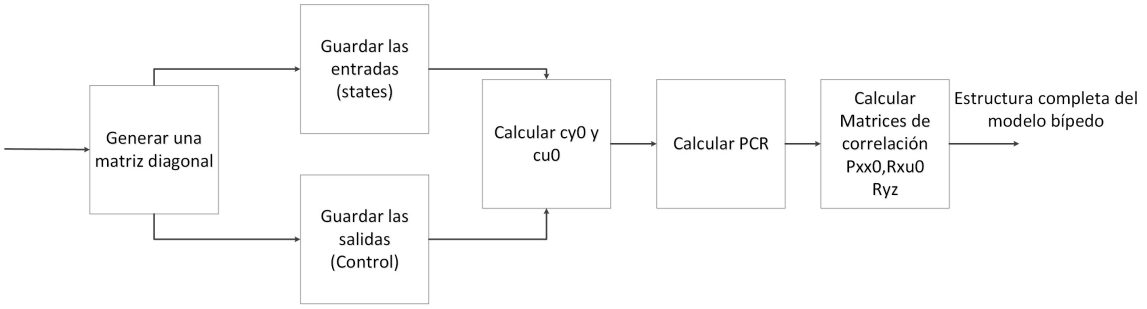

Figura 4.14: Diagrama a bloques de la generación de puntos de operación.

Se inicializa una matriz diagonal y se guardan las entradas-salidas para calcular C_{y0} y C_{u0}. Posteriormente se calculan los parámetros P_{xx0},R_{xu0} y R_{yz} para completar la estructura del modelo bípedo.

Se calculan los promedios los datos guardados, ver figura 4.15. Las variables que interesan son las posiciones X_0 y Y_0, los ángulos α,β_R,β_L, γ_R y γ_L. Todos los promedios de la parte de estados son guardados en un vector de valores denominado C_u^p. También se calcula los promedios de la parte de control, por lo que se toma en cuenta los valores de los momentos M_1,M_2,M_3 y M_4. Los promedios obtenidos son guardados en la variable C_y^p

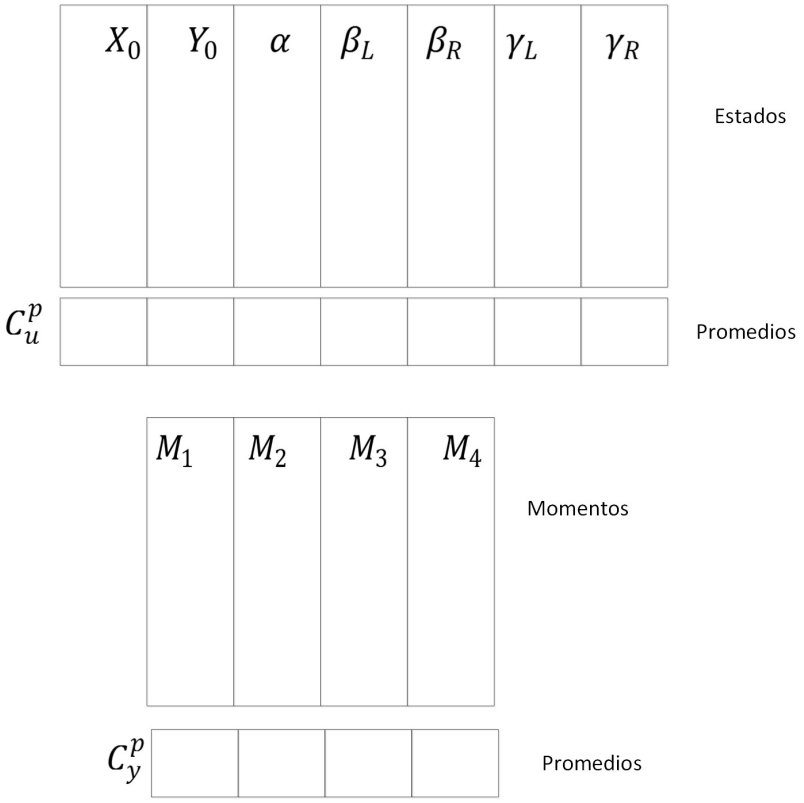

Figura 4.15: Cálculo de los promedios de la parte de estados y de control.

Posteriormente, se genera una matriz M conformada por dos matrices triangulares inferior y superior: la matriz triangular inferior tiene se describe con una diagonal de ceros y con valores ascendentes, por su parte la matriz triangular superior es el reflejo de la inferior, (ver figura 4.16).

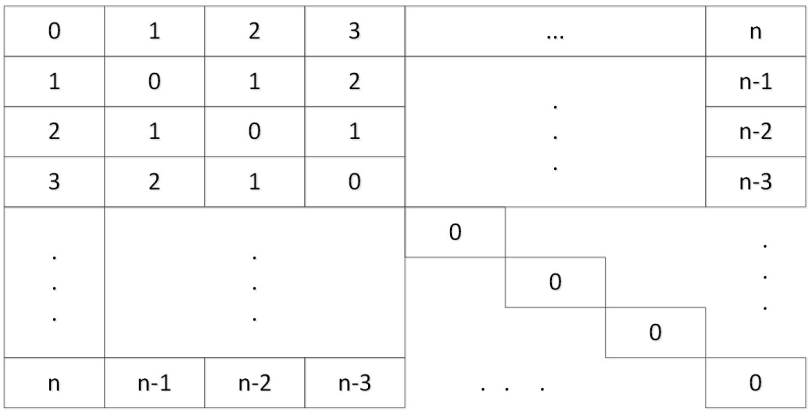

Figura 4.16: Matrices triangular M: inferior y superior.

Se recorre todos los registros guardados en la parte estados y control, el recorrido consta en tomar una fila de estados y de control, estos valores se guardan la variable

u y y respectivamente. Se calcula el vector J donde cada elemento es calculado por la ecuación 4.10 .

$$J = A + B \tag{4.10}$$

Donde A es calculado mediante la ecuación 4.11 :

$$Termino\ A = Vector_{estados} \times Matriz_{validacion} \times (Vector_{estados})^T \tag{4.11}$$

y donde el $vector_{estados}$ es calculado por la ecuación 4.12

$$vector_{estados} = u - C_u^p \tag{4.12}$$

La matriz de validación toma sólo en cuenta las variables a utilizar por lo que se selecciona la posición, ángulos y detección de soporte, discriminando las velocidades. Por su parte el termino B se calcula por medio de de la ecuación 4.13.

$$Termino\ B = Vector_{control} \times Matriz_{validacion} \times (Vector_{control})^T \tag{4.13}$$

y donde el $vector_{control}$ es calculado por 4.14

$$vector_{control} = y - C_y^p \tag{4.14}$$

La matriz de validación con tamaño(n x n) del termino B se describe como una diagonal que en cada casilla tiene valor es n+1. Resumiendo en la figura 4.17 se representa la generación del vector J.

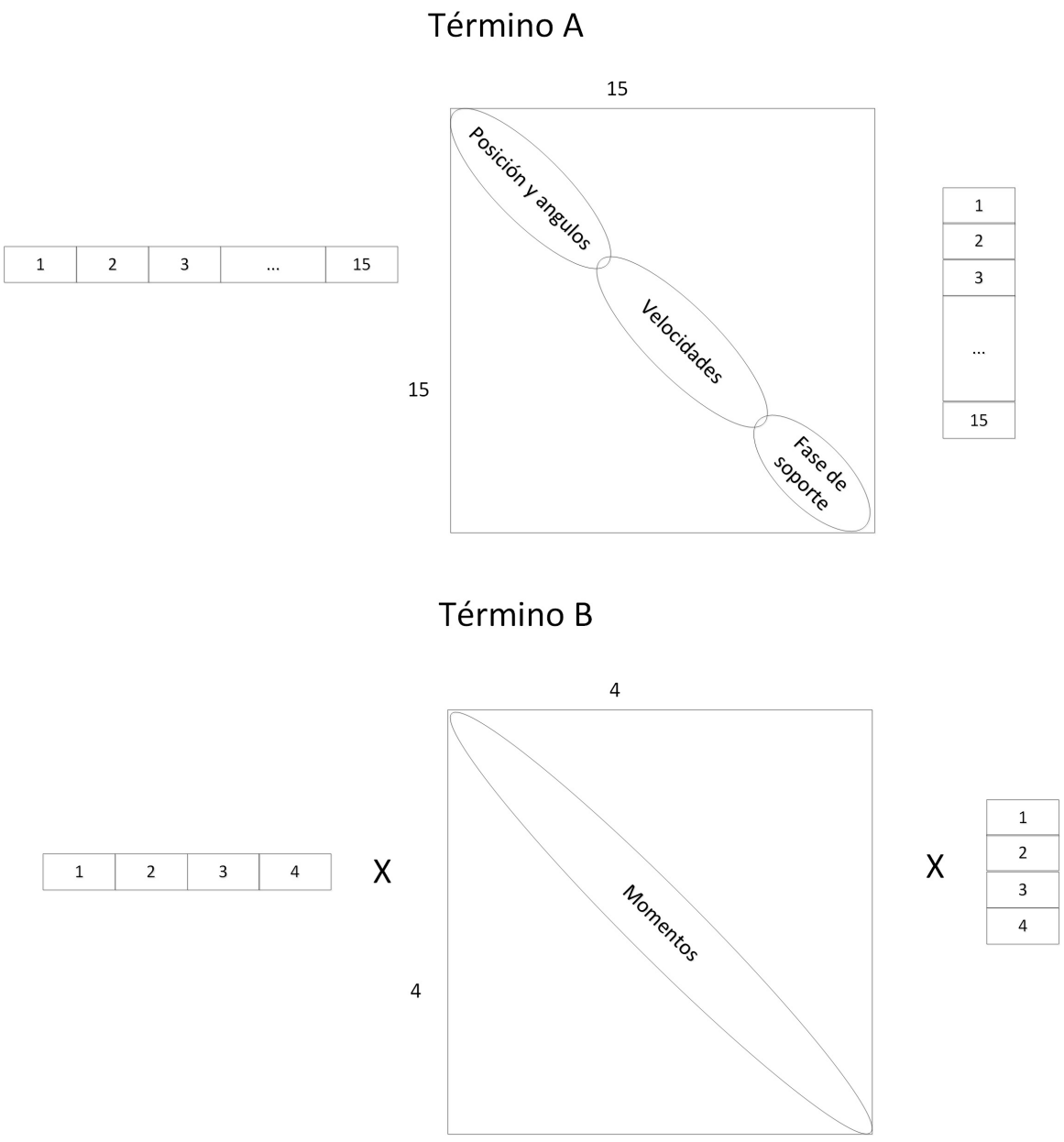

Figura 4.17: Cálculo del vector J

Una vez generado el vector J, se busca el valor ganador el cual se describe como el número de menor valor dentro del vector J. Una vez encontrado se guarda y se le denomina *winner* y se espera actualizar los parámetros C_u^p y C_y^p.

Con base a valor de *winner* se actualiza los vectores C_u^p y C_y^p. La actualización consiste en en calcular una nueva $lambda_{actual} = 1 - (1 - lambda_{inicial}) * H(k, winner)$ donde k es la posición actual del recorrido. Una vez calculado $lambda_{actual}$ se actua-

lizan C_u^p y C_y^p mediante

$$C_u^p = lambda_{actual} * C_u^p + (1 - lambda) * u \tag{4.15}$$

$$C_y^p = lambda_{actual} * C_u^p + (1 - lambda) * y \tag{4.16}$$

Una vez terminados estos pasos se procede a repetir E veces , donde E representa el número de épocas establecidas.

4.2.2.3. Cálculo de parámetros RC

Una vez calculados los puntos de operación de cada cluster, se calcula los parámetros estadísticos C_u^ρ , C_y^ρ, P_{xx}^ρ , $R_{x\tilde{u}}^\rho$, R_{yx}^ρ y R_{uu}^ρ. En la Figura 4.18 se muestra un bloque representativo de esta tarea.

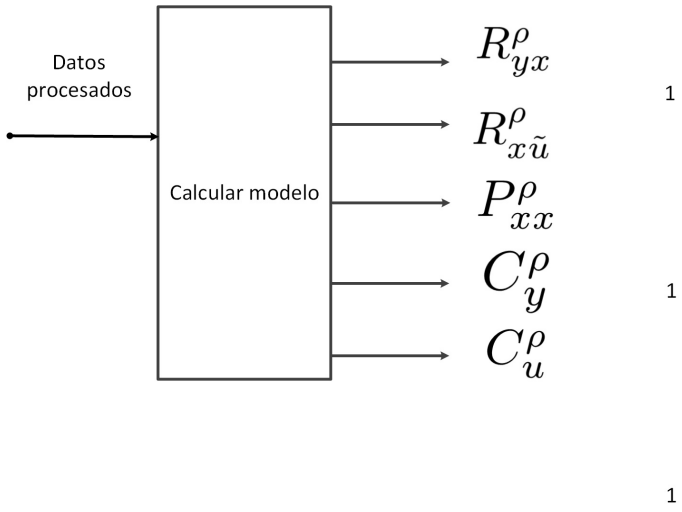

Figura 4.18: Cálculo de parámetros estadísticos

Una vez concluida la fase de aprendizaje y se cuenta con los parámetros estadísticos para el controlador RC. Se procede a establecer los datos para configurar correctamente el controlador RC. En la figura 4.19 se muestra un diagrama representativo de cada PCR, se utiliza las ecuaciones 4.1,4.2, 4.3,4.4 y 4.5 para calcular la salida del controlador.

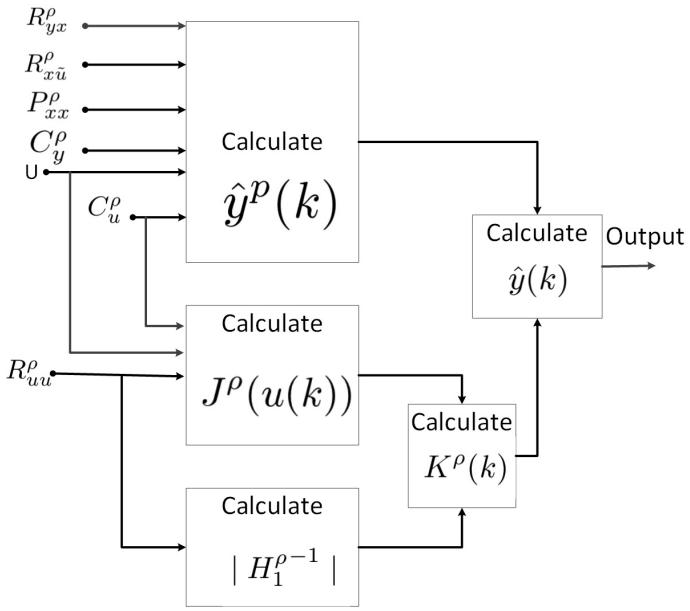

Figura 4.19: Diagrama de operación de cada PCR

Posteriormente se procede a calcular los parámetros de cada submodelo perteneciente a cada cluster. Por lo tanto se asignan los datos de la parte de estados a la variable U y la parte de control es guardada en la variable Y. Estas variables representan los datos de entrada y de salida respectivamente.

Se re-calculan los valores de C_u^p y C_u^p. Una vez que se tienen los nuevos valores se calcula nuevamente U y Y, ver figura .

Se realiza el análisis de componentes principales (ACP). ACP es una técnica utilizada para reducir la dimensionalidad de un conjunto de datos. Intuitivamente la técnica sirve para hallar las causas de la variabilidad de un conjunto de datos y ordenarlas por importancia.

Técnicamente, el ACP busca la proyección según la cual los datos queden mejor representados en términos de mínimos cuadrados. El ACP se emplea sobre todo en análisis exploratorio de datos y para construir modelos predictivos. El ACP comporta el cálculo de la descomposición en autovalores de la matriz de covarianza, normalmente tras centrar los datos en la media de cada atributo.

Se utiliza un comando de MatLab llamado *princomp*, esta función lleva a cabo el análisis de componentes principales (ACP) en la matriz de datos X n-por-p, y devuelve los coeficientes de componentes principales, también conocidos como cargas. COEF es una matriz p-por-p, cada columna que contiene los coeficientes para un componente principal. Las columnas están en orden decreciente de con base a la varianza.

SCORE, las actuales componentes principales; es decir, la representación de X en el espacio de componentes principales.

LATENT, un vector que contiene los valores propios de la matriz de covarianza de X.

Establecido lo anterior se configura el comando como se muestra a continuación:

```
[COEFF, SCORE, LATENT] = PRINCOMP(U)
```

Donde un representa los datos de entrada perteneciente a la parte de estados.

Posteriormente se toma el parámetro SCORE y con base en el número de componentes principales se escoge el número de columnas a utilizar y esta selección es guardada en la variable X.

Cuando se tiene X se procede a calcular la matriz de correlación de entrada R_{xu0} usando la siguiente ecuación:

$$R_{xu0} = \frac{X^T \times U}{N} \tag{4.17}$$

Donde N es el número de filas de la matriz U de los valores de entrada.

Es necesario generar una matriz triangular inferior con valor 1 de denominado RT con dimensiones n-por-n, donde n es la cantidad de componentes principales que se busca generar. Definidos RT se calcula una nueva matriz de correlación P_{xx0}, la cual se define como:

$$P_{xx0} = RT \times (\frac{X^T \times X}{N})^{-1} \tag{4.18}$$

Por último, se calcula el parámetro R_{yz0} definido por la ecuación siguiente:

$$R_{yz0} = \frac{Y^T \times (\sqrt{P_{xx0}})^T}{N} \tag{4.19}$$

Los parámetros R_{xu0},P_{xx0}, R_{yz0}, C_u^p y C_y^p componen un modelo final para utilizarlos dentro del controlador RC.

4.2.3. Resultados

En esta sección se describirán los resultados obtenidos de la realización de los experimentos sobre marcha normal, el cual se muestran el comportamiento de distintos parámetros que representan los ángulos de las piernas; en el experimento de aumento de velocidad se toma en cuenta la generación de un nuevo conjunto de datos ya que los datos humanos mantienen velocidad media o normal; por último en el cambio de angulo en el terreno se realiza en terreno plano y ascendente.

4.2.3.1. Marcha suave

Como se ha descrito anteriormente este experimento se realiza utilizando la caminata normal sobre terreno plano. Se muestran las gráficas de los parámetros angulares α, $\Delta\beta$, γ_L, γ_R,β_L y β_R. Ademas se compara con los controladores PD y PID para saber las diferencias y semejanzas que tienen en su comportamiento.

En la figura 4.20 se muestra una comparación de parámetros α del controlador PD, PID y RC. Este parámetro representa el movimiento del torso. Se observa que la señal del PD tiene unos picos de mayor magnitud, asimismo tanto en PD y PID tienen mayores fluctuaciones en comparación a al señal del controlador RC.

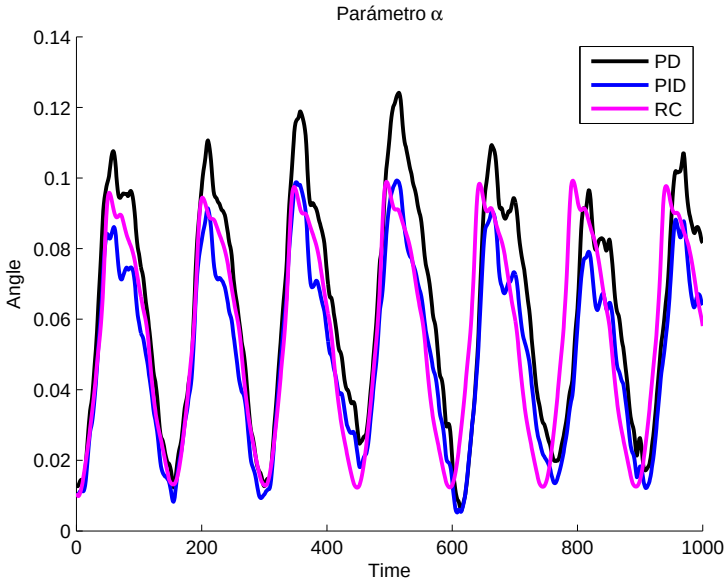

Figura 4.20: Parámetros $\alpha_{(PD)} vs \alpha_{(PID)} vs \alpha_{(RC)}$

Por otra parte, la figura 4.21 representa los parámetros γ_L y γ_R, estas dos señales representan el movimiento de la pierna cuando entra contacto con el suelo. La señal perteneciente al controlador RC cuenta con picos de menor valor que los picos de los PD y PID.

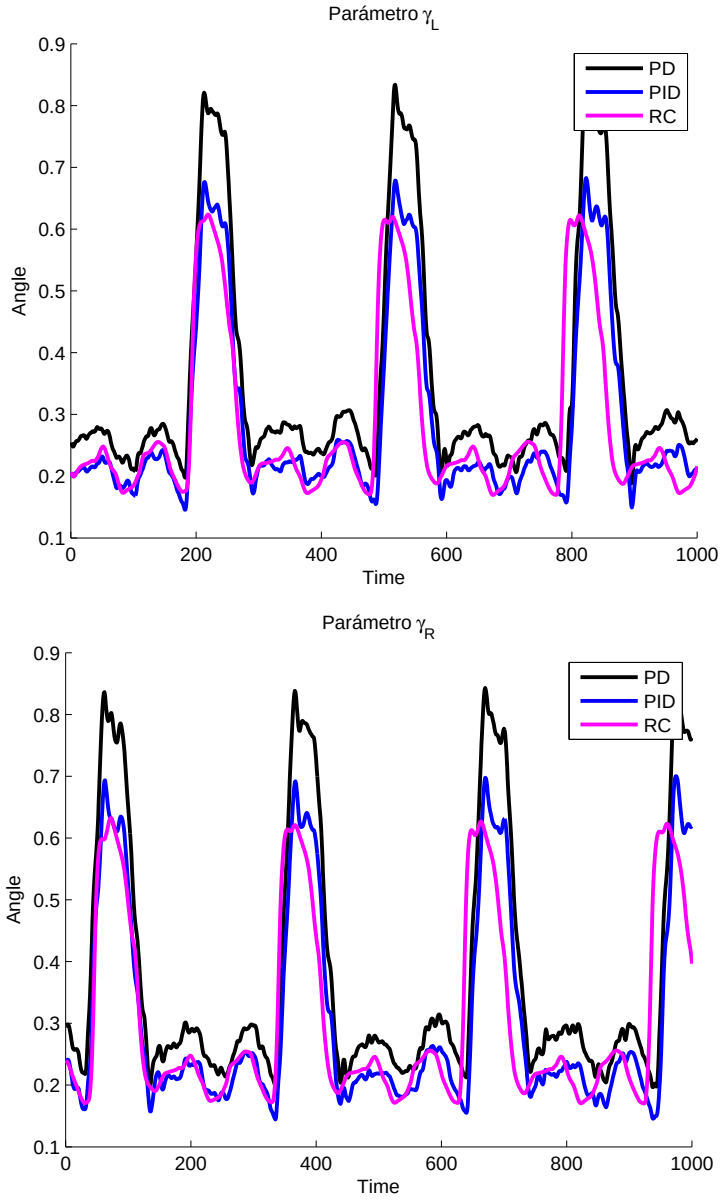

Figura 4.21: Comparación de los parámetros $\gamma_{L(PID)}$ y $\gamma_{L(RC)}$

Por ultimo en la figura 4.22 se muestra la comparación de los parámetros β_L y $beta_R$ las cuales representan el movimiento de las rodillas del robot bípedo. Se observa que se las señales del PID y del controlador RC se asemejan en mayor medida en comparación de la señal del PD.

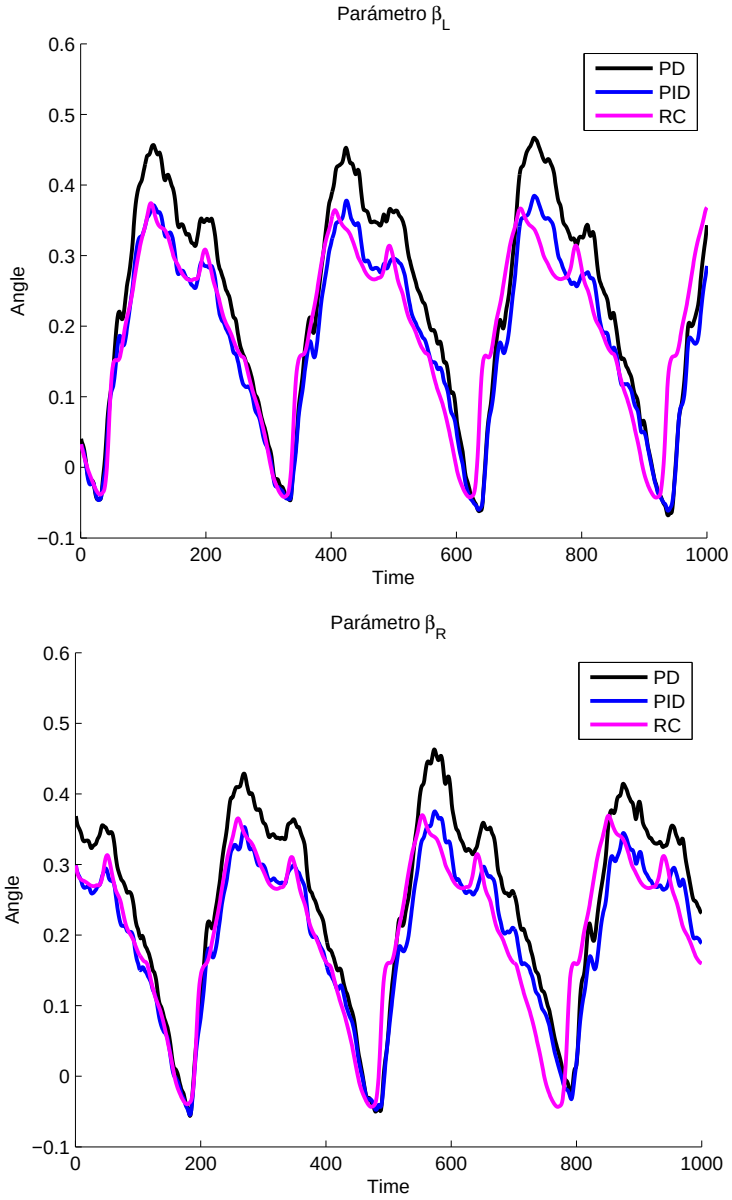

Figura 4.22: Comparación de los parámetros β_L y β_R de los controladores PD, PID y RC

Similarmente se calcula los parámetros estadísticos media, mediana, desviación estándar, valor mínimo y máximo. Estos valores ayudan a comparar de manera numérica las gráficas antes presentadas. En la tabla 4.2 se muestra los resultados obtenidos con base a las variables α,β_R,β_L, Δ_β,γ_R y γ_L.

	Media	Mediana	Desviación Estándar	Mínimo	Máximo
α	0.0405	0.0406	0.0281	2.7305e-006	0.0800
β_R	0.0644	0.0750	0.1694	-0.2089	0.3415
β_L	0.0436	0.0373	0.1762	-0.2089	0.3415
Δ_β	0.0208	0.0699	0.3308	-0.4719	0.4402
γ_R	0.2839	0.2246	0.1408	0.1696	0.6235
γ_L	0.3071	0.2300	0.1555	0.1702	0.6324

Tabla 4.2: Valores estadísticos de las distintas variables del método RC

De la tabla 4.2 se observa que el parámetro con la mayor desviación estándar es Δ_{beta}. La señal γ_L tiene el pico con mayor valor y por el contrario la señal Δ_{beta} tiene el pico más negativo.

4.2.3.2. Velocidad

Se realiza un aumento de velocidad a la marcha RC, sin embargo no se tienen conjuntos de datos que representen una velocidad mayor. Por tal motivo se procede a generar los conjuntos de datos con una velocidad superior. Se toma como muestra una señal de marcha y se replica con factor multiplicativo establecido dentro de la misma longitud de tiempo.

Una vez teniendo los conjuntos de datos se procedió a introducirlos al controlador basado en RC para que se calcularán los paramtros que describen la marcha humana. Tomando en cuenta la revisión realizada en [56], se tiene que la velocidad promedio de niños es de 114 cm/s convirtiéndolos a km/h y a m/s se tiene como resultado 1.14 m/s y 4.1 Km/h. Esta velocidad se toma como base para poder calcular las velocidades resultantes. El resultado de aumentar la velocidad de los datos humanos sobre terreno plano se muestra en la tabla 4.3.

	RC	PD	PID
	Terreno Plano 0 °	Terreno Plano 0 °	Terreno Plano 0°
Velocidad Normal	1.14 $\frac{m}{s}$	0.1303 $\frac{m}{s}$	0.1303 $\frac{m}{s}$
Velocidad Media	2.28 $\frac{m}{s}$	0.1957 $\frac{m}{s}$	0.2073 $\frac{m}{s}$
Velocidad Máxima	4.56 $\frac{m}{s}$	0.2612 $\frac{m}{s}$	0.2843 $\frac{m}{s}$

Tabla 4.3: Velocidad normal y máxima en 0 grados de inclinación

Como se han generado los datos de rampa ascendente idealmente se tiene que la velocidad normal en terreno plano es igual que la ascendente, sin embargo se sabe que no es así, por lo cual se revisó [57] y se concluye que la velocidad se reduce aproximadamente en 5 % sobre rampas, por tal motivo se le adiciona una tolerancia con este valor al momento de reportar los resultados en la tabla 4.4.

	RC	PD	PID
	Terreno Ascendente $\theta_{Ascmax} = 20.36$ °	Terreno Ascendente 2.2473 °	Terreno Ascendente 2.3451 °
Velocidad Normal	1.14 $\frac{m}{s}$ ± 5 %	0.0977 $\frac{m}{s}$	0.1153 $\frac{m}{s}$
Velocidad Media	2.28 $\frac{m}{s}$ ± 5 %	0.1707 $\frac{m}{s}$	0.1850 $\frac{m}{s}$
Velocidad Máxima	4.56 $\frac{m}{s}$ ± 5 %	0.2437 $\frac{m}{s}$	0.2547 $\frac{m}{s}$

Tabla 4.4: Velocidad normal, media y máxima en la máxima inclinación.

De la tabla 4.4 se observa que el controlador RC se puede aumentar en gran medida la velocidad en comparación a los controlador PD y PID que sólo soporta incrementos de velocidad de menor tamaño.

4.2.3.3. Robustez

Para analizar la robustez se tiene en cuenta el ángulo de inclinación del plano inclinado. Sin embargo el conjunto de datos de marcha humana representa una caminata sobre terreno plano. Por tal motivo se modifica los parámetros β_R, β_L, γ_R y γ_L para generar una marcha con cierto ángulo.

Utilizando el método de ajuste manual de las señales se llego a una inclinación que es estable y que a la perspectiva visual se asemejaba a un movimiento natural de caminata humana en pendiente ascendente. Para calcular el ángulo se tiene el siguiente sistema de coordenadas, ver figura 4.23:

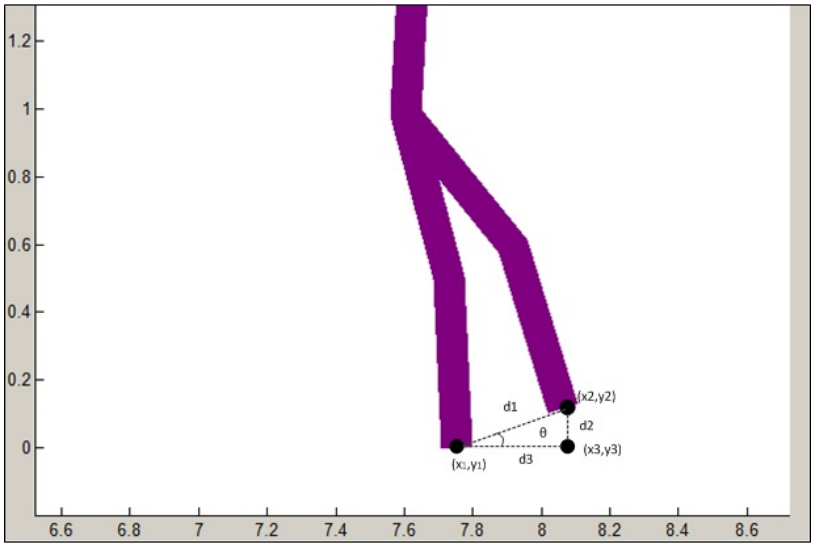

Figura 4.23: Comparación de los parámetros $\gamma_{R(PID)}$ y $\gamma_{R(RC)}$

Utilizando el cálculo de distancia entre dos puntos se calculan las distancias d_1, d_2 y d_3 por medio de

$$d1 = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \tag{4.20}$$

	PD	PID	RC
Velocidad normal	0.0977 $\frac{m}{s}$	0.1153 $\frac{m}{s}$	1.14 $\frac{m}{s}$
Velocidad máxima	0.2437 $\frac{m}{s}$	0.2547 $\frac{m}{s}$	4.56 $\frac{m}{s}$

Tabla 4.5: Velocidades normales y máximas para pendiente ascendente.

$$d2 = \sqrt{(x1 - x3)^2 + (y1 - y3)^2} \tag{4.21}$$

$$d3 = \sqrt{(x2 - x3)^2 + (y2 - y3)^2} \tag{4.22}$$

Posteriormente utilizando la ley se los senos se calcula en ángulo formado entre las dos rectas $d1$ y $d3$.

$$\theta = seno^{-1}(\frac{d_2}{d_1}) \tag{4.23}$$

Teniendo los resultados de los angulos obtenidos se procede a presentar los resultados, por tal motivo se establece las velocidades normales y máximas para pendiente ascendente en la tabla 4.5.

Una vez establecidos las velocidades se presenta en las gráficas de figura 4.24 una comparación de los ángulos máximos alcanzados en terreno ascendente.

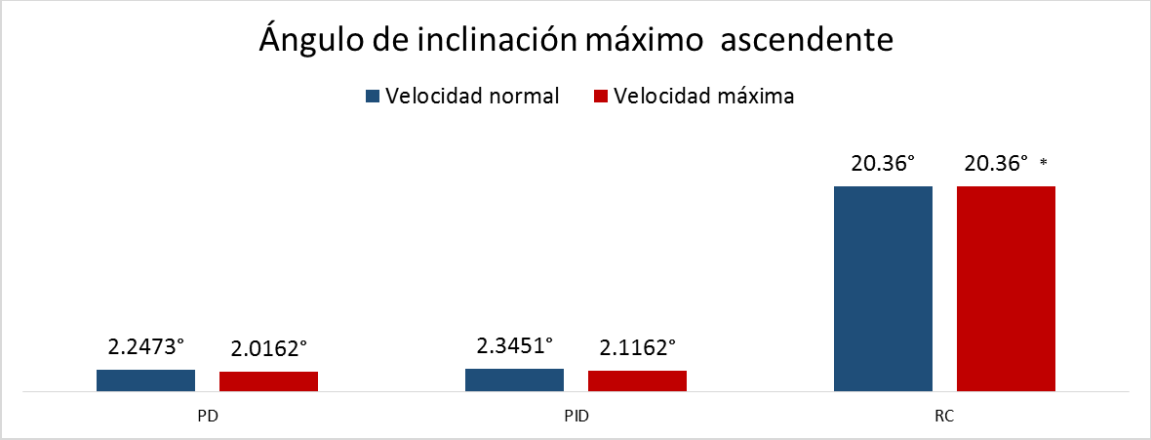

Figura 4.24: Comparación de los distintos ángulos de inclinación en los controladores PD,PID y RC.

Se observa que el ángulo de inclinación del controlador RC es muy superior en comparación del controladores PD y PID. Estos últimos tienen una mínima diferencia entre el angulo de inclinación a velocidad normal y velocidad máxima.

4.3. K-means para el análisis de datos de marcha humana

Se describe la utilización del algoritmo K-means para el control de la marcha bípeda de un robot, este algoritmo de agrupamiento es utilizado en gran medida ya que cuenta con una rapidez de implementación, eficiente y termina en óptimos locales. Se utilizará la estructura el controlador RC para realizar los experimentos modificando la etapa de clustering basado en SOM por la técnica K-means.

4.3.1. Descripción del algoritmo

K-means es uno de los más conocidos algoritmos de agrupamiento, sigue una forma fácil y simple para dividir una base de datos dada en k grupos (establecidos a priori). La idea principal es definir k centroides (uno para cada grupo) y luego tomar cada punto de la base de datos y situarlo en la clase de su centroide más cercano. El próximo paso es recalcular el centroide de cada grupo y volver a distribuir todos los objetos según el centroide más cercano. El proceso se repite hasta que ya no hay cambio en los grupos de un paso al siguiente.

Matemáticamente este algoritmo tiene como objetivo minimizar una función objetivo conocida como función de error al cuadrado dada por:

$$J(V) = \sigma_{i=1}^{c}\sigma_{i=1}^{ci}(||x_i - V_j||)^2 \tag{4.24}$$

donde $||x_i - v_j||$ es la distancia euclídea entre x_i y v_j. c_i el número de puntos de datos en i clúster. c es el número de centros de los conglomerados.

Pasos algorítmicos para k-means clustering

Establezca $X = x_1, x_2, x_3, ..., x_n$ el conjunto de puntos de datos y $V = v_1, v_2, ..., V_c$ el conjunto de los centros.

1) Seleccione al azar c centros de los conglomerados.

2) Calcular la distancia entre cada punto de datos y centros de los conglomerados.

3) Asignar el punto de datos a la agrupación centro cuya distancia desde el centro del cúmulo es mínimo de todos los centros de los conglomerados ..

4) Volver a calcular el nuevo centro de clúster mediante:

$$v_i = \frac{1}{c_i}\Sigma_{j=1}^{c_i}x_i \tag{4.25}$$

5) Vuelva a calcular la distancia entre cada punto de datos y nuevos centros de los conglomerados obtenidos.

6) Si fue reasignado ningún punto de datos y luego se detiene, de lo contrario repita desde el paso 3).

Siendo más representativos en la figura 4.25 se muestra el diagrama de flujo del algoritmo K-means.

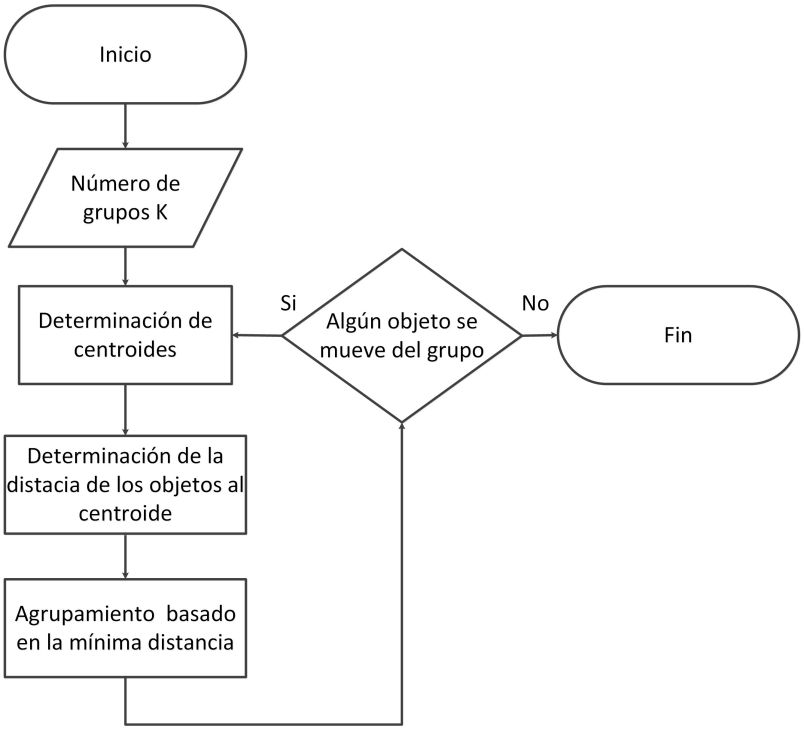

Figura 4.25: Diagrama de flujo del algoritmo K-means

Algunas ventajas del algoritmo K-means son las siguientes:

Ventajas

- Rápido, robusto y fácil de entender.
- Relativamente eficiente: donde n es el número de objetos, k es numero de clusters, d es número de dimensiones de cada objeto, y t es el numero de iteraciones. Normalmente, k, t, d $<<$ n.
- Se obtienen mejores resultados cuando los conjuntos de datos son distintas o bien separados unos de otros.

Desventajas

- El algoritmo de aprendizaje requiere la especificación a priori del número de centros de los clusters.
- Existe un uso de asignación exclusiva. Si hay dos datos muy juntos entonces K-means no será capaz de resolver que hay dos grupos.
- El algoritmo de aprendizaje no es invariante a transformaciones no lineales, es decir, con diferente representación de datos obtenemos diferentes resultados (datos representados en forma de coordenadas cartesianas y coordenadas polares darán resultados diferentes).

- El algoritmo de aprendizaje proporciona el óptimo local de la función de error al cuadrado.

- La elección al azar del centro de clúster en ocasiones no puede llevar a un resultado satisfactorio .

Utilización de herramienta de MatLab para K-means

El software MatLab contiene una gran cantidad de herramientas, para hacer el análisis de agrupamiento. En este trabajo se utiliza la función *kmeans*.

$(idx, C) = kmeans(X, k)$

Realiza el agrupamiento k-means para dividir las muestras de la matriz de datos X $n \times n$ en k grupos y devuelve idx que contiene indices de cluster, las filas de X corresponden a los puntos y las columnas a las variables. C devuelve la posición de los centros del clusters. Un ejemplo básico es el presentado a continuación:

```
X = [randn(100,2)+ones(100,2);...
randn(100,2)-ones(100,2)];
opts = statset('Display','final');

[idx,ctrs] = K-means(X,2,...
'Distance','city',...
'Replicates',5,...
'Options',opts);


plot(X(idx==1,1),X(idx==1,2),'r.','MarkerSize',12)
hold on
plot(X(idx==2,1),X(idx==2,2),'b.','MarkerSize',12)
plot(ctrs(:,1),ctrs(:,2),'kx',...
'MarkerSize',12,'LineWidth',2)
plot(ctrs(:,1),ctrs(:,2),'ko',...
'MarkerSize',12,'LineWidth',2)
legend('Cluster 1','Cluster 2','Centroids',...
'Location','NW')
```

El resultado de estas lineas de código es la que se muestra en la figura 4.26.

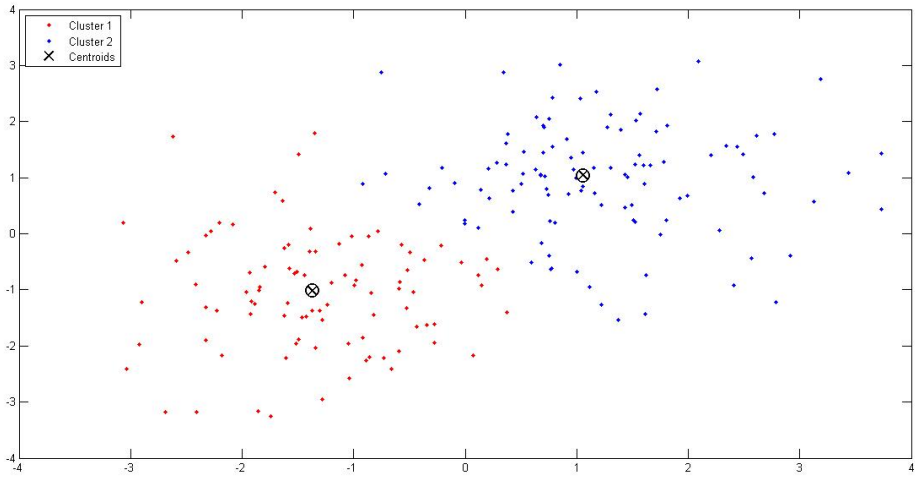

Figura 4.26: Ejemplo básico de manejo de herramienta K-means en MatLab

Se observa que el el algoritmo K-means es aplicado y divide el conjunto de datos en dos clusters de color azul y rojo, así también se muestra la localización de sus centros.

4.3.2. K-means aplicado a la marcha robótica bípeda

En este trabajo la aplicación del método k-means en robots bípedos se toma en cuenta la estructura del método *RC*. Por lo que se remplaza la parte de la separación de datos y cálculo de puntos de operación por la parte encargada de agrupamiento.

Para realizar el análisis de clustering se toma en cuenta los datos antes de la separación en el algoritmo RC, ya que se necesitan guardar todos los datos para aplicar agrupamiento K-means. A continuación se presenta lineas de código que realizan la agrupación K-means.

```
function I_K=FCMeans(data,I,Nop)
p1=[];
p2=[];
p3=[];
for i=1:Nop
cluster= I{:,i}; % toma los datos del cluster i
P1=data.state(cluster,1); %alfa
P2=data.state(cluster,2); %beta
P3=data.state(cluster,3); %beta
p1=[p1;P1]; %contatena la columna de P1 (alfas)
p2=[p2;P2]; %contatena la columna de P2 (betaL)
p3=[p3;P3]; %contatena la columna de P3 (betaR)
```

```
end
dato=[p1 p2 p3];

%figure(3)
[center,U,obj_fcn] = K-means(dato, Nop); % realiza
maxU = max(U);
index=[];
hold on
for i=1:Nop
index=find(U(i,:) == maxU);
I_K{i}=index;
scatter3(dato(index,1),dato(index,2),dato(index,3),'.') %grafica
index=[];
end
```

Lo primero que se realiza es generar todo un conjunto de datos tomando en cuenta los datos del controlador RC. Teniendo el conjunto de datos se procede a aplicar el algoritmo para generar los cluster. Una vez realizado la agrupación K-means se toman los nuevos indices que se utilizan para calcular los nuevos parámetros estadísticos que se utilizarán dentro del controlador. En la figura 4.27 se muestra la agrupación representada en las variables α, β_R y β_L.

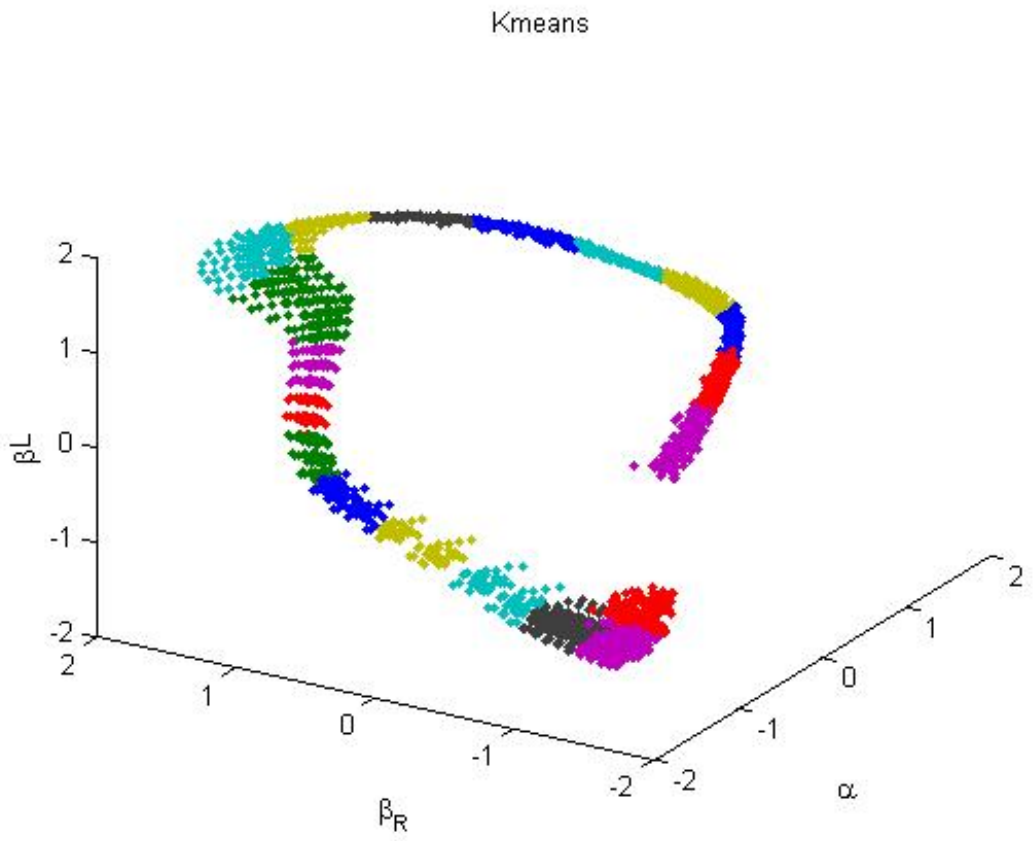

Figura 4.27: Agrupamiento K-means

Se observa que la distribución de los datos es diferente al presentado en la agrupación de RC y por consecuencia los valores de los parámetros estadísticos son diferentes. Estas diferencias hacen que la marcha cambie a la mostrada en el controlador RC.

4.3.3. Resultados

Los experimentos realizados son similares a los utilizados en el controlador RC. Se realiza un análisis en marcha normal sobre terreno plano, aumento de velocidad y un cambio de ángulo del terreno de marcha. Estos experimentos servirán para que se realice una comparación entre el controlador RC y el basado en K-means.

4.3.3.1. Marcha normal

Conforme la marcha normal se describe el comportamiento de los parámetros α, β_R y β_L, Δ_β,γ_R y γ_L, en donde cada parámetro representa los ángulos formados en el torso, rodillas y tobillo. Se comparará con los controladores clásicos PD y PID para obtener las diferencias entre las señales.

En la figura 4.29 se muestra el parámetro α y se describe como una señal periódica con una amplitud casi estable, además se tiene un incremento en los valores de los picos de la señal α en comparación del controlador RC.

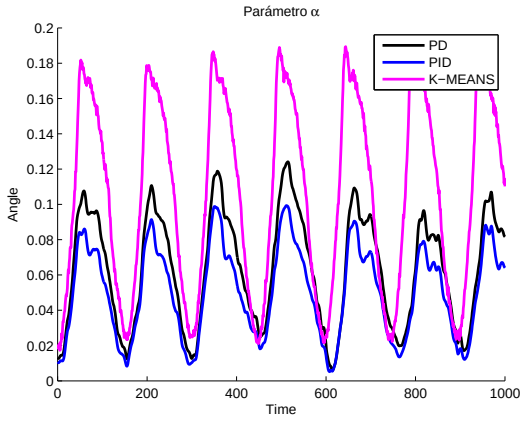

Figura 4.28: Comparación de parámetros α de los controladores PD, PID y K-means

Por otra parte en la figura 4.29 se muestra el parámetro β_L y β_R. Las parámetros se describen como señales periódicas cuyos picos de la señal de RC son mayores que los del PD y PID.

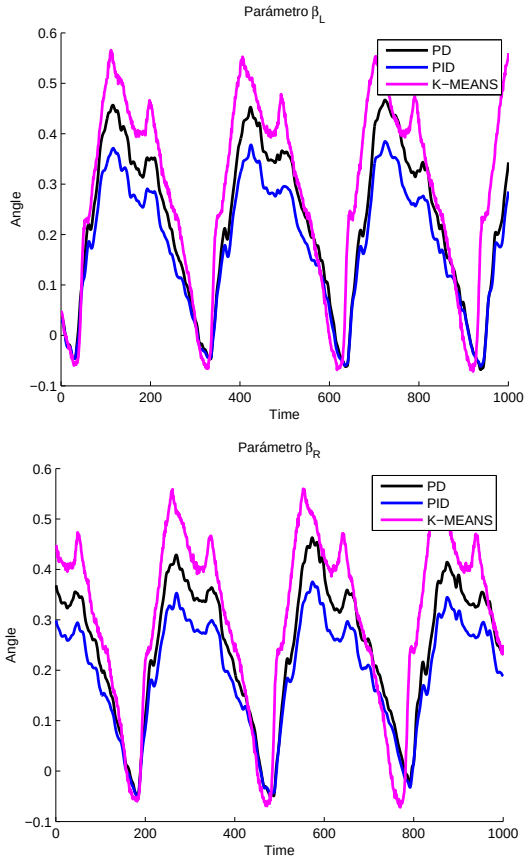

Figura 4.29: Comparación de parámetros β de los controladores PD, PID y K-means

Por último en la figura 4.30 se muestra los parámetros γ_R y γ_L estos representan el movimiento asociado a la parte de la pierna en contacto con el suelo. El valor de los picos del controlador RC se encuentra entre los valores de los picos de los controladores PD y PID.

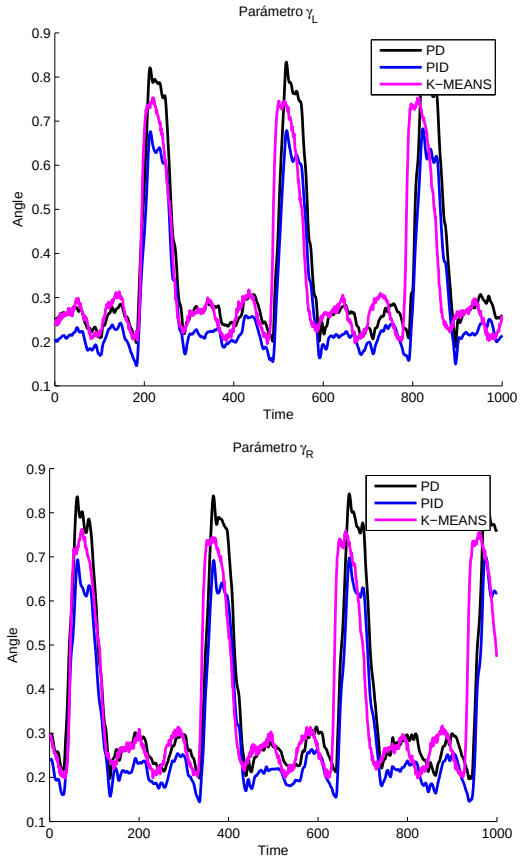

Figura 4.30: Comparación de parámetros γ de los controladores PD, PID y K-means

Las señales anteriores muestran el comportamiento de los parámetros en el tiempo y de como representan la marcha bípeda, sin embargo se desea conocer algunos parámetros como media, mediana, desviación estándar, el máximo y mínimo. Estos valores ayudan describir la marcha bípeda con K-means sin tomar en cuenta el tiempo. En la tabla 4.6se muestra los valores obtenidos en base a los parámetros α, β_R y β_L, Δ_β,γ_R y γ_L.

	Media	Mediana	Desviación Estándar	Mínimo	Máximo
α	0.1927	0.1981	0.1024	0.0269	0.3610
β_R	0.4856	0.5379	0.2983	-0.1500	0.9590
β_L	0.5011	0.5895	0.3023	-0.1413	0.9474
Δ_β	-0.0155	-0.0606	0.5460	-0.8598	0.8428
γ_R	0.4346	0.3411	0.2184	0.2205	0.9693
γ_L	0.4452	0.3436	0.2240	0.2175	0.9712

Tabla 4.6: Valores estadísticos de las distintas variables del controlador basado en K-means

De la tabla anterior se tiene que Δ_β tiene una mayor dispersión en base a su

promedio, esto se comprueba con el valor de su desviación estándar. La señal que tiene el mayor pico positivo es β_L, por lo contrario la señal con un pico negativo mayor es Δ_β.

4.3.3.2. Velocidad

Se aborda el aumento de velocidad en la marcha para el controlador basado con K-means. Se toman los mismos datos generados para operar dentro del controlador RC. Después de introducir estos datos generados se obtuvo que el controlador con K-means soporta un factor multiplicativo de 2 y 4 similar al RC.

Los resultados se presentan en la figura 4.31 con un factor de 2, asimismo en la figura 4.32 muestra un factor multiplicativo de 4. Ambas señales están distribuidas sobre una longitud de 1000 datos que representan un ciclo de marcha. Los factores son valores de potencia 2, sin embargo el factor 8 corresponde a una velocidad poco natural para un robot bípedo, por tal motivo se omite y se presenta factores factibles a utilizar en un robot bípedo.

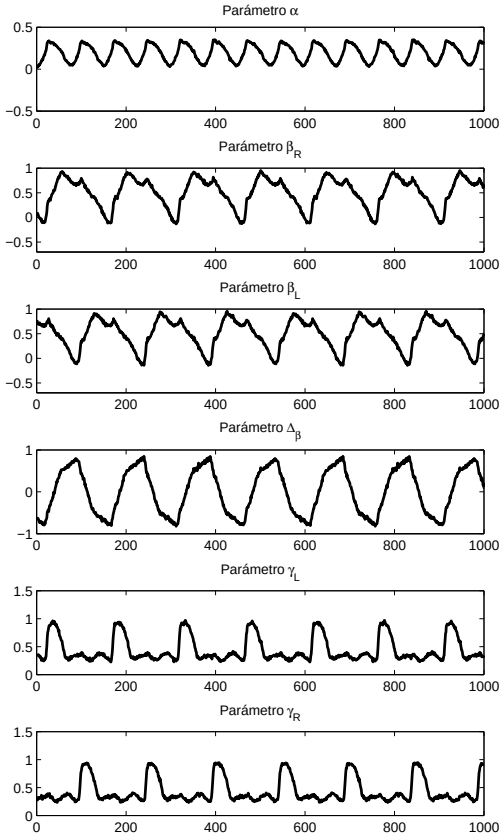

Figura 4.31: Velocidad aumentada con el factor 2 para K-means

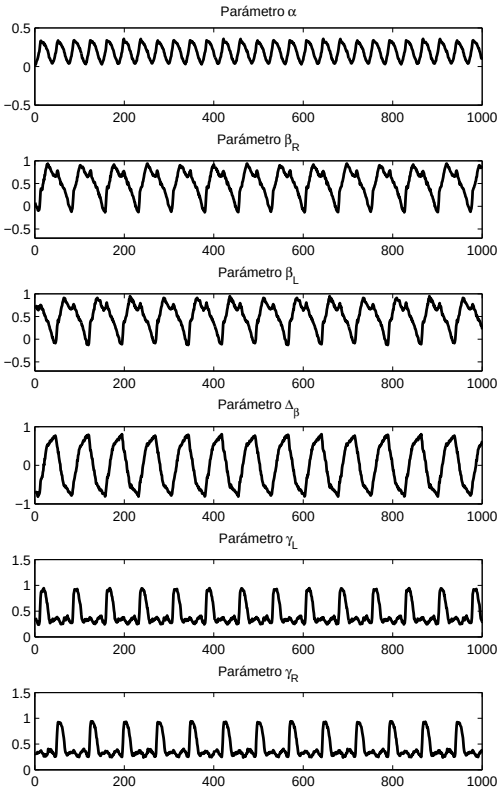

Figura 4.32: Velocidad aumentada con el factor 4 para K-means

Se observa que las velocidades aumentadas con factor 2 y 4 son las mismas que el controlador RC. Esto se debe a que se toma el mismo conjunto de datos para entrenar ambos controladores.

En la tabla 4.7 se muestra las velocidades reportadas del algoritmo K-means en comparación del PD y PID.

	K-MEANS	PD	PID
	Terreno Plano 0 °	Terreno Plano 0 °	Terreno Plano 0°
Velocidad Normal	1.14 $\frac{m}{s}$	0.1303 $\frac{m}{s}$	0.1303 $\frac{m}{s}$
Velocidad Media	2.28 $\frac{m}{s}$	0.1957 $\frac{m}{s}$	0.2073 $\frac{m}{s}$
Velocidad Máxima	4.56 $\frac{m}{s}$	0.2612 $\frac{m}{s}$	0.2843 $\frac{m}{s}$

Tabla 4.7: Velocidad normal, media y máxima en 0 grados de inclinación

Se observa que las velocidades del controlador basado en K-means son las mismas que las reportadas con el controlador, esto debido a que el conjunto de entrenamiento

	PD	PID	K-means
Velocidad normal	0.0977 $\frac{m}{s}$	0.1153 $\frac{m}{s}$	1.14 $\frac{m}{s}$
Velocidad máxima	0.2437 $\frac{m}{s}$	0.2547 $\frac{m}{s}$	4.56 $\frac{m}{s}$

Tabla 4.8: Velocidades normales y máximas para pendiente ascendente para el algoritmo K-means

es el mismo. Por lo tanto mejora en gran medida en comparación a los controladores PD y PID.

4.3.3.3. Robustez

Se realizan experimentos para conocer la robustez del controlador basado en K-means ante una pendiente ascendente y nuevamente se toman en cuenta los datos generados para el controlador RC. Se toman en cuenta las velocidades normal y máxima para realizar estos experimentos.

En la tabla 4.8 se muestra la velocidades normal y máxima utilizadas para el controlador K-means y los utilizados en PD y PID. Estas velocidades se toman en cuenta para conocer la velocidad y su respectivo ángulo de inclinacion.

Aunque las velocidades son similares al análisis anterior del controlador RC, el ángulo de inclinación es diferente debido a que los parámetros β y γ influyen en la posición de marcha bípeda, ya que los parámetros β son los que tienen la información de los ángulos de las rodillas y γ representa la parte de la pierna en contacto con el suelo.

En la figura 4.33 se muestra una comparación que se realiza al comparar los controladores PD y PID con el controlador basado en K-means.

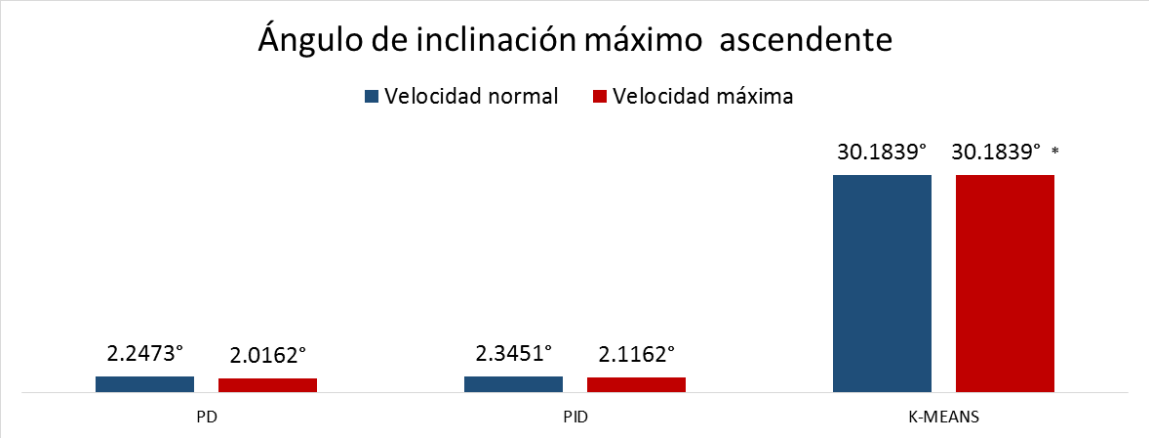

Figura 4.33: Comparación de los distintos ángulos de inclinación en los controladores PD,PID y K-means.

Se observa que usando K-means se tiene un ángulo de inclinacion superior al que

brindan los controladores clásicos (PD y PID), este ángulo reportado es incluso mayor que el presentado por el RC.

4.4. Fuzzy c-means para el análisis de datos de marcha humana

En esta sección se presenta la utilización del algoritmo Fuzzy c-means (FCM) para el control de la marcha de un robot bípedo. Este algoritmo es la mejora del controlador K-means anteriormente presentad. Sus mejoras principales radican en que un punto de la muestra tiene un grado de pertenencia a cada cluster generado contrario a lo que brinda el K-means que su pertenencia es exclusiva.

4.4.1. Descripción del algoritmo

Fuzzy c-means funciona mediante la asignación de la pertenencia a cada punto de datos correspondiente a cada centro de la agrupación sobre la base de la distancia entre el centro de la agrupación y el punto de datos.

Si los datos están cerca del centro de un cluster es más su pertenencia. Es evidente que la suma de los componentes de cada punto de datos debe ser igual a uno. Después de cada iteración centros de membresía y de racimo se actualizan de acuerdo con la fórmula:

$$\mu_{ij} = \frac{1}{\Sigma^c k = 1(d_{ij}/d_{ik})^{(\frac{2}{m-1})}} \tag{4.26}$$

$$v_j = \frac{\Sigma_{i=1}^n (\mu_{ij})^m x_i}{\Sigma_{i=1}^n (\mu_{ij})^m}, \forall j = 1, 2, 3, ..., c \tag{4.27}$$

Donde n es el número de puntos de datos.
v_j representa la j^{th} centro del cluster.
m es el índice de borrosidad, $m\epsilon[1, \infty]$.
c representa el número del centro de cluster.
μ_{ij} representa la miembros de i^{th} datos para j^{th} centro del cúmulo.
d_{ij} representa la distancia euclídea entre i^{th} datos y j^{th} centro del cluster.

El principal objetivo del algoritmo fuzzy c-means es minimizar la expresión:

$$J(U, V) = \Sigma_{i=1}^n \Sigma_{j=1}^c (\omega_{ij})^m ||x_i - v_j||^2 \tag{4.28}$$

Donde
$||x_i - v_j||$ es la distancia euclídea entre i^{th} datos y j^o centros del cúmulo.

Pasos algorítmicos para Fuzzy c-means clustering

Se establece $X = x_1, x_2, x_3..., x_n$ el conjunto de puntos de datos y $V = v_1, v_2, v_3..., v_c$ el conjunto de los centros.

- 1) Seleccione al azar c centros de los conglomerados.
- 2) Calcular la pertenencia difusa ω_{ij} usando:

$$\mu_{ij} = \frac{1}{\Sigma^c k = 1(d_{ij}/d_{ik})^{(\frac{2}{m-1})}} \tag{4.29}$$

- 3) Calcular los centros difusos v_j usando:

$$v_j = \frac{\Sigma_{i=1}^n (\omega_{ij})^m x_i}{\Sigma_{i=1}^n (\omega_{ij})^m}, \forall j = 1, 2, 3, ..., c \tag{4.30}$$

- Repita el paso 2) y 3) hasta que el mínimo J se consigue valor o $||T^{(k+1)} - U^{(k)}|| < \beta$ donde:
  k es el paso de iteración.
  β es el criterio de terminación entre $[0, 1]$.
  $U = (\mu_{ij})n \times c$ es la matriz de pertenencia difusa.
  J es la función objetivo.

Ventajas

- 1) Da mejor resultado para el conjunto de datos superpuestos y comparativamente mejor que el algoritmo K-means.
- 2) A diferencia de K-means en punto de datos debe pertenecer exclusivamente a un centro del cluster aquí un punto de datos se le asigna la pertenencia a cada centro de clúster como resultado el punto de datos puede pertenecer a más de un centro del cluster.

Desventajas

- Especificación inicial del número de clusters.
- Si se especifica una beta con menor valor de β se puede obtener un mejor resultado pero a expensas de un número mayor de iteraciones.

Utilización de herramienta de MatLab para K-means

Se utiliza la herramienta de MatLab para realizar el análisis de cluster Fuzzy c-means. El comando a utilizar es fcm y se utiliza de la siguiente manera.

$[centros, T, obj_fcn] = fcm(datos, cluster_n)$ aplica el método de agrupamiento fuzzy c-means para un determinado conjunto de datos.

Los argumentos de entrada de esta función son:

datos: conjunto de datos a agruparse; cada fila es un punto de datos de la muestra.

$cluster_n$: número de grupos (más de uno).

Los argumentos de salida de esta función son:

centros: matriz de centros de los conglomerados finales donde cada fila proporciona las coordenadas del centro

U: matriz final partición difusa (o matriz de función de pertenencia).

obj_fcn: valores de la función objetivo durante iteraciones.

Un ejemplo básico de utilización de la herramienta de MatLab se muestra a continuación.

```
data = rand(100, 2);
[center,U,obj_fcn] = fcm(data, 2);
plot(data(:,1), data(:,2),'o');
maxU = max(U);
index1 = find(U(1,:) == maxU);
index2 = find(U(2, :) == maxU);
line(data(index1,1),data(index1, 2),'linestyle','none',...
'marker','*','color','g');
line(data(index2,1),data(index2, 2),'linestyle','none',...
'marker', '*','color','r');
```

De este ejemplo se observa que el algoritmo FCM divide los datos en dos clusters y para graficarlos se buscan los mayores valores de pertenencia de los datos con base al cluster 1 o 2. La graficación de este ejemplo se observa en la figura 4.34.

4.4.2. FCM aplicado a la caminata de robot bípedo

En este trabajo la aplicación del método fuzzy c-means (FCM) en robots bípedos se toma en cuenta la estructura del método RC, de modo que se remplaza la parte de la separación de datos y cálculo de puntos de operación por la parte encargada de agrupamiento fcm.

Para realizar el análisis de clustering se toma en cuenta los datos de separación del RC de forma total, ya que se necesitan guardar todos los datos para después aplicar agrupamiento fuzzy c-means.

A continuación se presenta lineas de código que realizan la agrupación K-means.

```
function I_K=KM(data,I,Nop)
p1=[];
p2=[];
p3=[];
for i=1:Nop
cluster= I{:,i}; % toma los datos del cluster i
P1=data.state(cluster,1); %alfa
P2=data.state(cluster,2); %beta
P3=data.state(cluster,3); %beta
p1=[p1;P1]; %contatena la columna de P1 (alfas)
p2=[p2;P2]; %contatena la columna de P2 (betaL)
```

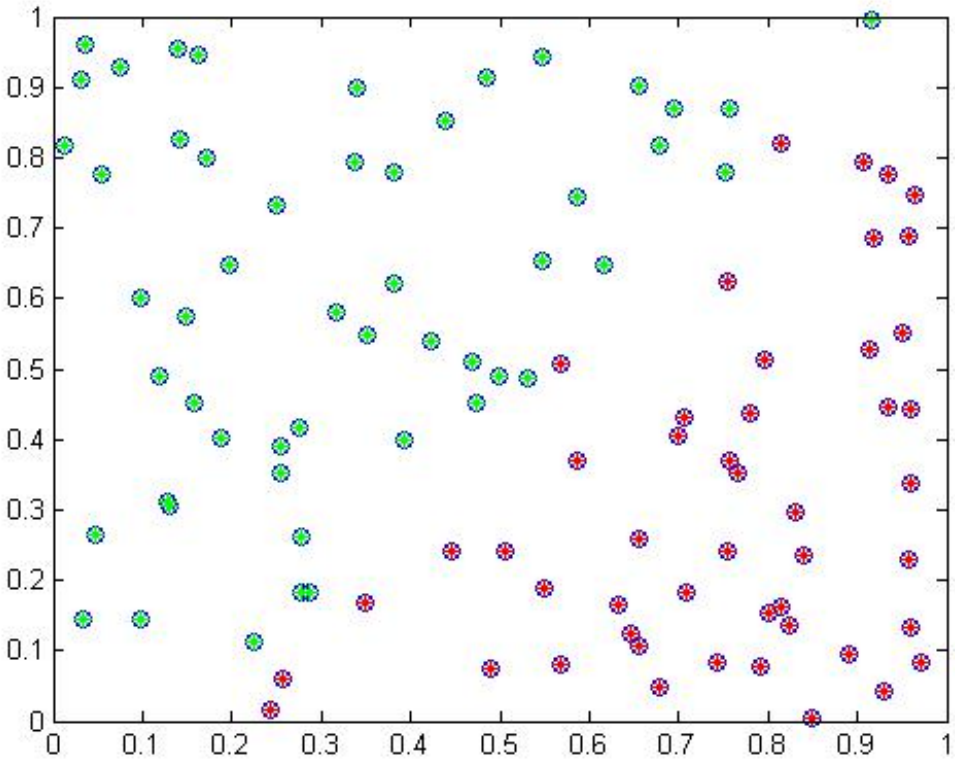

Figura 4.34: Ejemplo de uso del agrupamiento Fuzzy c-means con MatLab.

```
p3=[p3;P3]; %contatena la columna de P3 (betaR)
end
dato=[p1 p2 p3];
[index, centros]=K-means(dato,Nop);%index es el numero qu pertenece la tupla
%centros son los centros de los clusters
%figure(2)
%hold on
for i=1:Nop
k= find(index==i);
I_K{i}=k;           % tengo una matriz de donde estan los clusters
% scatter3(dato(k,1),dato(k,2),dato(k,3),'.')
end
```

Lo primero que se realiza es generar un conjunto de datos para después aplicar el algoritmo. Una vez aplicado se generan una matriz de pertenencia por cada punto de datos y devuelve los la posición de los centros. Para graficar se toma se recorre cada cluster asignando a ellos los puntos que le pertenecen.

En la figura 4.35 se muestra la agrupación representada en las variables α, β_R y β_L.

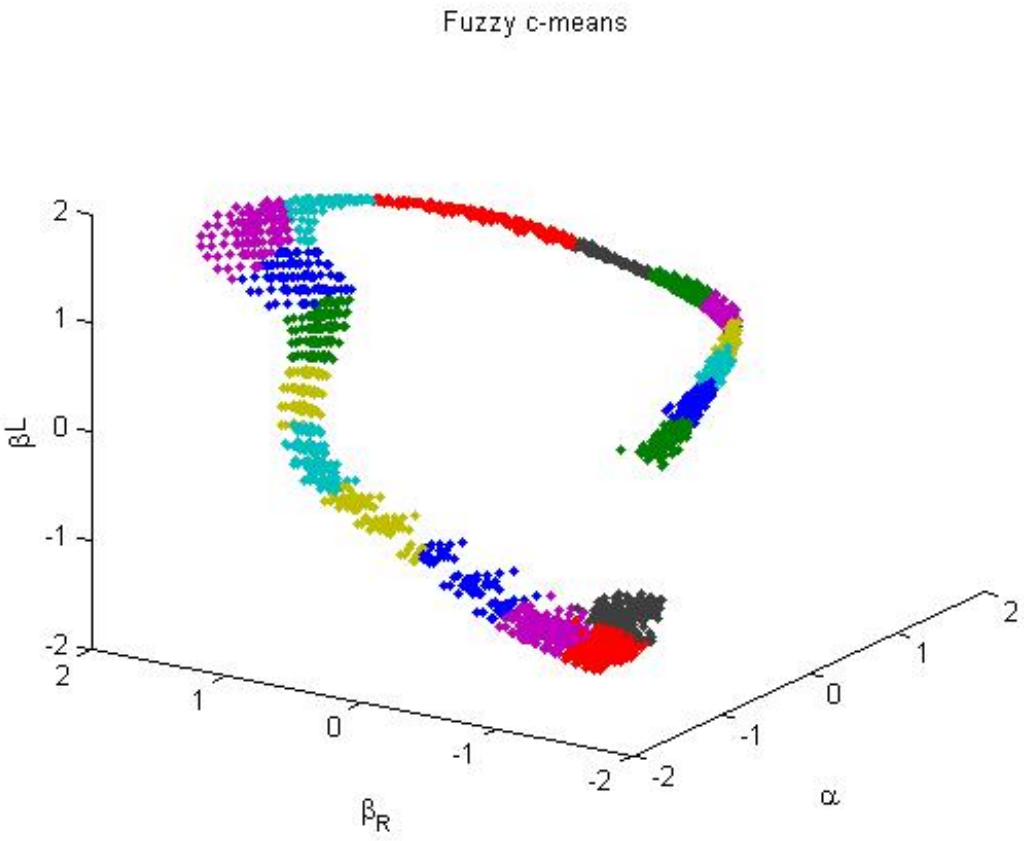

Figura 4.35: Agrupamiento Fuzzy c-means

Se observa que la distribución de los datos es diferente al presentado en la agrupación de RC y también los cálculos estadísticos se calculan de la misma manera.

Una vez realizado la agrupación fuzzy c-means se toman los nuevos clusters y se recalculan los parámetros estadísticos de salida, ya que estos valores se utilizarán posteriormente dentro dentro de la estructura del controlador RC.

4.4.3. Resultados

Se han realizado experimentos similares a los utilizado en RC y K-means, esto debido a que se pretende homogeneizar los análisis para su posterior comparación. Los experimentos se basan en un análisis en marcha normal, aumento de velocidad y cambio ángulo del terreno de marcha como anteriormente se había descrito.

4.4.3.1. Marcha normal

Conforme a la marcha normal se muestra el comportamiento de los parámetros α, β_R y β_L,γ_R y γ_L, los cuales representan a los ángulos formados del torso, ropillas y punto de contacto en el suelo

En la figura 4.36 se muestra el parámetro α y se describe como una señal periódica que representa el ángulo del movimiento entre el torso y la cadera.

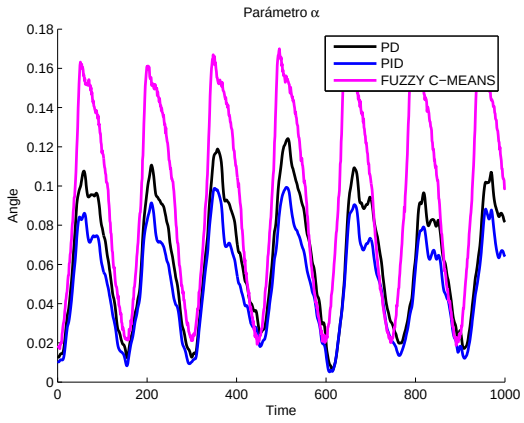

Figura 4.36: Parámetro α de los controladores PD vs PID vs FCM

Además se observa que el valor pico a pico de la señal del controlador FCM es mayor que las señales de los controladores PD, PID y RC, pero menor que el reportado por el controlador K-means.

Por otro lado en la figura 4.37 se muestra los parámetros β_L y β_R y se describen como señales periódica con picos menores a 0.7, también cabe destacar que son menores a las reportadas en el controlador RC.

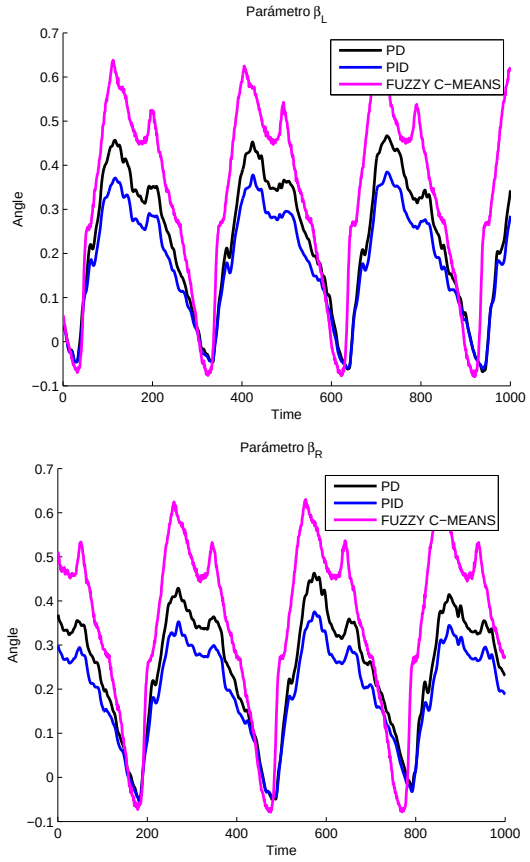

Figura 4.37: Parámetros β_L y β_R de los controladores PD vs PID vs FCM

En la figura 4.38 se muestra γ_L y γ_R, estos parámetros representan el comportamiento de la parte de la pierna que entra en contacto con el suelo. Se observa que el comportamiento de las señales del FCM se asemeja en gran medida al de la señal del controlador RC y K-means.

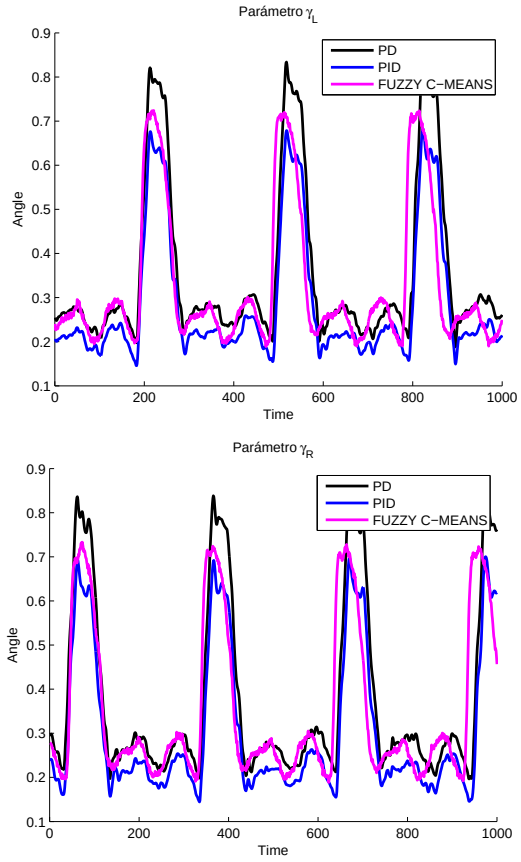

Figura 4.38: Parámetros γ_L y γ_R de los controladores PD vs PID vs FCM

Para conocer aun más el comportamiento de los parámetros α, β_R y β_L, Δ_β,γ_R y γ_L se realiza el cálculo de la como media, mediana, desviación estándar, el máximo y mínimo. Estos valores no toman en cuenta el tiempo y ayudan a describir y diferenciar los parámetros. En la tabla 4.9 se muestra los resultados obtenidos.

	Media	Mediana	Desviación Estándar	Mínimo	Máximo
α	0.1377	0.1416	0.0730	0.0208	0.2494
β_R	0.2914	0.3221	0.1784	-0.0705	0.5646
β_L	0.3006	0.3514	0.1811	-0.0714	0.5643
Δ_β	-0.0092	-0.0311	0.3268	-0.4744	0.4753
γ_R	0.1449	0.1133	0.0724	0.0814	0.3151
γ_L	0.1482	0.1140	0.0743	0.0807	0.3162

Tabla 4.9: Valores estadísticos de las distintas variables del controlador PD

El parámetro que tiene mayor desviación estándar y pico más bajo esΔ_β esto representa que tiene mayor dispersión al valor promedio.Por otra parte el parámetro que tiene el pico mas alto es β_R.

4.4.3.2. Velocidad

Se aplica el aumento de velocidad en la marcha bípeda y se toma los datos generados para operar dentro del controlador RC.

El controlador utilizando Fuzzy c-means funciona al duplicarle la velocidad de base con factor multiplicativo de 2 y 4. Los resultados se pueden observar mediante señales las cuales se presentan en la figura 4.39 con un factor de 2, asimismo la figura 4.40 muestra un factor de 4. Ambas señales están distribuidas sobre una longitud de 1000 datos.

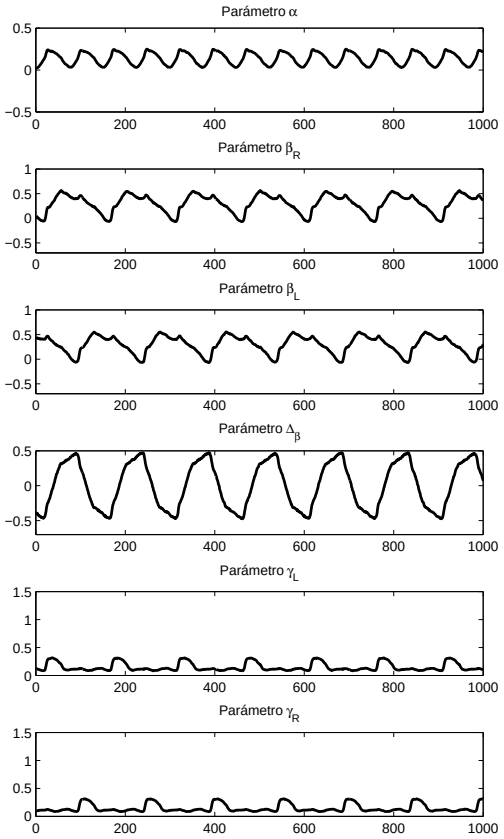

Figura 4.39: Velocidad aumentada con el factor 2 para FCM

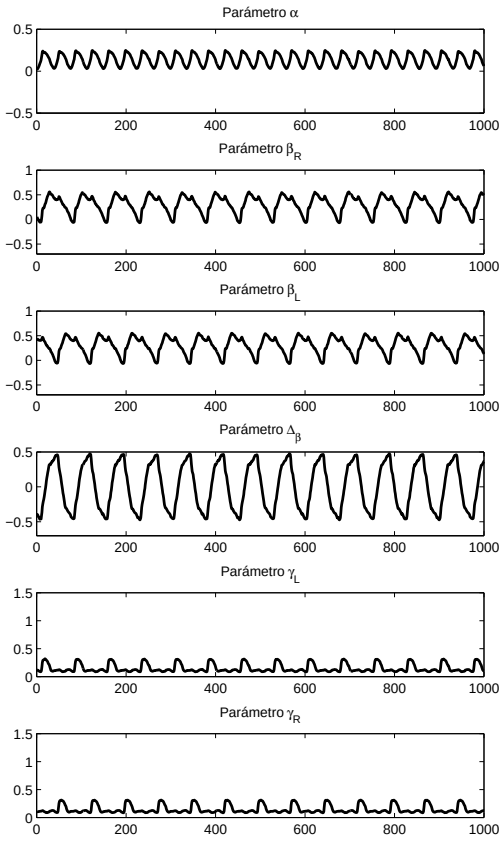

Figura 4.40: Velocidad aumentada con el factor 4 para FCM

En la tablas 4.10 y 4.11 se muestra las velocidades normal, media y máxima de controlador FCM en comparación de los controladores PD y PID.

	FCM	PD	PID
	Terreno Plano 0 °	Terreno Plano 0 °	Terreno Plano 0°
Velocidad Normal	1.14 $\frac{m}{s}$	0.1303 $\frac{m}{s}$	0.1303 $\frac{m}{s}$
Velocidad Media	2.28 $\frac{m}{s}$	0.1957 $\frac{m}{s}$	0.2073 $\frac{m}{s}$
Velocidad Máxima	4.56 $\frac{m}{s}$	0.2612 $\frac{m}{s}$	0.2843 $\frac{m}{s}$

Tabla 4.10: Velocidad normal y máxima en 0 grados de inclinación

	RC	PD	PID
	Terreno Ascendente $\theta_{Ascmax} = 37.64$ °	Terreno Ascendente 2.2473 °	Terreno Ascendente 2.3451 °
Velocidad Normal	1.14 $\frac{m}{s} \pm 5\,\%$	0.0977 $\frac{m}{s}$	0.1153 $\frac{m}{s}$
Velocidad Media	2.28 $\frac{m}{s} \pm 5\,\%$	0.1707 $\frac{m}{s}$	0.1850 $\frac{m}{s}$
Velocidad Máxima	4.56 $\frac{m}{s} \pm 5\,\%$	0.2437 $\frac{m}{s}$	0.2547 $\frac{m}{s}$

Tabla 4.11: Velocidad normal, media y máxima en la máxima inclinación.

De ambas tablas se observa que la velocidad en terreno plano del controlador basado en FCM es mayor en comparación a los reportados en PD y PID. Este comportamiento también es mostrado cuando existe rampa ascendente, cabe aclarar que estas velocidades son las mismas que en terreno plano con una tolerancia de 5 %.

4.4.3.3. Robustez

En esta subsección se presenta la robustez del controlador basado en Fuzzy c-means. Este análisis se realiza ante una pendiente ascendente. Se toman en cuenta los datos generados usados controlador RC y son aplicados al controlador basado en FCM.

Aunque las velocidades son similares al análisis anterior del controlador RC, el ángulo de inclinación es diferente ya que se aumenta en 17 ∘. En la figura 4.41 se muestra los resultados obtenidos cuando se realiza una compracion entre PD, PID y FCM.

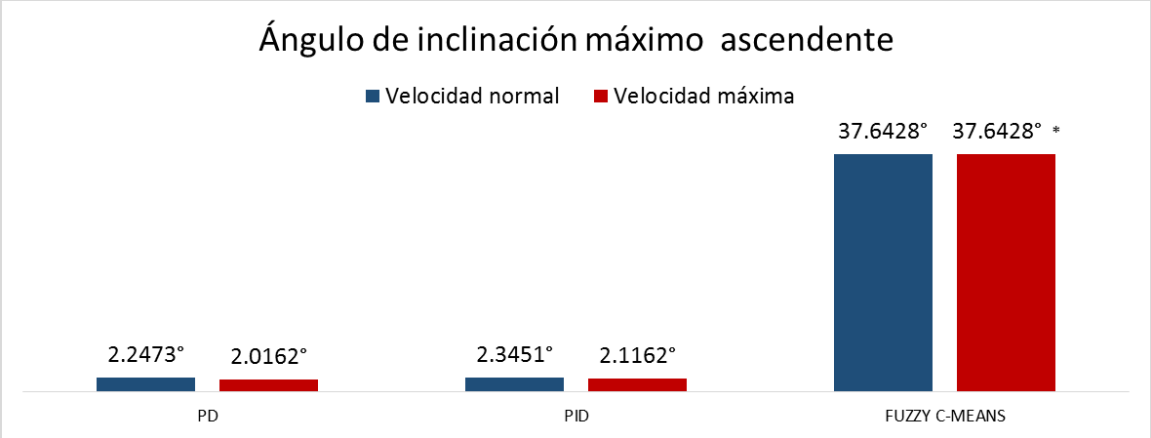

Figura 4.41: Comparación de los distintos ángulos de inclinación en los controladores PD,PID y fuzzy c-means.

Si tomamos en cuenta K-means, PD y PID se observa que la diferencia es amplia aproximadamente más de 15 veces el ángulo de inclinación del controlador PD y

PID, por lo tanto se tiene que el controlador FCM es más robusto ante el cambio de pendiente ascendente.

Ahora bien se han comparado de forma individual los controladores RC, el basado en K-means y FCM contra los controladores clásicos PID y PD, solo queda comparar entre si los métodos que procesan los datos de marcha humana.

4.5. Comparación de técnicas utilizadas

En esta sección se abordará una comparación realizada entre los controladores basados en métodos de agrupación : Regression clustering, K-means y Fuzzy c-means.

Una parte importante es el tiempo de ejecución en un equipo de computo. Para realizar este análisis se utilizo una computadora con un procesador INTEL core i5 450M con 4GB de memoria RAM. La simulación de cada algoritmo se divide en etapas las cuales son presentadas a continuación:

- **Configuración de controlador:** Esta etapa realiza el cambio de velocidad ya sea en un factor 2 o 4.
- **Aprendizaje:** Esta etapa comprende desde la carga del conjunto de datos de entrada hasta la generación de los parámetros estadísticos.
- **Ejecución del controlador:** Comprende la etapa de la ejecución del controlador y sus ciclos de retroalimentación.
- **Simulación:** esta etapa comprende la graficación de las coordenadas generadas para representar el robot bípedo.

El tiempo de la primera etapa promedia 3.8 segundos tomando en cuenta el peor caso, el cual es cuando se genera un conjunto de datos con un factor de 4.

Por su parte la etapa de aprendizaje reporta los siguientes tiempos, ver tabla 4.5.

Método	Tiempo
RC	53.2 segundos
K-means	4.7 segundos
Fuzzy c-means	12.1 segundos

Tabla 4.12: Tiempos de ejecución etapa de aprendizaje en métodos RC, K-means y FCM

Se observa que el método más rápido es el K-means, en segundo lugar es ocupado por el FCM y el menos rápido es el RC. Este resultado es debido a que contiene la ejecución del algoritmo SOM, este método requiere diversas iteraciones basados en un número de épocas.

La etapa de ejecución promedia un tiempo de ejecución de 8.9 segundos. Este realiza ciclos de realimentación que conforman un sistema cerrado.

Por último la etapa de la simulación promedia un tiempo 16.77 segundos. Esta simulación esta establecida para este análisis en 1000 datos de entrada.

Resumiendo se tienen los tiempos totales presentados en la tabla 4.5 .

Método	Tiempo
RC	82.67 segundos
K-means	34.17 segundos
Fuzzy c-means	41.57 segundos

Tabla 4.13: Tiempos de ejecución totales en métodos RC, K-means y FCM

Ante los resultados de tiempo de ejecución se observa que el controlador K-means es el más veloz debido a que toma en cuenta óptimos locales. Por su parte el método fuzzy c-means se tarda en mayor medida debido aque los puntos tienen pertenencia no total a los clusters de modo que no es exclusivo como el K-means y revisa todos los nodos para obtener su matriz de pertenencia. Por ultimo, el método RC ejecuta un algoritmo SOM, el cual contiene una cantidad establecida de iteraciones establecidas como épocas, entre mayores épocas mejor será el tiempo de ejecución.

Para hacer la comparación de la marcha se muestra la simulación de la marcha. En la figura 4.42 se muestra la caminata del robot bípedo basándose en el controlador K-means.

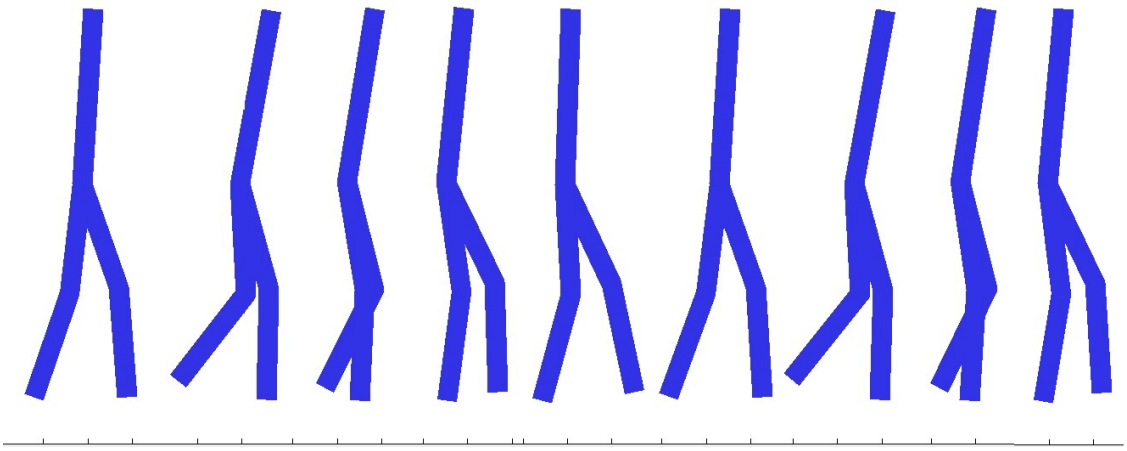

Figura 4.42: Simulación de la marcha robótica bípeda utilizando K-means

Se observa que una gran oscilación del torso como movimiento poco natural en comparación a la marcha humana. Por su parte en la figura 4.43 se muestra a la marcha del robot bípedo usando un controlador basado en fuzzy c-means. Se observa que la oscilación del torso es menor que el K-means, además de que el movimiento de las rodillas tiene menor angulo de apertura.

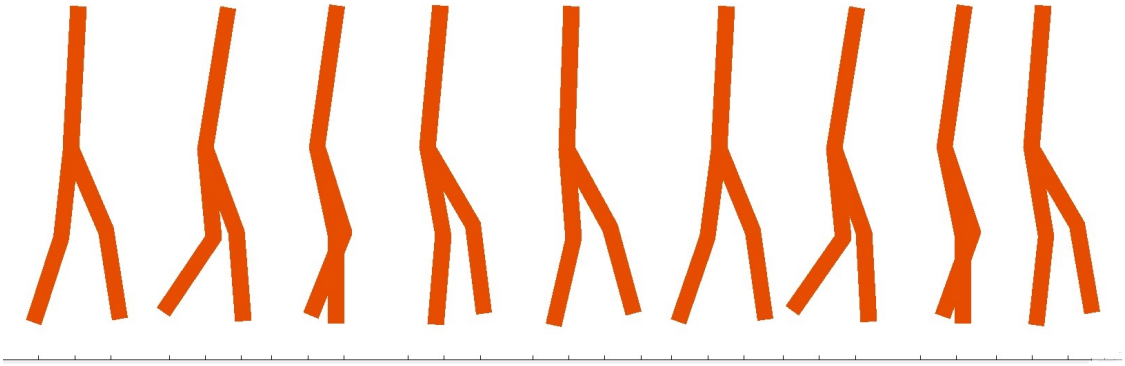

Figura 4.43: Simulación de la marcha robótica bípeda utilizando Fuzzy c-means

Por último en la figura 4.44 muestra la marcha del robot usando el método RC.

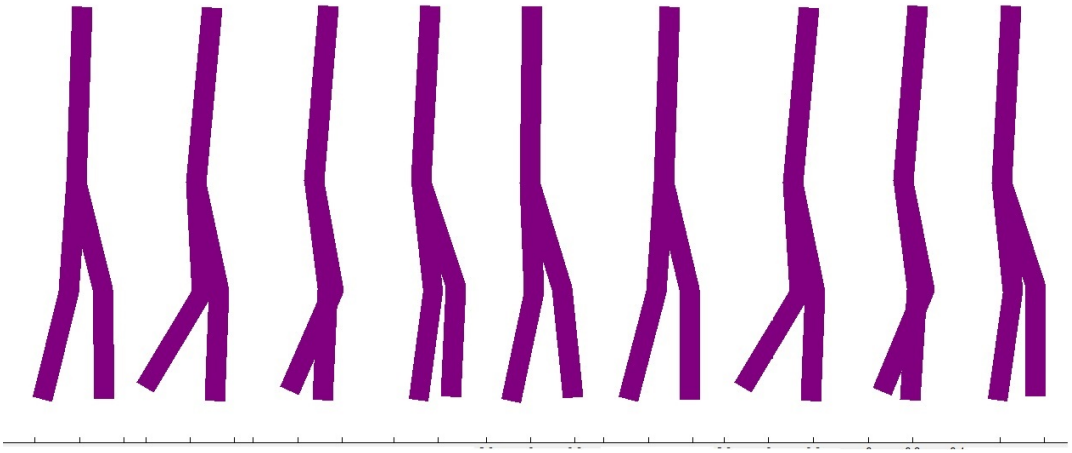

Figura 4.44: Simulación de la marcha robótica bípeda utilizando RC

Se observa que el movimiento de oscilación del torso es el mejor que los K-means y FCM, esto debido a la oscilación del torso casi mínima, por lo que se concluye que aunque el algoritmo RC tarde más tiempo de ejecución, este método tiene una marcha natural mejor debido a su menor oscilación en sus parámetros de la marcha.

Por otra parte si se toma en cuenta la robustez de marcha en terreno ascendente, se tienen diferentes ángulos de inclinación en los diferentes métodos de agrupamiento. En la tabla 4.5 se muestra los ángulos de inclinación recopilados en los experimentos.

Método	Ángulos
RC	20.36 °
K-means	30.18 °
Fuzzy c-means	37.64 °

Tabla 4.14: Tiempos de ejecución de los métodos RC,K-means y FCM

Se observa que el fuzzy c-means tiene un mayor ángulo,seguido de del K-means y por último el RC. Sin embargo por medio de la simulación se observa que el controlador RC muestra una marcha más parecida a la del ser humano tomando en cuenta el comportamiento de los ángulos de las rodillas, torso y tobillos. Los algoritmo K-means y FCM tienen mayor angulo de inclinacion pero se observa que a mayor angulo el movimiento se hace poco parecido a la caminata humana.

4.6. Conclusiones

Se han implementado tres métodos de control para un robot bípedo: regression clustering, k-means y fuzzy c-means. Estas técnicas utilizan el procesamiento de datos de la marcha humana, los datos que se utilizan son los parámetros angulares del torso, rodillas y pie. Para el análisis de aumento de velocidad se genera un conjunto de datos el cual representa una marcha acelerada y es similar en los tres controladores. Por su parte la robustez ante plano inclinado muestra que los tres métodos brindan en promedio 10 veces el valor del ángulo máximo reportado en los controladores PD y PID.

Capítulo 5

Conclusiones y trabajo a futuro

En este capítulo, se resumen los resultados principales a los que se llegaron después de realizar los experimentos. Así mismo, se consideran algunas recomendaciones que se deben de tomar en cuenta con base a lo obtenido en la realización de esta tesis Por último, se procede a comentar el trabajo futuro para ampliar más el conocimiento adquirido en esta tesis.

5.1. Conclusiones

En esta tesis se abordó el uso de minería datos para analizar los datos de la marcha humana aplicado a un robot bípedo.

En el capitulo 3 se implementaron los controladores PD y PID para controlar un sistema robótico bípedo, estos controladores están dentro de lo que denomina el control clásico. Sin embargo la implementación de estos sistemas de control es complicado ya que la sintonización se convierte en todo un reto, y una una mala realización de esta acción puede generar una inestabilidad que provocaría la caída del robot bípedo; tomando en cuenta la velocidad de simulación inicial solo se puede aumentar la velocidad en poca cantidad; asimismo la robustez no es muy amplia debido a que sólo se puede aumentar poco el ángulo de inclinación. De los dos controladores PD y PID el que tiene mejor velocidad y robustez es el PID sin embargo la diferencia entre estos dos controladores no es muy amplia.

En el capitulo 4 se implementó un controlador basado en *regression clustering*, este controlador hace uso de datos humanos los cuales son procesados para generar un conjunto de datos que representa la marcha humana. Estos datos son segmentados en cluster iniciales para poder utilizar el método SOM para realizar una última agrupación. Una vez establecidos los clusters se calculan parámetros útiles dentro del controlador para representar este conjunto de datos humanos. Los resultados de velocidad reportan que el controlador RC soporta 2 y 4 veces la velocidad inicial que se reporta en marcha normal. Con base a su robustez se reporta que el ángulo máximo de inclinación es 20.36 °, el cual es un valor mucho mayor que el controlador PD y PID. Ademas del controlador RC se aborda la implementación de los controladores basados

en kmeans y fuzzy c-means (FCM). Estos métodos toman como base la estructura del controlador RC. Por lo que la agrupación en vez de utilizar SOM se utiliza kmeans o Fuzzy c-means. Para el controlador basado en kmeans se observa una oscilación mayor en el torso en comparación al RC y FCM. La robustez del controlador en terreno ascendente alcanza 30.18 °, teniendo 10 ° más que los reportados en el controlador RC. Por su parte el controlador basado en fuzzy c-means tiene una oscilación media entre la mostrada por RC y kmeans. La robustez de este método es la mayor de los tres algoritmos de clustering logrando un ángulo máximo de inclinación con 37.64 °.

Conforme a tiempos de ejecución se concluye que el algoritmo con mayor rapidez es el k-means seguido del fuzzy c-means y por último el RC. Sin embargo con base al comportamiento dentro del simulador se observa que la marcha RC tiende ha ser más natural a la marcha del ser humano, en segundo lugar esta el fuzzy c-means y por último lugar el kmeans.

En resumen de los resultados obtenidos con las implementaciones de las técnicas que utilizan datos humanos son mejores en velocidad y la robustez que los presentados en las técnicas de control clásico PD y PID.

El simulador implementado es una herramienta de visualización, este simulador contiene los cinco métodos implementados: PD, PID, RC, Kmeans y Fuzzy c-means. Con esta herramienta de visualización se puede generar marchas del robot. La configuración por $default$ es en terreno plano y a velocidad normal, además tiene opciones para configurar el ángulo de ascendente y la velocidad de marcha.

Por último, en esta tesis se aborda una integración de tres tópicos los cuales son: información humana, minería de datos y robots bípedos. Existe poca bibliografía de trabajos que integran dos o incluso tres de estos temas en su estudio. Por lo que se piensa que este trabajo contribuye en el estudio de estas tres temas. De manera que los resultados obtenidos brindan una alternativa para controlar la marcha de robots bípedos.

5.2. Trabajo a futuro

Aún queda mucho trabajo por recorrer en ésta área del conocimiento. Se ha utilizado las técnicas de minería de datos que se han creído pertinentes para lograr el comportamiento estable durante la marcha de un robot bípedo. Sin embargo queda mucho trabajo a futuro por realizar para ampliar el conocimiento y contribuciones.

- Explorar otras técnicas de minería de datos como redes neuronales y ver el comportamiento de la marcha del robot bípedo.

- Establecer otros modelos robóticos bípedos para conocer el comportamiento en terreno ascendente y con aumento de velocidad.

- Llevar a la práctica mediante un robot bípedo físico los algoritmos empleados, cabe destacar que este trabajo llevara mucho tiempo ya que la construcción de

un robot bípedo se tiene que tener en cuenta muchas variables para lograr la estabilidad en las piernas.

Finalmente podemos establecer que los objetivos planteados en esta tesis han sido logrados. Y aunque existen al día de hoy una gran variedad de técnicas, todavía queda investigar para mejorar la efectividad, el rendimiento y la exactitud de las mismas.

Apéndice A

Simulador de robot bípedo

En este capítulo se describirá el sistema construido basado en un simulador de marcha robótica.

A.1. Componentes funcionales del simulador

Se ha realizado un simulador de un sistema bípedo, se toma en cuenta el modelo robótico propuesto en [48]. El software utilizado para desarrollar este simulador son las herramientas Simulink y GUI de MatLab . En la figura A.1 se muestra en diagrama de bloques representativo al simulador.

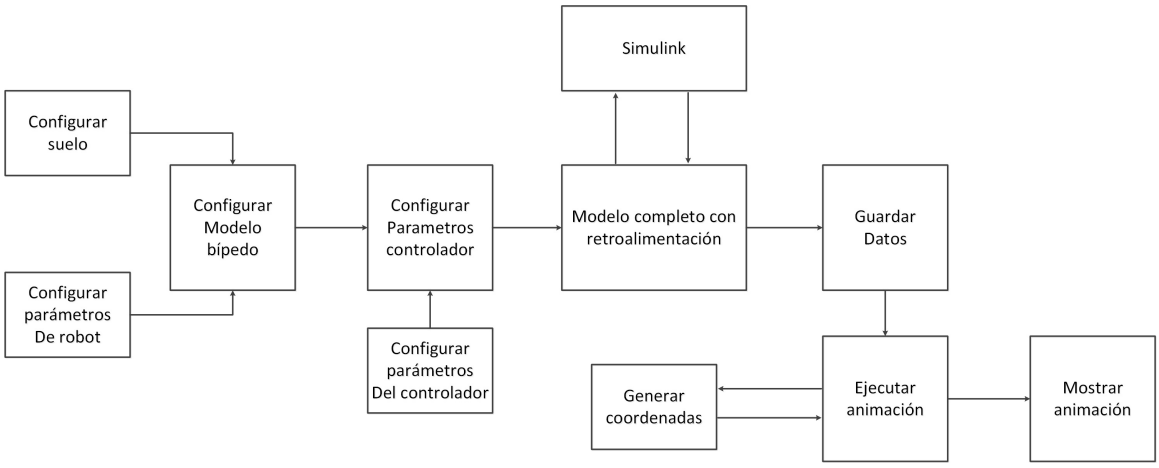

Figura A.1: Diagrama a bloques del software simulador.

Se tienen diferentes bloques los cuales se describirán a continuación:

- **Configurar el modelo bípedo**
  Para este bloque es necesario obtener datos de la descripción del modelo robótico y del suelo a utilizar.

- **Configurar parámetros del robot**
  En este bloque se configura algunos parámetros que describan al robot bípedo. La longitud de las piernas, centros de masas y velocidades iniciales,ver figura .

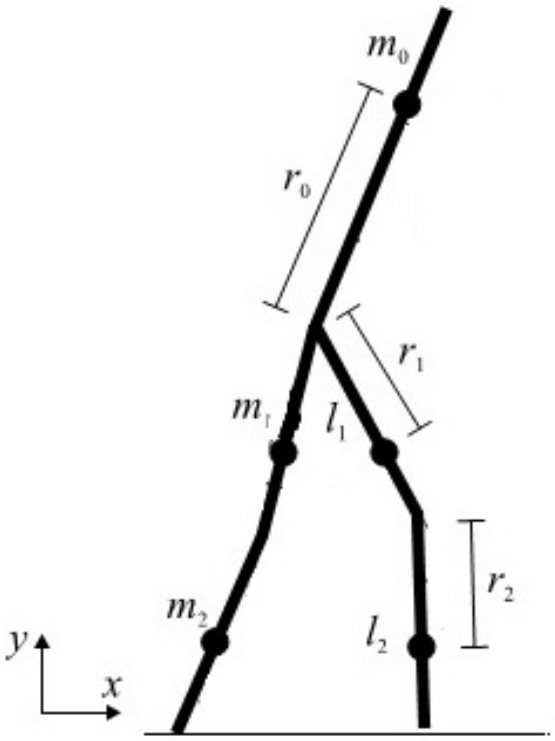

Figura A.2: Parámetros del robot

- **Configurar suelo del robot**
  Se genera dos vectores uno representa la longitud del suelo a recorrer y el otro es toma en cuenta la inclinación respecto a la longitud del suelo.

- **Configurar parámetros del controlador**
  Establece los parámetros que se utilizaran dentro del controlador, como longitud de paso de la simulación y los valores de las ganancias de los controladores PD o PID.

- **Elegir tipo de control**
  Este selector elige un tipo de control PD, PID o RC y en base a la selección se configura el controlador retro-alimentado.

- **Modelo completo con retroalimentación**
  Este bloque une las configuraciones del robot bípedo y del controlador seleccionado.

- **Simulink**
  Simulink es un diagrama de bloques entorno para simulación y diseño basado en modelos. Cuando se tiene completo y configurado los datos del controlador elegido (PD,PID o RC), estos se envían al entorno Simulink en donde se simulan los bloques de control, una vez terminada la simulación los datos son guardados y traídos hacia el *workspace* del GUI del simulador.

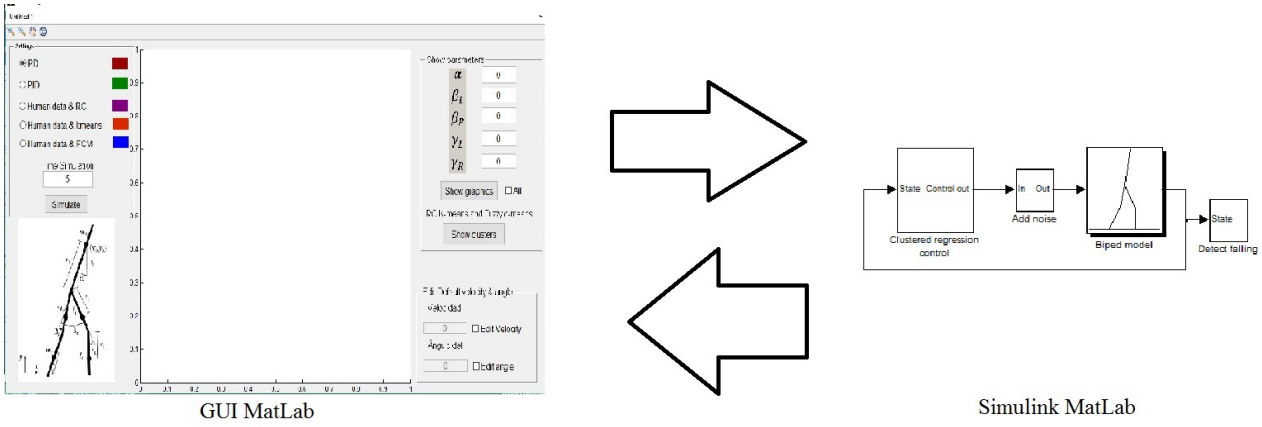

Figura A.3: Ciclo de transferencia bidireccional Simulink e interfaz gráfica

- **Guardar datos**
  Una vez realizada la simulación, los datos son guardados en un solo conjunto para procesarlos en el bloque de generar simulación.

- **Generar coordenadas**
  Se toman los datos generados de la simulación en Simulink, se toman parámetros α,β_R,β_L,γ_R,γ_L para generar las coordenadas del cuerpo humano del torso, cadera, rodillas y tobillos. Las ecuaciones utilizadas para calcula las coordenadas para dibujar son presentadas a continuación:

  - **Coordenadas de la cabeza**

$$xh = q(1,:) + (rdim(1) - rdim(4)) * sin(q(3,:)); \tag{A.1}$$

$$yh = q(2,:) + (rdim(1) - rdim(4)) * cos(q(3,:)) \tag{A.2}$$

  - **coordenadas de la cadera**

$$xj = q(1,:) - rdim(4) * sin(q(3,:)); \tag{A.3}$$

$$yj = q(2,:) - rdim(4) * cos(q(3,:)); \tag{A.4}$$

- **coordenadas de las rodillas**

$$xlj = xj - rdim(2) * sin(q(3,:) - q(4,:)); \tag{A.5}$$

$$ylj = yj - rdim(2) * cos(q(3,:) - q(4,:)); \tag{A.6}$$

$$xrj = xj - rdim(2) * sin(q(3,:) - q(5,:)); \tag{A.7}$$

$$yrj = yj - rdim(2) * cos(q(3,:) - q(5,:)); \tag{A.8}$$

- **Coordenadas de las piernas**

$$xlg = xlj - rdim(3) * sin(q(3,:) - q(4,:) + q(6,:)); \tag{A.9}$$

$$ylg = ylj - rdim(3) * cos(q(3,:) - q(4,:) + q(6,:)); \tag{A.10}$$

$$xrg = xrj - rdim(3) * sin(q(3,:) - q(5,:) + q(7,:)); \tag{A.11}$$

$$yrg = yrj - rdim(3) * cos(q(3,:) - q(5,:) + q(7,:)); \tag{A.12}$$

- **Mostrar animación**
  Este bloque es el encargado de dibujar las coordenadas calculadas sobre una gráfica con movimiento dinámico. Se utiliza un ciclo de iteraciones para acceder a las coordenadas y se grafican los puntos en cada iteración.

  En la Figura A.4 se muestra la ventana del simulador sin simular ningún controlador, la opción de "settings"sirve para elegir entre el controlador PD, PID o RC.

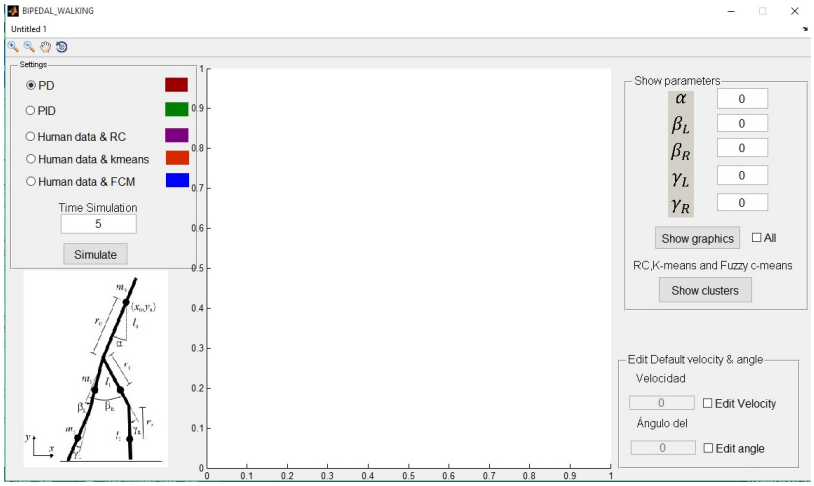

Figura A.4: Simulador de controladores PD, PID y RC

Se selecciona el *radiobutton* el cual pertenece al controlador PD,PID o RC, ademas se tiene que configurar el tiempo de simulación en la caja de texto. Se presiona el *button* de simular y se muestra la marcha bípeda de un robot. En la Figura A.5 se muestra la simulación unicamente del controlador PD. El color asignado par dibujar el robot es de color rojo.

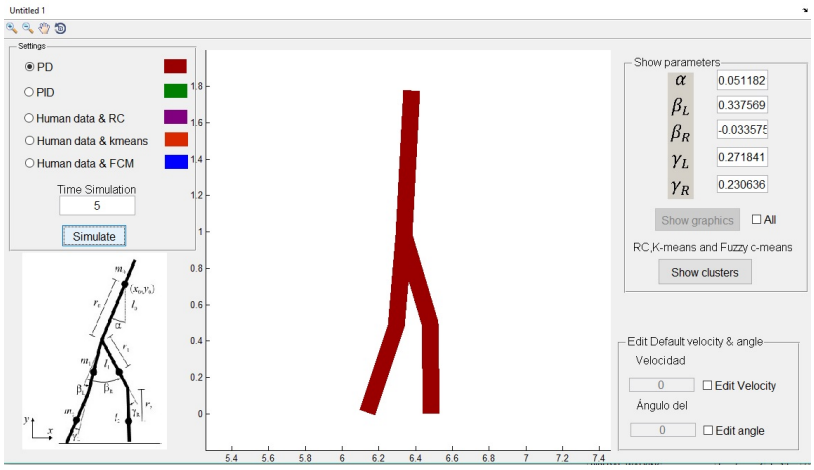

Figura A.5: Simulación del controlador PD

Por su parte en la Figura A.6 muestra la simulación del controlador PID. El robot esta coloreado de verde.

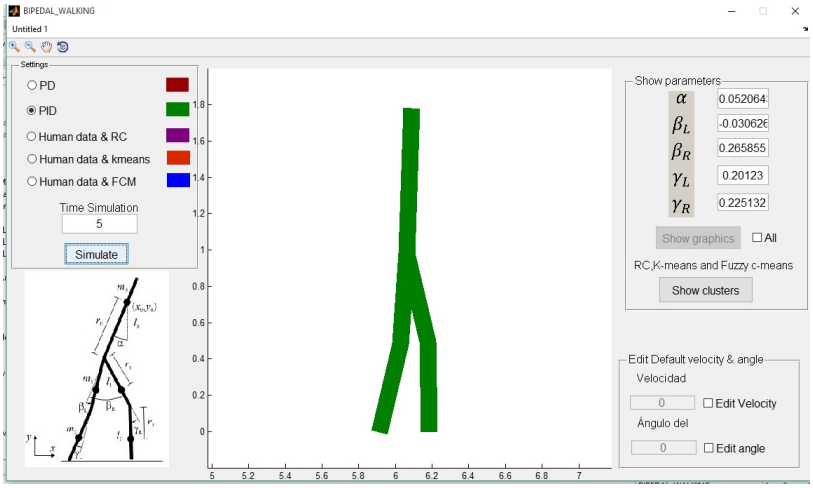

Figura A.6: Simulación del controlador PID

En la Figura A.7 se muestra la simulación controlador RC representado con el color morado.

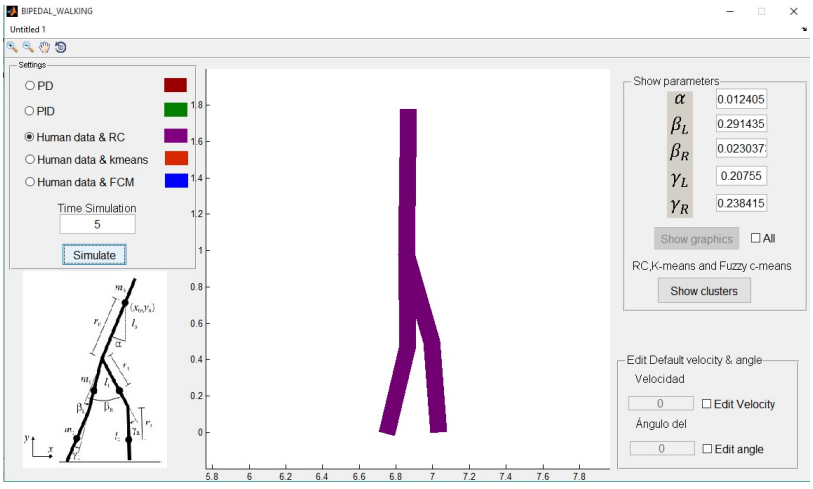

Figura A.7: Simulación del controlador RC

Conforme a los algoritmos de cluster kmeans y fuzzy c-means se muestra las ejecuciones en la figura A.8 y A.9 respectivamente.

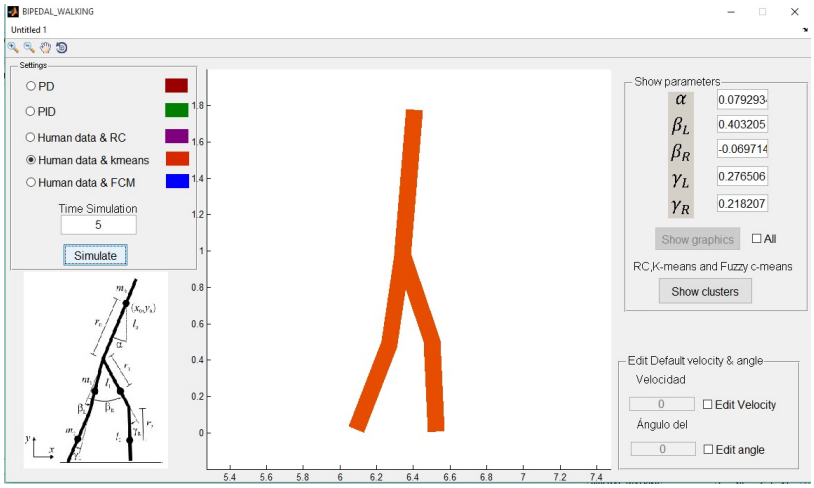

Figura A.8: Simulación del controlador Kmeans

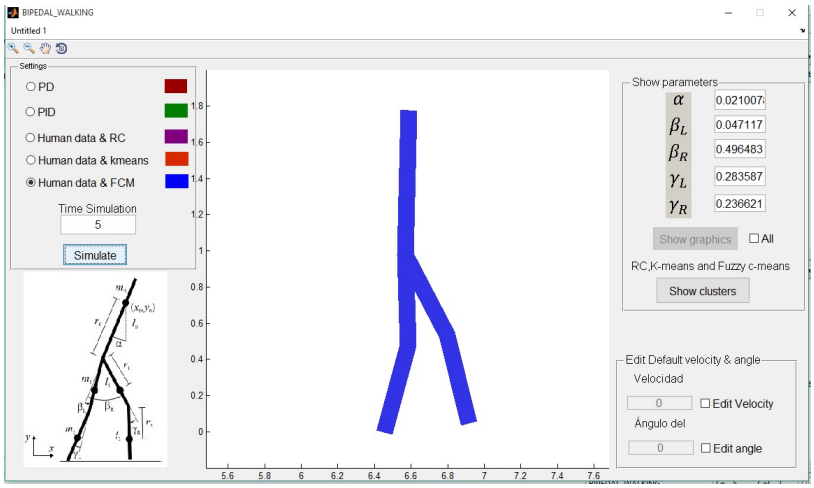

Figura A.9: Simulación del controlador FCM

A.2. Opciones de la interfaz gráfica

En esta sección se describirán las opciones dela interfaz gráfica en donde se realiza la simulación.

A.2.1. Panel principal de selección

El panel principal de la interfaz gráfica contiene tres opciones de simulación de los controladores PD, PID y RC, estas opciones son representados con *radiobutton*. Ademas este panel contiene un botón principal cuya función es comenzar la simulación.

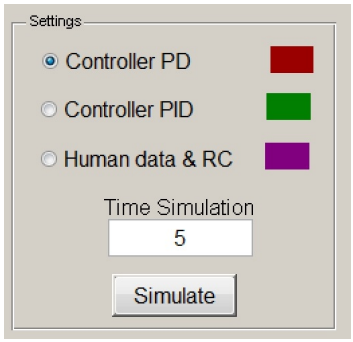

Figura A.10: Panel de configuración del simulador.

A.2.2. Panel de medición de parámetros y graficación

Este panel contiene las diferentes opciones para mostrar los datos de la marcha de los diferentes controladores. Se muestra los valores numéricos de los distintos parámetros α,β_R,β_L,γ_R y γ_L. También se muestra un botón de crear ventana con gráficas, ademas muestra un *checkbox* el cual es el encargado de seleccionar si se gráfica los datos del controlador actual o de todos los controladores simulados, ver figura A.11.

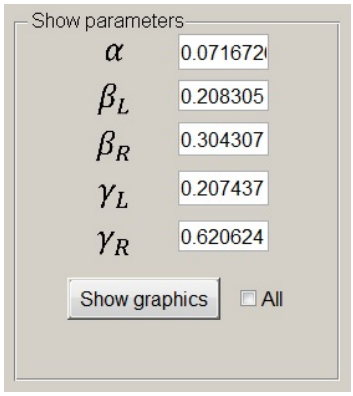

Figura A.11: Panel donde se muestran los parámetros α, β y γ

Esta opción gráfica los distintos parámetros angulares mostrados toma en cuenta la simulación actual del controlador seleccionado. En la figura A.12 se muestra un ejemplo de la graficación de todo el ciclo de marcha y el comportamiento en el tiempo de las distintos parámetros gráficas.

Una opción diferente de graficar los parámetros es mostrar todos los resultados actuales obtenidos de las simulación de los controladores. De modo que si se selecciona esta opción se mostrara las distintas gráficas como se muestra en la figura A.13.

A.2.3. Editar configuraciones de default

Este panel representa la parte de las configuraciones adicionales de la simulación. el cambio de la velocidad inicial y del ángulo de inclinación del suelo. En general,

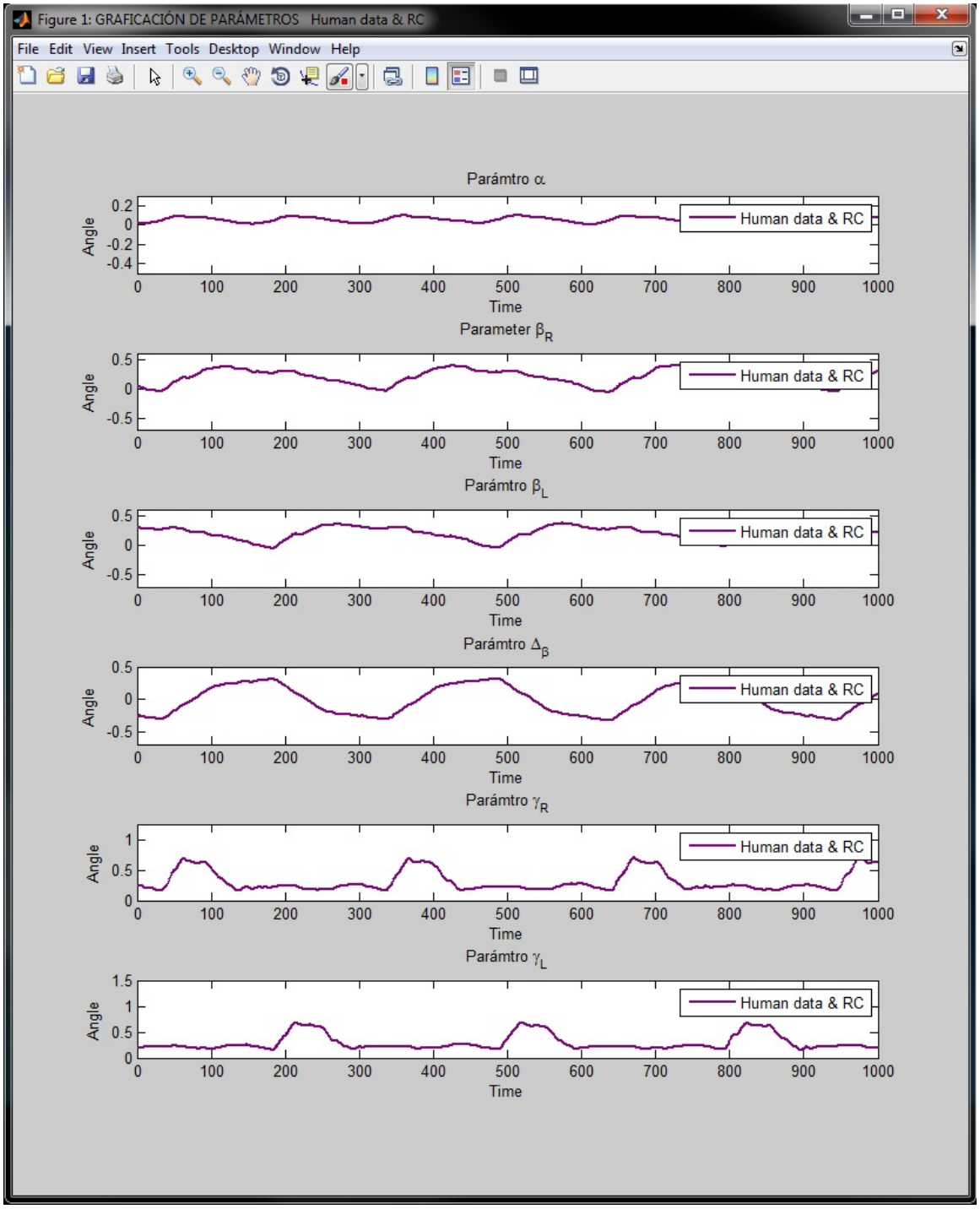

Figura A.12: Graficación de un controlador.

la simulación tiene como valores por *default* un angulo de inclinación cero y una velocidad inicial que garantiza la estabilidad. Por lo que editar estos parámetros abre

diversas posibilidades de simulación así como también se puede obtener condiciones inestables las cuales pueden hacer caer al robot bípedo.

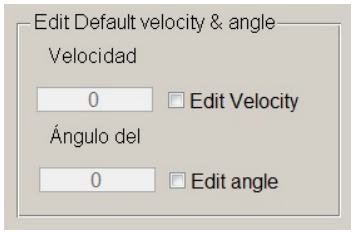

Figura A.14: Edición de velocidad y ángulo

En los experimentos se observó que la oscilación del tronco en el controlador RC fue menor en comparación con la oscilación mostrado por el controlador PID. Además, la longitud de los pasos generados por RC es mayor en comparación a los pasos del controlador PID.

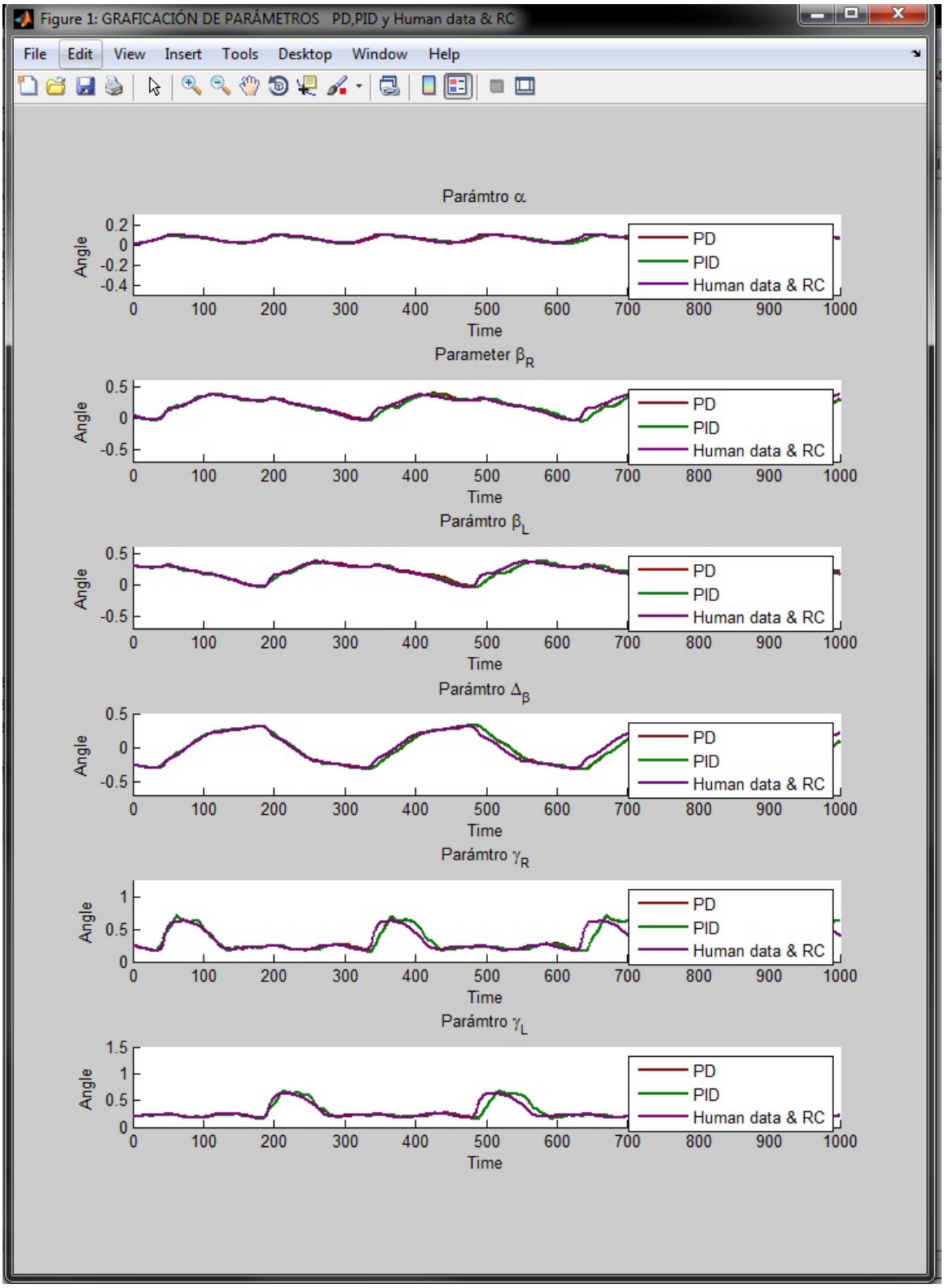

Figura A.13: Graficación de múltiples controladores.

Bibliografía

[1] Y. Harata, F. Asano, and T. Ikeda. Asymmetric gait for rimless wheel with knee joints. In *Control Applications (CCA), 2012 IEEE International Conference on*, pages 1026–1031, Dubrovnik (HRV), 3-5 Oct. 2012.

[2] F. Asano and J. Kawamoto. Passive dynamic walking of viscoelastic-legged rimless wheel. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2331–2336, Saint Paul MN (USA),14-18 May 2012.

[3] Herman van der Kooij, Ron Jacobs, Bart Koopman, and Frans van der Helm. An alternative approach to synthesizing bipedal walking. *Biological cybernetics*, 88(1):46–59, 2003.

[4] Christine Azevedo, Nicolas Andreff, Soraya Arias, and Bernard Espiau. Experimental bipedal walking. In *Experimental Robotics VIII*, pages 582–591. Springer, 2003.

[5] Anthony David and Olivier Bruneau. Bipedal walking gait generation based on the sequential method of analytical potential (smap). *Multibody System Dynamics*, 26(4):367–395, 2011.

[6] Andrew H Hansen, Dudley S Childress, and Steve C Miff. Roll-over characteristics of human walking on inclined surfaces. *Human movement science*, 23(6):807–821, 2004.

[7] F. Ali, A.C. Amran, and A. Kawamura. Bipedal robot walking strategy on inclined surfaces using position and orientation based inverse kinematics algorithm. In *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*, pages 181–186, Singapore (SGP),7-10 Dec 2010.

[8] Tad McGeer. Passive dynamic walking. *the international journal of robotics research*, 9(2):62–82, 1990.

[9] Tad McGeer. Passive walking with knees. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 1640–1645. IEEE, Cincinnati OH (USA),13-18 May 1990.

[10] Michael J Coleman, Anindya Chatterjee, and Andy Ruina. Motions of a rimless spoked wheel: a simple three-dimensional system with impacts. *Dynamics and Stability of Systems*, 12(3):139–159, 1997.

[11] Michael J Coleman. Dynamics and stability of a rimless spoked wheel: a simple 2d system with impacts. *Dynamical Systems*, 25(2):215–238, 2010.

[12] F. Asano, T. Sogawa, K. Tamura, and Y. Akutsu. Passive dynamic walking of rimless wheel with 2-Dof wobbling mass. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3120–3125, Tokyo (JPN),3-7 Nov 2013.

[13] D. Tanaka, F. Asano, and I. Tokuda. Gait analysis and efficiency improvement of passive dynamic walking of combined rimless wheel with wobbling mass. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 151–156, Vilamoura (PRT),7-12 Oct. 2012.

[14] A. Goswami, B. Espiau, and A. Keramane. Limit cycles and their stability in a passive bipedal gait. In *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 1, pages 246–251, Minneapolis MN (USA),22-28 Apr 1996.

[15] Mariano Garcia, Anindya Chatterjee, Andy Ruina, and Michael Coleman. The simplest walking model: stability, complexity, and scaling. *Journal of biomechanical engineering*, 120(2):281–288, 1998.

[16] AL Schwab and M Wisse. Basin of attraction of the simplest walking model. In *Proceedings of the ASME Design Engineering Technical Conference*, volume 6, pages 531–539, Pittsburgh PA (USA),9-12 Sept 2001.

[17] Arthur D Kuo. Energetics of actively powered locomotion using the simplest walking model. *Journal of biomechanical engineering*, 124(1):113–120, 2002.

[18] L.C. Visser, S. Stramigioli, and R. Carloni. Robust bipedal walking with variable leg stiffness. In *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, pages 1626–1631, Rome (ITA),24-27 June 2012.

[19] Yan Huang, Qi-Ning Wang, Yue Gao, and Guang-Ming Xie. Modeling and analysis of passive dynamic bipedal walking with segmented feet and compliant joints. *Acta Mechanica Sinica*, 28(5):1457–1465, 2012.

[20] Simon Mochon and Thomas A McMahon. Ballistic walking. *Journal of biomechanics*, 13(1):49–57, 1980.

[21] Simon Mochon and Thomas A McMahon. Ballistic walking: An improved model. *Mathematical Biosciences*, 52(3):241–260, 1980.

[22] Yan Huang, Baojun Chen, Qining Wang, Kunlin Wei, and Long Wang. Energetic efficiency and stability of dynamic bipedal walking gaits with different step lengths. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4077–4082, Taipei (CHN),18-22 Oct. 2010.

[23] Aaron D. Ames. First steps toward underactuated human-inspired bipedal robotic walking. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1011–1017, Saint Paul MN (USA),14-18 May 2012.

[24] A.D. Ames. Human-inspired control of bipedal walking robots. 59(5):1115–1130, 2014.

[25] Shu Jiang, S. Partrick, Huihua Zhao, and A.D. Ames. Outputs of human walking for bipedal robotic controller design. In *American Control Conference (ACC), 2012*, pages 4843–4848, Montreal (CAN), 27-29 June 2012.

[26] M.J. Powell, Huihua Zhao, and A.D. Ames. Motion primitives for human-inspired bipedal robotic locomotion: walking and stair climbing. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 543–549, Saint Paul MN (USA),14-18 May 2012.

[27] Shinya Aoi and Kazuo Tsuchiya. Generation of bipedal walking through interactions among the robot dynamics, the oscillator dynamics, and the environment: Stability characteristics of a five-link planar biped robot. *Autonomous Robots*, 30(2):123–141, 2011.

[28] Christophe Sabourin, Olivier Bruneau, and Jean-Guy Fontaine. Learning of the dynamic walk of an under-actuated bipedal robot: Improvement of the robustness by using cmac neural networks. In *Climbing and Walking Robots*, pages 543–550. Springer, 2005.

[29] Soraya Bououden, Foudil Abdessemed, and Benali Abderraouf. Control of a bipedal walking robot using a fuzzy precompensator. In *Agent and Multi-Agent Systems: Technologies and Applications*, pages 853–862. Springer, 2009.

[30] L.F. Gunderson and J.P. Gunderson. Using data-mining to allow robots to discover the preferences of humans. In *Integration of Knowledge Intensive Multi-Agent Systems, 2003. International Conference on*, pages 438–443, Boston MA (USA),30 Sept.-4 Oct. 2003.

[31] H. Wang, T.-T. Lee, and W.A. Gruver. A neuromorphic controller for a three-link biped robot. 22(1):164–169, 1992.

[32] Jih-Gau Juang and Chun shin Lin. Gait synthesis of a biped robot using backpropagation through time algorithm. In *Neural Networks, 1996., IEEE International Conference on*, volume 3, pages 1710–1715, Washington DC (USA), 3-6 Jun 1996.

[33] Jun Nakanishi, Jun Morimoto, Gen Endo, Gordon Cheng, Stefan Schaal, and Mitsuo Kawato. Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems*, 47(2):79–91, 2004.

[34] Kevin Loken. *Imitation-based learning of bipedal walking using locally weighted learning*. PhD thesis, The University Of British Columbia, 2006.

[35] Jeong-Jung Kim, Tae-Yong Choi, and Ju-Jang Lee. Falling avoidance of biped robot using state classification. In *Mechatronics and Automation, 2008. ICMA 2008. IEEE International Conference on*, pages 72–76, Takamatsu (JPN), 5-8 Aug 2008.

[36] J.P. Ferreira, M.M. Crisostomo, and A.P. Coimbra. Adaptive pd controller modeled via support vector regression for a biped robot. 21(3):941–949, 2013.

[37] T. Miyashita, K. Shinozawa, N. Hagita, and H. Ishiguro. Behavior selection and environment recognition methods for humanoids based on sensor history. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3468–3473, Beijing (CHN),9-15 Oct 2006.

[38] Aram W Salatian, Keon Young Yi, and Yuan F Zheng. Reinforcement learning for a biped robot to climb sloping surfaces. *Journal of Robotic Systems*, 14(4):283–296, 1997.

[39] Takeshi Mori, Yutaka Nakamura, Masa-Aki Sato, and Shin Ishii. Reinforcement learning for cpg-driven biped robot. In *AAAI*, volume 4, pages 623–630, San Jose CAL (USA),25-29 July 2004.

[40] Gen Endo, Jun Morimoto, Takamitsu Matsubara, Jun Nakanishi, and Gordon Cheng. Learning cpg-based biped locomotion with a policy gradient method: Application to a humanoid robot. *The International Journal of Robotics Research*, 27(2):213–228, 2008.

[41] Jianjuen Hu, Jerry Pratt, and Gill Pratt. Stable adaptive control of a bipedal walking; robot with cmac neural networks. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1050–1056. IEEE, Detroit MI (USA),10-15 May 1999.

[42] Bernd Porr, Christian von Ferber, and Florentin Wörgötter. Iso learning approximates a solution to the inverse-controller problem in an unsupervised behavioral paradigm. *Neural computation*, 15(4):865–884, 2003.

[43] Poramate Manoonpong, Tao Geng, Tomas Kulvicius, Bernd Porr, and Florentin Wörgötter. Adaptive, fast walking in a biped robot under neuronal control and learning. *PLoS Computational Biology*, 3(7):e134, 2007.

[44] Poramate Manoonpong and Florentin Wörgötter. Efference copies in neural control of dynamic biped walking. *Robotics and Autonomous Systems*, 57(11):1140–1153, 2009.

[45] Utku Seven, Tunc Akbas, Kaan Can Fidan, and Kemalettin Erbatur. Bipedal robot walking control on inclined planes by fuzzy reference trajectory modification. *Soft Computing*, 16(11):1959–1976, 2012.

[46] N Bigdeli, K Afshar, BI Lame, and A Zohrabi. Modeling of a five-link biped robot dynamics using neural networks. *Applied Sci*, 8:3612–3620, 2008.

[47] N. Onn, M. Hussein, C.H.H. Tang, M.Z.M. Zain, M. Mohamad, and W.Y. Lai. Motion control of human bipedal model in sagittal plane. *WSEAS Transactions on Systems and Control*, 10:160–171, 2015.

[48] O. Haavisto and H. Hyotyniemi. Data-based modeling and control of a biped robot. In *Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on*, pages 427–432, Espoo (FIN),27-30 June 2005.

[49] N.K. Jaiswal and V. Kumar. Comparison between conventional PID and fuzzy PID supervisor for 3-Dof scara type robot manipulator. In *Electrical, Electronics and Computer Science (SCEECS), 2014 IEEE Students' Conference on*, pages 1–5, Bhopal (IND),1-2 March 2014.

[50] Ahmed Z Alassar, Iyad M Abuhadrous, Hatem Elaydi, et al. Comparison between flc and pid controller for 5dof robot arm. In *Advanced Computer Control (ICACC), 2010 2nd International Conference on*, volume 5, pages 277–281. IEEE, Shenyang (CHN), 27-29 March 2010.

[51] Ben W Stansfield, Susan J Hillman, M Elizabeth Hazlewood, Alastair A Lawson, Alison M Mann, Ian R Loudon, and James E Robb. Normalized speed, not age, characterizes ground reaction force patterns in 5-to 12-year-old children walking at self-selected speeds. *Journal of Pediatric Orthopaedics*, 21(3):395–402, 2001.

[52] BW Stansfield, SJ Hillman, ME Hazlewood, and JE Robb. Regression analysis of gait parameters with speed in normal children walking at self-selected speeds. *Gait & posture*, 23(3):288–294, 2006.

[53] David W Aha. Editorial. In *Lazy Learning*, pages 7–10. Springer, 1997.

[54] Sethu Vijayakumar, Aaron D'souza, and Stefan Schaal. Incremental online learning in high dimensions. *Neural computation*, 17(12):2602–2634, 2005.

[55] Christopher G Atkeson, Andrew W Moore, and Stefan Schaal. Locally weighted learning for control. In *Lazy learning*, pages 75–113. Springer, 1997.

[56] Tomas Herrera Valenzuela. Actividad fisica, caminata y costo energetico en ninos y adolescentes: una revision, 2010.

[57] David Martín Bermejo et al. Comparación de tiempos de trayectos metro-a pie-bici en la zona urbana de barcelona. 2007.