



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA ELECTRÓNICA
MAESTRÍA EN CIENCIAS DE LA ELECTRÓNICA
OPCIÓN EN AUTOMATIZACIÓN

**“Medición y modelado bilineal del Zero Moment Point (ZMP)
mediante la técnica de subespacios recursiva”***

T E S I S

Presentada para obtener el título de:
Maestro en Ciencias de la Electrónica Opción en Automatización

Presenta:

Lic. Daniel Gomez Rivera*

Directores:

Dra. Amparo Palomino Merino (FCE-BUAP)
Dr. Gibran Etcheverry Doger (EDEI-UDLAP)
Dra. Aurora Vargas Treviño (FCE-BUAP)

Puebla, México

Diciembre 2016

*BECARIO CONACYT
*BECARIO VIEP

BUAP[®]

Agradecimientos

Durante el tiempo en el que realicé la maestría pude constatar el apoyo brindado, por lo cual quiero expresar mi más sincero agradecimiento.

A mis padres y hermano que me apoyaron en todo el transcurso de la maestría.

A mis directores de tesis, Dra. Amparo Palomino Merino, Dra. Aurora Vargas Treviño y Dr. Gibran Etcheverry, por apoyarme con sus conocimientos para la realización de esta tesis.

A mis sinodales, ya que con sus acertados comentarios y correcciones, contribuyeron a enriquecer este trabajo.

A mis compañeros de generación Juan Carlos, Ricardo Torres, Fernando Chevez, Miguel, Luis, Ricardo, Froylan, Araceli.

Al CONACYT por el apoyo económico brindado.

Dedicatoria

Dedico esta tesis a todos aquellos que
me apoyaron en todo momento durante
el transcurso de este trabajo

En este trabajo se utiliza la técnica de identificación de subespacios para obtener un modelo en el espacio de estados de la marcha humana; para simplificar este problema se aproxima el movimiento que se produce en la marcha humana mediante el modelo *Cart-Table*, el cual consiste en el desplazamiento de una masa en una superficie plana, lo cual es una aproximación del movimiento del centro de masa de cuerpo humano que se da cuando se ejerce el movimiento de la marcha. Para implementar este modelo físicamente, se realiza un dispositivo de medición, basado principalmente en un acelerómetro colocado en la cintura de una persona. Con este dispositivo se obtiene información del desplazamiento del centro de masa, la cual puede ser adquirida digitalmente y procesada en una computadora. Estos datos se aplican a un algoritmo de identificación estocástica, ya que solo se tienen datos experimentales de salida.

El algoritmo de identificación utiliza herramientas geométricas, de estadística y descomposiciones de matrices. Del primer grupo mencionado se tienen las proyecciones ortogonales y las proyecciones oblicuas. Del segundo grupo se utilizan las matrices de covarianza de las señales de ruido. Con respecto a la descomposición de matrices, se utiliza la factorización QR , LQ y la descomposición en valores singulares. Además, por tratarse de un algoritmo de identificación estocástica, también se utiliza el filtro de Kalman.

El algoritmo de identificación de subespacios que se propone, está conformado por dos partes, la primera parte consiste en un algoritmo de identificación lineal, con lo que se obtiene un modelo matemático y un error de estimación, ya que el sistema de interés en este trabajo es un sistema dinámico no lineal. La segunda parte consta de un algoritmo de identificación cuadrática que reduce el error de estimación que se obtiene en la parte lineal.

La obtención de los datos experimentales del movimiento del centro de masa en el proceso de la caminata humana, se realizó mediante un arreglo implementado con un acelerómetro, un Arduino Nano y un módulo Bluetooth.

Contenido

Índice de figuras	IV
Introducción	VI
CAPÍTULO 1	1
Estado del arte	1
1.1. Zero Moment Point (ZMP)	1
1.2. Módulo de medición	2
1.3. Técnica de modelado	3
1.3.1. Técnicas de identificación del ZMP	3
1.3.2. Modelo <i>Cart–Table</i>	4
1.3.3. Técnica de subespacios	5
CAPÍTULO 2	6
<i>Zero Moment Point (ZMP)</i> y caminata humanoide.....	6
2.1. ¿Qué es el ZMP?.....	6
2.2. Simplificación de la dinámica de la marcha humana mediante el Modelo <i>Cart–Table</i>	7
2.3. Caminata humanoide	8
2.3.1. Robots humanoides	8
2.3.2. Caminata humana	9
CAPÍTULO 3	11
Arreglo de medición.....	11
3.1. Acelerómetro MMA7361	11
3.2. Implementación de código en Arduino.....	12
3.3. Módulo Bluetooth HC-06	15
3.4. Primer prototipo del módulo de medición	16
CAPÍTULO 4	21
Algoritmo de identificación de subespacios	21
4.1. Herramientas para la identificación de subespacios.....	22
4.1.1. Proyecciones ortogonales y oblicuas	22
4.1.2. Filtro de Kalman	24
4.1.3. Matriz de Hankel	26
4.1.4. Descomposición QR y LQ.....	27
4.1.5. Descomposición en valores singulares (SVD).....	28
4.2. Identificación estocástica	28

4.3. Identificación determinista	32
4.4. Serie de Volterra y modelo bilineal.....	35
CAPÍTULO 5	38
Análisis caótico.....	38
5.1. El exponente de Lyapunov	38
5.2. Algoritmo para calcular exponentes de Lyapunov.....	38
CAPÍTULO 6	40
Resultados	40
6.1. Identificación de subespacios de la marcha humana	50
6.2. Análisis caótico de la marcha humana	59
Conclusiones y trabajo a futuro	65
Conclusiones generales.....	65
Trabajo a futuro	66
Apéndice A. Código en Matlab del algoritmo implementado.....	67
Apéndice B. Publicaciones.....	77
Referencias.....	81

Índice de figuras

Figura 0.1. Diagrama a bloques del módulo propuesto.	VII
Figura 1.1. Generador de Patrones de Marcha, Control de seguimiento del ZMP.	2
Figura 2.1. Fuerzas y momentos que actúan en el pie.	6
Figura 2.2. Modelo <i>Cart-Table</i>	8
Figura 2.3. Evolución de los robots humanoides de Honda [14].	9
Figura 2.4. Polígono de soporte para la estabilidad de la marcha humana.	10
Figura 3.1. Diagrama de bloques del sistema de adquisición de datos para la marcha humana..	11
Figura 3.2. Acelerómetro MMA7361 [17].	12
Figura 3.3. Diagrama de Flujo que describe el procedimiento para obtener la información del acelerómetro.	14
Figura 3.4. Módulo de comunicación Bluetooth HC-06.	15
Figura 3.5. Diagrama esquemático del módulo de adquisición de datos.	16
Figura 3.6. Módulo de adquisición de datos para la medición del desplazamiento, conformado por acelerómetro, Arduino Nano y módulo Bluetooth.	16
Figura 3.7. Orientación de los ejes cuando se coloca en la cintura de la persona.	17
Figura 3.8. Ventana del programa PuTTY para realizar la configuración del módulo HC-06 (mediante comunicación inalámbrica).	17
Figura 3.9. Ventana de datos obtenidos.	18
Figura 3.10. Señal obtenida con el dispositivo de medición implementado.	19
Figura 3.11. Señal filtrada.	19
Figura 3.12. Movimiento de la marcha humana representado como un péndulo invertido.	20
Figura 4.1. Diagrama del proceso de la identificación de sistemas.	21
Figura 4.2. Proyección ortogonal de matrices.	23
Figura 4.3. Proyección oblicua.	24
Figura 4.4. Diagrama de bloques de un sistema estocástico.	29
Figura 4.5. Diagrama de bloques de un sistema determinista	33
Figura 4.6. Diagrama del algoritmo cuadrático.	36
Figura 6.1. Desplazamiento del centro de masa en el eje <i>x</i> y en el eje <i>y</i> de la persona 1.	41
Figura 6.2. Desplazamiento del centro de masa en el eje <i>x</i> y en el eje <i>y</i> de la persona 1 cuando se tiene un tropiezo.	42
Figura 6.3. Desplazamiento del centro de masa en el eje <i>y</i> de la persona 1 cuando se tiene un desbalance.	43
Figura 6.4. Desplazamiento del centro de masa en el eje <i>x</i> y en el eje <i>y</i> de la persona 2.	44
Figura 6.5. Desplazamiento del centro de masa en el eje <i>y</i> de la persona 2 cuando se tiene un tropiezo.	45
Figura 6.6. Desplazamiento del centro de masa en el eje <i>x</i> y en el eje <i>y</i> de la persona 2 cuando se tiene un desequilibrio.	46
Figura 6.7. Desplazamiento del centro de masa en el eje <i>y</i> de la persona 3.	47
Figura 6.8. Desplazamiento del centro de masa en el eje <i>y</i> de la persona 3 cuando se tiene un tropiezo.	48
Figura 6.9. Desplazamiento del centro de masa en el eje <i>y</i> de la persona 3 cuando se tiene un desequilibrio.	49
Figura 6.10. Gráfica de comparación entre la estimación y la señal real de ambos ejes.	51

Figura 6.11. Gráfica de comparación entre la estimación y la señal real de ambos ejes.	52
Figura 6.12. Gráfica de comparación entre la estimación y la señal real de ambos ejes.	53
Figura 6.13. Gráfica de comparación de la señal original con las estimaciones lineal y cuadrática de los datos de tropiezo.	54
Figura 6.14. Gráfica de comparación de la señal original con las estimaciones lineal y cuadrática de los datos de tropiezo.	56
Figura 6.15. Gráfica de comparación de la señal original con las estimaciones lineal y cuadrática de los datos de tropiezo.	58
Figura 6.16. Gráfica de fase caminata normal en el eje x de la persona 1.	59
Figura 6.17. Gráfica de fase tropiezo en la caminata eje x de la persona 1.	60
Figura 6.18. Gráfica de fase desequilibrio en la caminata eje x de la persona 1.	60
Figura 6.19. Gráfica de fase caminata normal en el eje x de la persona 2.	61
Figura 6.29. Gráfica de fase tropiezo en la caminata eje x de la persona 2.	61
Figura 6.21. Gráfica de fase desequilibrio en la caminata eje x de la persona 2.	62
Figura 6.22. Gráfica de fase caminata normal en el eje x de la persona 3.	62
Figura 6.23. Gráfica de fase tropiezo en la caminata eje x de la persona 3.	63
Figura 6.24. Grafica de fase desequilibrio en la caminata eje x de la persona 3.	63

En el presente trabajo se realizó la medición del *Zero Moment Point* (ZMP por sus siglas en inglés) mediante un módulo inalámbrico creado a partir de acelerómetros y microcontroladores. Con este módulo se obtuvieron los datos experimentales del ZMP necesarios para obtener un modelo matemático del mismo, utilizando una técnica geométrica de identificación de subespacios.

La identificación de sistemas consiste en construir modelos matemáticos de sistemas dinámicos a partir de un análisis de los datos experimentales, mediciones u observaciones obtenidas del sistema en cuestión. Los sistemas dinámicos son muy comunes en diversas áreas donde se requiere tener un modelo preciso para fines de análisis, predicción, simulación, diseño y control.

En particular, los algoritmos de control para crear una forma de marcha humana artificial en un robot humanoide, requieren de modelos matemáticos complejos; estos modelos no pueden ser obtenidos de forma sencilla a partir de las leyes que rigen el proceso, además puede ser muy caro y difícil experimentar sobre el sistema real.

En la literatura se han propuesto diversos modelos lineales para la solución al proceso de identificación. Algunos de estos son: ARX (*AutoRegresive with eXternal input*) [1], ARMAX (*AutoRegresive Moving Average with eXternal input*) [1] y OE (*Output Error*) [1], los cuales utilizan la optimización de parámetros del sistema minimizando una función de costo conveniente. La aplicación principal de estos modelos y algoritmos es obtener funciones de transferencia entrada-salida para sistemas de una sola variable (SISO del inglés *Single Input Single Output*) [1].

Cuando se busca obtener modelos matemáticos de sistemas multivariantes (MIMO del inglés *Multiple Input Multiple Output*) [11], los modelos antes mencionados presentan varias desventajas, ya que en general requieren el uso de técnicas no lineales iterativas para minimizar la función de costo, resultando en un incremento en el tiempo de procesamiento. Otra desventaja se debe a la existencia de mínimos locales y no convexidades, resultando en una alta sensibilidad debida a la elección de parámetros y al procedimiento de optimización, por lo que no se asegura una optimización global.

Para poder solucionar algunos de los problemas mencionados, se han desarrollado métodos denominados geométricos o de subespacios, basados en teoría estocástica. Estos métodos consisten en una serie de proyecciones, en conjunto con otras herramientas geométricas.

Las técnicas de identificación basadas en subespacios son de gran utilidad para obtener modelos matemáticos en variables de estado en tiempo discreto para sistemas lineales e invariantes en el tiempo del tipo MIMO con las siguientes características: no necesitan realizar ninguna parametrización canónica, no requieren una optimización iterativa no lineal, solo se requiere de herramientas estadísticas y de álgebra lineal como lo son las descomposiciones matriciales *LQ* y *SVD* [24]. Por último, los modelos en espacio de estados

obtenidos por métodos basados en subespacios pueden describir una gran variedad de procesos.

El sistema de interés en este trabajo y del cual se obtendrá un modelo matemático mediante el método de identificación de subespacios, es el de la marcha humana. Este sistema se simplifica considerando que toda la masa del cuerpo está concentrada en una masa localizada en el centro de masa (COM) del cuerpo. El movimiento de la marcha es entonces aproximado al movimiento de un péndulo invertido.

Para obtener los datos experimentales necesarios, que requieren los métodos de identificación de subespacios, se requiere un dispositivo que pueda medir los movimientos del centro de masa de una persona mientras camina; para lograr esto, se utiliza un acelerómetro colocado en la cintura para simular el movimiento del centro de masa.

Los acelerómetros tienen la desventaja de que sus mediciones tienen una gran cantidad de ruido, por lo que no son totalmente confiables. Una opción para evitar este problema es usar un sensor más complejo, como lo es una unidad de medición de inercial, la cual contiene un acelerómetro y un giroscopio integrado. Con estos dos sensores en conjunto se pueden corregir los datos del acelerómetro, utilizando los del giroscopio, aunque también aumenta la complejidad del dispositivo. Para tratar el problema del ruido, en este trabajo se optó por hacer uso de un filtro que reduce el ruido en la señal.

Se propone el arreglo de medición formado por un acelerómetro para medir los movimientos del centro de masa, arduino para procesar la información y un módulo bluetooth, como se muestra en el diagrama de bloques de la figura 0.1.

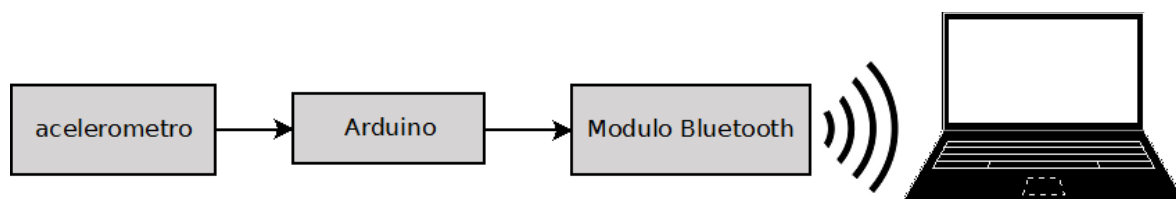


Figura 0.1. Diagrama a bloques del módulo propuesto.

Con el dispositivo de medición se realizaron diversas pruebas en diferentes casos de la marcha humana con diferentes personas y estos datos también se utilizaron para realizar un análisis caótico de las series de tiempo medidas.

Organización del documento de tesis

El presente documento se encuentra dividido en 5 capítulos:

En el **capítulo 1** se menciona el estado del arte referente al *Zero Moment Point* y las técnicas de identificación.

En el **capítulo 2** se explican los conceptos más importantes involucrados en el proceso de la marcha humanoide.

En el **capítulo 3** se describe la implementación del arreglo de medición y el proceso para obtener los datos del movimiento del centro de masa de una persona en el proceso de la marcha.

En el **capítulo 4** se explica la identificación de sistemas y las herramientas que se utilizan para realizar algoritmos de identificación de subespacios.

En el **capítulo 5** se presentan los resultados obtenidos al aplicar la identificación de subespacios a los datos experimentales obtenidos mediante el arreglo de medición presentado en el capítulo 3. Además se presenta un análisis del comportamiento caótico de los mismos datos usados para la identificación.

CAPÍTULO 1

Estado del arte

La importancia de obtener un modelo del *Zero Moment Point* reside en la posibilidad de crear algoritmos de control para la marcha humanoide. Diversos autores retoman el modelo más utilizado en la creación de los algoritmos antes mencionados; este modelo consiste en la simplificación del cuerpo de un robot humanoide a una forma más básica, como lo es el péndulo invertido [8]. La efectividad de este modelo es mejorada por diversos métodos, como por ejemplo usando técnicas de identificación de sistemas. Además se han propuesto algoritmos de control para seguir una trayectoria deseada del ZMP. Para mostrar la eficiencia del algoritmo se han utilizado robots humanoides como el HRP-4C o NAO para las pruebas experimentales [7].

1.1. Zero Moment Point (ZMP)

El concepto de *Zero Moment Point* fue presentado por M. Vukobratovic en conjunto con D. Juricic en el periodo de 1969 – 1973 [12], basado en el problema de modelar una forma de marcha artificial para un robot humanoide. El uso de este concepto permaneció durante mucho tiempo como el único para generar algoritmos de síntesis de la marcha humana.

La primera demostración práctica del ZMP se llevó a cabo en Japón en el año de 1984, en la Universidad de Waseba con el primer robot equilibrado dinámicamente llamado WL-10RD de la familia WABOT [2]. Este trabajo fue realizado 16 años después de la aparición del concepto de ZMP. Este concepto se aplica en la creación de algoritmos para lograr una marcha artificial estable en robots humanoides y exoesqueletos.

Shuuji Kajita ha utilizado el concepto del ZMP en varios de sus trabajos, en [3] representa la dinámica de un robot humanoide con el modelo del *Cart-Table*, con esto se obtiene una representación conveniente para obtener el ZMP. Además, diseña un controlador para seguir la trayectoria del ZMP usando información futura de este para corregir el error que se genera debido a la diferencia que existe entre el modelo de la caminata humana y el *Cart-Table*. Para generar un patrón de caminata para un robot humanoide con el ZMP obtenido, define una nueva variable que resulta de la derivada de la aceleración que se produce en el eje horizontal del movimiento del centro de masa

$$\frac{d}{dt} \dot{x} = u_x . \quad (1.1)$$

Tomando esta variable como entrada para la ecuación del ZMP del eje horizontal,

$$p_x = x - \frac{z_c}{g} \dot{x} , \quad (1.2)$$

donde x es el desplazamiento horizontal, z_c es la distancia del suelo a la posición de la masa móvil del modelo *Cart-Table*, g es la constante de gravedad y \ddot{x} es la aceleración.

Esta ecuación se puede transformar en un sistema dinámico

$$\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_x, \quad (1.3)$$

$$p_x = \begin{bmatrix} 1 & 0 & \frac{-z_c}{g} \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}. \quad (1.4)$$

Posteriormente utiliza la dinámica de este sistema para obtener un controlador que sigue la trayectoria del ZMP para construir un generador de patrones de caminata como lo muestra el diagrama de la Figura 1.1. El sistema genera la trayectoria del centro de masa tal que el ZMP resultante sigue la referencia dada. Para obtener el patrón de caminata implementa un control que usa información futura.

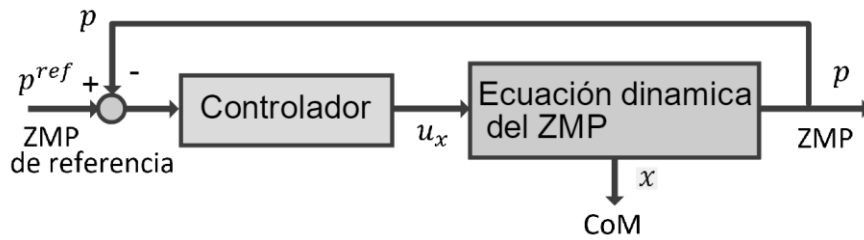


Figura 0.1. Generador de Patrones de Marcha, Control de seguimiento del ZMP.

Finalmente realiza simulaciones en OpenHRP [4] para demostrar el desempeño del generador de patrones de marcha.

Las aplicaciones más importantes utilizando el concepto de ZMP son realizadas por compañías como HONDA y su robot humanoide ASIMO (2000) [14], Kawada industries Inc. con su robot humanoide HRP (1998) [13] y la Universidad de Tsukuba en la implementación de exoesqueletos (2001) [13].

1.2. Módulo de medición

En [5] se propone el uso de una unidad de medición inercial que contiene un acelerómetro triaxial, un giroscopio triaxial y un magnetómetro triaxial, colocados en la cintura de una persona de edad avanzada para medir la postura de la cadera y así prever cuando se va a presentar una caída, al momento de cumplirse ciertas condiciones dadas por este dispositivo. Adicionalmente se tiene un soporte formado por dos estructuras que ayudan a las piernas a caminar y mantener el equilibrio sin ninguna manipulación directa. La estabilidad es analizada mediante el ZMP basado en las mediciones de este dispositivo.

El Instituto de Investigación Científica e Industrial (ISIR, por sus siglas en inglés) de la Universidad de Osaka en Japón, recolectó una base de datos de la marcha humana para ayudar en el desarrollo, prueba y evaluación de algoritmos de identificación de dicha marcha [6]. El dispositivo utilizado para recabar esta información consiste de 4 sensores: 3 unidades de medición inercial, las cuales están formadas por 1 acelerómetro triaxial y un giroscopio triaxial cada una, 1 Smartphone Motorola ME860, que incluye únicamente un acelerómetro triaxial.

Los sensores se colocan alrededor de la cintura de la persona por medio de un cinturón diseñado para cargar dichos sensores, de modo que queda colocado un sensor en el lado izquierdo, otro en el lado derecho, uno en el centro enfrente y el Smartphone en el centro en la espalda. Las unidades de medición inercial tienen una frecuencia de muestreo de 100 Hz.

1.3. Técnica de modelado

La identificación de sistemas consiste en construir modelos matemáticos de sistemas dinámicos a partir de las entradas y salidas observadas. Existen diversas técnicas de identificación de sistemas que pueden clasificarse por varios criterios: se tienen los métodos paramétricos y no paramétricos dependiendo del tipo de modelo empleado. Se tienen métodos de identificación *off-line* y *on-line* dependiendo de la aplicación.

1.3.1. Técnicas de identificación del ZMP

La técnica de identificación usada por Wael Suleiman en [7], parte del modelo de un robot humanoide considerado como un sistema cuadrático:

$$X_{k+1} = A_1 X_k + B_1 u_k, \quad (1.5)$$

$$Z_{k+1} = A_2 Z_k + B_2 (u_k \otimes X_k), \quad (1.6)$$

$$p_k = C_1 X_k + C_2 Z_k, \quad (1.7)$$

donde u_k es la entrada del sistema y p_k es la salida del mismo (ZMP). $\{X_k, Z_k\}$ son los estados y \otimes es el producto Kronecker que se define mediante la siguiente expresión:

$$\text{Sean } X \in \mathbb{R}^{m \times n} \text{ y } Y \in \mathbb{R}^{m \times n}: X \otimes Y \in \mathbb{R}^{mp \times nl} = \begin{bmatrix} x_{11}Y & \dots & x_{1n}Y \\ \vdots & \ddots & \vdots \\ x_{m1} & \dots & x_{mn}Y \end{bmatrix}. \quad (1.8)$$

El sistema está definido por los coeficientes de las matrices de las ecuaciones (1.5) (1.6) y (1.7), principalmente por las matrices desconocidas A_2 , B_2 y C_2 , lo que significa que estos parámetros pueden ser descritos por

$$\theta = \begin{bmatrix} \text{vec}(A_2) \\ \text{vec}(B_2) \\ \text{vec}(C_2) \end{bmatrix}, \quad (1.9)$$

donde $\text{vec}(\ast)$ denota el operador definido como

$$\text{vec}(M) = \text{vec}[m_1 \quad m_2 \quad \cdots \quad m_n] = [m_1^T \quad m_2^T \quad \cdots \quad m_n^T]^T. \quad (1.10)$$

Dadas las señales de entrada y salida del sistema real, lo que se requiere es que la salida del siguiente sistema

$$\hat{X}_{k+1} = A_1 \hat{X}_k + B_1 u_k, \quad (1.11)$$

$$\hat{Z}_{k+1}(\theta) = A_2(\theta) \hat{Z}_k(\theta) + B_2(\theta)(u_k \otimes \hat{X}_k), \quad (1.12)$$

$$\hat{p}_k = C_1 \hat{X}_k + C_2(\theta) \hat{Z}_k(\theta), \quad (1.13)$$

se aproxime a la salida p_k del sistema real con la mayor precisión posible.

Este criterio puede ser transformado en la minimización del error de la salida con respecto de los parámetros de θ . Si se considera un conjunto de datos con N muestras, la función correspondiente al error de la salida está dada por

$$J_N(\theta) = \frac{1}{N} \sum_{k=1}^N \|p_k - \hat{p}_k(\theta)\|_2^2 = \frac{1}{N} E_N(\theta)^T E_N(\theta), \quad (1.14)$$

La solución numérica de este problema se puede calcular por diversos algoritmos. Por ejemplo, el método de búsqueda mediante el gradiente iterativo, el cual está basado en actualizar los parámetros de θ mediante

$$\hat{\theta}^i = \hat{\theta}^{i-1} - (\psi_N^T(\hat{\theta}^{i-1}) \psi_N(\hat{\theta}^{i-1}) + \lambda^i I)^{-1} \psi_N^T(\hat{\theta}^{i-1}) E_N(\hat{\theta}^{i-1}), \quad (1.15)$$

donde λ^i es el parámetro de regularización y

$$\psi_N(\theta) = \frac{\partial E_N(\theta)}{\partial \theta^T}, \quad (1.16)$$

es el Jacobiano del vector de error $E_N(\theta)$.

1.3.2. Modelo *Cart-Table*

El modelo *Cart-Table* se utiliza para aproximar toda la masa de un robot humanoide a una masa localizada en su centro de masa, con lo que el problema de controlar el movimiento del robot humanoide se reduce a controlar un péndulo invertido. Este modelo es el más utilizado para la creación de patrones estables de marcha usando una trayectoria específica del ZMP.

En [8], Kajita asume que la caminata bípeda es equivalente al problema de balancear una masa sobre una superficie que se encuentra sobre un soporte inestable, ya que en la fase de la caminata en la que se está apoyando con una sola pierna, puede ser presentada por el modelo del péndulo invertido o del *Cart-Table*.

La caminata bípeda puede entonces ser modelada mediante el movimiento del ZMP y el balanceo del cuerpo. Un robot humanoide o una persona están en balance, cuando la posición del ZMP se encuentra dentro del polígono de soporte. Cuando el ZMP sale del polígono de soporte, el cuerpo pierde el balance. La marcha bípeda puede ser generada modelando el ZMP deseado para la posible trayectoria del centro de masa (CoM, Center of Mass) y el movimiento del cuerpo puede ser aproximado por el modelo dinámico del *Cart-Table*. Por

ejemplo, en [9] y [10] se utiliza este modelo para obtener distintos algoritmos de control para generar diversos tipos de marcha bípeda.

1.3.3. Técnica de subespacios

Las técnicas de subespacios son variadas y han sido utilizadas para la resolución de diversos problemas. Uno de los precursores de los algoritmos de identificación de subespacios es Peter Van Overschee cuyo trabajo es de gran importancia, ya que unifica este tipo de técnicas y las lleva a la práctica [27].

Hay una gran variedad de trabajos en la literatura con respecto a la identificación de subespacios, algunos de los autores más importantes en esta área son:

- Overschee y Moor
- Larimore
- Boets
- Viberg
- Chiuso y Picci
- Sima y Hoffel
- Cock
- Goethals

Algunas de las técnicas más importantes de identificación de sistemas basadas en subespacios son las siguientes [11]:

- N4SID (*Numerical Algorithms for Subspace State Space System Identification*) desarrollado por Overschee y Moor. Este algoritmo es útil para determinar el orden del sistema y la secuencia de estados, pero es sensible cuando se realizan cambios en las entradas o salidas.
- CVA (*Canonical Variate Analysis*) creado por Larimore. Este algoritmo es óptimo para determinar el orden del sistema a partir de muestras finitas. A diferencia del N4SID, esta técnica no es sensible a cambios en las entradas y salidas.
- MOESP (*Multivariable Output-Error State sPace*) desarrollado por Verhaegen. Esta técnica utiliza la descomposición LQ para determinar la matriz de observabilidad extendida y es sensible a cambios en las entradas o salidas.

En este capítulo se revisó el estado del arte de los conceptos de principal interés para la identificación de marcha humana mediante identificación de subespacios.

Zero Moment Point (ZMP) y caminata humanoide

El concepto de *Zero Moment Point* es de gran importancia para la robótica humanoide, ya que mediante este, se puede dar solución al problema del robot al moverse establemente mientras mantiene el contacto entre su pie y el suelo.

2.1. ¿Qué es el ZMP?

Primero se explica lo que es el polígono de soporte, el cual es un concepto de vital importancia para comprender el ZMP. Matemáticamente el polígono de soporte es una envolvente convexa, además es el conjunto convexo más pequeño que incluye todos los puntos de contacto. Aplicando este concepto a un robot humanoide, el polígono de soporte es la región formada por los puntos del perímetro del pie que está en contacto con el suelo.

El ZMP es un punto dentro del polígono de soporte en el cual el momento debido a las fuerzas inerciales y de gravedad son cero en el eje horizontal.

Para comprender este concepto se considera la estructura del cuerpo en una postura en la que carga su peso en un solo pie de apoyo, para hacer más simple el análisis se toma en cuenta solo el pie de apoyo, omitiendo el resto de la estructura que está encima del tobillo como se muestra en la representación del pie de la Figura 2.1, sustituyendo su influencia por una fuerza \vec{F}_A y un momento \vec{M}_A . Además, el peso del pie de apoyo está aplicado sobre su centro de gravedad. El pie también experimenta la fuerza de reacción del suelo en algún punto de la planta, esto es lo que mantiene todo el cuerpo en equilibrio.

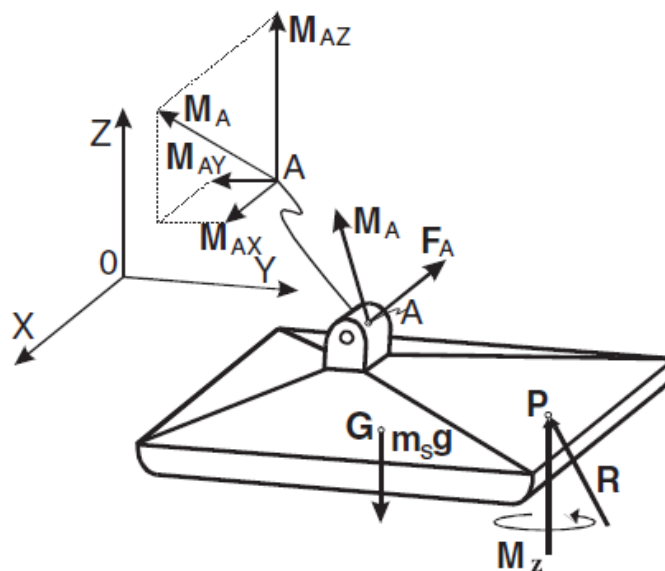


Figura 2.1. Fuerzas y momentos que actúan en el pie.

La fuerza de reacción con el suelo consiste de tres componentes de una fuerza denominada $\vec{R}(\vec{R}_x, \vec{R}_y, \vec{R}_z)$ y un momento $\vec{M}(\vec{M}_x, \vec{M}_y, \vec{M}_z)$. Ya que la fuerza de fricción actúa en el punto de contacto del pie con el suelo, las componentes de la fuerza R y el momento M que actúan en el plano horizontal serán balanceadas por la fricción, por lo tanto la fuerza de reacción en el plano horizontal (\vec{R}_x, \vec{R}_y) representa la fuerza de fricción que está equilibrando la componente horizontal de la fuerza \vec{F}_A , mientras que la componente vertical del momento \vec{M}_z representa la fuerza que balancea el momento inducido por la fuerza \vec{F}_A . En consecuencia, si se asume que en el contacto del pie con el suelo, éste no se puede resbalar, entonces la fricción compensará las componentes (\vec{R}_x, \vec{R}_y) y la reacción vertical \vec{M}_z . La componente vertical \vec{R}_z representa la reacción del suelo que equilibra las fuerzas verticales. Las componentes de los momentos se compensan únicamente al cambiar la posición de la fuerza de reacción \vec{R} dentro del polígono de soporte. Por lo tanto, la componente horizontal del momento \vec{M}_A cambiará la fuerza de reacción a la posición correspondiente. La fuerza de reacción siempre se encuentra dentro del área cubierta por el pie; el incremento en el movimiento del tobillo se compensará por el cambio de posición en esta fuerza de reacción y por lo tanto, no habrá componentes horizontales de los momentos \vec{M}_x y \vec{M}_y .

En consecuencia, la condición necesaria y suficiente para que un cuerpo se encuentre en equilibrio dinámico mientras camina, es que en un punto de la planta del pie donde la fuerza de reacción del suelo esté actuando, se cumpla que $\vec{M}_x = 0$ y $\vec{M}_y = 0$ [12].

2.2. Simplificación de la dinámica de la marcha humana mediante el Modelo *Cart-Table*

El modelo *Cart-Table* consiste en una masa móvil que se desplaza sobre una superficie cuya masa es lo suficientemente pequeña y entonces despreciable. Esta superficie tiene un soporte con una base tan pequeña, tal que, no podría mantener el balance cuando la masa móvil se encuentre en la orilla de la superficie; sin embargo, se puede mantener el equilibrio momentáneamente mientras la masa se mueva con cierta aceleración [13].

La dinámica de la marcha humana se puede aproximar por la masa móvil desplazándose sobre una superficie como se muestra en la Figura 2.2; ya que el centro de masa está representado por la masa M , Z_c es la distancia del suelo al centro de masa y \ddot{x} es la aceleración con la cual se desplaza la masa. La posición de la masa determina el centro de presión que actúa sobre el suelo y por lo tanto se obtiene el ZMP.

En este modelo, el ZMP está definido por la siguiente expresión:

$$p = x - \frac{Z_c}{g} \ddot{x}, \quad (2.1)$$

donde x es el desplazamiento de la masa a lo largo de la superficie, Z_c es la distancia del suelo a la masa, g es la constante de gravedad, \ddot{x} es la aceleración de la masa y p es el *Zero Moment Point*.

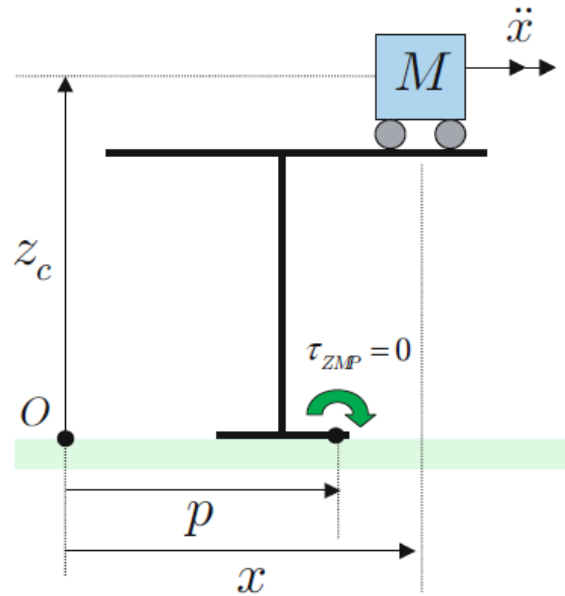


Figura 2.2. Modelo *Cart-Table*.

Otro modelo que se utiliza para aproximar la marcha humana es el del péndulo invertido. En este modelo el movimiento del centro de masa se calcula a partir del ZMP, mientras que en el modelo del Cart – Table, el ZMP es generado por el movimiento del centro de masa. Estos modelos, aunque son parecidos, tienen una relación entrada – salida opuesta.

2.3. Caminata humanoide

En este trabajo, la caminata humanoide se refiere a la forma de desplazarse de una persona y de cualquier robot bípedo.

2.3.1. Robots humanoides

En la caminata de robots humanoides, el uso del ZMP es fundamental para la creación de algoritmos de control dinámico para con esto, planificar una secuencia de movimiento estable en cualquier robot bípedo.

La evolución de la caminata en robots humanoides se puede apreciar de manera general en la evolución de los robots humanoides de la marca japonesa HONDA. El primer prototipo de esta compañía consistía en un par de piernas sin la parte superior del cuerpo, a la cual le tomaba 5 segundos dar un paso, con la limitante de que solo podía caminar en línea recta. Con los prototipos siguientes, la serie (E1, E2, E3), el objetivo que se buscaba era el de lograr una forma de caminar más parecida a la del ser humano. En los prototipos que siguieron (E4, E5, E6) se enfocaron en darle más capacidades de movimiento a su robot, como lo son el poder moverse más rápido, subir y bajar escaleras, además de pendientes. En el prototipo P1 crearon un robot humanoide completo al montar un torso al robot de dos piernas que ya

habían creado. El prototipo P2 caminaba con suficiente estabilidad, tomando en cuenta el peso que cargaba al añadirle el peso del torso, lo que hacía su caminata más parecida a la del ser humano. El modelo P3 fue el robot más compacto que habían creado, nuevamente para darle un parecido más cercano a un ser humano. Este podía caminar más rápido que sus antecesores, con mejor balance y de una manera más natural. Después de más de 15 años de investigación y desarrollo, en el año 2000 se presentó la primera versión de ASIMO; una de sus características más importantes fue su habilidad para caminar, ya que podía girar, subir escaleras, ejecutar patrones complicados de marcha además de poder cambiar la velocidad y longitud de sus pasos. En versiones posteriores de ASIMO, éste presenta la habilidad de correr en línea recta a una velocidad de 6km/h con la capacidad de poder cambiar su dirección mientras corre. Todo esto con un movimiento más dinámico y fluido [14]. La evolución de los robots de HONDA se muestra en la Figura 2.3.



Figura 2.3. Evolución de los robots humanoides de Honda [14].

2.3.2. Caminata humana

La marcha humana consiste en una serie de movimientos repetidos, regulares que desplazan el centro de masa a lo largo de una trayectoria deseada. Se puede considerar como un cambio continuo entre estabilidad y movilidad.

Algunas características generales que debe de tener un sistema locomotor para lograr una marcha estable son:

- Cada pierna debe ser capaz de soportar el peso del cuerpo sin colapsar,
- El equilibrio se debe de mantener tanto en forma estática como en forma dinámica,

- La pierna oscilante debe poder convertirse en la pierna de soporte,
- Debe de poder proporcionar la fuerza suficiente para realizar el movimiento de las extremidades y avanzar el torso.

El equilibrio es un concepto importante cuando se trata el tema de la marcha humana, éste se puede dividir en equilibrio estático y dinámico. El equilibrio estático se presenta cuando se mantiene el control de una postura sin desplazamiento. El equilibrio dinámico se establece cuando el centro de masa sale de la vertical del cuerpo y este es devuelto a la base de sustentación mediante una acción que lo equilibre.

Una característica importante del estado de equilibrio es la estabilidad, ya que cuando se produce una perturbación sobre un sistema en un determinado estado de equilibrio y este ya no vuelve al mismo, entonces se tiene un estado de equilibrio inestable. Por otro lado, si después de cesar la perturbación el sistema vuelve al estado original, entonces se tiene un estado de equilibrio estable [15].

La estabilidad en la marcha humana está directamente relacionada con el polígono de soporte. Cada que cambia la posición de los pies también cambia la forma del polígono de soporte, así que cuando el cuerpo se sustenta con un solo pie de apoyo, se presenta un estado con cierto grado de estabilidad en comparación con el caso cuando se encuentra apoyado por ambos pies, esto debido a que el polígono de soporte es más pequeño en el primer caso. En la Figura 2.4 se muestran los polígonos de soportes de los casos mencionados [16].

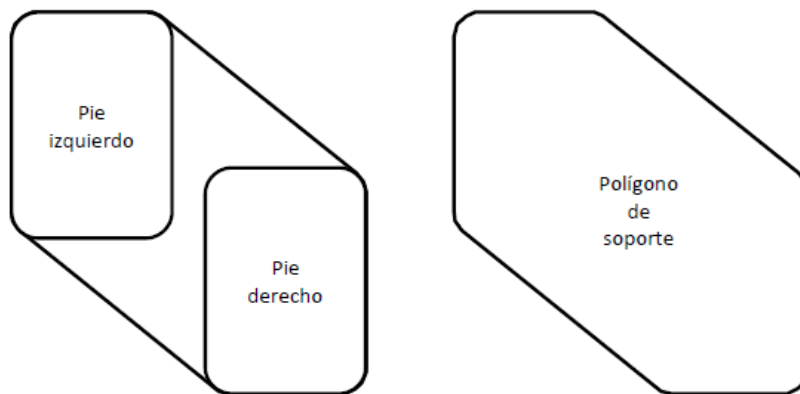


Figura 2.4. Polígono de soporte para la estabilidad de la marcha humana.

En este capítulo se revisaron los conceptos relacionados con la marcha humanoide para poder comprender este proceso y poder reproducirlo experimentalmente mediante un dispositivo que se presentara en el capítulo siguiente

CAPÍTULO 3

Arreglo de medición

El objetivo principal del arreglo de medición es obtener datos experimentales del movimiento del centro de masa en la caminata humana, para después procesarlos y enviarlos a una computadora vía inalámbrica para su futuro análisis. Este procedimiento se muestra en el diagrama a bloques de la Figura 3.1

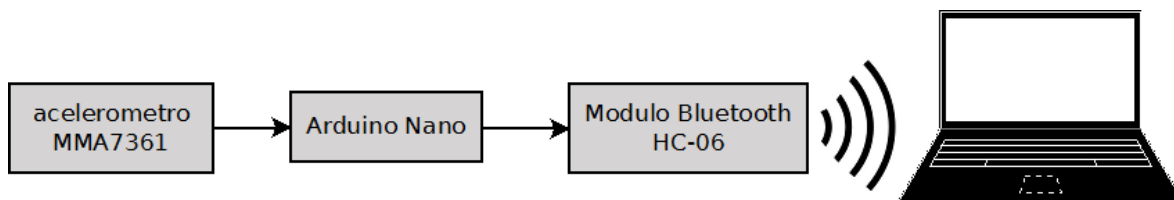


Figura 3.1. Diagrama de bloques del sistema de adquisición de datos para la marcha humana.

Para medir el desplazamiento del centro de masa se utiliza un acelerómetro. Un acelerómetro es un dispositivo electromecánico que mide las fuerzas de aceleración; existen diversos tipos dependiendo del tipo de tecnología que utilicen para medir esta magnitud: mecánicos, piezoeléctricos, capacitivos, MEMS, entre otros. La unidad de medida de estos dispositivos es el g, el cual es equivalente a $9.8m/s^2$. Estos dispositivos son utilizados para medir los cambios de velocidad con respecto del tiempo y para el control de mecanismos de vibración.

3.1. Acelerómetro MMA7361

El acelerómetro que se usó en este trabajo es el MMA7361, el cual es de tipo MEMS, además es un acelerómetro analógico de 3 ejes [17].

Las características principales del sensor MMA7361 son:

- Bajo consumo de corriente: $400\mu A$
- Modo de espera: $3\mu A$
- Bajo voltaje de operación: $2.2V - 3.6V$
- Alta sensibilidad: $800mV/1.5g$
- Sensibilidad seleccionable: $\pm 1.5g, \pm 6g$
- $0g$ detect para protección de caída libre
- Acondicionamiento de señales con filtro pasa bajas.
- Diseño robusto
- Dimensiones: $3mm \times 5mm \times 1mm$

En la Figura 3.2 se muestra la organización de los pines del MMA7361.

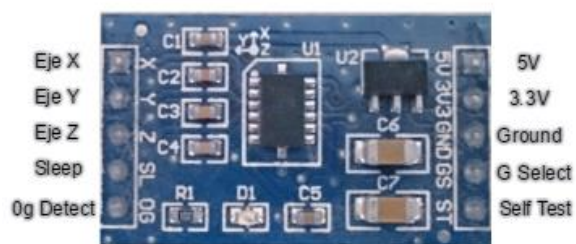


Figura 3.2. Acelerómetro MMA7361 [17].

A continuación, se muestra la descripción de estos pines:

Eje x. Es una señal de salida analógica correspondiente al valor de la aceleración en el eje x.

Eje y. Es una señal de salida analógica correspondiente al valor de la aceleración en el eje y.

Eje z. Es una señal de salida analógica correspondiente al valor de la aceleración en el eje z.

Sleep (SL). Este pin se activa de forma negada y el integrado pasará a modo “sleep”, en este modo, no se enviará nada a sus salidas.

Detect (0g). Este pin se activa cuando se detecta 0g en los tres ejes. Útil para detectar caída libre.

5V. Este pin está conectado a un regulador que transforma 5V a 3.3V que requiere el integrado para funcionar.

3.3V. Este pin no tiene regulador, se usa cuando se tiene el voltaje de 3.3V regulado previamente.

Ground (GND). Este pin se conecta a la tierra del circuito.

G Select (GS). Este pin se utiliza para seleccionar entre las dos sensibilidades. Si este pin está en bajo, se encuentra en modo 1.5g y si está en alto se cambia al modo 6g.

Self Test (ST). Este pin sirve para verificar que tanto las piezas mecánicas y eléctricas en el interior del integrado estén funcionando correctamente.

3.2. Implementación de código en Arduino

El acelerómetro tiene la capacidad de obtener el ángulo de inclinación al que está sometido, esto debido a que también detecta la fuerza de gravedad que afecta al integrado, conociendo

la dirección de esta fuerza en conjunto con algunas funciones trigonométricas, se obtiene el ángulo de inclinación.

Para realizar el procedimiento que obtiene el ángulo, se utiliza la plataforma Arduino, en específico se utiliza el Arduino Nano debido a que se necesita un dispositivo que sea portable y que se pueda colocar en la cintura de una persona para realizar la medición de los desplazamientos del centro de masa.

Las características principales de Arduino son:

- Voltaje de operación: 5V
- Voltaje de entrada recomendado: 7V – 12V
- Voltaje de entrada límite: 6V – 20V
- 14 entradas/salidas digitales
- 8 entradas analógicas
- Corriente por pin entrada/salida: 40mA
- Memoria Flash: 32 KB de los cuales 2 KB son usados por el “bootloader”
- SRAM: 2 KB
- EEPROM: 1 KB
- Velocidad de reloj: 16 MHz
- Dimensiones (largo): 45mm
- Dimensiones (ancho): 18mm
- Peso: 5g

La tarea a realizar es adquirir los datos del acelerómetro y procesarlos para obtener la inclinación (en grados) del dispositivo. El código realizado en el software de Arduino consiste en una primera parte donde se adquieren los valores analógicos del sensor, para posteriormente convertir estos valores mediante el ADC del Arduino Nano, el cual tiene una resolución de 10 bits, con lo que se tienen 1024 diferentes valores para cada variable obtenida. Estos valores se ajustan a una escala que varía entre números positivos y negativos. El ADC tiene un voltaje de referencia de 3.3 V, para convertir el valor de ADC a voltaje se utiliza la siguiente expresión

$$V_{adc} = \frac{Valor_{adc} * V_{ref}}{1023} . \quad (3.1)$$

El siguiente paso es convertir el valor del voltaje obtenido a su correspondiente valor en g. Esto se logra al utilizar la sensibilidad del acelerómetro, en el caso del acelerómetro usado es de 800 mv/g con lo que resulta

$$Acc = \frac{V_{adc}}{0.8} . \quad (3.2)$$

Este procedimiento se describe en el diagrama de flujo que se muestra en la Figura 3.3.

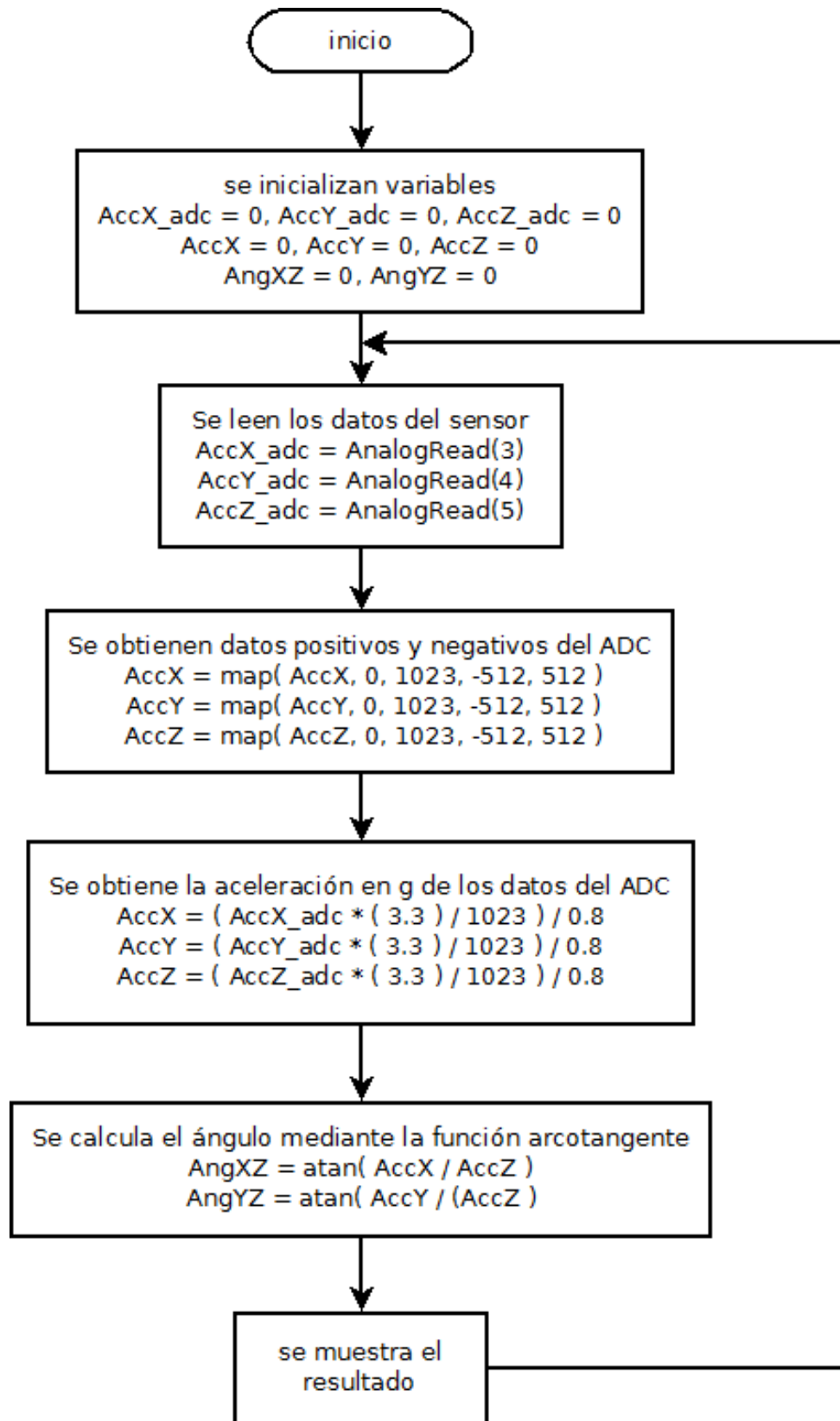


Figura 3.3. Diagrama de Flujo que describe el procedimiento para obtener la información del acelerómetro.

3.3. Módulo Bluetooth HC-06

Otro aspecto importante es el tipo de comunicación entre el dispositivo y la computadora. Para poder realizar las mediciones del centro de masa sin restricciones de espacio que podrían generar si se conectara el dispositivo de medición, mediante un cable hacia la computadora, se desea que sea inalámbrico. Para añadir esta característica, además de los elementos antes mencionados, se utilizó el módulo de comunicación Bluetooth HC-06 que se muestra en la Figura 3.4.



Figura 3.4. Módulo de comunicación Bluetooth HC-06.

Las especificaciones del módulo HC-06 [18] son:

- Sensibilidad (tasa de error de bits) que puede alcanzar -80dB
- Cuenta con un módulo EDR (Enhanced Data Rate) para acelerar la transferencia de datos
- Cuenta con una antena integrada a 2.4GHz
- Cuenta con una memoria FLASH de 8Mbit
- Voltaje de operación: $3.1\text{V} \sim 4.1\text{V}$
- El consumo de corriente cuando se está sincronizando el dispositivo, se encuentra en el rango de $30\text{mA} \sim 40\text{mA}$. Mientras se encuentra en comunicación, la corriente es de 8mA
- Tiene un puerto estándar HCI (Host Controller Interface)
- Tiene un transceptor inalámbrico digital a 2.4GHz
- Tecnología Bluetooth CSR BC04
- De dimensiones: $27\text{mm} \times 13\text{mm} \times 2\text{mm}$
- Temperatura de funcionamiento: -25° a $+75^\circ$
- Bajo consumo de potencia
- Bajo costo

3.4. Primer prototipo del módulo de medición

El módulo de comunicación Bluetooth HC-06 se integra con el acelerómetro y el Arduino Nano para formar el primer prototipo de medición de desplazamiento del centro de masa. El diagrama esquemático se muestra en la Figura 3.5, mientras que el diagrama de conexiones se muestra la Figura 3.6

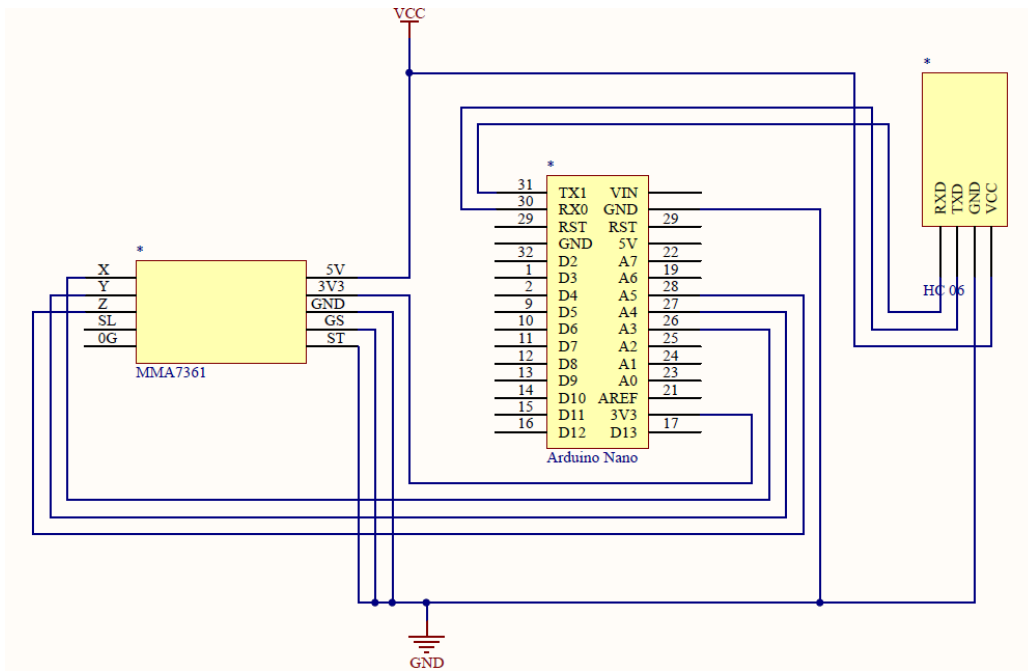


Figura 3.5. Diagrama esquemático del módulo de adquisición de datos.

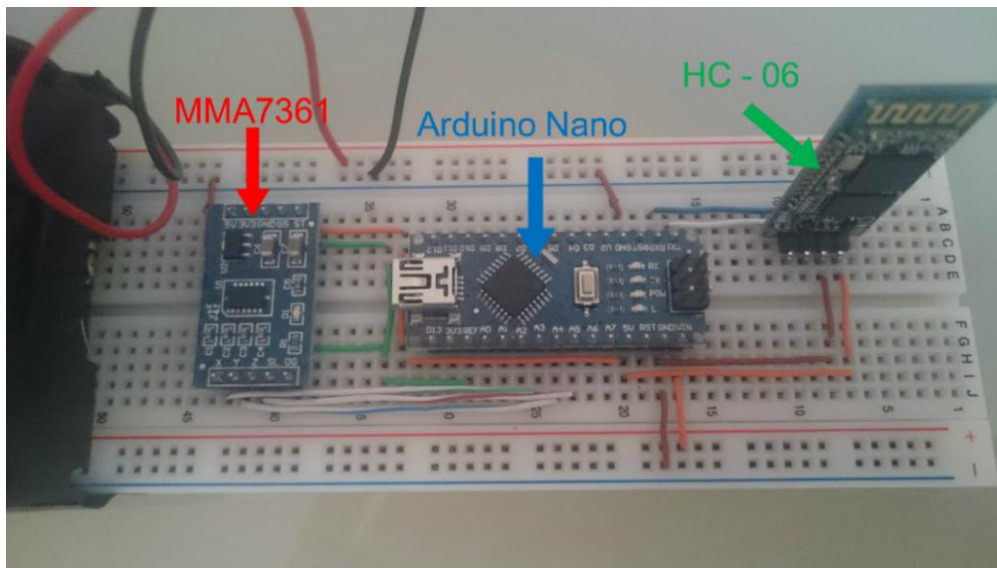


Figura 3.6. Módulo de adquisición de datos para la medición del desplazamiento, conformado por acelerómetro, Arduino Nano y módulo Bluetooth.

La orientación de los ejes x , y y z , al colocar el dispositivo en la cintura de la persona, es la que se presenta en la Figura 3.7

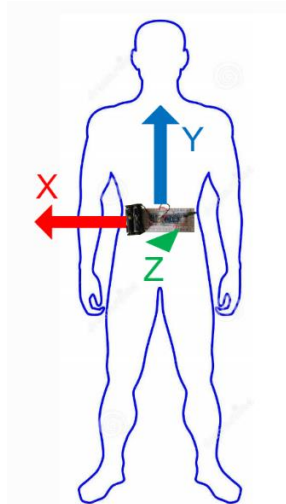


Figura 3.7. Orientación de los ejes cuando se coloca en la cintura de la persona.

Para realizar la comunicación entre el dispositivo de medición y la computadora, con el fin de obtener los datos de la marcha humana, se utiliza un software para diversos protocolos de red llamado PuTTY. Este software permite configurar el módulo HC-06 como se muestra en la Figura 3.8

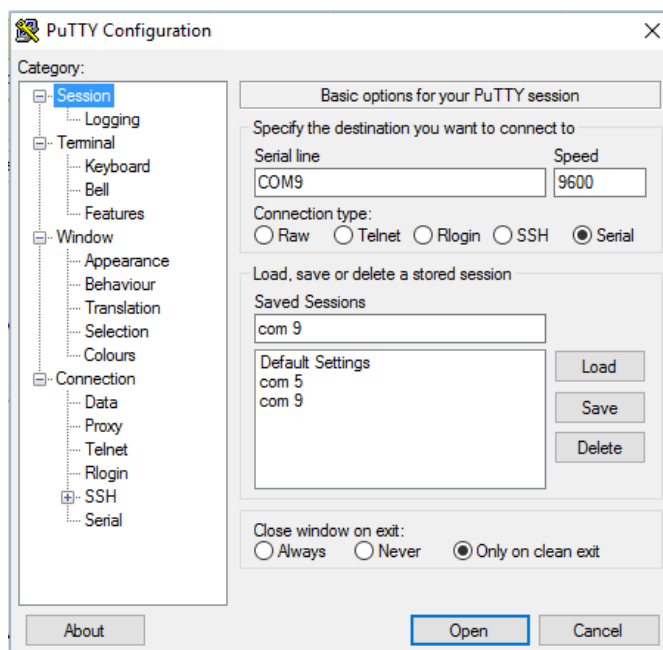


Figura 3.8. Ventana del programa PuTTY para realizar la configuración del módulo HC-06 (mediante comunicación inalámbrica).

Para comenzar la comunicación vía inalámbrica en la computadora se oprime el botón "open". Después de realizar la configuración, los datos del acelerómetro se obtendrán en

tiempo real, además de poder guardarlos en un archivo de texto al finalizar la comunicación, esto último con el fin de poder analizarlos en el futuro.

La ventana donde se muestra la información se muestra en la Figura 3.9.

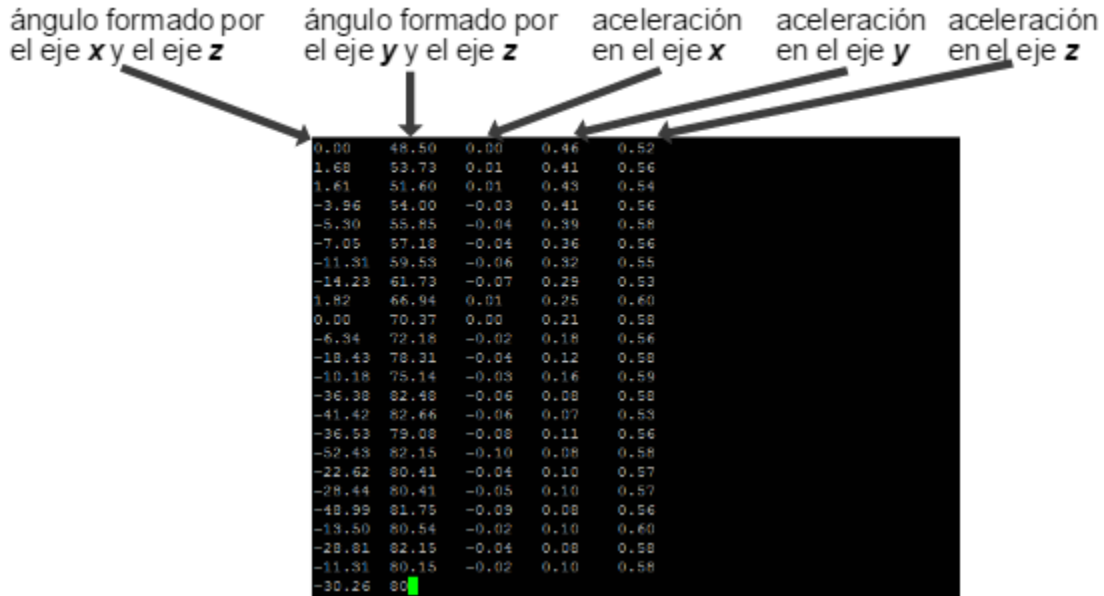


Figura 3.9. Ventana de datos obtenidos.

Los datos desplegados son, en la primera columna el ángulo en el eje x, en la segunda columna el ángulo en el eje y, en la tercera columna se despliega la aceleración en el eje x, en la cuarta columna la aceleración en el eje y y por último, en la quinta columna se tiene la aceleración en el eje z.

Los datos obtenidos tienen mucho ruido debido que los acelerómetros en general tienen esta desventaja, a esto se le agrega el hecho de que los acelerómetros detectan la aceleración provocada por cualquier movimiento provocando un error en las mediciones adquiridas. En la Figura 3.10 se muestra una señal obtenida mediante el dispositivo de medición.

Ya que las frecuencias importantes en las mediciones del acelerómetro son bajas, se propone usar un filtro pasa-bajas IIR [19] de orden uno con la siguiente función de transferencia:

$$H(z) = \frac{1-\alpha}{1-\alpha z^{-1}}, \quad (3.3)$$

que se representa con la siguiente ecuación en diferencias:

$$y[n] = (1 - \alpha)x[n] + \alpha y[n - 1], \quad (3.4)$$

donde $y[n]$ representa el valor de la aceleración filtrada actual, $x[n]$ es la muestra actual tomada del acelerómetro, $y[n - 1]$ es la muestra filtrada en el instante anterior. El valor de α determina la frecuencia de corte del filtro.

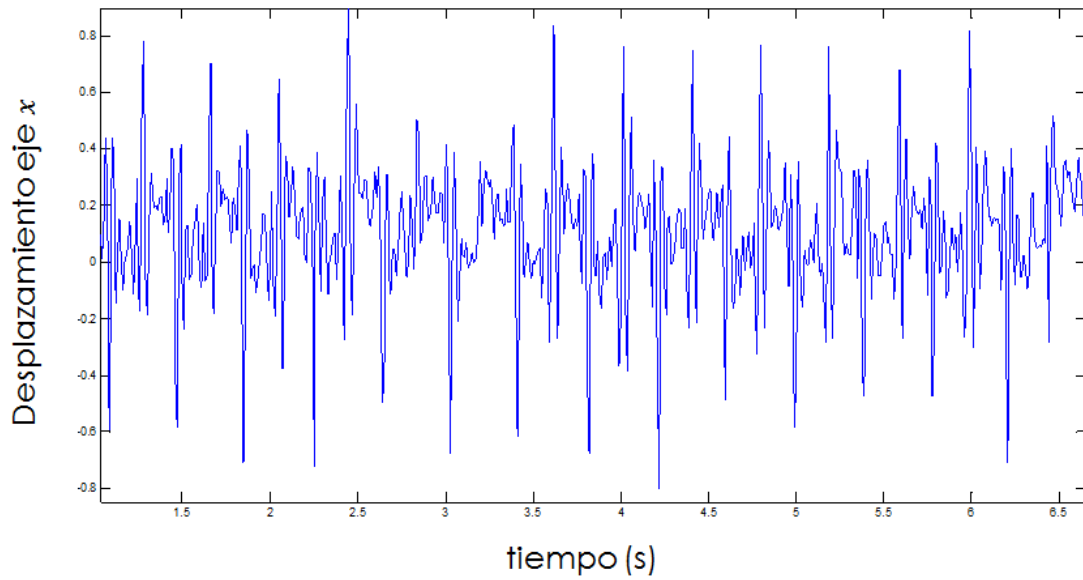


Figura 3.10. Señal obtenida con el dispositivo de medición implementado.

Se desarrolló un código en MATLAB para implementar el filtro de la ecuación (3.4) considerando como entrada al filtro, el vector de datos obtenidos por el acelerómetro.

Utilizando este filtro en la señal de la figura 3.9, se obtiene la señal filtrada que se muestra en la figura 3.11.

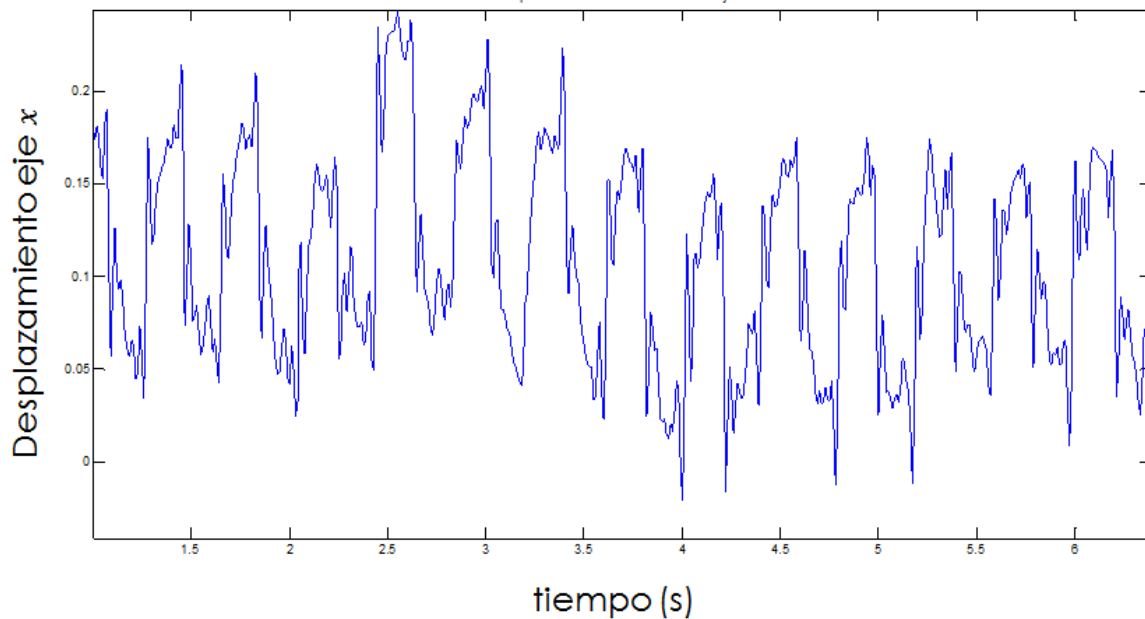


Figura 3.11. Señal filtrada.

El desplazamiento del centro de masa se obtuvo en unidades de grados, ya que se calculó el ángulo formado al caminar con el eje x , al realizar los movimientos de la caminata; esto es

similar al movimiento que se da en un péndulo invertido. Después, estos datos se convirtieron en un desplazamiento como el que se obtiene en el modelo Cart-Table, ya que se conoce la distancia del suelo al centro de masa (en este caso el acelerómetro). Con estos datos y mediante relaciones trigonométrica, se obtuvo el valor de la variable que se buscaba como lo muestra la Figura 3.12.

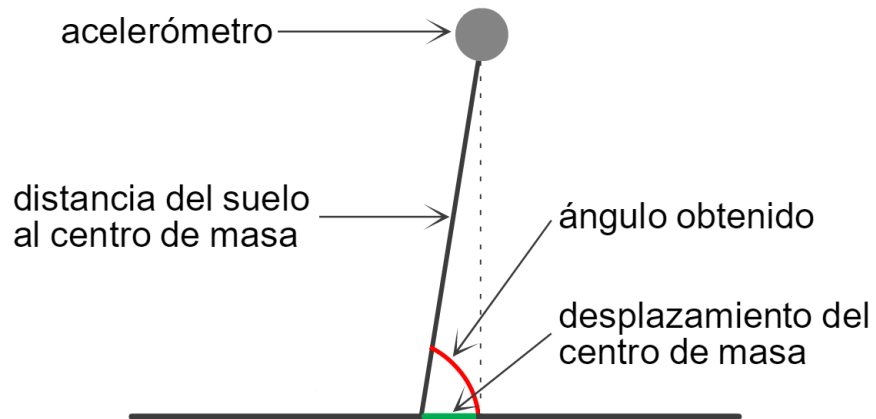


Figura 3.12. Movimiento de la marcha humana representado como un péndulo invertido.

Algoritmo de identificación de subespacios

El problema a resolver con la identificación de subespacios es el de encontrar modelos dinámicos a partir de datos de entrada/salida, los cuales son generados por sistemas lineales invariantes en el tiempo y de dimensión finita [20]. Este proceso se describe en la Figura 4.1

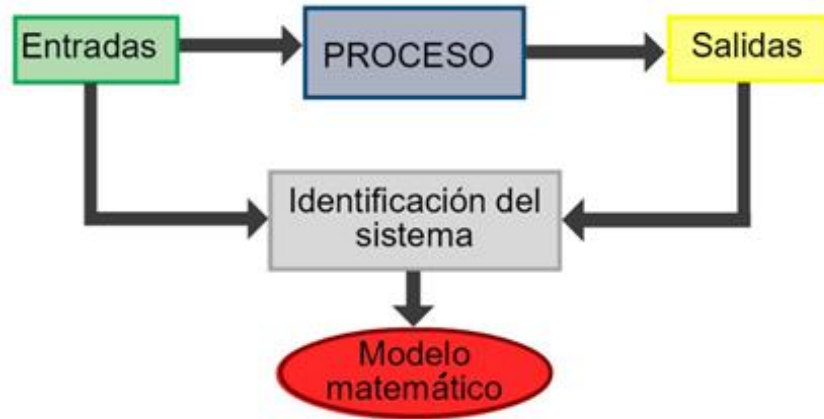


Figura 4.1. Diagrama del proceso de la identificación de sistemas.

Este tipo de técnicas de identificación tienen diversas ventajas, como por ejemplo, que no necesita un modelo de parametrización específico y además el hecho de tener una mejor eficiencia computacional en comparación con otros algoritmos.

El modelo matemático usado está descrito por la siguiente ecuación en variables de estado:

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad (4.1)$$

$$y_k = Cx_k + Du_k + v_k, \quad (4.2)$$

$$E \left[\begin{pmatrix} w_p \\ v_p \end{pmatrix} \begin{pmatrix} w_q^T & v_q^T \end{pmatrix} \right] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \delta_{pq} \geq 0, \quad (4.3)$$

donde $x_k \in \mathbb{R}^n$ es el vector de estados en el tiempo k , $u_k \in \mathbb{R}^m$ son las mediciones de las m entradas, $y_k \in \mathbb{R}^l$ son las mediciones de las l salidas, $v_k \in \mathbb{R}^l$ y $w_k \in \mathbb{R}^n$ son secuencias de ruido. La matriz $A \in \mathbb{R}^{n \times n}$ contiene la dinámica del sistema, $B \in \mathbb{R}^{n \times m}$ es la matriz de entrada que representa la transformación lineal por la cual las entradas tienen influencia en el siguiente estado, $C \in \mathbb{R}^{l \times n}$ es la matriz de salida que describe como el estado interno es transferido al exterior en los valores de y_k , $D \in \mathbb{R}^{l \times m}$ es la matriz de transmisión directa, $Q \in \mathbb{R}^{n \times n}$, $S \in \mathbb{R}^{n \times l}$ y $R \in \mathbb{R}^{l \times l}$ son las matrices de covarianza de las secuencias de ruido w_k y v_k . E es el operador de los valores esperados y δ_{pq} es el operador delta Kronecker.

El par de matrices $\{A, B\}$ se asumen observables, lo que implica que todos los estados del sistema pueden ser observados en la salida y_k y por lo tanto pueden ser identificados.

Las matrices $\{A, [B \quad Q^{1/2}]\}$ se asumen controlables, lo que implica que todos los estados del sistema son excitados por la entrada determinística u_k o la entrada estocástica w_k .

El problema de identificación de subespacios consiste en lo siguiente: dados los datos experimentales de las entradas $u_1 \dots u_s$ y las salidas $y_1 \dots y_s$, se deben encontrar las matrices del sistema A, B, C, D, Q, S, R .

4.1. Herramientas para la identificación de subespacios

Las herramientas necesarias para la implementación de los algoritmos de identificación de subespacios son principalmente geométricas, de entre las que destacan las proyecciones ortogonales y oblicuas; también se hace uso de matrices de Hankel, la pseudoinversa Moore Penrose, la descomposición de matrices y el filtro de Kalman; estas herramientas se describen a continuación.

4.1.1. Proyecciones ortogonales y oblicuas

Este tipo de identificación usa frecuentemente conceptos geométricos, ya que diversas características de un sistema se pueden obtener mediante una manipulación geométrica de los espacios fila de algunas matrices [21].

En las siguientes secciones se utilizan las matrices A y B , para describir diversos operadores, estas matrices no deben confundirse con las matrices A y B del modelo en espacio de estados.

Proyecciones ortogonales

El operador \prod_B denota la proyección del espacio fila de una matriz en el espacio fila de la matriz $B \in \mathbb{R}^{q \times j}$:

$$\prod_B \triangleq B^\dagger B, \quad (4.4)$$

donde B^\dagger denota la pseudoinversa de Moore-Penrose de la matriz B . Esta matriz se define como $B^\dagger = B^T (BB^T)^{-1}$ si $q < j$ y $B^\dagger = (BB^T)^{-1} B^T$ si $q > j$, siempre y cuando exista la inversa $(BB^T)^{-1}$ [1].

Una **proyección ortogonal** A/B se define como la proyección del espacio fila de una matriz $A \in \mathbb{R}^{p \times j}$ en el espacio fila de la matriz B :

$$A/B \triangleq A \prod_B = AB^\dagger B. \quad (4.5)$$

Esta proyección se puede interpretar geométricamente en el ambiente de dos dimensiones representado en la Figura 4.2.

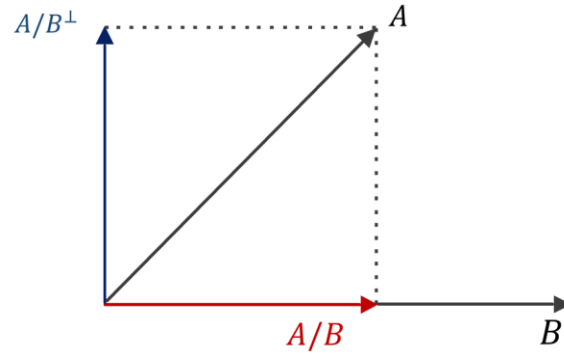


Figura 4.2. Proyección ortogonal de matrices.

El operador geométrico Π_{B^\perp} denota la proyección del espacio fila de una matriz en el complemento ortogonal del espacio fila de la matriz B :

$$A/B^\perp \triangleq A\Pi_{B^\perp}, \quad (4.6)$$

donde

$$\Pi_{B^\perp} = I_j - \Pi_B. \quad (4.7)$$

La combinación de los dos operadores geométricos Π_B y Π_{B^\perp} descompone la matriz A en dos matrices cuyo espacio fila es ortogonal:

$$A = A\Pi_B + A\Pi_{B^\perp}. \quad (4.8)$$

Proyecciones oblicuas

Se puede descomponer la matriz A como una combinación lineal de dos matrices no ortogonales B y C , además del complemento ortogonal de B y C . Las filas de la matriz A son descompuestas como una combinación lineal de las filas de B y C además de las filas de una tercera matriz ortogonal a B y C . Lo anterior es representado por la expresión:

$$A = L_B B + L_C C + L_{B^\perp, C^\perp} \begin{pmatrix} B \\ C \end{pmatrix}^\dagger. \quad (4.9)$$

La matriz $L_C C$ se define como la proyección oblicua del espacio fila de A a lo largo del espacio fila de B en el espacio fila de C :

$$A /_B C \triangleq L_C C. \quad (4.10)$$

Lo anterior se interpreta como la proyección ortogonal del espacio fila de A dentro del espacio fila de B y C , para finalmente descomponer el resultado a lo largo del espacio fila de C . Matemáticamente, la proyección del espacio fila de A dentro del espacio fila de B y C se define como sigue:

$$A / \begin{pmatrix} C \\ B \end{pmatrix} = A \begin{pmatrix} C \\ B \end{pmatrix}^\dagger \begin{pmatrix} C \\ B \end{pmatrix}. \quad (4.11)$$

Al descomponer la expresión anterior a lo largo de los espacios fila de B y C se tiene que la proyección oblicua del espacio fila de A a lo largo del espacio fila de B sobre el espacio fila de C es representado por la expresión

$$A /_B C \triangleq A \left[\begin{pmatrix} C \\ B \end{pmatrix}^\dagger \right]_{(:,1:r)} C. \quad (4.12)$$

Esta proyección se puede interpretar geoméricamente en el ambiente de dos dimensiones representado en la Figura 4.3

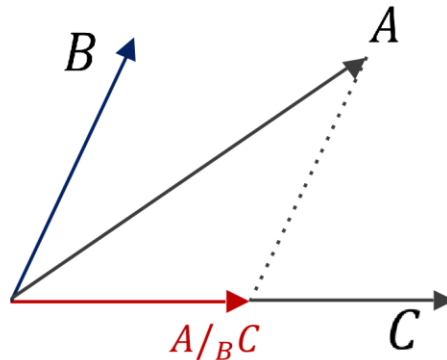


Figura 4.3. Proyección oblicua.

4.1.2. Filtro de Kalman

El **filtro de Kalman** aborda el problema de obtener un estimador del estado x de un sistema en tiempo discreto que está definido por la siguiente ecuación en diferencias:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}, \quad (4.13)$$

$$z_k = Hx_k + v_k. \quad (4.14)$$

La variable w_k representa el ruido en el proceso y v_k representa el ruido en las mediciones, estas se suponen independientes una de la otra, de ruido blanco y con distribución de probabilidad normal:

$$p(w) \sim N(0, Q), \quad (4.15)$$

$$p(v) \sim N(0, R). \quad (4.16)$$

Ahora se define \hat{x}_k^- como la estimación del estado *a priori* en el tiempo k , dado cierto conocimiento del proceso antes del tiempo k y \hat{x}_k como la estimación del estado *a posteriori* en el tiempo k dado z_k . Se pueden definir los errores de estimación del estado *a priori* y *a posteriori* como sigue:

$$e_k^- = x_k - \hat{x}_k^-, \quad (4.17)$$

$$e_k = x_k - \hat{x}_k . \quad (4.18)$$

La covarianza del error de estimación *a priori* es:

$$P_k^- = E[e_k^- e_k^{-T}] \quad (4.19)$$

y la covarianza del error de estimación *a posteriori* es:

$$P_k = E[e_k e_k^T] . \quad (4.20)$$

Se encuentra una ecuación que calcula la estimación del estado *a posteriori* \hat{x}_k como una combinación lineal de la estimación del estado *a priori* x_k^- y la diferencia de la medición actual z_k y la predicción de la medición Hx_k^- :

$$\hat{x}_k = x_k^- + K(z_k - Hx_k^-) . \quad (4.21)$$

La diferencia $(z_k - Hx_k^-)$ se denomina el **residuo** y este representa la discrepancia entre la predicción de la medición Hx_k^- y la medición actual z_k .

La matriz K es la **ganancia** que minimiza la covarianza del error *a posteriori*:

$$K_k = P_k^- H^T (HP_k^- + R)^{-1} . \quad (4.22)$$

El filtro de Kalman estima un proceso usando una estructura parecida a la del control con retroalimentación, ya que el filtro estima el estado del proceso en algún tiempo y entonces obtiene retroalimentación en forma de mediciones. Las ecuaciones que conforman el filtro de Kalman están en dos grupos: ecuaciones de actualización en tiempo y ecuaciones de actualización en las mediciones. Las primeras son responsables de proyectar el estado actual y la covarianza del error de estimación para obtener la estimación *a priori* para el siguiente paso en el tiempo. Las segundas ecuaciones son responsables de la retroalimentación.

Las ecuaciones de actualización en el tiempo también se denominan ecuaciones de *predicción* y a las ecuaciones de actualización en las mediciones se les conoce como ecuaciones de *corrección*.

Las ecuaciones de predicción son:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} , \quad (4.23)$$

$$P_k^- = AP_{k-1}A^T + Q . \quad (4.24)$$

Las ecuaciones de corrección son:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} , \quad (4.25)$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) , \quad (4.24)$$

$$P_k = (I - K_k H)P_k^- . \quad (4.25)$$

La primera tarea durante la actualización de la medición es calcular la ganancia de Kalman K_k ; el siguiente paso es conocer el proceso en el instante actual para obtener z_k para entonces

generar una estimación del estado *a posteriori*. Por último, se obtiene la covariancia del error de estimación [22].

Después de cada actualización en el tiempo y en la medición, el proceso se repite con la estimación previa *a posteriori* usada para predecir la nueva estimación *a priori*.

4.1.3. Matriz de Hankel

Se considera un sistema lineal, discreto, invariante en el tiempo y en reposo para tiempos anteriores a cero, donde las entradas y salidas de datos están organizados de la siguiente manera:

$$u = (u(0), u(1), \dots, u(n-1)), \quad (4.26)$$

$$y = (y(0), y(1), \dots, y(n-1)). \quad (4.27)$$

Con n igual al total de datos, con n suficientemente grande. Para $k > 0$ se obtiene la siguiente expresión:

$$[y(0), y(1), \dots, y(n-1)] = [g_{k-1} \dots g_0] \begin{bmatrix} 0 & \dots & 0 & u(0) & \dots & u(n-k) \\ \vdots & \ddots & \vdots & u(1) & \dots & u(n-k+1) \\ 0 & \ddots & \vdots & \vdots & \dots & \vdots \\ u(0) & u(1) & \dots & u(k-1) & \dots & u(k-1) \end{bmatrix}. \quad (4.28)$$

Suponiendo que la matriz del lado derecho de la expresión formada por las entradas es de rango completo¹, entonces la respuesta al impulso $g_{k-1} \dots g_0$ puede obtenerse resolviendo la ecuación (4.28) mediante el método de mínimos cuadrados. Con esto se puede calcular una expresión mínima para un sistema lineal e invariante en el tiempo, mediante el conjunto de datos de entrada y salida sin la necesidad de utilizar la respuesta al impulso. Los datos de entrada y salida del sistema se pueden organizar de tal manera que se obtiene lo que se conoce como matriz de Hankel [23].

Los datos de entrada y salida generalmente se organizan de la siguiente manera:

$$[u(0), u(1), \dots, u(k+n-2)], \quad (4.29)$$

$$[y(0), y(1), \dots, y(k+n-2)]. \quad (4.30)$$

Para construir las matrices de Hankel, donde k es más grande que el orden del sistema; estas matrices son de gran importancia para los algoritmos de identificación de subespacios. Una **matriz de Hankel** es una matriz cuadrada con todas las diagonales de derecha a izquierda paralelas, estas se construyen fácilmente con los datos de entrada y salida; los subíndices indican los argumentos del primer y el último elemento de la primera columna y el número total de columnas está determinado por la cantidad de datos n . Las matrices de Hankel para la entrada y la salida se definen como sigue:

¹ Una matriz tiene rango completo si todas sus columnas son linealmente independientes.

Entradas,

$$U_{0|k-1} = \begin{bmatrix} u(0) & u(1) & \dots & u(n-1) \\ u(1) & u(2) & \dots & u(n) \\ \vdots & \vdots & \dots & \vdots \\ u(k-1) & u(k) & \dots & u(k+n-2) \end{bmatrix}, \quad (4.31)$$

$$U_{k|2k-1} = \begin{bmatrix} u(k) & u(k+1) & \dots & u(k+n-1) \\ u(k+1) & u(k+2) & \dots & u(k+n) \\ \vdots & \vdots & \dots & \vdots \\ u(2k-1) & u(2k) & \dots & u(2k+n-2) \end{bmatrix}, \quad (4.32)$$

Salidas,

$$Y_{0|k-1} = \begin{bmatrix} y(0) & y(1) & \dots & y(n-1) \\ y(1) & y(2) & \dots & y(n) \\ \vdots & \vdots & \dots & \vdots \\ y(k-1) & y(k) & \dots & y(k+n-2) \end{bmatrix}, \quad (4.33)$$

$$Y_{k|2k-1} = \begin{bmatrix} y(k) & y(k+1) & \dots & y(k+n-1) \\ y(k+1) & y(k+2) & \dots & y(k+n) \\ \vdots & \vdots & \dots & \vdots \\ y(2k-1) & y(2k) & \dots & y(2k+n-2) \end{bmatrix}. \quad (4.34)$$

El índice k determina el número de filas de la matriz de Hankel y n corresponde a los datos y al número de columnas.

Las dimensiones de estas matrices son, $U_{0|k-1} \in \mathbb{R}^{km \times n}$, $U_{k|2k-1} \in \mathbb{R}^{km \times n}$, $Y_{0|k-1} \in \mathbb{R}^{kp \times n}$, $Y_{k|2k-1} \in \mathbb{R}^{kp \times n}$, donde m es el número de entradas y p el número de salidas.

Por definición se tiene que

$$U_{0|k-1} = U_p, \quad (4.35)$$

$$U_{k|2k-1} = U_f, \quad (4.36)$$

$$Y_{0|k-1} = Y_p, \quad (4.37)$$

$$Y_{k|2k-1} = Y_f. \quad (4.38)$$

Los subíndices p y f en las variables denotan el pasado o el futuro de los datos de entrada y salida del sistema.

4.1.4. Descomposición QR y LQ

Dada una matriz $A \in \mathbb{C}^{n \times m}$, con $n \geq m$, existen dos matrices Q y R , ambas dependientes de A de manera que:

$$A = QR, \quad (4.39)$$

tal que se cumple lo siguiente:

- $Q \in \mathbb{C}^{n \times m}$ tiene columnas ortonormales ($Q * Q = I_m$)
- $R \in \mathbb{C}^{m \times m}$ es triangular superior con entradas diagonales no repetidas.

Si además las columnas de A son linealmente independientes, entonces se puede considerar la matriz R con todos sus elementos diagonales reales positivos.

Si $A \in \mathbb{R}^{n \times m}$ entonces tanto Q como R pueden tomarse con elementos reales y $Q^T Q = I_m$.

El cálculo de la descomposición LQ se realiza al tomar la transpuesta de la descomposición QR , que es la que se usa en el algoritmo de identificación.

Si se toma la descomposición LQ de la matriz de datos dada por

$$\begin{bmatrix} U_{0|k-1} \\ Y_{0|k-1} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}, \quad (4.40)$$

donde $L_{11} \in \mathbb{R}^{km \times km}$, $L_{21} \in \mathbb{R}^{kp \times km}$, $L_{22} \in \mathbb{R}^{kp \times kp}$, además L_{11} y L_{22} son matrices triangulares inferiores y $Q_1 \in \mathbb{R}^{n \times km}$, $Q_2 \in \mathbb{R}^{n \times kp}$ son ortogonales [24].

4.1.5. Descomposición en valores singulares (SVD)

La SVD es un método para transformar variables correlacionadas en otro conjunto que no están, pero que expresan la relación entre el conjunto original de los datos. Además, es un método que sirve para identificar y ordenar las dimensiones en donde los datos muestran mayor variación; también puede ser considerada para aplicarla en la reducción de datos [25].

Sea A_{SVD} una matriz rectangular que se puede descomponer en tres matrices, una matriz ortogonal U , una matriz diagonal S y la transpuesta de una matriz ortogonal V , tal que

$$A_{SVD} = USV^T, \quad (4.41)$$

donde

$$U^T U = I, \quad (4.42)$$

$$V^T V = I. \quad (4.43)$$

Las columnas de U son los eigenvectores ortonormales de $A_{SVD} A_{SVD}^T$, las columnas de V son eigenvectores ortonormales de $A_{SVD}^T A_{SVD}$ y S es una matriz diagonal que contiene las raíces de los eigenvalores de U o V en orden descendente.

4.2. Identificación estocástica

La identificación estocástica consiste en calcular las matrices del sistema cuando no se tiene ninguna señal de entrada, esto es cuando solo se tienen los datos de salida. El diagrama de bloques de un sistema estocástico se muestra en la Figura 4.4

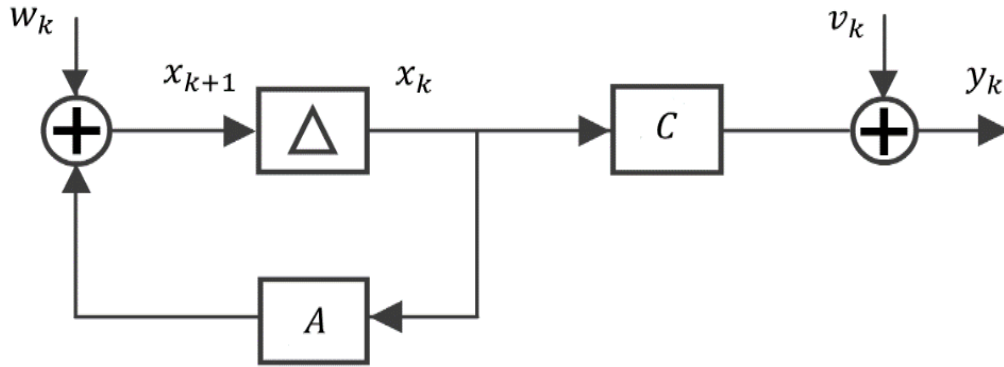


Figura 4.4. Diagrama de bloques de un sistema estocástico.

Dadas p muestras de la señal de salida y_k generadas por un sistema estocástico de orden n , se tiene el sistema dinámico en variables de estado dado como:

$$x_{k+1} = Ax_k + w_k, \quad (4.44)$$

$$y_k = Cx_k + v_k, \quad (4.45)$$

con las secuencias de ruido blanco w_k y v_k cuya media es cero y su matriz de covariancia dada por:

$$E \left[\begin{pmatrix} w_p \\ v_p \end{pmatrix} \begin{pmatrix} w_q^T & v_q^T \end{pmatrix} \right] = \begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} \delta_{pq}. \quad (4.46)$$

Con esta información es posible determinar el valor de las matrices A , C , Q , S y R .

A continuación se describe el procedimiento numérico para la implementación de algoritmos de identificación basados en subespacios usando todas las herramientas presentadas en esta sección.

El primer paso consiste en organizar los datos experimentales en matrices de Hankel, en este caso al no tener datos de entrada, solo se ocupan las expresiones (4.33) y (4.34).

En seguida se procede a calcular la descomposición QR de las matrices de Hankel previamente obtenidas, para el caso estocástico se representa de la siguiente manera:

$$Y = \frac{1}{\sqrt{j}} (Y_{0|2k-1}), \quad (4.47)$$

$$Y = RQ^T, \quad (4.48)$$

donde $Q^T \in \mathbb{R}^{2(m+l)i \times j}$ es una matriz ortonormal y $R \in \mathbb{R}^{2(m+l)i \times 2(m+l)i}$ es una matriz triangular inferior. Esta descomposición tiene varias ventajas, la más importante es que todas las operaciones geométricas utilizadas se pueden expresar de una manera sencilla en términos de la descomposición QR .

Esta descomposición puede ser fragmentada de la siguiente forma:

$$\frac{1}{\sqrt{j}}(Y_{0|2k-1}) = \frac{1}{\sqrt{j}} \begin{pmatrix} Y_p \\ Y_f \end{pmatrix} = \frac{1}{\sqrt{j}} \begin{pmatrix} Y_p^+ \\ Y_f^- \end{pmatrix}, \quad (4.49)$$

$$= RQ^T = \frac{1}{\sqrt{j}} \begin{matrix} li & & & j \\ & l & & \\ & & l(i-1) & \end{matrix} \begin{pmatrix} Y_{0|k-1} \\ Y_{k|k} \\ Y_{k+1|2k-1} \end{pmatrix}, \quad (4.50)$$

$$= \begin{matrix} li & l & l(i-1) & j \\ & & & \end{matrix} \begin{pmatrix} R_{11} & 0 & 0 \\ R_{21} & R_{22} & 0 \\ R_{31} & R_{32} & R_{33} \end{pmatrix} \begin{pmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \end{pmatrix}, \quad (4.51)$$

De esta descomposición se pueden obtener fácilmente los datos de salida pasados y futuros para su uso en diversas operaciones

Una de las operaciones que se pueden realizar son las proyecciones ortogonales; la proyección ortogonal de las salidas futuras Y_f con las salidas pasadas da como resultado la secuencia de estado \hat{X}_i .

Las propiedades principales de un proceso estocástico lineal e invariante en el tiempo son las siguientes:

- Si se asume que el proceso estocástico es estacionario entonces $E[x_k^s] = 0$, donde el operador E denota el promedio de un número infinito de experimentos, por lo tanto se define:

$$E[x_k^s (x_k^s)^T] = \Sigma^s, \quad (4.52)$$

donde la matriz de covarianza de los estados Σ^s es independiente del tiempo k , lo que implica que la matriz A del sistema es una matriz estable².

- Por lo tanto se define la ecuación de Lyapunov de la matriz de covarianza de los estados como:

$$\Sigma^s = E[(Ax_k^s + w_k)(Ax_k^s + w_k)^T] = A\Sigma^s A^T + Q, \quad (4.53)$$

- De la misma forma se pueden definir las siguientes matrices de covarianza:

$$\Lambda_0 = C\Sigma^s C^T + R, \quad (4.54)$$

² Una matriz es estable si todos sus eigenvalores pertenecen al semiplano derecho del plano complejo.

$$G = A\Sigma^s C^T + S, \quad (4.55)$$

- En los algoritmos de identificación de subespacios estocásticos, el filtro de Kalman es de gran importancia, si se aplica este filtro a las ecuaciones (4.44) y (4.45) se obtienen las expresiones:

$$x_{k+1} = Ax_k + Ke_k, \quad (4.56)$$

$$y_k = Cx_k + e_k, \quad (4.57)$$

$$E[e_k(e_k)^T] = (\Lambda_0 - CPC^T), \quad (4.58)$$

donde K es la ganancia de Kalman y está dada por la ecuación:

$$K = (G - APC^T)(\Lambda_0 - CPC^T)^{-1}, \quad (4.59)$$

P es la matriz de covarianza, la cual puede ser determinada mediante la solución de la ecuación de Riccati:

$$P = APA^T + (G - APC^T)(\Lambda_0 - CPC^T)^{-1}(G - APC^T)^T. \quad (4.60)$$

- Para resolver esta ecuación se considera el problema de eigenvalores generalizado [26].

El procedimiento para realizar el algoritmo de identificación estocástica basado en el trabajo de Overschee [27] en el lenguaje de programación de MATLAB, se muestra a continuación:

I) Se calculan las matrices de Hankel de los datos de salida con las expresiones (4.33) y (4.34)

II) Se obtiene el factor R de la matriz de Hankel mediante la descomposición QR :

$$R = qr(Y), \quad (4.61)$$

se aplica la transpuesta a la matriz obtenida.

III) Del factor R obtenido se extraen las matrices: Y_f, Y_p, Y_f^-, Y_p^+ .

IV) Se calculan las siguientes proyecciones:

$$O_i = Y_f/Y_p, \quad (4.62)$$

$$O_{i-1} = Y_f^-/Y_p^+. \quad (4.63)$$

V) Se calcula la SVD de la proyección O_i :

$$O_i = USV^T. \quad (4.64)$$

VI) Se determinan Γ_i y Γ_{i-1} :

$$\Gamma_i = U_1 S_1^{1/2}, \quad (4.65)$$

$$\Gamma_{i-1} = \underline{\Gamma}_i. \quad (4.66)$$

VII) Se determinan los estados \hat{X}_i y \hat{X}_{i+1} :

$$\hat{X}_i = \Gamma_i^\dagger O_i, \quad (4.67)$$

$$\hat{X}_{i+1} = \Gamma_{i-1}^\dagger O_{i-1}. \quad (4.68)$$

VIII) Se resuelve la ecuación formada por los elementos anteriores y se obtienen las matrices A y C :

$$\begin{pmatrix} \hat{X}_{i+1} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A \\ C \end{pmatrix} \hat{X}_i + \begin{pmatrix} \rho_w \\ \rho_v \end{pmatrix}. \quad (4.69)$$

IX) Se determinan las matrices Q , S y R

$$\begin{pmatrix} Q & S \\ S^T & R \end{pmatrix} = Ej \left[\begin{pmatrix} \rho_w \\ \rho_v \end{pmatrix} \begin{pmatrix} \rho_w^T & \rho_v^T \end{pmatrix} \right]. \quad (4.70)$$

X) Se determinan Σ^s , G y Λ_0 utilizando las ecuaciones (4.53) (4.54) y (4.55) respectivamente.

XI) Se calcula P y la ganancia de Kalman K resolviendo la ecuación de Riccati:

$$P = APA^T + (G - APC^T)(\Lambda_0 - CPC^T)^{-1}(G - APC^T)^T, \quad (4.71)$$

$$K = (G - APC^T)(\Lambda_0 - CPC^T)^{-1}. \quad (4.72)$$

XII) Finalmente se obtiene el modelo estocástico dado por la ecuación:

$$x_{k+1} = Ax_k + Ke_k, \quad (4.73)$$

$$y_k = Cx_k + e_k, \quad (4.74)$$

$$E[e_k(e_k)^T] = R. \quad (4.75)$$

4.3. Identificación determinista

El problema a resolver con la identificación determinista es el de encontrar las matrices de un sistema que no tiene influencia de ruido ($v_k = 0$ y $w_k = 0$). El diagrama a bloques de un sistema determinista se muestra en la Figura 4.5

Dadas m mediciones de la señal de entrada u_k y de salida y_k , generadas por el siguiente sistema desconocido:

$$x_{k+1} = Ax_k + Bu_k, \quad (4.76)$$

$$y_k = Cx_k + Du_k. \quad (4.77)$$

se deben determinar las matrices A , B , C y D .

$$Z_{i-1} = Y_f^- / \begin{pmatrix} W_p^+ \\ U_f^- \end{pmatrix}. \quad (4.87)$$

V) Se calcula la *SVD* de la proyección oblicua O_i :

$$O_i = USV^T. \quad (4.88)$$

VI) Se determina Γ_i y Γ_{i-1} :

$$\Gamma_i = U_1 S_1^{1/2}, \quad (4.89)$$

$$\Gamma_{i-1} = \underline{\Gamma}_i, \quad (4.90)$$

donde $\underline{\Gamma}_i$ representa a Γ_i sin la última columna.

VII) Se resuelve la siguiente ecuación y se obtienen las matrices A y C :

$$\begin{pmatrix} \Gamma_{i-1}^\dagger Z_{i+1} \\ Y_{i|i} \end{pmatrix} = \begin{pmatrix} A \\ C \end{pmatrix} \Gamma_i^\dagger Z_i + K U_f + \begin{pmatrix} \rho_w \\ \rho_v \end{pmatrix}. \quad (4.91)$$

VIII) Se vuelven a calcular Γ_i y Γ_{i-1} a partir de las matrices obtenidas A y C .

IX) Se obtienen las matrices B y D a partir de

$$B, D = \arg \min_{B, D} \left\| \begin{pmatrix} \Gamma_{i-1}^\dagger Z_{i+1} \\ Y_{i|i} \end{pmatrix} - \begin{pmatrix} A \\ C \end{pmatrix} \Gamma_i^\dagger Z_i - \mathcal{K}(B, D) U_f \right\|_F^2 \quad (4.92)$$

donde

$$\mathcal{P} = \begin{pmatrix} \Gamma_{i-1}^\dagger Z_{i+1} \\ Y_{i|i} \end{pmatrix} - \begin{pmatrix} A \\ C \end{pmatrix} \Gamma_i^\dagger Z_i, \quad (4.93)$$

$$Q = U_f. \quad (4.94)$$

X) Se obtiene el siguiente modelo determinista

$$x_{k+1} = A x_k + B u_k, \quad (4.95)$$

$$y_k = C x_k + D u_k. \quad (4.96)$$

4.4. Serie de Volterra y modelo bilineal

Las series de Volterra se utilizan frecuentemente para representar la relación entrada-salida de sistemas dinámicos no lineales, en problemas de procesamiento de señales y en identificación de sistemas. En este trabajo se hace uso de este concepto para realizar la identificación de un kernel cuadrático a través de un sistema bilineal.

Una serie de Volterra finita se expresa mediante la siguiente ecuación:

$$y_t = \sum_{d=1}^D \sum_{\tau_1=1}^t \sum_{\tau_2}^{\tau_1} \dots \sum_{\tau_d}^{\tau_{d-1}} (t, \tau_1, \dots, \tau_d) u_{\tau_1} u_{\tau_2} \dots u_{\tau_d}, \quad (4.97)$$

donde la entrada es u_t y la salida es y_t , K_d son los kernels; esta serie se puede representar por un sistema bilineal “nilpotente” si los kernels se definen como una suma de productos finitos de funciones de una variable donde el grado no lineal es controlable [29].

Utilizando este concepto se construye un sistema bilineal para realizar la identificación, la cual consta de los bloques del siguiente diagrama:

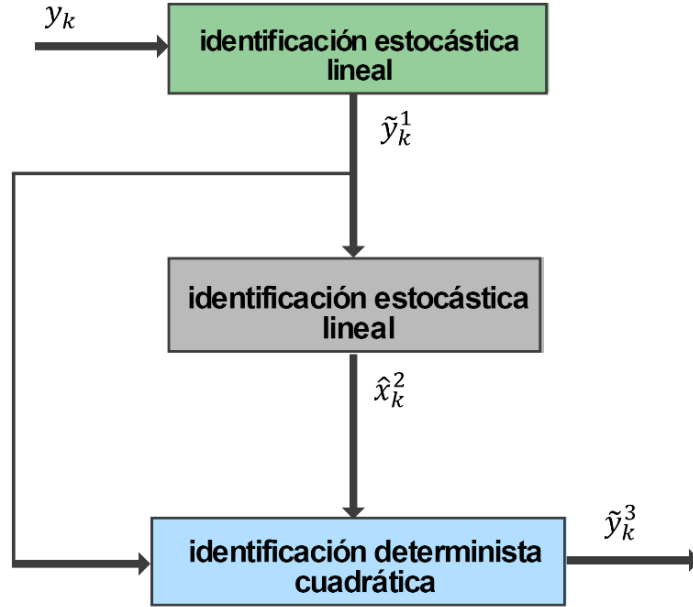


Figura 4.6. Diagrama del algoritmo cuadrático.

El sistema de ecuaciones se forma de la siguiente manera:

$$\hat{x}_{k+1}^{1,1} = A^{1,1}\hat{x}_k^{1,1} + K^{1,1}\tilde{y}_k \quad (4.98)$$

$$\hat{y}_k^{1,1} = C^{1,1}\hat{x}_k^{1,1} \quad (4.99)$$

$$\hat{x}_{k+1}^{1,2} = A^{1,2}\hat{x}_k^{1,2} + K^{1,2}\tilde{y}_k^{1,1} \quad (4.100)$$

$$\hat{y}_k^{1,2} = C^{1,2}\hat{x}_k^{1,2} \quad (4.101)$$

$$\hat{x}_{k+1}^{2,2} = A^{2,2}\hat{x}_k^{2,2} + D^2\tilde{y}_{k+1}^{1,2} \otimes \hat{x}_{k+1}^{1,2} \quad (4.102)$$

$$\hat{y}_k^{2,2} = C^{2,2}\hat{x}_k^{2,2} \quad (4.103)$$

donde el primer superíndice indica el número de capa homogénea (afín en el estado), y el segundo indica el grado no lineal del kernel estimado.

Finalmente, la estimación estocástica-determinista no lineal resulta de la suma de los kernels lineal y cuadrático estimados:

$$\hat{y}_k = \hat{y}_k^{1,1} + \hat{y}_k^{2,2} . \quad (4.104)$$

En este capítulo se explicaron las herramientas más utilizadas para implementar un algoritmo de identificación de subespacios. Además se describieron los algoritmos de identificación estocástica y determinista que en conjunto se utilizan para identificar sistemas cuadráticos.

CAPÍTULO 5

Análisis caótico

La dinámica de la marcha humana ha sido estudiada para poder conocer su comportamiento caótico mediante métodos de análisis no lineales [30]. Este análisis es de gran importancia, ya que con esto se pueden detectar patologías comunes que se presentan en la marcha humana.

Una de las formas más comunes para analizar el comportamiento caótico de cualquier sistema es mediante el cálculo de los exponentes de Lyapunov. Una versión mejorada del método desarrollado por K. Briggs para calcular estos exponentes se presenta en este trabajo para analizar el comportamiento caótico de la marcha humana [31], cuyas señales son obtenidas por el modulo desarrollado en el presente trabajo y presentado en el capítulo 3.

5.1. El exponente de Lyapunov

Para calcular los exponentes de Lyapunov [32], se consideran dos puntos dentro del plano fase: X y X_f , tales que:

$$\frac{\|X_f\|}{\|X\|} = e^{\lambda_i t}, \quad i = 1, 2, \dots, n \quad (5.1)$$

De esta manera, λ_i se obtiene de la expresión:

$$\lambda_i = \lim_{t \rightarrow \infty} \left(\frac{1}{t} \log \frac{\|X_f\|}{\|X\|} \right). \quad (5.2)$$

Si cualquiera de los exponentes obtenidos es positivo, entonces el sistema es caótico. Esto significa que cualquier par de puntos cercanos dentro del estado inicial se separan abruptamente. Además, el sistema es sensible a las condiciones iniciales, el cual es uno de los aspectos principales de un sistema caótico [30].

5.2. Algoritmo para calcular exponentes de Lyapunov

El algoritmo que se programa en MATLAB para calcular los exponentes de Lyapunov se describe a continuación:

- I) Se definen las siguientes constantes:
 - m : la dimensión embebida
 - n : número de muestras
 - del : retraso en el tiempo

- II) Se definen los vectores $X \in \mathbb{R}^{n \times m}$ y $X_f \in \mathbb{R}^{n \times m}$ mediante los datos experimentales:

$$X(i, b) = y(i + (b - 1) * del) , \quad (5.3)$$

$$X_f(i, b) = y(i + td + (b - 1) * del) , \quad (5.4)$$

donde td es el periodo del ciclo, además $i = 1, \dots, n$ y $b = 1, \dots, m$.

- III) Los vectores de diferencia $dX \in \mathbb{R}^{n \times m-1}$ y $dX_f \in \mathbb{R}^{n \times m-1}$ se definen como:

$$dX(:, bd) = X(:, bd + 1) - X(:, 1) , \quad (5.5)$$

$$dX_f(:, bd) = X_f(:, bd + 1) - X_f(:, 1) , \quad (5.6)$$

donde $bd = 1, \dots, m - 1$.

- IV) Se calcula T_i :

$$T_i = dX_f \cdot pinv(dX) \quad (5.7)$$

- V) T_i se factoriza mediante la descomposición QR , donde Q resulta en una matriz ortogonal y R en una matriz superior triangular con elementos no negativos en la diagonal:

$$T_i = QR \quad (5.8)$$

- VI) Se calculan los exponentes de Lyapunov mediante la siguiente expresión:

$$Lyap = \log(diag(R)) \quad (5.9)$$

- VII) Finalmente se muestran en pantalla los exponentes obtenidos.

En este capítulo se describió un método para calcular los exponentes de Lyapunov, el cual es utilizado para determinar el comportamiento caótico de un sistema dinámico. Además se explicó el algoritmo que se implementa en MATLAB para el cálculo numérico del exponente de Lyapunov.

CAPÍTULO 6

Resultados

En esta sección se presentan los resultados experimentales que se obtuvieron con los datos del dispositivo de medición del desplazamiento del centro de masa (CoM), así como la identificación de un sistema para la marcha humana y su análisis caótico.

Mediante el dispositivo de medición se obtuvieron diferentes patrones de marcha para su estudio, estos son:

- i) caminata normal.
- ii) tropiezo mientras se camina.
- iii) desbalance cuando se camina.

Estas mediciones se realizaron en tres personas diferentes en una superficie plana y sin ningún tipo de irregularidad que pudiera hacer tropezar a la persona.

i) Caminata normal, datos persona 1.

Para la **persona 1** se tiene la gráfica de la Figura 6.1, para el desplazamiento del centro de masa en el **eje x** y para el desplazamiento del centro de masa en el **eje y** .

Se puede observar la forma como se desplaza el centro de masa cuando se camina de forma normal a lo largo de una trayectoria recta.

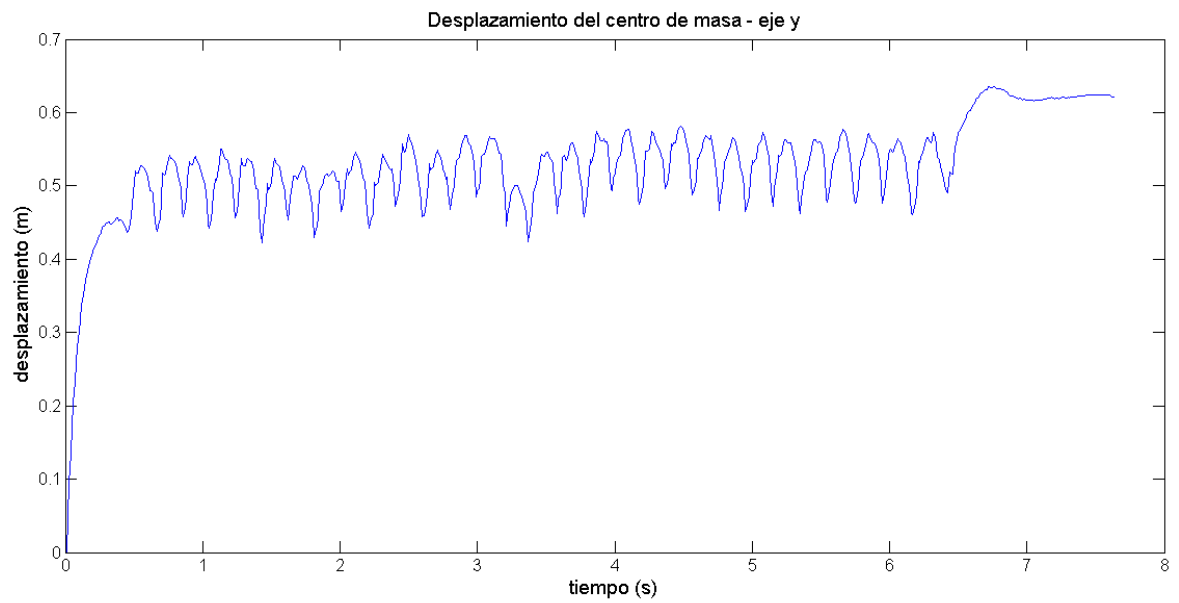
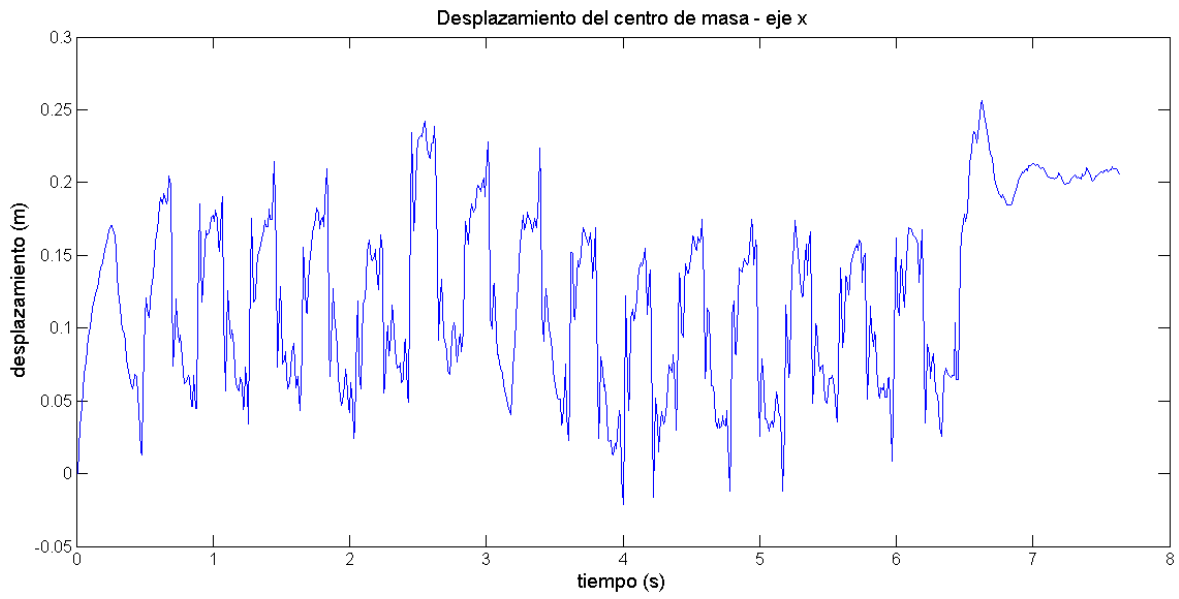


Figura 6.1. Desplazamiento del centro de masa en el eje x y en el eje y de la persona 1.

ii) Tropiezo en la marcha, datos persona 1.

Para el caso del tropiezo en la marcha se tiene la gráfica de la Figura 6.2 para los datos del eje x y el mismo caso visto desde la perspectiva del desplazamiento del centro de masa en eje y .

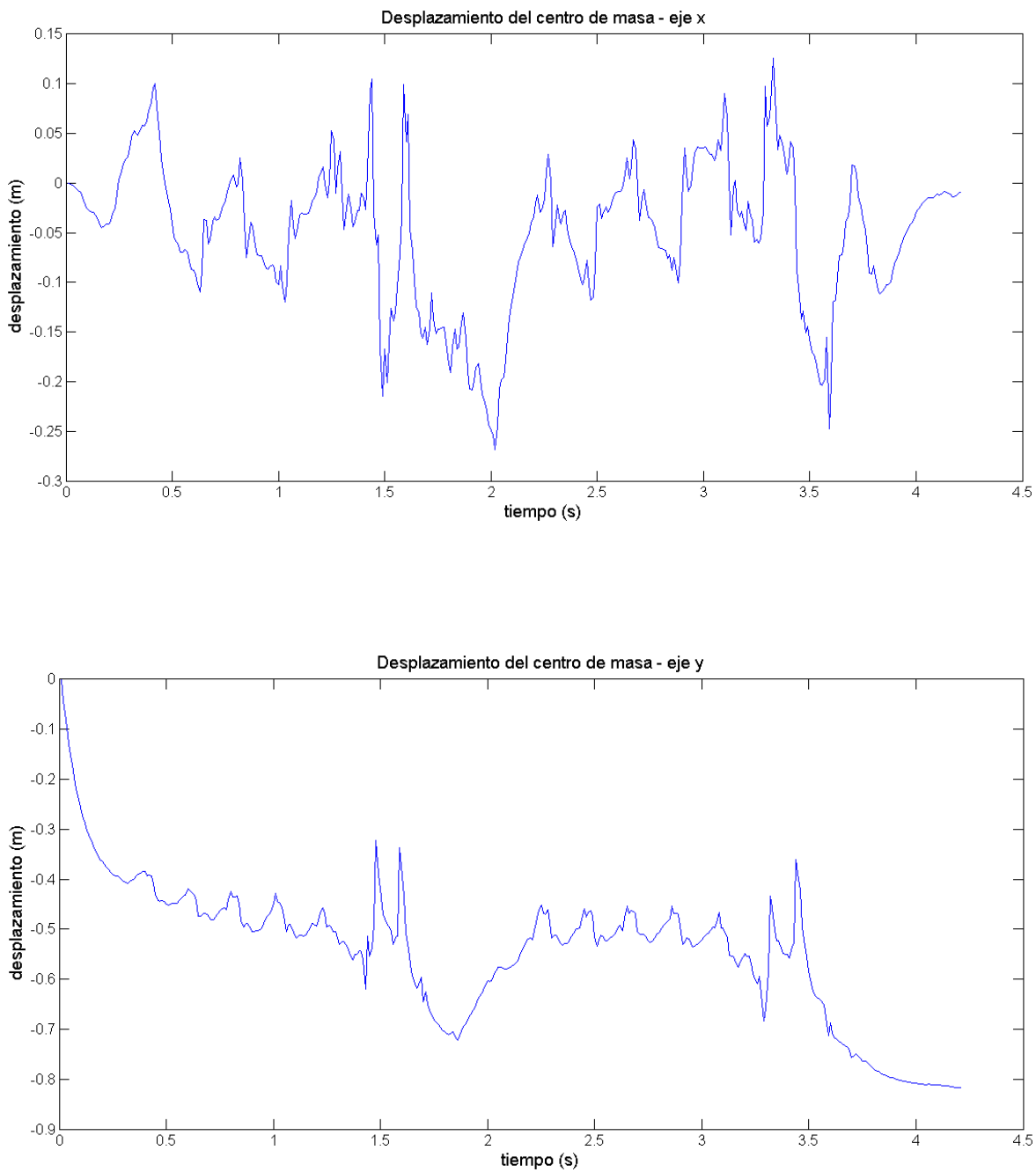


Figura 6.2. Desplazamiento del centro de masa en el eje x y en el eje y de la persona 1 cuando se tiene un tropiezo.

iii) Desequilibrio en la marcha, datos persona 1

Para el desequilibrio en la marcha se tiene la gráfica de la Figura 6.3 para el desplazamiento del centro de masa en el **eje x** y para el caso cuando el desplazamiento del centro de masa ocurre en el **eje y**.

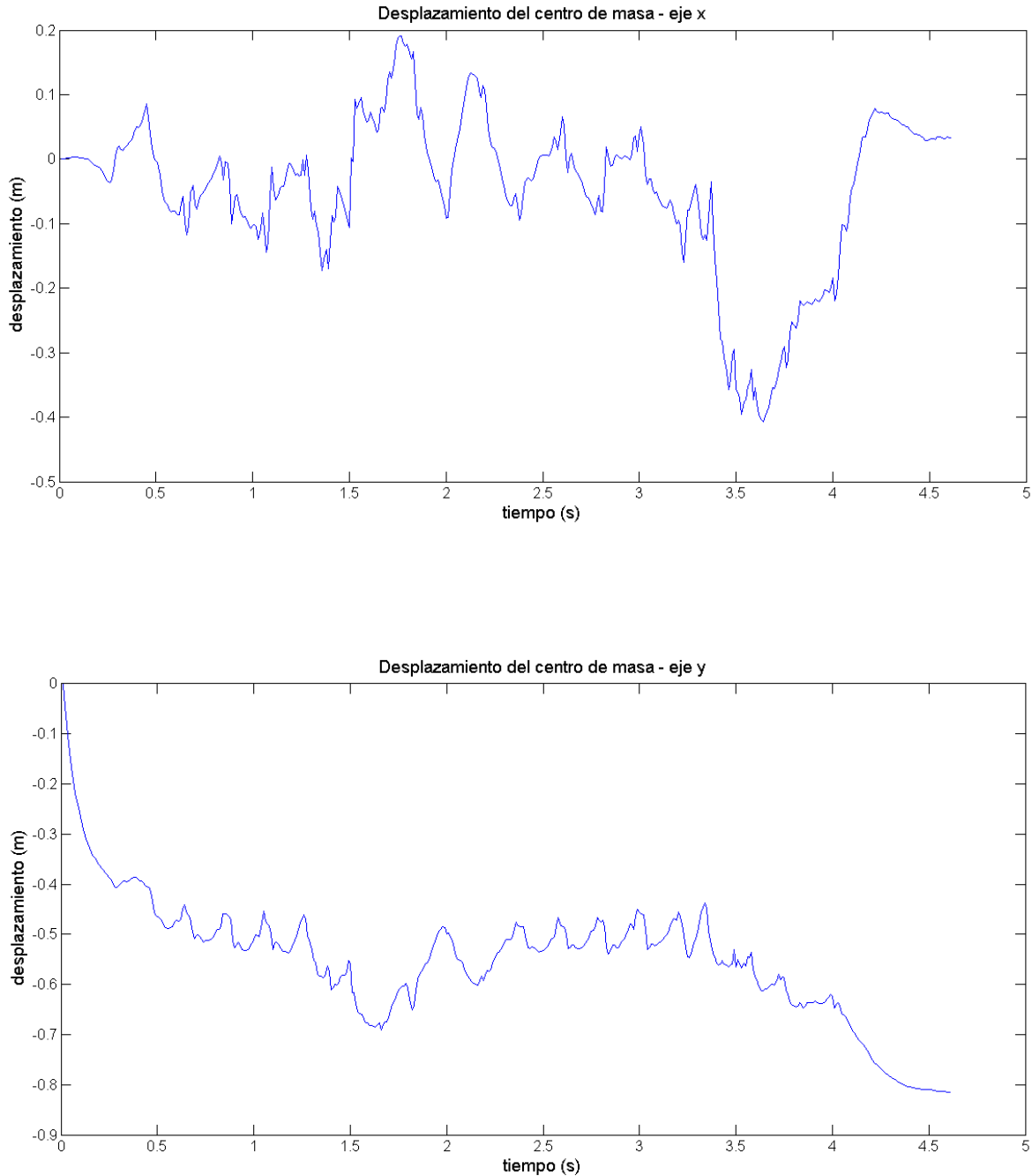


Figura 6.3. Desplazamiento del centro de masa en el eje y de la persona 1 cuando se tiene un desbalance.

i) Caminata normal, datos persona 2

Para la **persona 2** el desplazamiento del centro de masa está representado por la gráfica de la Figura 6.4, en el **eje x** y en el caso en el cual el desplazamiento del centro de masa se da en el **eje y**.

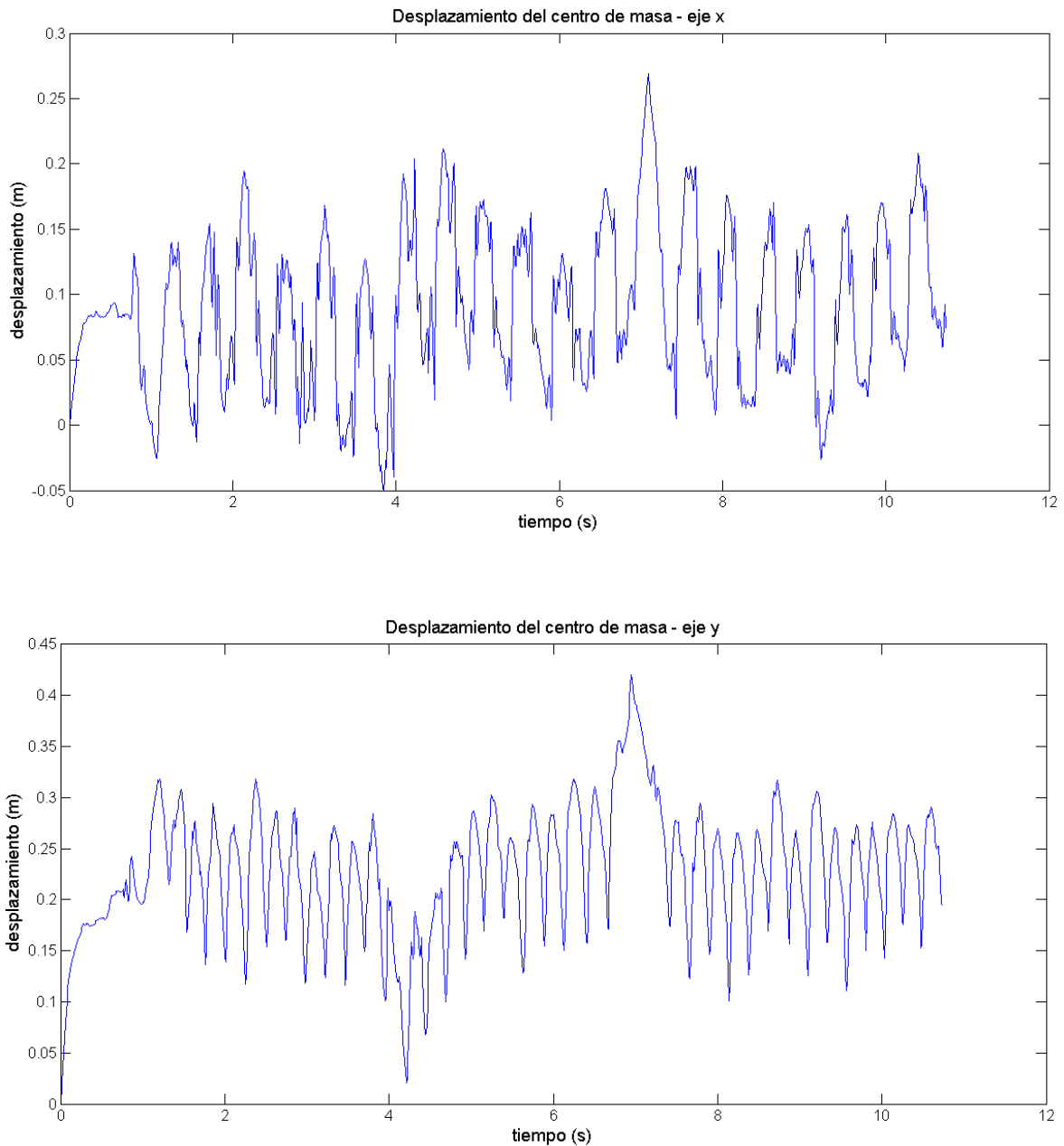


Figura 6.4. Desplazamiento del centro de masa en el eje x y en el eje y de la persona 2.

ii) Tropiezo en la marcha, datos persona 2.

Para el tropiezo en la marcha se tiene la gráfica de la Figura 6.5, para los datos en el **eje x** y con el mismo caso para los datos en el **eje y**.

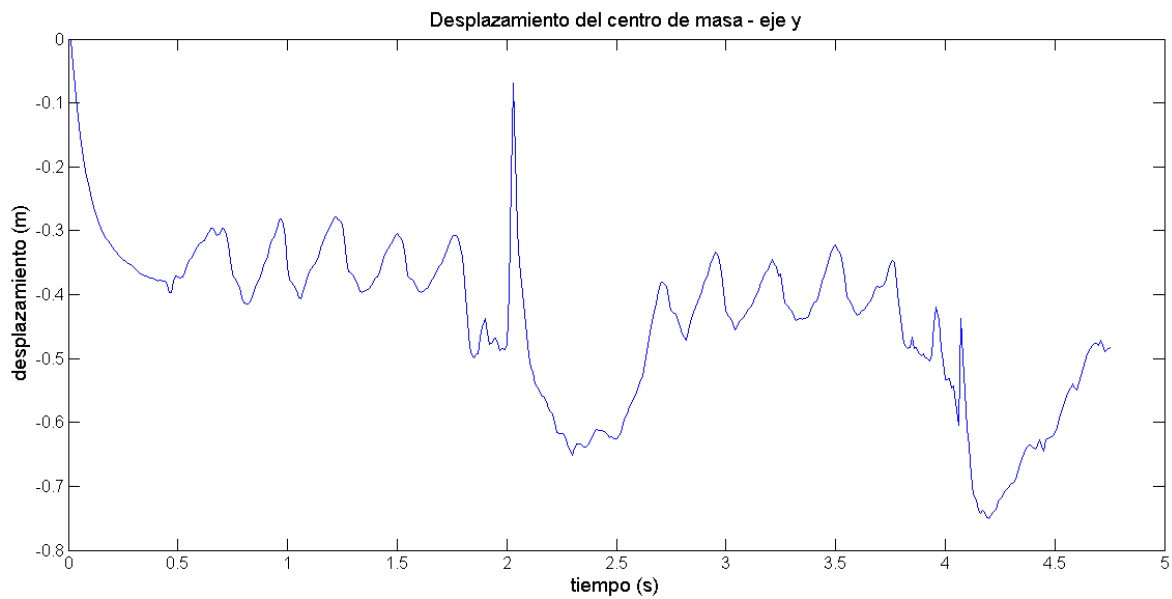
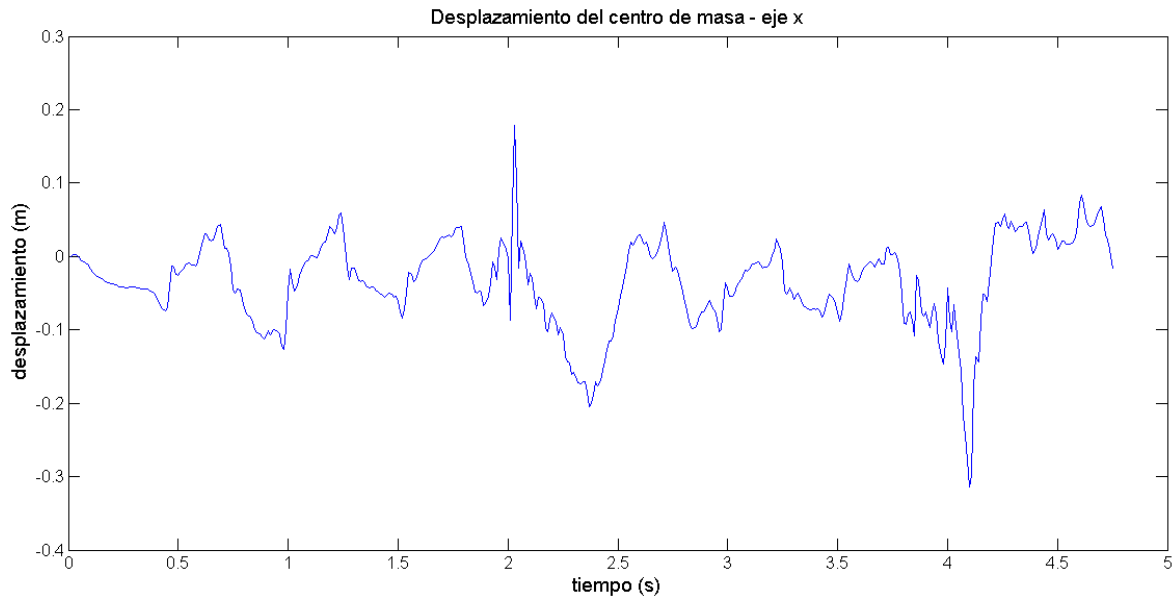


Figura 6.5. Desplazamiento del centro de masa en el eje y de la persona 2 cuando se tiene un tropiezo.

iii) Desequilibrio en la marcha, datos persona 2

Para el desequilibrio en la marcha se tiene la gráfica de la Figura 6.6 para el desplazamiento en el **eje x** y para el mismo caso con los datos obtenidos del **eje y**.

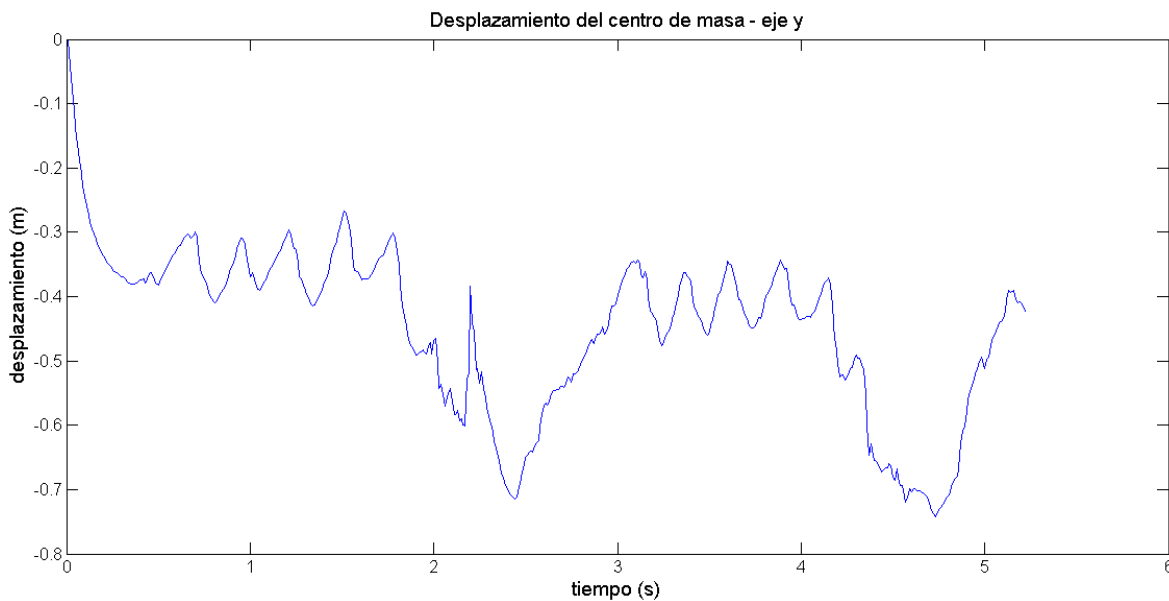
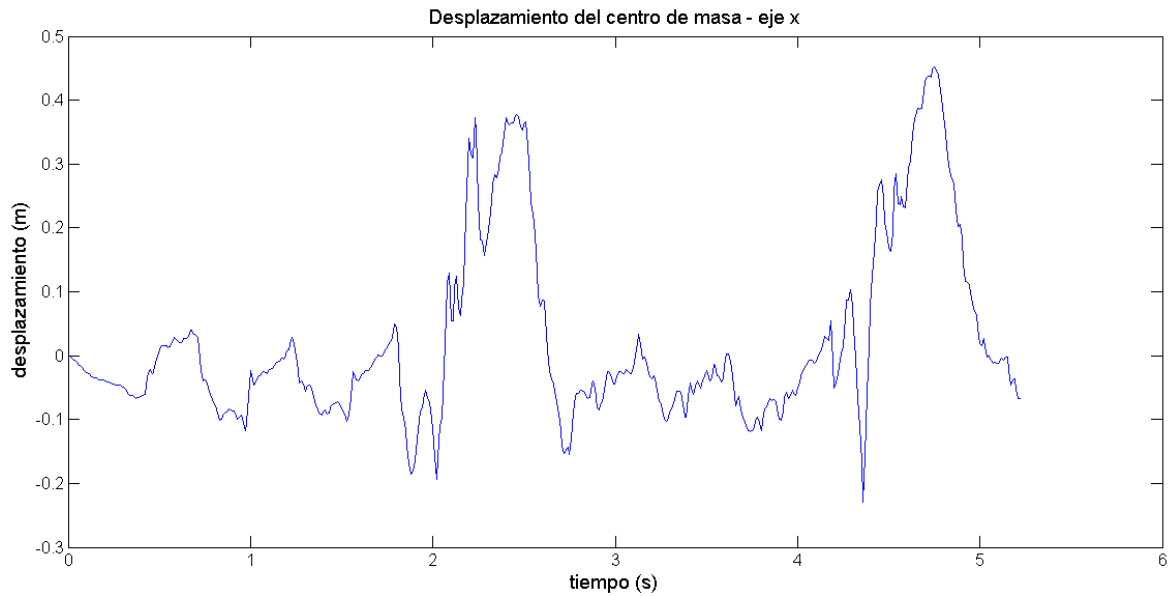


Figura 6.6. Desplazamiento del centro de masa en el eje x y en el eje y de la persona 2 cuando se tiene un desequilibrio.

i) Caminata normal, datos persona 3

Para la **persona 3** se tiene la gráfica de la Figura 6.7 para los datos a lo largo del **eje x** y para la caminata de la misma persona, con los datos del **eje y**.

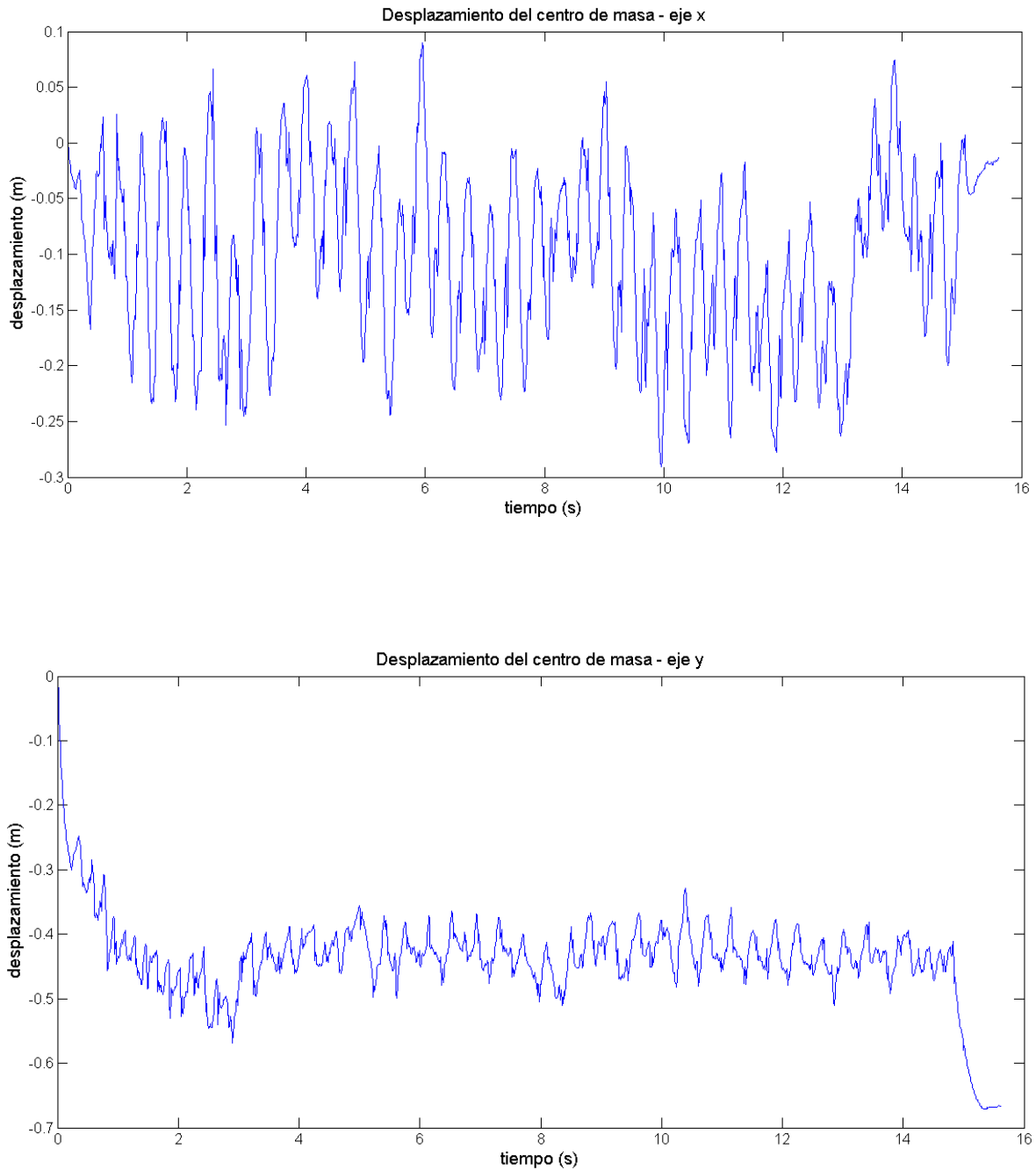


Figura 6.7. Desplazamiento del centro de masa en el eje y de la persona 3.

ii) Tropiezo en la marcha, datos persona 3

Para el tropiezo en la marcha se tiene la gráfica de la Figura 6.8 con los datos obtenidos del eje x y para el mismo caso con los datos del eje y .

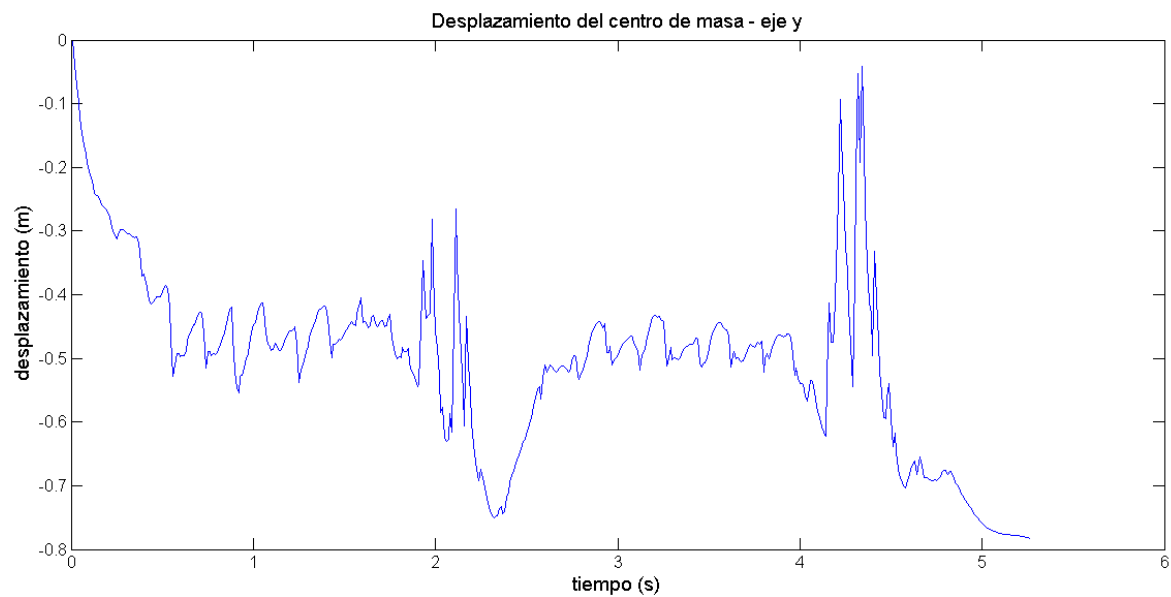
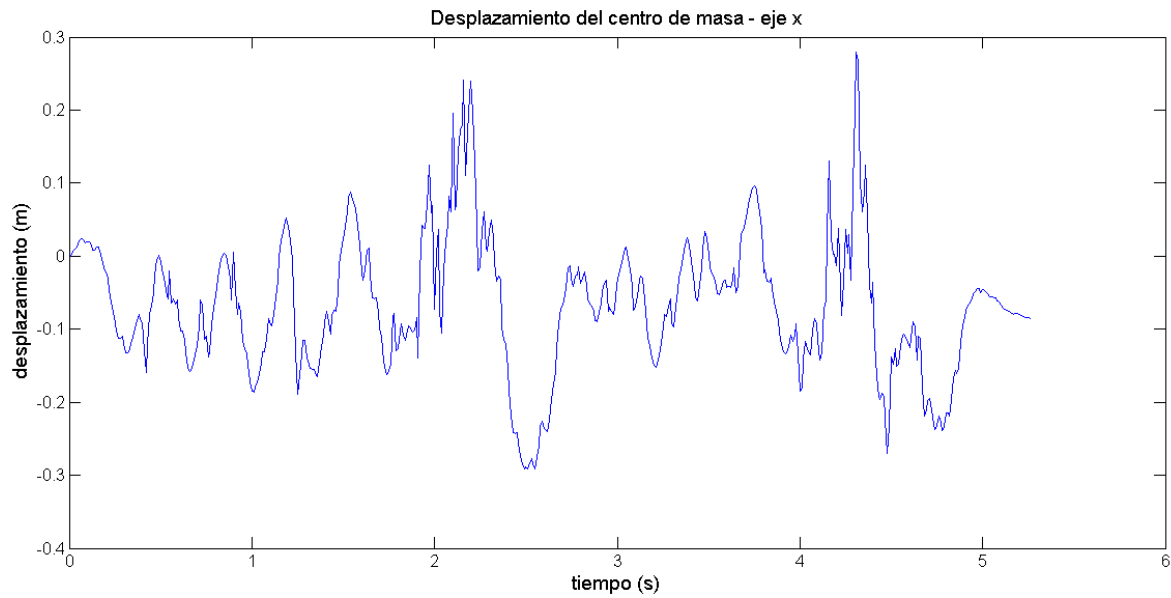


Figura 6.8. Desplazamiento del centro de masa en el eje y de la persona 3 cuando se tiene un tropiezo.

iii) Desequilibrio en la marcha, datos persona 3

Para el desequilibrio en la marcha se obtiene la gráfica de la Figura 6.9 para los datos del **eje x** y para la misma acción realizada pero con los datos del **eje y**.

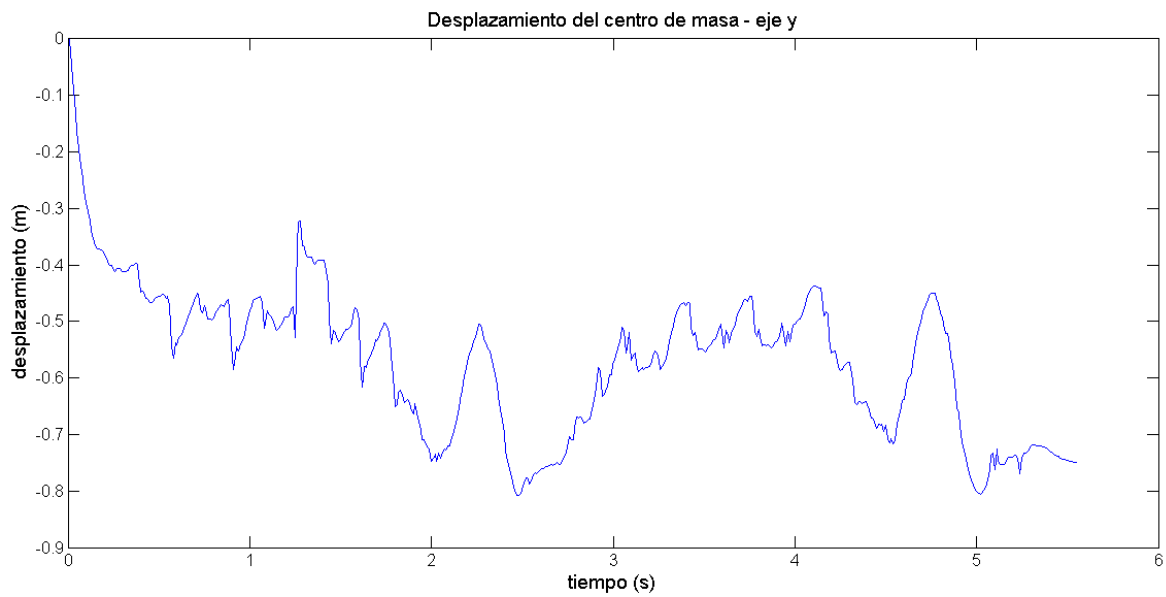
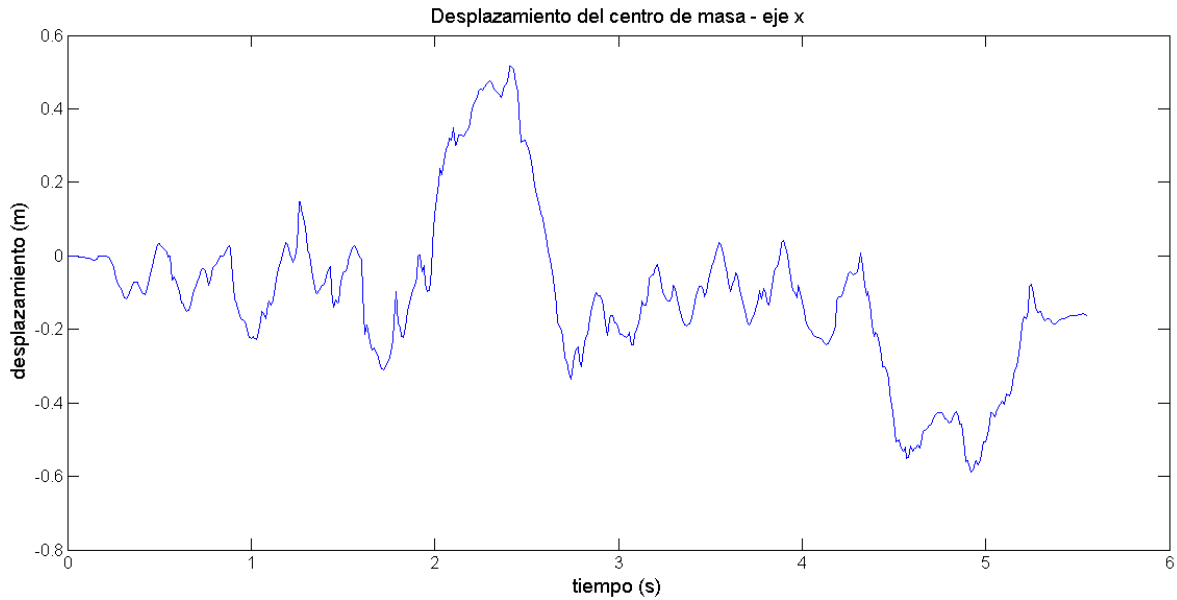


Figura 6.9. Desplazamiento del centro de masa en el eje y de la persona 3 cuando se tiene un desequilibrio.

Con estos datos se pueden realizar varios análisis, para por ejemplo identificar características que ayuden a detectar cuando se esté cayendo una persona y poder usar esta información en algún dispositivo que ayude a prevenir caídas en el futuro.

Las aplicaciones de interés en este trabajo son dos:

- Utilizar los datos de la marcha normal de una persona para realizar la identificación de subespacios y con esto encontrar un modelo matemático que describa adecuadamente este sistema,
- Utilizar los datos de la marcha normal, del caso cuando se tiene un tropiezo y del caso cuando hay un desbalance mientras se está caminando, para poder analizar su comportamiento caótico.

6.1. Identificación de subespacios de la marcha humana

Utilizando el algoritmo descrito en el capítulo 4, se obtuvo un modelo en variables de estado del proceso de la marcha con los datos experimentales filtrados.

Una de las ventajas mencionadas de la identificación de subespacios a diferencia de otros métodos es la posibilidad de identificar sistemas con varias señales de entrada y salida (MIMO), la siguiente prueba consiste en obtener un modelo en el espacio de estados para los datos del movimiento del centro de masa de la **persona 1** en el **eje x** y en el **eje y** simultáneamente.

Utilizando el mismo procedimiento para la mitad de los datos de ambos ejes se obtiene el modelo en espacio de estados que se presenta a continuación con los siguientes parámetros: $n = 4, i = 8$

$$x_{k+1} = \begin{bmatrix} 1.0001 & -0.0050 & 0.0039 & -0.0486 \\ 0.0003 & 0.9593 & -0.2428 & -0.0034 \\ 0.0005 & 0.0647 & 0.9683 & -0.0204 \\ 0.0010 & 0.0164 & -0.0194 & 1.0788 \end{bmatrix} x_k + \begin{bmatrix} -0.4510 & -5.5895 \\ -4.5402 & 3.6318 \\ -0.9056 & -1.6258 \\ 0.5443 & 14.3645 \end{bmatrix} e_k,$$

$$y_k = \begin{bmatrix} -0.0982 & -0.1205 & -0.1126 & -0.0194 \\ -0.4035 & 0.0220 & 0.0283 & -0.0655 \end{bmatrix} x_k,$$

La segunda mitad de los datos se utiliza para validar el algoritmo, introduciéndolos al modelo generado y obtener la señal de salida que se espera. Los datos se dividen en 50% para la identificación y 50% para la validación, ya que al obtener un modelo a partir de las señales de entrada, se requiere comprobar que el modelo obtenido es capaz de reproducir el comportamiento de las entradas utilizadas. En la gráfica de la figura 6.10 se muestra la comparación entre la señal original y la identificada

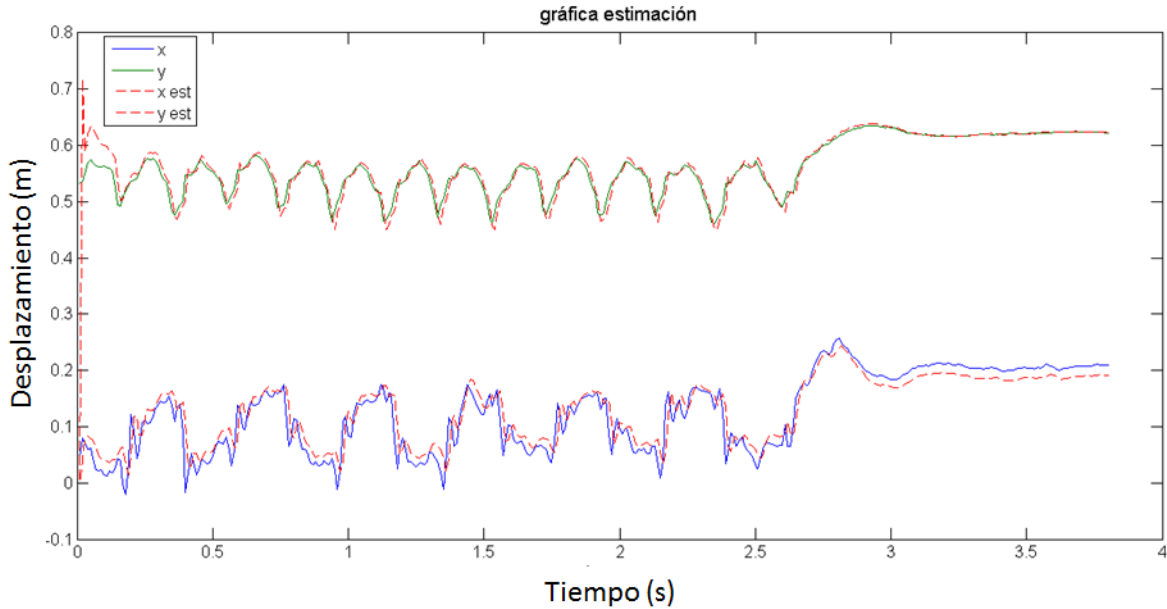


Figura 6.10. Gráfica de comparación entre la estimación y la señal real de ambos ejes

En la gráfica, la línea azul es la señal original en x , la línea verde es la señal original en y y la línea punteada roja es la estimación de ambos casos.

Utilizando la expresión (6.1) se analiza la precisión con la que el sistema identificado obtiene la señal de salida deseada:

$$R = \left(1 - \frac{\sum_{t=1}^N |y_t - \hat{y}_t|^2}{\sum_{t=1}^N |y_t|^2} \right) \times 100 \quad (6.1)$$

En este caso resulta:

$$R = 96.7955\% \quad (6.2)$$

Nuevamente se analiza el caso para el sistema multivariable obtenido de los datos del movimiento del centro de masa de la **persona 2** en el **eje x** y el **eje y** con los siguientes parámetros:

$n = 4$ y $i = 8$

$$x_{k+1} = \begin{bmatrix} 0.9973 & -0.0021 & -0.0768 & -0.0496 \\ 0.0014 & 0.9505 & -0.1239 & 0.1322 \\ 0.0018 & 0.0158 & 1.0564 & -0.0045 \\ 0.0022 & -0.0193 & 0.0819 & 1.0404 \end{bmatrix} x_k + \begin{bmatrix} -2.7518 & -6.7477 \\ 7.9883 & -1.6838 \\ 1.4100 & 5.3222 \\ 5.0355 & 7.2474 \end{bmatrix} e_k,$$

$$y_k = \begin{bmatrix} -0.0992 & 0.1381 & 0.0347 & -0.0763 \\ -0.2607 & -0.0541 & -0.0878 & -0.0251 \end{bmatrix} x_k.$$

La segunda mitad de los datos se utiliza para validar el algoritmo, introduciéndolos al modelo generado y obtener la señal de salida que se espera. La comparación entre la señal original y la identificada es mostrada en la Figura 6.11.

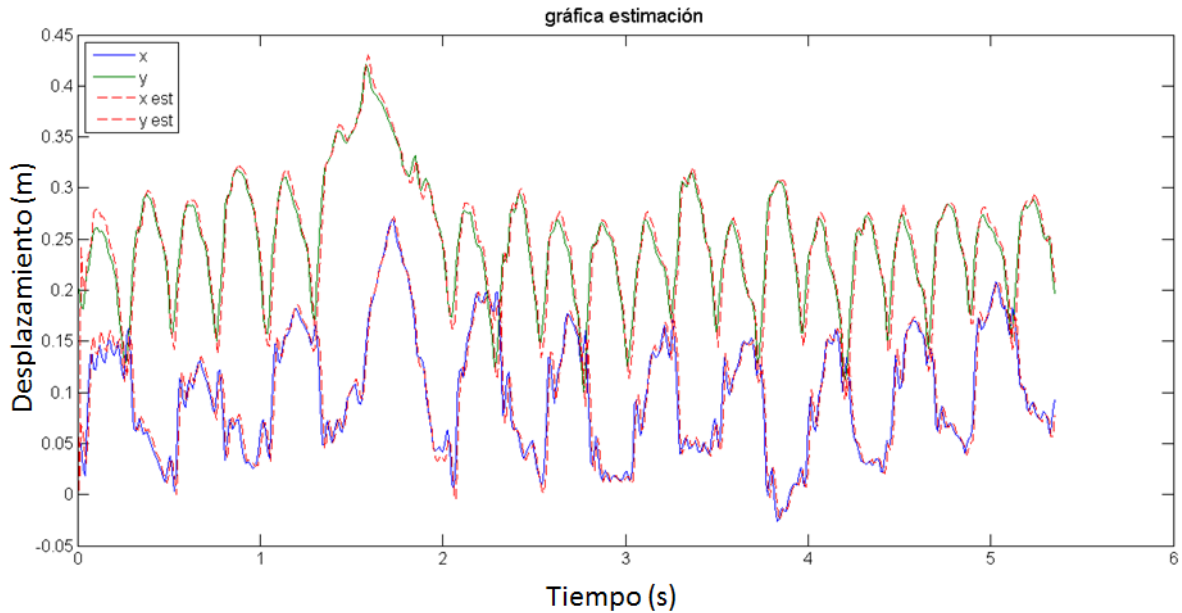


Figura 6.11. Gráfica de comparación entre la estimación y la señal real de ambos ejes.

En la gráfica, la línea azul es la señal original en x , la línea verde es la señal original en y y la línea punteada roja es la estimación de ambos casos.

Utilizando nuevamente la expresión (6.1) se analiza la precisión con la que el sistema identificado obtiene la señal de salida deseada, la cual resulta:

$$R = 98.2791\% . \quad (6.3)$$

Se utiliza el mismo procedimiento para la mitad de los datos de ambos ejes y se obtiene el modelo en espacio de estados que se presenta a continuación con los siguientes parámetros: $n = 4$ y $i = 8$.

$$x_{k+1} = \begin{bmatrix} 1.0001 & -0.0050 & 0.0039 & -0.0486 \\ 0.0003 & 0.9593 & -0.2428 & -0.0034 \\ 0.0005 & 0.0647 & 0.9683 & -0.0204 \\ 0.0010 & 0.0164 & -0.0194 & 1.0788 \end{bmatrix} x_k + \begin{bmatrix} -0.4510 & -5.5895 \\ -4.5402 & 3.6318 \\ -0.9056 & -1.6258 \\ 0.5443 & 14.3645 \end{bmatrix} e_k ,$$

$$y_k = \begin{bmatrix} -0.0982 & -0.1205 & -0.1126 & -0.0194 \\ -0.4035 & 0.0220 & 0.0283 & -0.0655 \end{bmatrix} x_k .$$

La segunda mitad de los datos se utiliza para validar el algoritmo, introduciéndolos al modelo generado y obtener la señal de salida que se espera. En la gráfica de la figura 6.12 se muestra la comparación entre la señal original y la identificada.

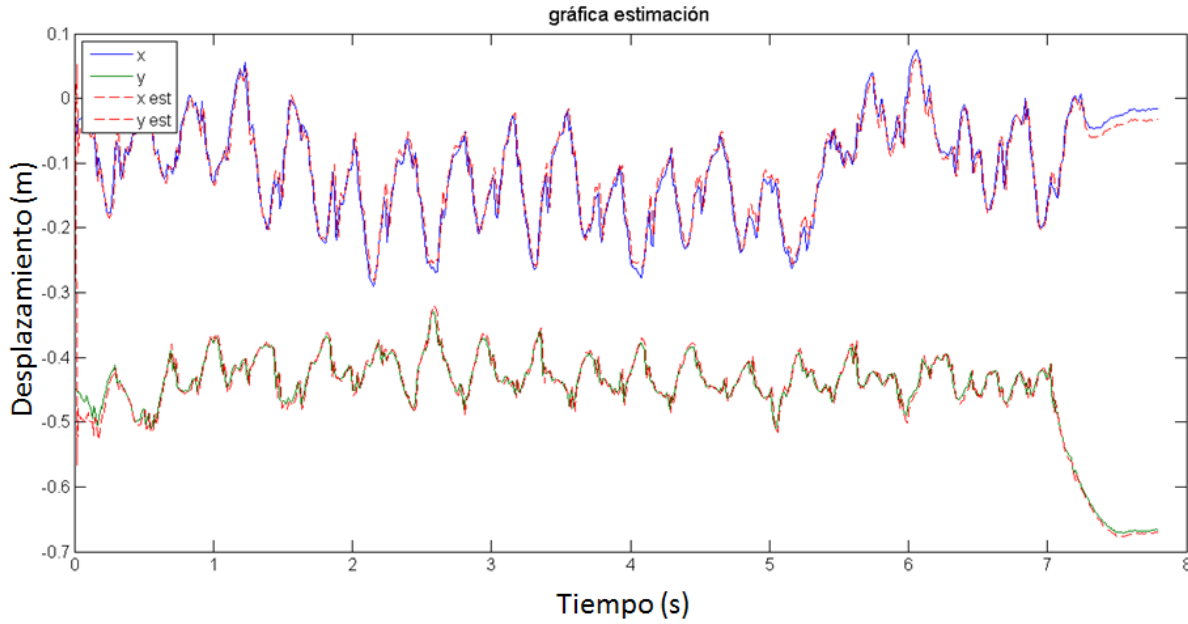


Figura 6.12. Gráfica de comparación entre la estimación y la señal real de ambos ejes.

Utilizando nuevamente la expresión (6.1) se analiza la precisión con la que el sistema identificado obtiene la señal de salida deseada, la cual resulta:

$$R = 98.3989\% . \quad (6.4)$$

A continuación, se realiza la identificación de la señal en la que se tiene un tropiezo mientras se está caminando, esto para comprobar el desempeño de la capa cuadrática del algoritmo. Los datos utilizados son los correspondientes al **eje x** de la **persona 1**, ya que en estos es donde se tiene una variación abrupta cuando ocurre el tropiezo, se toman entonces la mitad de éstos para obtener el modelo en espacio de estados de la forma que se muestra en la ecuaciones (4.98) – (4.103).

Se obtiene la estimación lineal con los siguientes parámetros:

$$n = 4 \text{ y } i = 8$$

$$x_{k+1}^{1,1} = \begin{bmatrix} 1.0064 & 0.0362 & -0.0055 & 0.0386 \\ 0.0190 & 0.9235 & -0.1677 & -0.0968 \\ 0.0283 & 0.0732 & 0.9594 & -0.2772 \\ -0.0279 & 0.1004 & 0.1932 & 0.9142 \end{bmatrix} x_k^{1,1} + \begin{bmatrix} -3.5207 \\ 2.4663 \\ 1.1828 \\ -1.1159 \end{bmatrix} e_k ,$$

$$y_k^{1,1} = [-0.1037 \quad 0.1119 \quad 0.0583 \quad 0.0542] x_k^{1,1} + e_k .$$

El siguiente paso es obtener la parte correspondiente a la primera capa de la parte cuadrática, que está representada por la siguiente ecuación en espacios de estados:

$$x_{k+1}^{1,2} = \begin{bmatrix} 0.9689 & -0.1203 & 0.1098 & -0.1321 \\ 0.1217 & 0.8642 & -0.4688 & 0.0682 \\ -0.0647 & 0.4723 & 0.7902 & -0.3096 \\ 0.0396 & 0.0585 & 0.2602 & 0.9268 \end{bmatrix} x_k^{1,2} + \begin{bmatrix} -1.1316 \\ -0.7138 \\ 0.3346 \\ -0.3049 \end{bmatrix} e_k^{1,1},$$

$$y_k^{1,2} = [-0.0427 \quad -0.0227 \quad -0.0335 \quad -0.0290] x_k^{1,2} + e_k.$$

El último paso consiste en obtener la representación en espacio de estados de la segunda capa de la parte cuadrática:

$$x_{k+1}^{2,2} = \begin{bmatrix} -0.2349 & 0.2353 & -0.2015 & -0.0295 \\ -0.4596 & -0.0435 & -0.0019 & -0.1147 \\ 0.4767 & 0.2846 & -0.0716 & 0.1139 \\ 0.0281 & 0.0192 & -0.1186 & 0.2832 \end{bmatrix} x_k^{2,2}$$

$$+ \begin{bmatrix} 0.5246 & 0 & 0 & 0 \\ 0 & -1.9942 & 0 & 0 \\ 0 & 0 & -3.3199 & 0 \\ 0 & 0 & 0 & 0.0554 \end{bmatrix} y_{k+1}^{1,2} \otimes x_{k+1}^{1,2},$$

$$y_k^{2,2} = [-0.0427 \quad -0.0227 \quad -0.0335 \quad -0.0290] x_k^{2,2} + e_k.$$

La segunda mitad de los datos se utilizó para validar el algoritmo, introduciéndolos al modelo generado y obteniendo así la señal de salida esperada.

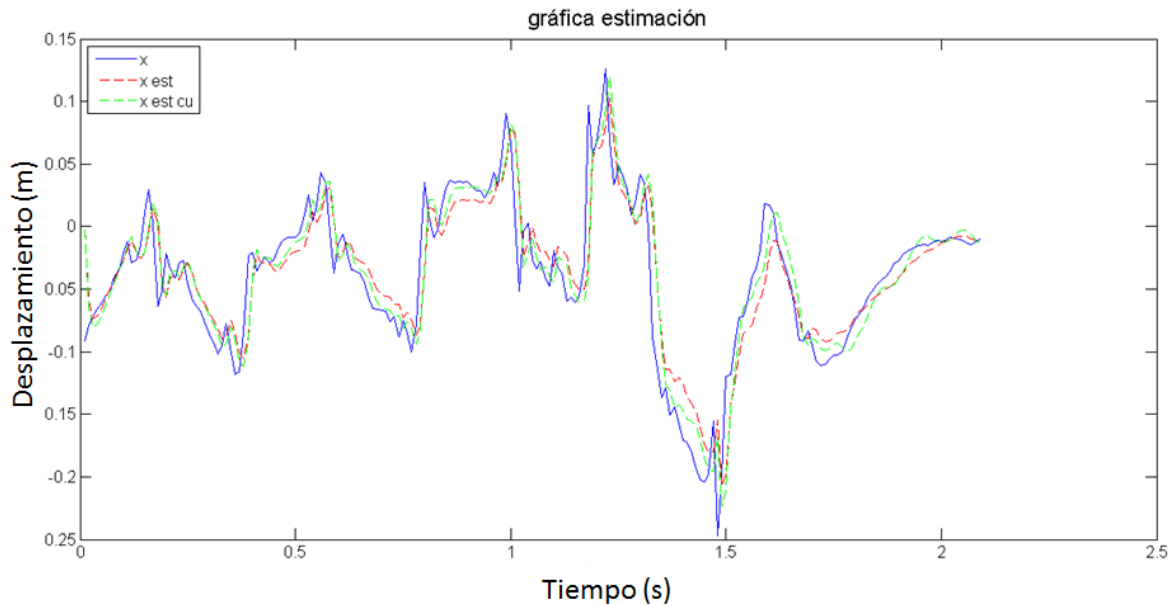


Figura 6.13. Gráfica de comparación de la señal original con las estimaciones lineal y cuadrática de los datos de tropiezo.

En la gráfica de la figura 6.13 se puede observar el comportamiento de la señal original (línea azul) en comparación con la estimación lineal (línea roja punteada) y la estimación cuadrática (línea verde punteada).

Se utiliza la expresión (4.104) y (6.1) para medir el desempeño del algoritmo, en este caso obtiene una R para la parte lineal y una R_{cu} para la parte cuadrática:

$$R = 85.0671 , \quad (6.5)$$

$$R_{cu} = 86.5420 . \quad (6.6)$$

Con lo que se obtiene una mejora del 1.4750 al usar la parte cuadrática del algoritmo.

La siguiente prueba consiste en utilizar los datos correspondientes al **eje x** de la **persona 2**, ya que en estos es donde se tiene una variación abrupta cuando ocurre el tropiezo, se toman entonces la mitad para obtener el modelo en espacio de estados de la forma que se muestra en las ecuaciones (4.98) – (4.103).

Se obtiene la estimación lineal con los siguientes parámetros:

$$n = 4 \text{ y } i = 8$$

$$x_{k+1}^{1,1} = \begin{bmatrix} 1.0192 & 0.0491 & 0.0467 & -0.0064 \\ -0.0330 & 0.9474 & 0.1289 & 0.0528 \\ -0.0108 & -0.0834 & 0.9978 & -0.0820 \\ 0.0297 & -0.0001 & -0.0090 & 0.8203 \end{bmatrix} x_k^{1,1} + \begin{bmatrix} -5.3694 \\ 4.8830 \\ 2.1752 \\ -2.1382 \end{bmatrix} e_k ,$$

$$y_k^{1,1} = [-0.0153 \quad 0.1159 \quad -0.0776 \quad -0.0620] x_k^{1,1} + e_k .$$

El siguiente paso es obtener la parte correspondiente a la primera capa de la parte cuadrática, que está representada por la siguiente ecuación en espacios de estados:

$$x_{k+1}^{1,2} = \begin{bmatrix} 0.2370 & -0.7145 & -0.3633 & 0.0677 \\ 0.5819 & 0.5424 & -0.5841 & -0.0901 \\ 0.0393 & 0.2647 & 0.1437 & 0.6728 \\ -0.3940 & 0.0848 & -0.3963 & 0.5308 \end{bmatrix} x_k^{1,2} + \begin{bmatrix} 3.6586 \\ 0.6381 \\ 1.1661 \\ -0.0970 \end{bmatrix} e_k^{1,1} ,$$

$$y_k^{1,2} = [0.0898 \quad -0.0201 \quad 0.0324 \quad 0.0214] x_k^{1,2} + e_k .$$

El último paso consiste en obtener la representación en espacio de estados de la segunda capa de la parte cuadrática:

$$x_{k+1}^{2,2} = \begin{bmatrix} -0.3884 & 0.0880 & -0.1003 & 0.0224 \\ -0.0419 & -0.3362 & -0.0535 & -0.0597 \\ -0.1371 & 0.2021 & -0.0741 & -0.1289 \\ -0.0933 & 0.1608 & 0.3685 & 0.0039 \end{bmatrix} x_k^{2,2} + \begin{bmatrix} 8.3433 & 0 & 0 & 0 \\ 0 & 3.2118 & 0 & 0 \\ 0 & 0 & 2.6412 & 0 \\ 0 & 0 & 0 & -1.2081 \end{bmatrix} y_{k+1}^{1,2} \otimes x_{k+1}^{1,2},$$

$$y_k^{2,2} = [0.0011 \quad -0.0001 \quad 0.0033 \quad -0.0027] x_k^{2,2} + e_k.$$

La segunda mitad de los datos se utilizó para validar el algoritmo, introduciéndolos al modelo generado y obteniendo así la señal de salida esperada.

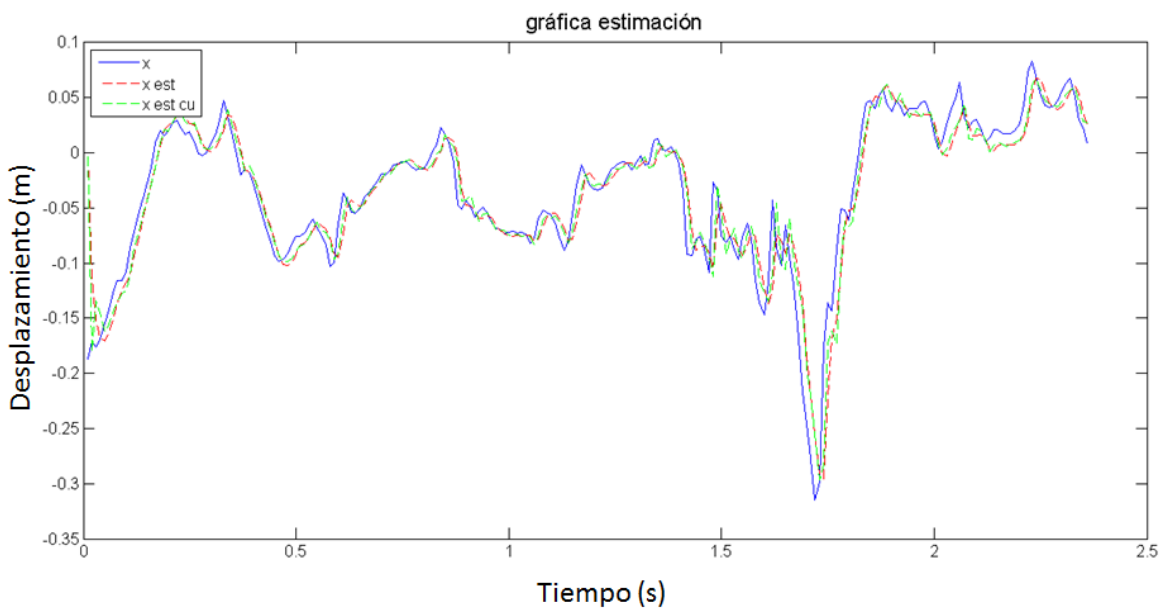


Figura 6.14. Gráfica de comparación de la señal original con las estimaciones lineal y cuadrática de los datos de tropiezo.

En la gráfica de la figura 6.14 se puede observar el comportamiento de la señal original (línea azul) en comparación con la estimación lineal (línea roja punteada) y la estimación cuadrática (línea verde punteada).

Se utiliza la expresión (6.1) para medir el desempeño del algoritmo, en este caso obtiene una R para la parte lineal y una R_{cu} para la parte cuadrática:

$$R = 87.9993, \tag{6.7}$$

$$R_{cu} = 90.1364. \tag{6.8}$$

Con lo que se obtiene una mejora del 2.1370 al usar la parte cuadrática del algoritmo.

A continuación se utilizan los datos correspondientes al **eje x** de la **persona 3**, se toman entonces la mitad de estos para obtener el modelo en espacio de estados de la forma que se muestra en la ecuaciones (4.98) – (4.103).

Se obtiene la estimación lineal con los siguientes parámetros:

$$n = 4 \text{ y } i = 8$$

$$x_{k+1}^{1,1} = \begin{bmatrix} 0.9481 & -0.1013 & 0.0018 & -0.0363 \\ 0.0225 & 0.9491 & 0.0827 & -0.0330 \\ -0.0467 & 0.0513 & 1.1042 & 0.2217 \\ -0.0013 & 0.0371 & -0.2169 & 0.9571 \end{bmatrix} x_k^{1,1} + \begin{bmatrix} -4.8213 \\ -11.4928 \\ -9.3192 \\ 23.8038 \end{bmatrix} e_k ,$$

$$y_k^{1,1} = [-0.1669 \quad -0.1223 \quad 0.0113 \quad -0.0469] x_k^{1,1} + e_k .$$

El siguiente paso es obtener la parte correspondiente a la primera capa de la parte cuadrática, que está representada por la siguiente ecuación en espacios de estados:

$$x_{k+1}^{1,2} = \begin{bmatrix} 1.0019 & 0.0483 & -0.0330 & 0.0575 \\ -0.0731 & 0.1765 & 0.7451 & 0.5287 \\ 0.0467 & -0.8321 & 0.1348 & 0.2895 \\ -0.0418 & -0.3678 & -0.3095 & 0.2920 \end{bmatrix} x_k^{1,2} + \begin{bmatrix} -0.6149 \\ -0.8223 \\ -0.2021 \\ 0.1499 \end{bmatrix} e_k^{1,1} ,$$

$$y_k^{1,2} = [-0.0347 \quad 0.0101 \quad -0.0467 \quad 0.0320] x_k^{1,2} + e_k .$$

El último paso consiste en obtener la representación en espacio de estados de la segunda capa de la parte cuadrática:

$$x_{k+1}^{2,2} = \begin{bmatrix} 0.1853 & 0.0827 & 0.0125 & -0.1518 \\ 0.0144 & 0.0221 & 0.0201 & 0.0381 \\ 0.1143 & -0.0800 & -0.0636 & -0.3152 \\ 0.0860 & 0.0542 & -0.3312 & 0.3833 \end{bmatrix} x_k^{2,2} + \begin{bmatrix} -1.1776 & 0 & 0 & 0 \\ 0 & 0.0485 & 0 & 0 \\ 0 & 0 & -0.4619 & 0 \\ 0 & 0 & 0 & -0.1464 \end{bmatrix} y_{k+1}^{1,2} \otimes x_{k+1}^{1,2} ,$$

$$y_k^{2,2} = [0.0001 \quad -0.0014 \quad 0.0003 \quad -0.0023] x_k^{2,2} + e_k ,$$

La segunda mitad de los datos se utilizó para validar el algoritmo, introduciéndolos al modelo generado y obteniendo así la señal de salida esperada.

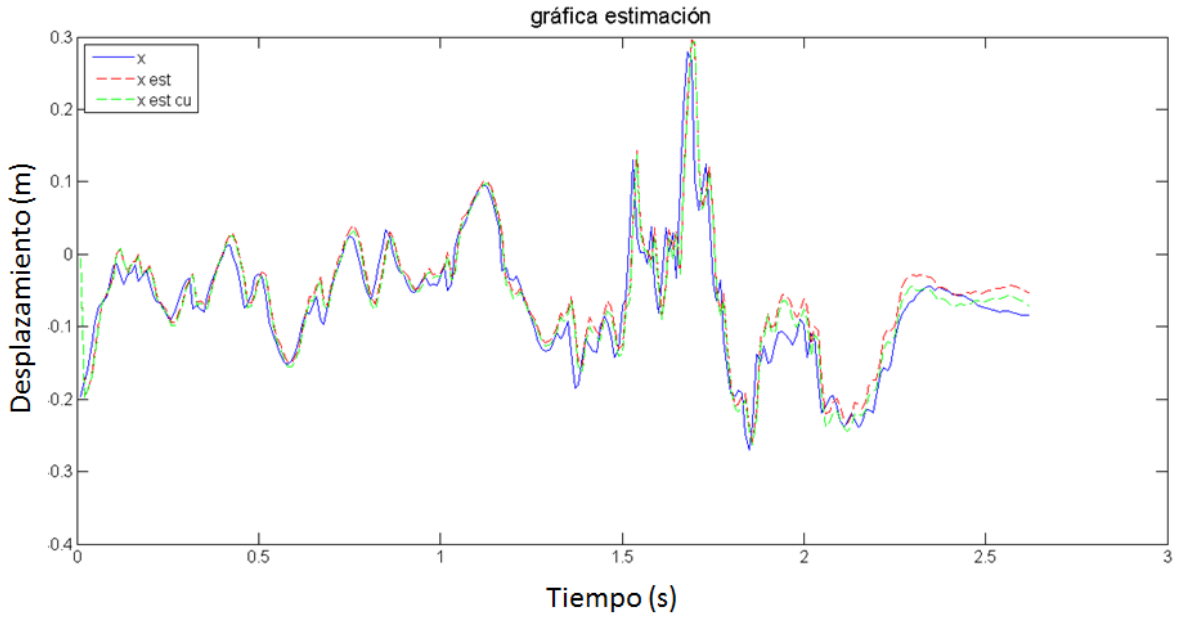


Figura 6.15. Gráfica de comparación de la señal original con las estimaciones lineal y cuadrática de los datos de tropiezo.

Análogamente a los casos anteriores, se utiliza la expresión (6.1) para medir el desempeño del algoritmo, en este caso obtiene una R para la parte lineal y una R_{cu} para la parte cuadrática:

$$R = 88.1956 , \tag{6.8}$$

$$R_{cu} = 89.5446 . \tag{6.9}$$

Con lo que se obtiene una mejora del 1.3489 al usar la parte cuadrática del algoritmo.

Los resultados anteriores se resumen en la tabla 1.

		Identificación lineal	Identificación cuadrática
Persona 1	Caminata normal	96.7955	
	Tropiezo	85.0671	86.5420
Persona 2	Caminata normal	98.2791	
	Tropiezo	87.9993	90.1364
Persona 3	Caminata normal	98.3989	
	Tropiezo	88.1956	89.5446

Tabla 1. Resumen de los resultados de identificación

6.2. Análisis caótico de la marcha humana

Utilizando el algoritmo descrito en el capítulo 5, se llevó a cabo el análisis del comportamiento caótico de diferentes personas en la marcha humana, obteniendo su gráfica de fase y los coeficientes de Lyapunov. Para ello, se tomaron solo los datos del eje x sin haber sido filtrados, ya que se observó que se perdía información importante para este análisis con el filtro pasabajas IIR.

Se utilizan los datos de la caminata normal de la **persona 1** sin filtrar; para obtener un coeficiente positivo de Lyapunov, se debe de tener un sistema caótico. Con la gráfica de fase se puede observar el comportamiento del sistema como lo muestra la Figura 6.16.

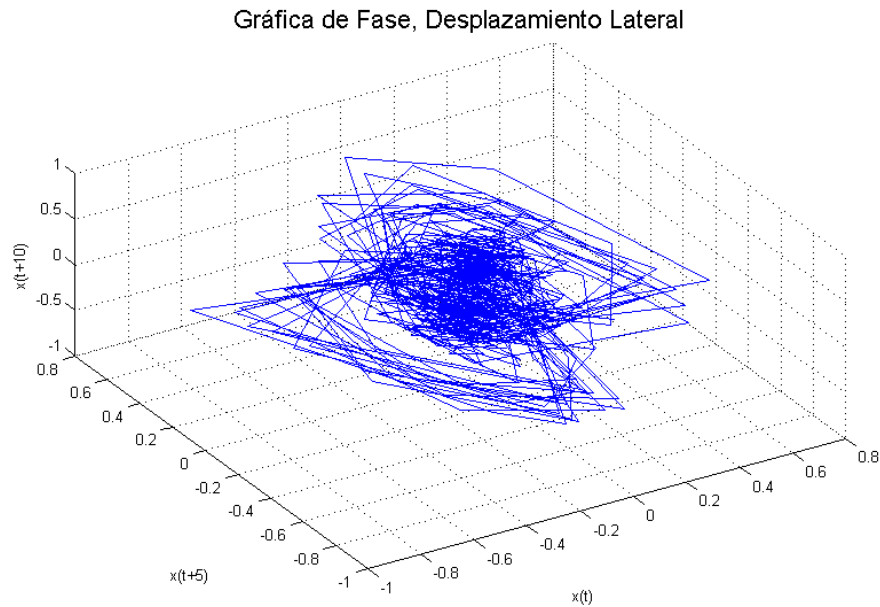


Figura 6.16. Gráfica de fase caminata normal en el eje x de la persona 1.

Con el algoritmo se obtienen los coeficientes de Lyapunov positivos $[0.5550, 0.4453]$.

Otra señal a analizar es la de los datos obtenidos de la persona que tropieza mientras camina, su gráfica de fase se presenta en la Figura 6.17.

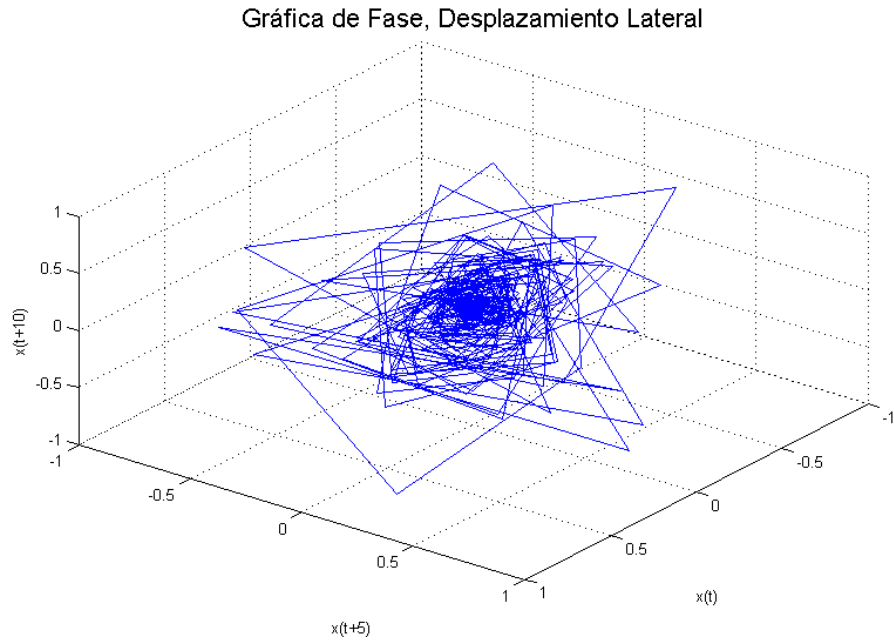


Figura 6.17. Gráfica de fase tropiezo en la caminata eje x de la persona 1.

El coeficiente de Lyapunov obtenido es [0.3214] .

También se obtiene la gráfica de fase de los datos correspondientes al desbalance de la persona mientras camina, ver figura 6.18.

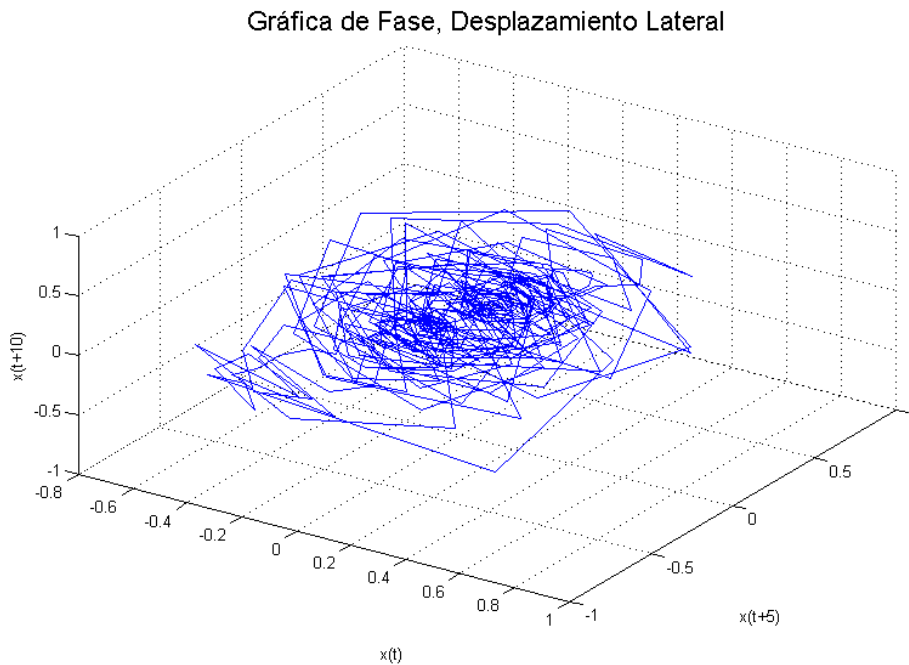


Figura 6.18. Gráfica de fase desequilibrio en la caminata eje x de la persona 1.

El coeficiente de Lyapunov obtenido es: [0.6324] .

Para la **persona 2** se obtiene la gráfica de fase de la caminata normal, que se muestra en la Figura 6.19.

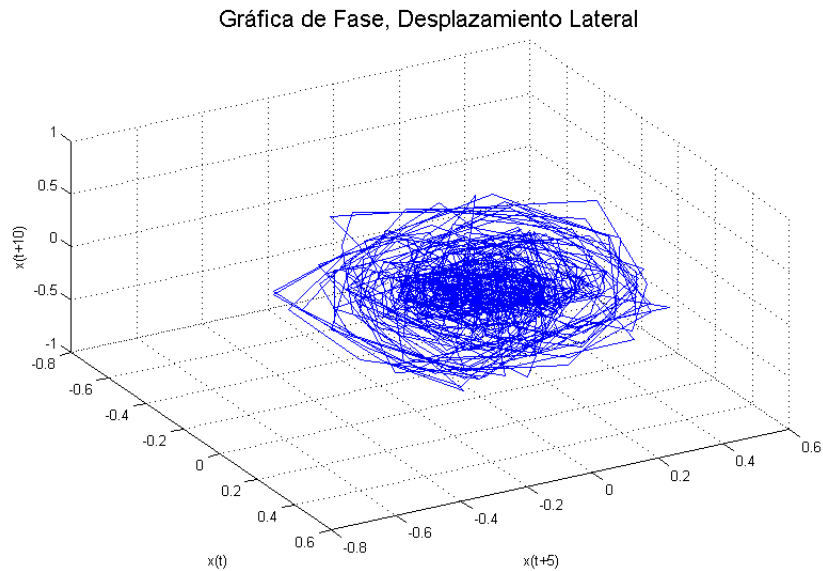


Figura 6.19. Gráfica de fase caminata normal en el eje x de la persona 2.

Con el algoritmo presentado en el capítulo 5 se obtienen los coeficientes de Lyapunov positivos [0.8829, 1.1488] .

También se analiza la señal correspondiente al tropiezo de la persona en la marcha, cuya gráfica de fase se presenta en la Figura 6.29.

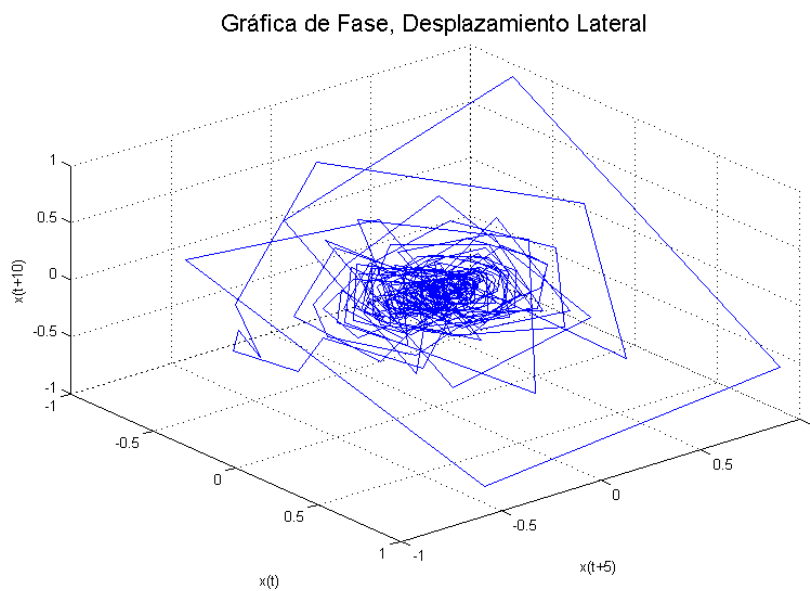


Figura 6.20. Gráfica de fase tropiezo en la caminata eje x de la persona 2.

El coeficiente de Lyapunov obtenido es [0.2700] .

Se obtiene la gráfica de fase de los datos correspondientes al desbalance de la persona mientras camina, Figura 6.21.

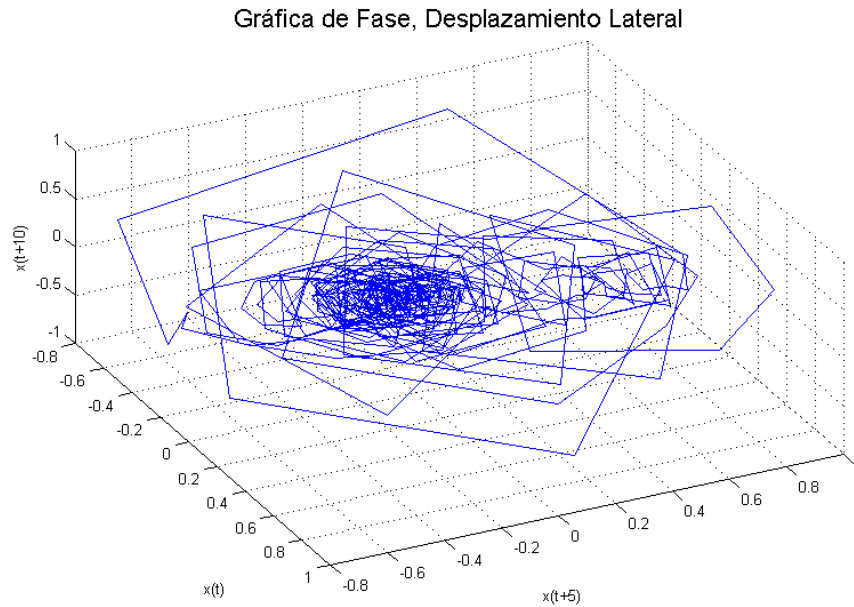


Figura 6.21. Gráfica de fase desequilibrio en la caminata eje x de la persona 2.

Los coeficientes de Lyapunov obtenidos son $[1.12, 0.9387]$.

Para la **persona 3** se obtiene la gráfica de fase de la caminata normal que se muestra en la Figura 6.22.

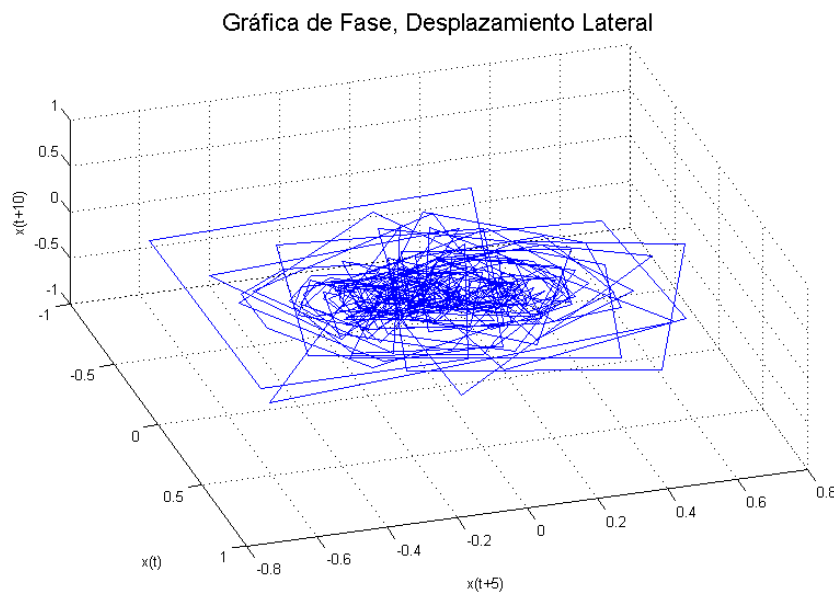


Figura 6.22. Gráfica de fase caminata normal en el eje x de la persona 3.

Con el algoritmo se obtienen los coeficientes de Lyapunov positivos $[1.5464, 1.2145]$.

Se analiza también la señal que corresponde al tropiezo de la persona mientras camina, obteniendo su gráfica de fase de la Figura 6.23.

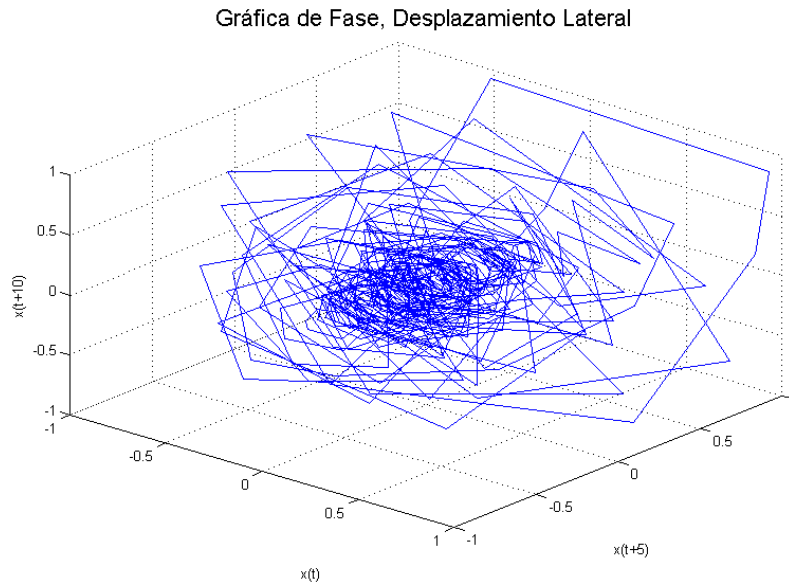


Figura 6.23. Gráfica de fase tropiezo en la caminata eje x de la persona 3.

Los coeficientes de Lyapunov obtenidos son

$$[0.0930, 0.0403] .$$

Se obtiene la gráfica de fase de la Figura 6.24 con los datos del desbalance de la persona mientras camina:

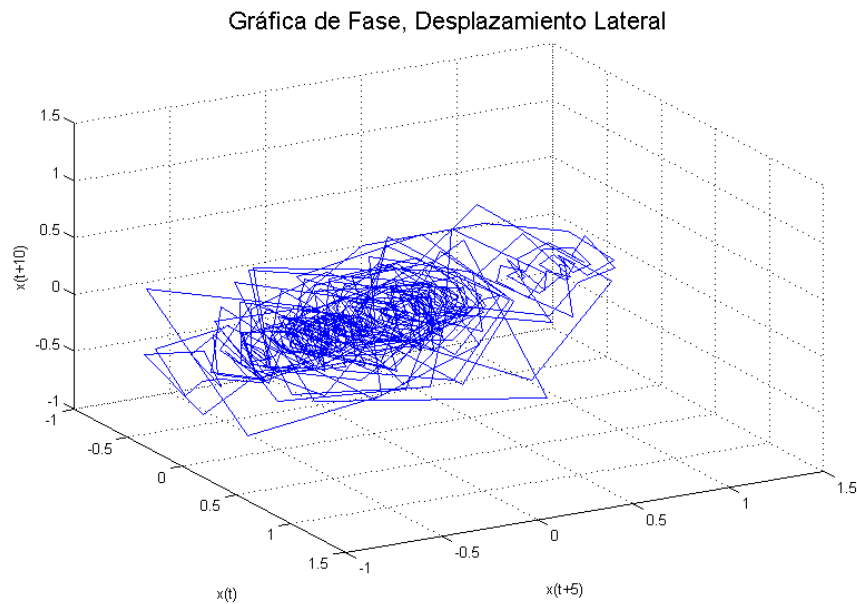


Figura 6.24. Grafica de fase desequilibrio en la caminata eje x de la persona 3.

El coeficiente de Lyapunov obtenido es $[1.57]$.

Se resumen los resultados obtenidos en la Tabla 2 [31]. En esta tabla se muestran los coeficientes de Lyapunov en los distintos casos de la marcha, caminata normal, tropiezo en la marcha y desequilibrio en la marcha.

		Coeficientes de <u>Lyapunov</u>
Persona 1	Caminata normal	0.555, 0.4453
	Tropiezo	0.3214
	Desequilibrio	0.6324
Persona 2	Caminata normal	0.8829, 1.1488
	Tropiezo	0.27
	Desequilibrio	1.12, 0.9387
Persona 3	Caminata normal	1.5464, 1.2145
	Tropiezo	0.0930, 0.0403
	Desequilibrio	1.57

Tabla 2. Coeficientes de Lyapunov para diferentes casos en la marcha.

Estos coeficientes positivos indican que el movimiento del centro de masa que se produce en el proceso de la marcha tiene un comportamiento caótico.

Conclusiones y trabajo a futuro

Conclusiones generales

En este trabajo se planteó el obtener un modelo matemático de un modelo simplificado de la marcha humana mediante datos obtenidos experimentalmente que se utilizaron en algoritmo de identificación de sistemas que está formado por dos bloques, uno de identificación lineal y otro de identificación cuadrática, esto debido a la naturaleza del sistema de interés.

Para obtener los datos experimentales se implementó un módulo de medición que permite obtener datos del centro de masa de una persona mientras está caminando. Estos datos son de gran utilidad ya que permiten describir un modelo simplificado de la marcha humana. En este punto se tuvo el problema de una señal con una cantidad considerable de ruido al usar un acelerómetro para obtener los datos, ya que los valores obtenidos eran inexactos; esto se solucionó con el uso de un filtro, al hacer un análisis de las frecuencias de mayor importancia en la señal se observó que éstas son principalmente bajas frecuencias por lo que se implementó un filtro pasa-bajas IIR de orden 1. A pesar de los problemas mencionados, el acelerómetro permitió construir un dispositivo de medición de bajo costo. Además, los datos experimentales obtenidos contienen las características necesarias para describir el movimiento del centro de masa durante la marcha humana.

Posteriormente se utilizó un algoritmo de identificación de subespacios para obtener un modelo en la representación del espacio de estados del sistema simplificado de la caminata humana mediante los datos experimentales obtenidos con el módulo de medición mencionada anteriormente. En un comienzo se pensó estimar el modelo de la marcha humana mediante un algoritmo de identificación cuadrático ya que se suponía que un algoritmo de identificación no tendría un buen desempeño con señales de este tipo. Al realizar las pruebas experimentales se obtuvo un resultado arriba de 90% en la estimación lineal por lo que ya no fue necesario la estimación cuadrática en las señales de la caminata. Para comprobar el desempeño de la parte cuadrática del algoritmo se procedió a obtener señales de caminata con cambios abruptos como por ejemplo el caso cuando una persona tropieza mientras está caminando, en estas señales se obtuvo un rango entre 85% y 87% en la identificación lineal y una mejora de entre el 2% y 3% cuando se aplicaba la parte cuadrática. El algoritmo de identificación de subespacios implementado presentó relativa facilidad en la programación debido al uso de operadores geométricos y descomposición de matrices como los principales elementos para construir la mayor parte del código en MATLAB. En comparación con otros algoritmos de identificación como los ARX, ARMAX u OE, el algoritmo de identificación de subespacios tiene la ventaja de poder obtener sistemas dinámicos SISO o MIMO sin ninguna modificación en el código, los algoritmos ARX, ARMAX u OE tienen un mejor desempeño al obtener sistemas dinámicos SISO, sin embargo cuando se requiere obtener un sistemas MIMO mediante estos métodos, se tiene la desventaja de necesitar técnicas no lineales iterativas para minimizar la función de costo, esto tiene como consecuencia un incremento en el tiempo de procesamiento, en la complejidad del algoritmo y por consecuencia en los requerimientos del equipo de cómputo a utilizar.

Por último, se tuvo la inquietud de utilizar los datos de la marcha humana para analizar su comportamiento mediante los exponentes de Lyapunov, los valores obtenidos de estos coeficientes muestran que la caminata tiene comportamiento caótico moderado por su propia naturaleza; esto al comparar con diversos sistemas caóticos presentados en la literatura. Estos coeficientes son de gran utilidad para analizar comportamientos irregulares. Esta solo es una pequeña parte del potencial que tiene este tipo de análisis, ya que se pueden examinar diversos aspectos de la marcha humana con diferentes enfoques.

Trabajo a futuro

Para el trabajo a futuro se pretende mejorar el módulo de medición, remplazando el acelerómetro por una unidad de medición inercial (IMU), la cual ya tiene integrado un acelerómetro y un giroscopio. El acelerómetro es sensible a vibraciones y ruido mecánico en general que contamina la señal deseada causando mediciones inexactas, el giroscopio es menos sensible al tipo de ruido que afecta al acelerómetro, sin embargo el giroscopio tiene el problema de deriva (el cual consiste en que no regresa a un valor de referencia cero cuando la rotación termina). Estos dos elementos en conjunto pueden evitar los problemas que tienen individualmente obteniendo mediciones más exactas.

Otro punto que se puede realizar en el futuro es la creación de un algoritmo de generación de patrones de marcha para un robot humanoide con el modelo obtenido mediante la identificación de subespacios en este trabajo. El robot humanoide que se podría utilizar es el modelo NAO de la compañía Aldebaran, ya que es relativamente accesible.

Apéndice A. Código en Matlab del algoritmo implementado

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%      ALGORITMO DE IDENTIFICACIÓN DE SISTEMAS BASADO EN SUBESPACIOS
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%      Primera capa - estimación lineal - estocástica
%
%Se definen las constantes m y ny que indicaran si se trata de un sistema
%SISO o MIMO y modificaran la dimensión de las matrices de Hankel según
%sea el caso, también se definen i y j los cuales determinan las filas y
%columnas de la matriz de Hankel

[m,ny] = size(y);

i = 8;
j = ny-2*i+1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      Paso 1 del algoritmo de la sección 4.2
%
%Se calcula la matriz de Hankel de los datos experimentales

Hy = zeros(m*2*i,j);
for k = 1:2*i
    Hy((k-1)*m+1:k*m,:) = y(:,k:k+j-1);
end

Y = Hy/sqrt(j);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      Paso 2 del algoritmo de la sección 4.2
%
%Se calcula el factor R de la matriz de Hankel obtenida

R = triu(qr(Y'))';
R = R(1:2*i*(m),1:2*i*(m));
```

```

%////////////////////////////////////
%                               Paso 3 del algoritmo de la sección 4.2
%////////////////////////////////////

%Se extraen las matrices Ryf correspondiente a los datos futuros de
%salida, Ryp correspondiente a los datos pasados de salida y Ryfa Rype
%para los datos futuros extendidos y pasados disminuidos.

Ryp = R(1:m*i,:);
Ryf = R(m*i+1:(m+1)*i,:);

Rype = R(1:m*i+1,:);
Ryfa = R(m*i+2:(m+1)*i,:);

%////////////////////////////////////
%                               Paso 4 del algoritmo de la sección 4.2
%////////////////////////////////////

%Se calculan las proyecciones Oi y Oim

Oi = Ryf*(pinv(Ryp))*Ryp;
Oi = Oi(1:i*(m),1:i*(m));

Oim = Ryfa*(pinv(Rype))*Rype;
Oim = Oim(1:i*(m)-1,1:i*(m)+1);

%////////////////////////////////////
%                               Paso 5 del algoritmo de la sección 4.2
%////////////////////////////////////

%Se calcula la SVD de la proyección Oi

[U,S,V] = svd(Oi);

vaux_s = diag(S);

U1 = U(:,1:n);

%////////////////////////////////////
%                               Paso 6 del algoritmo de la sección 4.2
%////////////////////////////////////

%Se determinan gama y gamam1 con las matrices obtenidas de la SVD y sus
%respectivas pseudoinversas

gama = U1 * diag(sqrt(vaux_s(1:n)));
gamam1 = U1(1:m * (i-1),:) * diag(sqrt(vaux_s(1:n)));

gama_inv = pinv(gama);
gamam1_inv = pinv(gamam1);

```

```

%////////////////////////////////////
%                               Paso 7 del algoritmo de la sección 4.2
%////////////////////////////////////

%Se determinan los estados mediante las matrices gama y gamam1 en
%conjunto con las proyecciones obtenidas anteriormente Oi y Oim

Xi = gama_inv * Oi;
Xip = gamam1_inv * Oim;

%////////////////////////////////////
%                               Paso 8 del algoritmo de la sección 4.2
%////////////////////////////////////

%Se obtiene y se resuelve la ecuación formada mediante minios cuadrados,
%se extraen las matrices A y C de la solución

Rhs = [Xi , zeros(n,m)];
Lhs = [Xip ; R(m*i+1:m*(i+1),1:m*(i+1))];

sol = Lhs/Rhs;

A = sol(1:n,1:n);
C = sol(n+1:n+m,1:n);

%////////////////////////////////////
%                               Paso 9 del algoritmo de la sección 4.2
%////////////////////////////////////

%Se determinan las matrices de covarianza Q S y R

res = Lhs - sol*Rhs;
cov = res*res';
Qs = cov(1:n,1:n);
Ss = cov(1:n,n+1:n+m);
Rs = cov(n+1:n+m,n+1:n+m);

%////////////////////////////////////
%                               Paso 10 del algoritmo de la sección 4.2
%////////////////////////////////////

%Se calcula sigma, G y Lambda0

sigma = dlyap(A,Qs);
G = A * sigma*C' + Ss;
L0 = C * sigma*C' + Rs;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Paso 11 del algoritmo de la sección 4.2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Se calcula P y la ganancia de Kalman resolviendo la ecuación de Riccati

L0i = inv(L0);

AA = [A'-C' * L0i*G' zeros(n,n);-G * L0i * G' eye(n)];
BB = [eye(n) -C' * L0i * C;zeros(n,n) A-G * L0i * C];

[v,d] = eig(AA,BB);
ew = diag(d);

[dum,I]=sort(abs(ew));

Po = real(v(n+1:2*n,I(1:n))/v(1:n,I(1:n)));

Kf = (G - A*Po*C')*(inv(L0 - C*Po*C'));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Paso 12 del algoritmo de la sección 4.2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Se obtiene la estimación de la salida mediante el modelo estocástico
%obtenido con las matrices A, C y Kf

xst = Xi;
yest = zeros(m,ny);

for ipe = 1:ny
    yest(:,ipe) = C * xst(:,ipe) ;
    xst(:,ipe+1) = A * xst(:,ipe) + Kf * (y(:,ipe) - yest(:,ipe));
end

%Se obtiene la variable R1 que nos indica el desempeño del algoritmo al
%comparar la similitud en la señal estimada y la señal real (ecuación
%(6.1))

ve1 = y;
ve2 = yest;

ver = (abs(ve1 - ve2)).^2;
ver2 = (abs(ve1)).^2;

vef1 = cumsum(ver);
vef2 = cumsum(ver2);

R1 = (1-(vef1/vef2))*100

%Se obtiene el error entre la señal estimada y la señal real

er = y - yest;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Segunda capa - estimación cuadrática - estocástica
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%En esta primera parte de la estimación cuadrática se toman la parte de
%los datos que no se pudieron estimar con la parte lineal y se utilizan
%en el mismo algoritmo de identificación estocástica

Hy21 = zeros(m*2*i,j);
for k = 1:2*i
    Hy21((k-1)*m+1:k*m,:) = er(:,k:k+j-1);
end

Y21 = Hy21/sqrt(j);

R21 = triu(qr(Y21'))';
R21 = R21(1:2*i*(m),1:2*i*(m));

Ryp21 = R21(1:m*i,:);
Ryf21 = R21(m*i+1:(m+1)*i,:);

Rype21 = R21(1:m*i+1,:);
Ryfa21 = R21(m*i+2:(m+1)*i,:);

Oi21 = Ryf21*(pinv(Ryp21))*Ryp21;
Oi21 = Oi21(1:i*(m),1:i*(m));

Oim21 = Ryfa21*(pinv(Rype21))*Rype21;
Oim21 = Oim21(1:i*(m)-1,1:i*(m)+1);

[U21,S21,V21] = svd(Oi21);

vaux_s21 = diag(S21);

U121 = U21(:,1:n);

gama21 = U121 * diag(sqrt(vaux_s21(1:n)));
gamam121 = U121(1:m * (i-1),:) * diag(sqrt(vaux_s21(1:n)));

gama_inv21 = pinv(gama21);
gamam1_inv21 = pinv(gamam121);

Xi21 = gama_inv21 * Oi21;
Xip21 = gamam1_inv21 * Oim21;

Rhs21 = [Xi21 , zeros(n,m)];
Lhs21 = [Xip21 ; R21(m*i+1:m*(i+1),1:m*(i+1))];

sol21 = Lhs21/Rhs21;

A21 = sol21(1:n,1:n);
C221 = sol21(n+1:n+m,1:n);

```

```

res21 = Lhs21 - sol21*Rhs21;
cov21 = res21*res21';
Qs21 = cov21(1:n,1:n);
Ss21 = cov21(1:n,n+1:n+m);
Rs21 = cov21(n+1:n+m,n+1:n+m);

sigma21 = dlyap(A21,Qs21);
G21 = A21 * sigma21*C21' + Ss21;
L021 = C21 * sigma21*C21' + Rs21;

AA21 = [A21'-C21' * L021\G21' zeros(n,n);-G21 * L021\G21' eye(n)];
BB21 = [eye(n) -C21' * L021\C21;zeros(n,n) A21-G21 * L021\C21];

[v21,d21] = eig(AA21,BB21);
ew21 = diag(d21);

[dum21,I21] = sort(abs(ew21));

Po21 = real(v21(n+1:2*n,I21(1:n))/v21(1:n,I21(1:n)));

Kf21 = (G21 - A21*Po21*C21')*(inv(L021 - C21*Po21*C21'));

xst21 = Xi21;
yest21 = zeros(m,ny);

for ipe21 = 1:ny
    yest21(:,ipe21) = C21 * xst21(:,ipe21) ;
    xst21(:,ipe21+1) = A21 * xst21(:,ipe21) + Kf21 * (er(:,ipe21) -
yest21(:,ipe21));
end

%Se calcula el error de la segunda capa

er2 = er - yest21;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Tercera capa - estimación cuadrática - determinista
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%En esta última capa se utiliza la señal de error de la segunda capa como
%datos de salida y el producto kronecker de los estados de la segunda
%capa con el error de la segunda capa, se utiliza como entrada

ecc = kron(er2,Xi21);

[l,nu] = size(ecc);

```

```

%////////////////////////////////////
%                               Paso 1 del algoritmo de la sección 4.3
%////////////////////////////////////

%Se obtienen las matrices de Hankel de los datos mencionados
%anteriormente como entrada y como salida

Hucu = zeros(1*2*i,j);
for k = 1:2*i
    Hucu((k-1)*1+1:k*1,:) = ecc(:,k:k+j-1);
end

l=1;

Hycu = zeros(1*2*i,j);
for k = 1:2*i
    Hycu((k-1)*1+1:k*1,:) = er2(:,k:k+j-1);
end

Uecu = Hucu/sqrt(j);

Yecu = Hy/sqrt(j);

%////////////////////////////////////
%                               Paso 2 del algoritmo de la sección 4.3
%////////////////////////////////////

%Se obtiene el factor R de las matrices de Hankel obbtenidas

Rcu = triu(qr([Uecu;Yecu]'))';
Rcu = Rcu(1:2*i*(m+1),1:2*i*(m+1));

%////////////////////////////////////
%                               Paso 3 del algoritmo de la sección 4.3
%////////////////////////////////////

%Del factor R se obtienen las matrices Rfcu, Rpcu, Rucu

Rfcu = Rcu((2*m+1)*i+1:2*(m+1)*i,:);

Rpcu = [Rcu(1:m*i,:); Rcu(2*m*i+1:(2*m+1)*i,:)];

Rucu = Rcu(m*i+1:2*m*i,1:2*m*i);

%////////////////////////////////////
%                               Paso 4 del algoritmo de la sección 4.3
%////////////////////////////////////

%Se calcula las proyecciones ortogonales Zi y la proyección oblicua Oi

Zi = Rfcu * pinv([Rpcu;Rucu]) * [Rpcu;Rucu];

Vppo = Ryfd * pinv([Rwpd;Rufd]);

```

```

Vppo = Vppo(:,1:rd);

Obcu = Vppo*Rwpd;

WOWcu = [Obcu(:,1:mi2) - (Obcu(:,1:mi2)/Rucu)*Rucu, Obcu(:,mi2+1:2*(m+1)*i)];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Paso 5 del algoritmo de la sección 4.3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Se calcula la SVD de la matriz WOWcu

[Ucu,Scu,Vcu] = svd(WOWcu);

vaux_scu = diag(Scu);

U1cu = Ucu(:,1:n);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Paso 6 del algoritmo de la sección 4.3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Se determinan las matrices gamacu y gamaim1cu con sus respectivas
%pseudoinversas

gamacu = U1cu * diag(sqrt(vaux_scu(1:n)));
gamaim1cu = gamacu(1:1*(i-1),:);

gama_invcu = pinv(gamacu);
gamaim1_invcu = pinv(gamaim1cu);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Paso 7 del algoritmo de la sección 4.3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Se resuelve la ecuación obtenida mediante mínimos cuadrados para obtener
%las matrices A y C

Rhscu = [gama_invcu*Rcu((2*m+1)*i+1:2*(m+1)*i,1:(2*m+1)*i), zeros(n,1);...
         Rcu(m*i+1:2*m*i,1:(2*m+1)*i+1)];

Lhscu = [gamaim1_invcu*Rcu((2*m+1)*i+1:2*(m+1)*i,1:(2*m+1)*i+1) ; ...
         Rcu((2*m+1)*i+1:(2*m+1)*i+1,1:(2*m+1)*i+1)];

solcu = Lhscu/Rhscu;

Acu = solcu(1:n,1:n);
Ccu = solcu(n+1:n+1,1:n);
rescu = Lhscu - solcu*Rhscu;

```

```

%////////////////////////////////////
%                               Paso 8 del algoritmo de la sección 4.3
%////////////////////////////////////

%Se calcula nuevamente gamacu a partir de A y C

gamacu = Ccu;

for kcu=2:i
    gamacu((kcu-1)*l+1:kcu*l,:) = gamacu((kcu-2)*l+1:(kcu-1)*l, :)*Acu;
end

gamaimlcu = gamacu(1:l*(i-1), :);

gama_invcu = pinv(gamacu);
gamaiml_invcu = pinv(gamaimlcu);

%////////////////////////////////////
%                               Paso 9 del algoritmo de la sección 4.3
%////////////////////////////////////

%Se obtiene la matriz D resolviendo la ecuación (105)

Rhscu= [gama_invcu*Rcu((2*m+1)*i+1:2*(m+1)*i, 1:(2*m+1)*i), zeros(n,l); ...
        Rcu(m*i+1:2*m*i, 1:(2*m+1)*i+1)];

Lhscu = [gamaiml_invcu*Rcu((2*m+1)*i+1:2*(m+1)*i, 1:(2*m+1)*i+1); ...
        Rcu((2*m+1)*i+1:(2*m+1)*i+1, 1:(2*m+1)*i+1)];

Pcu = Lhscu-[Acu;Ccu]*Rhscu(1:n, :);
Pcu = Pcu(:, 1:2*m*i);
Qcu = Rcu(m*i+1:2*m*i, 1:2*m*i);

L1cu = Acu*gama_invcu;
L2cu = Ccu*gama_invcu;

Mcu = [zeros(n,l), gamaiml_invcu];
Kcu = [eye(l), zeros(l,n); zeros(l*(i-1), l), gamaimlcu];

pkrcu = 0;
for k=1:i
    Ncu = [[Mcu(:, (k-1)*l+1:l*i)-L1cu(:, (k-1)*l+1:l*i), zeros(n, (k-1)*l)]
           [-L2cu(:, (k-1)*l+1:l*i), zeros(l, (k-1)*l)]];
    if k == 1;
        Ncu(n+1:n+1, 1:l)=eye(l)+Ncu(n+1:n+1, 1:l);
    end
    Ncu = Ncu*Kcu;
    pkrcu = pkrcu+kron(Qcu((k-1)*m+1:k*m, :)', Ncu);
end

Pcu = Pcu(:);
solcu = pkrcu\Pcu;

sol_bdcu = reshape(solcu, (n+1), m);

```

```

Dcut = sol_bdcu(1+1:1+n,:);

%se calcula la matriz para el producto kronecker

[nb,mb] = size(Dcut);
result = zeros(nb,nb);

for ik=1:nb
    Dcu(ik,ik) = Dcut(ik);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                               Paso 10 del algoritmo de la sección 4.3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xst4 = zeros(n,ny);
yest4 = zeros(m,ny);

for ipe4 = 1:ny-1
    yest4(:,ipe4) = Ccu * xst4(:,ipe4);
    xst4(:,ipe4+1) = Acu * xst4(:,ipe4) + Dcu *
(kron(er2(:,ipe4+1),xst2l(:,ipe4+1)));
end

%Por último se suma la estimación que se obtuvo de en la primera capa y
%la que se obtuvo en la segunda capa cuadrática

yestf = yest + yest4;

%Se obtiene R1cu para medir el desempeño de la estimación en la capa
%cuadrática

ve1cu = y(1);
ve2cu = yestf(1);

vercu = (abs(ve1cu - ve2cu)).^2;
ver2cu = (abs(ve1cu)).^2;

vef1cu = cumsum(vercu);
vef2cu = cumsum(ver2cu);

R1cu = (1 - (vef1cu/vef2cu))*100

```

Estimation of the Zero Moment Point through stochastic identification

D. Gomez Rivera, G. Etcheverry, A. D. Palomino Merino, M. A D. Vargas Treviño.

Abstract: In this paper a state space model of the human gait is estimated by using subspace based identification; to achieve this, the Zero Moment Point (ZMP) concept is used based on the cart table model. In order to implement this model, we realize an experimental setup which detects the Center of Mass (COM) motion of a person. The experimental setup use a triaxial accelerometer to measure the acceleration. The stochastic identification method is employed for the system recovery.

Keywords: ZMP, stochastic identification, subspace

Resumen: En este artículo se utiliza la identificación de sistemas basada en subespacios para obtener un modelo en el espacio de estados de la marcha humana; para esto, se utiliza el concepto del Zero Moment Point, el cual se obtiene mediante el modelo del cart table. Para implementar este modelo, se desarrolló una plataforma experimental que detecta el movimiento del centro de masa de una persona, este dispositivo está conformado por un acelerómetro triaxial, usado para detectar la aceleración real. Para la identificación del sistema en espacio de estados se utiliza la identificación estocástica.

Palabras Clave: ZMP, identificación estocástica, subespacio

I. INTRODUCCIÓN

El concepto de zero moment point (ZMP por sus siglas en inglés) fue presentado por Vukobratovic en el periodo entre 1969 a 1973. Este concepto se aplica principalmente para la creación de algoritmos que implementen una marcha artificial en robots humanoides.

La tarea más importante de un cuerpo en la marcha es conservar el balance dinámico, este balance se consigue cuando toda la superficie del pie está en contacto con el suelo, ya que de esta forma, el único contacto se realiza vía la fuerza de fricción y la componente vertical de la fuerza de reacción con el suelo.

Para realizar el análisis del caso que se da cuando un cuerpo está apoyado sobre un pie, se desprecia la parte superior del pie y en su lugar se considera una fuerza F_A y un momento M_A , el peso del pie actúa como su centro de gravedad. El pie también experimenta la fuerza de reacción del suelo en algún punto de la planta, esto es lo que mantiene todo el cuerpo en equilibrio.

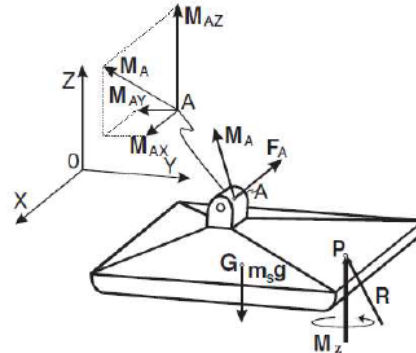


Fig. 1 Fuerzas que actúan en el pie [1]

En general, la fuerza de reacción con el suelo consiste de tres componentes de una fuerza denominada $R(R_x, R_y, R_z)$ y un momento $M(M_x, M_y, M_z)$. Ya que la fuerza de fricción actúa en el punto de contacto del pie con el suelo, las componentes de la fuerza R y el momento M que actúan en el plano horizontal serán balanceadas por la fricción, por lo tanto la fuerza de reacción en el plano horizontal (R_x, R_y) representa la fuerza de fricción que está equilibrando la componente horizontal de la fuerza F_A , mientras que la componente vertical del momento M_z representa la fuerza que balancea el momento inducido por la fuerza F_A . En consecuencia, si se asume que en el contacto del pie con el suelo éste no se puede resbalar, entonces la fricción compensará las componentes (R_x, R_y) y la reacción vertical (M_z) . La componente vertical R_z representa la reacción del suelo que equilibra las fuerzas verticales. Las componentes de los momentos se compensan únicamente al cambiar la posición de la fuerza de reacción R dentro del polígono de soporte. Por lo tanto, la componente horizontal del momento M_A cambiará la fuerza de reacción a la posición correspondiente. La fuerza de reacción siempre se encuentra dentro del área cubierta por el

Daniel Gomez Rivera, Benemérita Universidad Autónoma de Puebla, Av. San Claudio y 18 sur, Ciudad Universitaria, Puebla, México. (e-mail: danielgomez@ece.buap.mx).

Amparo D. Palomino Merino, Benemérita Universidad Autónoma de Puebla, Av. San Claudio y 18 sur, Ciudad Universitaria, Puebla, México. (e-mail: palomino@ece.buap.mx).

Gibrán Etcheverry, Universidad de las Américas Puebla, Sta. Catarina Mártir, Cholula, Puebla, México. (e-mail: gibran.etccheverry@udlap.mx).

Maria Aurora D. Vargas Treviño, Benemérita Universidad Autónoma de Puebla, Av. San Claudio y 18 sur, Ciudad Universitaria, Puebla, México. (e-mail: mavargas@ece.buap.mx).



THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS

CONSTANCIA

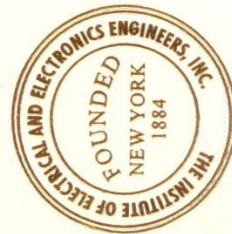
Artículo: “*Estimation of the Zero Moment Point through stochastic identification*”

Autores: Daniel Gomez, Gibran Etcheverry, Amparo Palomino and Aurora Vargas

Id. artículo: 50

El Comité Técnico del XIII Congreso Internacional Sobre Innovación y Desarrollo Tecnológico CIINDET 2016, que se llevó a cabo en la Ciudad de Cuernavaca, Morelos, México, del 7 al 9 de septiembre de 2016, hace constar que el artículo citado fue presentado de acuerdo con el programa técnico del congreso e incluido en las memorias del mismo.

La presente constancia se expide para los fines legales que a los autores convengan.
Cuernavaca, Morelos, México a 9 de septiembre de 2016.



Atentamente

Dr. Jorge Guiferrero Calderón Guizar
Presidente del Comité Técnico CIINDET 2016

Chaotic Analysis on Human Gait Time-Series Signals

D. Gomez Rivera, A. D. Palomino Merino, M. A.D. Vargas Treviño, G. Etcheverry

Abstract— Human Gait analysis is an important subject given its application to the study of pathologies of the human locomotor system. The study of the chaotic behavior of this complex system can help to understand in deep the variability of the human gait patterns. This work explains how to develop an acquisition and analysis tool in a twofold manner: first, a simple and practical setup is implemented in order to achieve the measurement of a person Center of Mass (CoM) when walking; second, an improved method for estimating Lyapunov exponents is described in order to analyze the recorded time-series chaotic behavior.

Index Terms—Chaotic Time-Series, Human Gait, Lyapunov Exponents.

I. INTRODUCTION

The dynamics of the human gait has been studied in order to establish its chaotic behavior based on simple nonlinear time-series analysis methods [1]. This analysis is performed in order to study early diagnose common gait pathologies through unconstrained slow, normal, and fast paces [2, 3].

The importance of having an acquisition system for measuring the human gait becomes important at elder people when walking needs assistance for falling injury avoidance [4]. Therefore in this work, a simple and straightforward wireless setup is developed in order to log data into a personal computer.

A common manner to analyze human gait nonlinear and chaotic behavior is through Lyapunov exponents computation. An improved and simple method developed by K. Briggs is hereby implemented to show that the lateral walking movement is responsible for the aforementioned chaotic pattern [5].

II. LATERAL WALKING TIME-SERIES AND PHASE PLOT

The lateral time-series $x_t, x_{t+1}, \dots, x_{t+i}$ are responsible for 38-40% of gait cycle (swing) [2], which is shown in Fig. 1. This time-series was obtained from experimental data from an accelerometer unit placed at a person waist in order to obtain the Center of Mass (CoM) and therefore, capture the human

gait system dynamic behavior. The finding of a positive Lyapunov coefficient means that the system is chaotic; see Fig. 2 where the phase plot shows a moderated chaotic behavior [5].

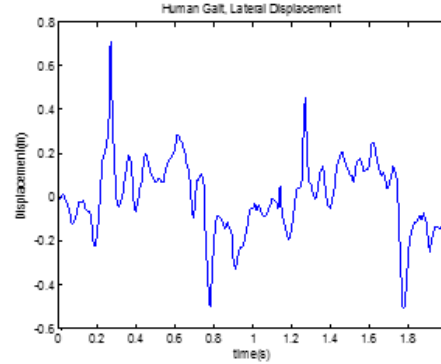


Fig. 1. Gait cycle

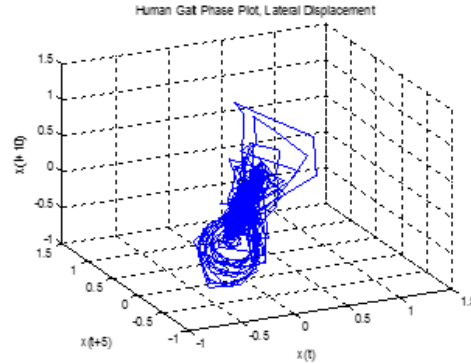


Fig. 2. Human gait phase plot

III. THE LYAPUNOV EXPONENT

In order to compute the Lyapunov exponent, two points are considered within the phase plane: X and X_f , such as [2]:

$$\frac{\|X_f\|}{\|X\|} = e^{\lambda_i t} \quad (1)$$

In this manner, λ_i is obtained from expression:

$$\lambda_i = \lim_{t \rightarrow \infty} \left(\frac{1}{t} \log \frac{\|X_f\|}{\|X\|} \right) \quad (2)$$

Manuscript received April 25, 2016; revised.
 Daniel Gomez Rivera, Benemérita Universidad Autónoma de Puebla, Av. San Claudio y 18 sur, Ciudad Universitaria, Puebla, México. (e-mail: danielgomez@ece.buap.mx).
 Amparo D. Palomino Merino, Benemérita Universidad Autónoma de Puebla, Av. San Claudio y 18 sur, Ciudad Universitaria, Puebla, México. (e-mail: palomino@ece.buap.mx).
 María Aurora D. Vargas Treviño, Benemérita Universidad Autónoma de Puebla, Av. San Claudio y 18 sur, Ciudad Universitaria, Puebla, México. (e-mail: mavargas@ece.buap.mx).
 Gibran Etcheverry, Universidad de las Américas Puebla, Sta. Catarina Mártir, Cholula, Puebla, México. (e-mail: gibran.etccherry@udlap.mx).

ICCSSL 2016

CERTIFICATE

of

PARTICIPATION

Paper Title: Chaotic Analysis on Human Gait Time-Series Signals (S016)

This is to certify Daniel Gomez Rivera from Benemerita Universidad Autonoma de Puebla, Mexico has attended 2016 6th International Conference on Circuits, System and Simulation(ICCSSL 2016) will be held in Mexico City during August 16-18, 2016.

***Conference Committee
ICCSSL 2016***

Aug.16-18,2016, Mexico

Referencias

- [1] J. H. Ramírez Cortes, “Identificación de Sistemas en Representación de Espacio de Estados” (tesis de pregrado). Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Morelos, México. 2007.
- [2] Development of Waseda Robot, <http://www.humanoid.waseda.ac.jp/>, fecha de consulta 29/09/2016.
- [3] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, H. Hirokawa, “Biped Walking Pattern Generation by using Preview Control of Zero Moment Point”, International Conference on Robotics and Automation, vol. 2, pp 1620-1626, 2003.
- [4] OpenHRP3 Official Site, <http://fkanehiro.github.io/openhrp3-doc/en/index.html>, fecha de consulta 29/09/2016
- [5] C. Yang, & T. Murakami, “Walking-Assist Principle Analysis for a Multi-Legged System,” IEEJ Journal of Industry Applications, vol. 4, no. 3, pp. 294–300, 2015.
- [6] T.T. Ngo, et al., The largest inertial sensor-based gait data base and performance evaluation of gait-based personal authentication Pattern Recognition (2013), <http://dx.doi.org/10.1016/j.patcog.2013.06.028>
- [7] W. Suleiman, F. Kanehiro, K. Miura and E. Yoshida, "Improving ZMP based Control Using System Identification Techniques", Paris, France, International Conference on Humanoid Robots, Article number 5379600, pp. 74-80, 2009.
- [8] S. Kajita, F. Kanehiro, K. Kaneko, K. Yok and H. Hirukawa, "The 3D Linear Inverted Pendulum Mode A simple modeling for a biped walking pattern generation", International Conference on Intelligent Robots and Systems, vol. 1, pp. 239-240, 2001.
- [9] R. Ferreira, N. Shafii, N. Lau, L. P. Reis, A. Abdolmaleki, “Diagonal Walk Reference Generator based on Fourier Approximation of ZMP Trajectory”, Article number 6623534 2013.
- [10] Y. D. Hong, C. S. Park, and J. H. Kim, “Stable Bipedal Walking With a Vertical Center-of-Mass Motion by an Evolutionary and Optimized Central Pattern Generator”, IEEE Transactions on Industrial Electronics, VOL. 61, 2014.
- [11] C. A. Cortes Garzon, “Identificación de Sistemas Lineales y No-Lineales Multivariables por Espacio de Estados” (tesis de pregrado), Universidad Tecnológica de Pereira, Pereira, 2013.
- [12] M. Vukobratovic and B. Borovac, "Zero-Moment Point |Thirty Five Years of its Life," International Journal of Humanoid Robotics, vol. 1, 2004.
- [13] Kajita, Shuuji., Hirukawa, Hirohisa., Harada, Kensuke., Yokoi, Kazuhito. Introduction to humanoid robotics, Springer, 2005.

- [14] ASIMO, official site, <http://asimo.honda.com/>, fecha de consulta 29/09/2016
- [15] Christopher L Vaughan, Biran L Davis, Jeremy C O'Connor, Dynamics on Human Gait, Kiboho Publishers, 1999.
- [16] M.H.P. Dekker, "Zero Moment Point Method for Stable Biped Walking", University of Technology, Eindhoven, 2009.
- [17] Freescale Semiconductor, MMA7361L technical data
- [18] Guangzhou HC Information Technology, Module HC-06 datasheet
- [19] Juan Vignolo Barchiesi, "Introducción al Procesamiento Digital de Señales", Ediciones Universitarias de Valparaíso, 2008.
- [20] E. B. Lopez Montero, "Identificación de Sistemas en Lazo Cerrado con Estructura tipo CLOE", (tesis de pregrado), Centro Nacional de Investigación y Desarrollo Tecnológico, Cuernavaca, Morelos, México, 2007.
- [21] K. De Cock, B. De Moor, K.U.Leuven, "Subspace Identification Methods", Department of Electrical Engineering (ESAT – SCD), Leuven, Belgium.
- [22] G. Welch & G. Bishop, "An Introduction to the Kalman Filter", University of North Carolina, 2001.
- [23] D. Di Ruscio, "Subspace System Identification - Theory and applications", Telemark University College, Porsgrunn, Norway, 6th edition, December 11, 2009.
- [24] Tohru Katayama, "Subspace Methods for System Identification", Springer, Kyoto, Japan, 2005.
- [25] V. C. Klema & A. J. Laub, "The Singular Value Decomposition: its Computation and Some Applications", IEEE Transactions on Automatic Control, Vol AC-25, No 2, 1980.
- [26] W. F. Arnold & A. J. Laub, "Generalized Eigen Problem Algorithms and Software for Algebraic Riccati Equations", Proceedings of the IEEE, vol. 72, No 12, 1984.
- [27] P. Van Overschee and B. De Moore, "Subspace Identification for Linear Systems Theory - Implementation - Applications", Katholieke University Leuven, Belgium, 1996.
- [28] G. Etcheverry, W. Suleiman, A. Monin, "Quadratic System Identification by Hereditary Approach", ICASSP, 2006.
- [29] M. Perc, "The dynamics of human gait," Eur. J. Phys., vol. 26, pp. 525-534, 2005.
- [30] A. Goshvarpour & A. Goshvarpour, "Nonlinear Analysis of Human Gait Signals", I. J. Information Engineering and Electronic Business, vol. 2, pp. 15-21, 2012.
- [31] S. Iqbal, "Nonlinear Time-Series Analysis of Different Human Walking Gaits" IEEE International Conference on Electro/Information Technology (EIT), pp. 25-30, 2015.

[32] S. Sato, M. Sano & Y. Sawada, "Practical Methods of Measuring the Generalized Dimension and the Largest Lyapunov Exponent in High Dimensional Chaotic Systems" Prog. Theor. Phys. Vol 77. 1987.