

Proyecto Fin de Carrera

Ingeniería de Tecnologías Industriales

Modelado y Estudio Cinemático y Dinámico de un Waveboard

Autor: Eduardo Arroyo Martín

Tutor: Daniel García Vallejo

Dpto. de Ingeniería Mecánica y Fabricación
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2018



Proyecto Fin de Carrera
Ingeniería de Tecnologías Industriales

Modelado y Estudio Cinemático y Dinámico de un Waveboard

Autor:

Eduardo Arroyo Martín

Tutor:

Daniel García Vallejo

Profesor titular

Dpto. de Ingeniería Mecánica y Fabricación

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2018

Proyecto Fin de Carrera: Modelado y Estudio Cinemático y Dinámico de un Waveboard

Autor: Eduardo Arroyo Martín

Tutor: Daniel García Vallejo

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal

A mi familia
A mis maestros

Agradecimientos

Después de un intenso período de siete meses, hoy es el día: escribo este apartado de agradecimientos para finalizar mi trabajo de fin de grado. Ha sido un período de aprendizaje intenso, no solo en el campo científico, sino también a nivel personal. Escribir este trabajo ha tenido un gran impacto en mí y es por eso que me gustaría agradecer a todas aquellas personas que me han ayudado y apoyado durante este proceso.

Eduardo Arroyo Martín

Sevilla, 2018

Debido al avance tecnológico, se han creado nuevas variantes de los que conocemos por monopatines, como son los longboards, waveboards, etc. Estos últimos, los waveboards (llamados así por su similitud en el movimiento con el surf) son un nuevo tipo de skateboards que constan de dos plataformas pivotables unidas por una barra flexible con una rueda de capacidad de giro 360° cada una.

El funcionamiento de este aparato es muy interesante, ya que para avanzar hay que hacerlo en forma de “eses” transformando el movimiento lateral en inercia hacia adelante, siendo el uso del impulso mucho menos necesario que en otros monopatines. Por lo que sería interesante estudiar la simulación cinemática y dinámica de este mecanismo, para entender mejor su funcionamiento.

Agradecimientos	v
Resumen	vi
Índice	vii
Índice de Figuras	viii
Notación	x
1 Introducción: El Waveboard	1
2 Herramientas Matemáticas	3
2.1. <i>Cinemática del Sólido Rígido</i>	3
2.2. <i>Ecuaciones de Newton-Euler</i>	4
2.3. <i>Ecuaciones de Movimiento</i>	5
3 Modelado del Waveboard	6
3.1. <i>Cinemática del Waveboard</i>	6
3.2. <i>Ecuaciones de movimiento del Waveboard</i>	11
4 Simulación Cinemática y Dinámica	14
4.1. <i>Simulación Cinemática y Dinámica Inversa</i>	14
4.1.1. <i>Caso 1 de Dinámica Inversa</i>	15
4.1.2. <i>Caso 2 de Dinámica Inversa</i>	20
4.1.3. <i>Caso 3 de Dinámica Inversa</i>	21
4.1.4. <i>Caso 4 de Dinámica Inversa</i>	21
4.1.5. <i>Caso 5 de Dinámica Inversa</i>	24
4.2. <i>Dinámica Directa</i>	27
4.2.1. <i>Caso 1 de Dinámica Directa</i>	27
4.2.2. <i>Caso 2 de Dinámica Directa</i>	28
5 Conclusiones	30
Bibliografía	31
Anexo I. Códigos de Matlab	32

ÍNDICE DE FIGURAS

Figura 1-1. El waveboard	1
Figura 3-1.El waveboard y sus sólidos	6
Figura 3-2.Sistema de Referencia Local del Sólido 2	7
Figura 3-3.Representación del giro θ_{int}	7
Figura 3-4.Representación del giro θ_{32}	8
Figura 3-5.Sistema de Referencia Local del Sólido 3	8
Figura 3-6.Representación del giro θ_{43}	9
Figura 3-7.Sistema de Referencia Local del Sólido 4	9
Figura 3-8.Representación del giro θ_{52}	10
Figura 3-9.Distance entre los sólidos del waveboard 1	10
Figura 3-10.Distance entre los sólidos del waveboard 2	10
Figura 3-11.Representación del giro ξ	11
Figura 4-1. x_{21} en Caso 1,2 y 3	16
Figura 4-2. \dot{x}_{21} en Caso 1,2 y 3	16
Figura 4-3. \ddot{x}_{21} en Caso 1,2 y 3	17
Figura 4-4. θ_{52} en Caso 1,2 y 3	17
Figura 4-5. z_{21} en Caso 1,2 y 3	17
Figura 4-6. θ_{32} en Caso 1,2 y 3	18
Figura 4-7. F_{2x} en Caso 1	18
Figura 4-8. F_{2y} en Caso 1	19
Figura 4-9. M_{2x} en Caso 1	19
Figura 4-10. F_{5y} en Caso 1	19
Figura 4-11. M_{5x} en Caso 1	20
Figura 4-12. M_{2x} en Caso 1 y 2	20
Figura 4-13. M_{5x} en Caso 1 y 2	20
Figura 4-14. M_{2x} en Caso 1,2 y 3	21
Figura 4-15. M_{5x} en Caso 1,2 y 3	21
Figura 4-16. z_{21} en Caso 1,2,3 y 4	22
Figura 4-17. θ_{32} en Caso 1,2,3 y 4	22
Figura 4-18. F_{2x} en Caso 1 y 4	22
Figura 4-19. F_{2y} en Caso 1 y 4	23
Figura 4-20. M_{2x} en Caso 1,2,3 y 4	23
Figura 4-21. F_{5y} en Caso 1 y 4	23
Figura 4-22. M_{5x} en Caso 4	24

Figura 4-23. $M5x$ en Caso 1,2,3 y 4	24
Figura 4-24. z_{21} en Caso 1,2,3,4 y 5	25
Figura 4-25. θ_{32} en Caso 1,2,3,4 y 5	25
Figura 4-26. $F2x$ en Caso 1,4 y 5	25
Figura 4-27. $F2y$ en Caso 1,4 y 5	26
Figura 4-28. $M2x$ en Caso 1,2,3,4 y 5	26
Figura 4-29. $F5y$ en Caso 1,4 y 5	26
Figura 4-30. $M5x$ en Caso 5	27
Figura 4-31. $M5x$ en Caso 1,2,3,4 y 5	27
Figura 4-32. x_{21} en Caso 1 de Dinamica Directa	28
Figura 4-33. z_{21} en Caso 1 de Dinamica Directa	28
Figura 4-34. x_{21} en Caso 1 y 2 de Dinamica Directa	29
Figura 4-35. z_{21} en Caso 1 y 2 de Dinamica Directa	29

Notación

cos	Función coseno
sen	Función seno

1 INTRODUCCIÓN:EL WAVEBOARD

A lo largo de los años se ha estudiado mucho el funcionamiento de la bicicleta, mejorando su mecanismo y optimizándolo para mejorar su rendimiento. El mecanismo de la bicicleta es uno de los primeros que se aprenden en el ámbito de cinemática y dinámica de máquinas, pero hay otros mecanismos de los que no hay muchos estudios, cómo por ejemplo el del waveboard. Por lo que sería interesante estudiar este mecanismo, tomando como referencia uno de los muchos estudios de simulación de la dinámica de la bicicleta que hay en internet [1].

El mecanismo del waveboard es muy poco conocido, encontrándose muy poca información acerca de este, sólo algunas indicaciones sobre medidas y materiales en la página web del fabricante [2]. El waveboard se caracteriza por tener dos plataformas de plástico unidas por una barra que permite un movimiento de torsión entre las plataformas en el eje longitudinal del “wave”. Cada plataforma tiene una rueda, similar a la de los patines en línea, colocada sobre una estructura giratoria (de rueda loca). Esta estructura se encuentra orientada 30° hacia el tail o cola, tanto en la plataforma delantera como en la trasera. La inclinación de estas estructuras hace que este tipo de tablas sean direccionales, solamente avancen en un sentido.

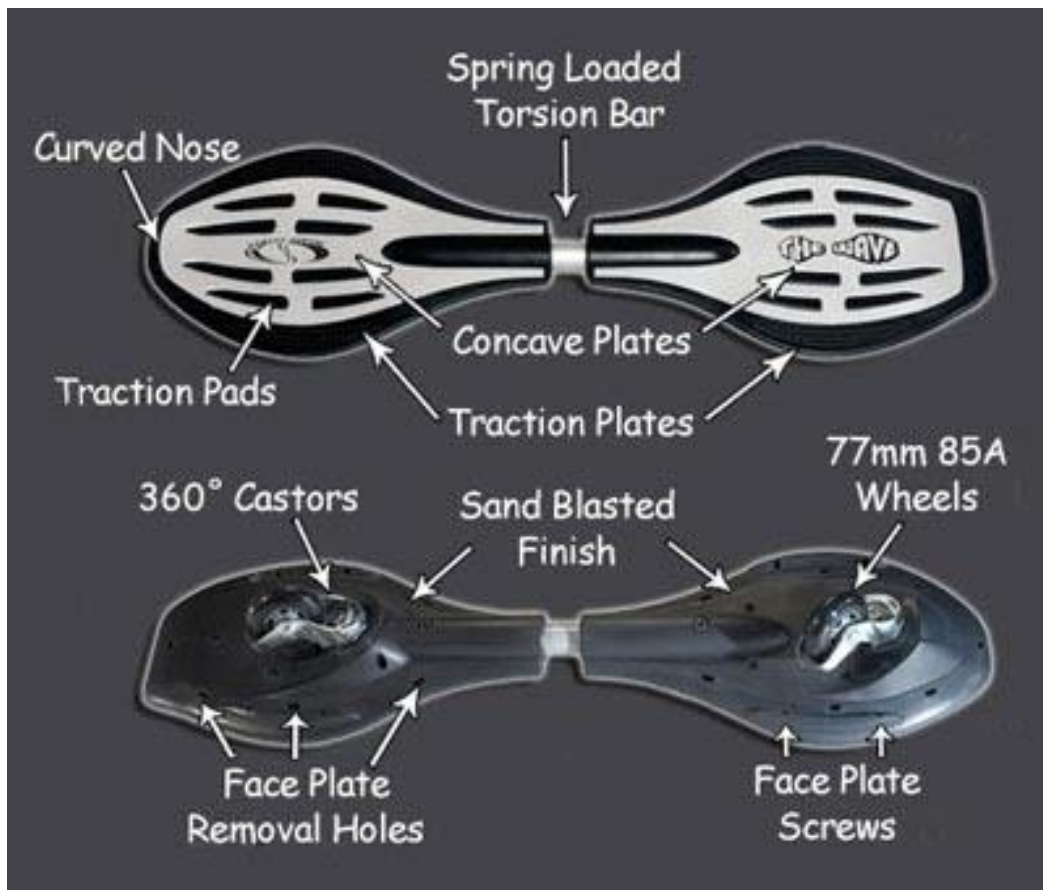


Figura 1-1.El waveboard

El funcionamiento de este tipo de monopatín es muy diferente a los otros monopatines que sólo pueden ganar velocidad mediante impulso o por desnivel del terreno, con un waveboard sólo se necesitará el impulso inicial y a continuación mediante un movimiento en forma de “eses”, en el que los dos pies tienen que mover cada una de las plataformas en dirección perpendicular al avance del waveboard, se consigue transformar la energía cinética transversal al movimiento en impulso hacia adelante.

Es curioso conocer el comportamiento de un aparato utilizado para el ocio, pero del que poca gente sabe su complejo funcionamiento. El propósito de este proyecto es la simulación cinemática y dinámica de este

mecanismo bajo acciones que un humano tendría que hacer para el correcto movimiento de este. Se implementará un programa en MATLAB para modelar el mecanismo mediante herramientas matemáticas, para posteriormente integrar sus ecuaciones de movimiento y así obtener el valor de todos sus grados de libertad a lo largo del tiempo. Con esto conseguiremos comprobar el grado de importancia de todos los parámetros físicos que intervienen en el movimiento del mecanismo.

2 HERRAMIENTAS MATEMÁTICAS

2.1 Cinemática Del Sólido Rígido

Para el modelado del waveboard, se necesitarán una serie de herramientas matemáticas, del ámbito de cinemática y dinámica de máquinas [3], que ayuden a conseguir las ecuaciones de movimiento del mecanismo (Ecuaciones de Newton-Euler + ecuaciones de restricción).

El primer paso para conseguir estas ecuaciones es entender la cinemática del sólido rígido, conociendo la orientación de este en el espacio y sus respectivos parámetros cinemáticos.

Para conocer el vector posición de un punto del mecanismo en coordenadas globales de referencia utilizaremos la siguiente relación:

$$\text{Coordenadas: } q^i = [r_x^i \ r_y^i \ r_z^i \ \alpha^i \beta^i \gamma^i] = [r^i \alpha^i \beta^i \gamma^i]$$

$$\text{Posición de P: } r_p^i = r^i + A^i \bar{u}_p^i$$

Siendo la matriz $A^i = A_{\alpha^i} A_{\beta^i} A_{\gamma^i}$, y siendo estas las matrices de giro de cada uno de los giros asociados al sólido i . El vector \bar{u}_p^i es la distancia entre el origen y el punto a definir en coordenadas locales del sólido i .

Así como se pueden obtener los vectores posición de cualquier punto del sólido, derivando las relaciones antes vistas, podremos obtener los vectores velocidad y aceleración de cualquier punto del sólido.

Otra manera de calcular las velocidades y aceleraciones de un punto del sólido rígido, es a través del jacobiano de su vector posición:

$$V_p^i = H_p^i \dot{q}^i$$

$$a_p^i = H_p^i \ddot{q}^i + h_p^i \dot{q}^i$$

La matriz H_p^i es el jacobiano de la posición, es decir las derivadas parciales de las tres componentes de la posición respecto de todas las coordenadas del sistema, y la matriz h_p^i es la derivada respecto al tiempo de la matriz H_p^i . Estas matrices tienen dimensiones de $3 \times n$, siendo n el número de coordenadas del mecanismo.

Como ya se ha comentado, cada sólido del mecanismo lleva asociado una matriz de giro que relaciona el sistema de referencia local del mismo con el sistema de referencia global del mecanismo. A partir de estas matrices podemos calcular las velocidades angulares y aceleraciones angulares de cada uno de los sólidos en función de las coordenadas del sistema:

$$\tilde{w}_i = \dot{A}_i A_i^T \quad \tilde{\tilde{w}}_i = A_i^T \dot{A}_i$$

Siendo \tilde{w}_i y $\tilde{\tilde{w}}_i$ las matrices antisimétricas de la velocidad angular del sólido i , en coordenadas globales y coordenadas locales, respectivamente.

$$\tilde{w} = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}$$

Una vez calculadas las velocidades angulares de los sólidos, podemos calcular las matrices \bar{G} de los sólidos (muy importantes para la posterior formación de las ecuaciones de Newton-Euler), y a partir de la derivada temporal de estas, se calcularán las aceleraciones angulares de los sólidos.

$\bar{w}_i = \begin{bmatrix} \bar{w}_{x_i} \\ \bar{w}_{y_i} \\ \bar{w}_{z_i} \end{bmatrix} = \bar{G}_i \dot{q}$, siendo \bar{G}_i el jacobiano de la velocidad angular en locales respecto a las coordenadas del sistema derivadas en el tiempo: $\bar{G}_i = \frac{\partial \bar{w}_i}{\partial \dot{q}}$

Se calcularán las aceleraciones angulares de la siguiente manera:

$\bar{\alpha}_i = \bar{G}_i \ddot{q} + \bar{g}_i \dot{q}$, siendo \bar{g}_i la derivada temporal de $\bar{G}_i \rightarrow \bar{g}_i = \dot{\bar{G}}_i$

Tanto \bar{G}_i , como \bar{g}_i , tendrán dimensiones de $3 \times n$.

2.2 Ecuaciones De Newton-Euler

Una vez comprendida la cinemática del sólido rígido, nos centraremos en la formación de las ecuaciones de Newton-Euler. Estas son ecuaciones vectoriales independientes del sistema de referencia al que se refieren y las cuales representan las ecuaciones de equilibrio del mecanismo en cuestión.

$$\begin{bmatrix} m^i 1 & 0 \\ 0 & \bar{I}^i \end{bmatrix} \begin{bmatrix} a_G^i \\ \bar{\alpha}^i \end{bmatrix} = \begin{bmatrix} F^i \\ \bar{M}^i \end{bmatrix} + \begin{bmatrix} 0 \\ -\bar{w}^i x \bar{I}^i \bar{w}^i \end{bmatrix}, i = 2, 3, 4 \dots nb$$

$$\begin{bmatrix} m^{21} & & & & & \\ & \ddots & & & & \\ & & m^{nb1} & & & \\ & & & \bar{I}^2 & & \\ & & & & \ddots & \\ & & & & & \bar{I}^{nb} \end{bmatrix} = \hat{M} ; \quad \begin{bmatrix} F^2 \\ \vdots \\ F^{nb} \\ \bar{M}^2 \\ \vdots \\ \bar{M}^{nb} \end{bmatrix} = \hat{Q} ; \quad \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -\bar{w}^2 x \bar{I}^2 \bar{w}^2 \\ \vdots \\ -\bar{w}^{nb} x \bar{I}^{nb} \bar{w}^{nb} \end{bmatrix} = \hat{Q}_v$$

$$\hat{M} \begin{bmatrix} a_G^2 \\ \vdots \\ a_G^{nb} \\ \alpha_G^2 \\ \vdots \\ \alpha_G^{nb} \end{bmatrix} = \hat{Q} + \hat{Q}_v$$

La matriz \hat{M} contiene las matrices de masas y de tensores de inercia locales para cada sólido, por lo que sus dimensiones son de $6nb \times 6nb$, siendo nb el número de sólidos del mecanismo. Esta es multiplicada por el vector que contiene la aceleración del centro de gravedad y la aceleración angular, según el sistema de referencia local, para cada sólido del mecanismo. En el otro miembro de la igualdad tenemos la matriz \hat{Q} , que representa las fuerzas aplicadas y de reacción de cada sólido según el sistema de referencia global y los momentos aplicados y de reacción de cada sólido según los sistemas de referencias locales, por lo que tendrá una dimensión de $6nb \times 1$. También nos encontramos el término de coriolis, con otros momentos aplicados localmente a cada sólido del mecanismo.

Aunque estas ecuaciones determinan la dinámica de los sólidos que forman una máquina, se quiere expresar estas ecuaciones en función de las coordenadas del mecanismo. Para ello se utilizarán las ecuaciones cinemáticas con jacobianos que permiten escribir las ecuaciones de Newton-Euler en términos de las coordenadas del mecanismo y sus derivadas. Con esto conseguiremos que estas ecuaciones dependan de en qué posición se encuentre el mecanismo.

Entonces, las aceleraciones de los sólidos se calcularían de la siguiente manera:

$$\begin{bmatrix} a_G^2 \\ \vdots \\ a_G^{nb} \\ \bar{\alpha}^2 \\ \vdots \\ \bar{\alpha}^{nb} \end{bmatrix} = \begin{bmatrix} H^2 \\ \vdots \\ H^{nb} \\ \bar{G}^2 \\ \vdots \\ \bar{G}^{nb} \end{bmatrix} \ddot{q} + \begin{bmatrix} h^2 \\ \vdots \\ h^{nb} \\ \bar{g}^2 \\ \vdots \\ \bar{g}^{nb} \end{bmatrix} \dot{q} = L\ddot{q} + l\dot{q}$$

Por lo que los términos de las ecuaciones de Newton-Euler, se multiplicarán por las matrices L y l para que

estos dependan de las coordenadas del sistema. La nueva estructura de estas ecuaciones será:

$$M\ddot{q} = Q + Q_v$$

$$M = L^T \widehat{M} L \quad ; \quad Q = L^T \widehat{Q} \quad ; \quad Q_v = L^T (\widehat{Q}_v - \widehat{M} l \dot{q})$$

Las dimensiones de las matrices L y l son de 6nbXn. Por lo que la matriz M será de dimensiones nXn y las matrices Q y Q_v de dimensiones nX1.

2.3 Ecuaciones De Movimiento

Las ecuaciones de equilibrio tienen que ser aumentadas con las ecuaciones de restricción para obtener las ecuaciones de movimiento.

Cada mecanismo en movimiento conlleva una serie de ecuaciones de restricción, con una fuerza de reacción asociada a cada una de ellas. Las ecuaciones de restricción pueden ser holónomas o no-holónomas, según dependan de las derivadas de las coordenadas o no. Como en el waveboard vamos a requerir de ambos tipos de restricciones, se explicará el caso de las ecuaciones de movimiento en el que tengamos los dos tipos de restricciones.

$$\left. \begin{array}{l} C(q, t) = 0 \\ C(q, \dot{q}, t) = B(q)\dot{q} = 0 \end{array} \right\} \rightarrow \begin{bmatrix} C_q \\ B \end{bmatrix} \dot{q} + \begin{bmatrix} C_t \\ 0 \end{bmatrix} = 0 \rightarrow D\dot{q} + E = 0$$

$$\begin{bmatrix} M & D^T \\ D & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} Q_{apl} + Q_v \\ -\dot{D}\dot{q} - \dot{E} \end{bmatrix}$$

La matriz D tiene unas dimensiones de mXn, siendo m el número de ecuaciones de restricción tanto holónomas como no-holónomas, y la matriz E tendrá unas dimensiones de mX1.

La matriz Q_{apl} ahora solo representan las fuerzas aplicadas a cada sólido, ya que las fuerzas de reacción vienen representadas por la multiplicación de D^T y λ. La matriz λ son los multiplicadores de Lagrange, que representan la fuerza que hay que ejercer para que se cumpla esa ecuación de restricción pero en sentido contrario, es decir λ = -F, siendo F la fuerza comentada anteriormente.

Si integráramos estas ecuaciones a lo largo de un tiempo determinado, obtendríamos las coordenadas del mecanismo y las fuerzas de reacción a lo largo de este tiempo. Pero experimentalmente se ha observado que al ser una integración de segundo grado, se crean ciertos errores en la integración que provocan que se dejen de cumplir las ecuaciones de restricción. Este problema se puede solucionar mediante el método de Baumgarte[4], en el que habría que añadir dos términos mas a las ecuaciones de restricción, estando estos dos términos multiplicados cada uno por un coeficiente adimensional de valor ajustable para evitar los errores en la integración. Por lo que las ecuaciones de movimiento quedarían de la siguiente manera:

$$\begin{bmatrix} M & D^T \\ D & 0 \end{bmatrix} \begin{bmatrix} \ddot{q} \\ \lambda \end{bmatrix} = \begin{bmatrix} Q_{apl} + Q_v \\ -\dot{D}\dot{q} - \dot{E} - 2\alpha \begin{bmatrix} \dot{C}(q, t) \\ C(q, \dot{q}, t) \end{bmatrix} - \beta^2 \begin{bmatrix} C(q, t) \\ 0 \end{bmatrix} \end{bmatrix}$$

Siendo α y β los coeficientes adimensionales que se variarán según los requisitos de la integración.

3 MODELADO DEL WAVEBOARD

En este apartado se aplicarán las herramientas matemáticas antes vistas al mecanismo del waveboard, para obtener sus ecuaciones de movimiento.

3.1 Cinemática del waveboard

Para modelar este mecanismo, primero lo orientaremos y lo modelaremos en el aire, para posteriormente mediante ecuaciones de restricción situarlo en el suelo. Nuestro mecanismo va a constar de 6 sólidos, los sólidos 2 y 5 serán las dos plataformas pivotables, los sólidos 3 y 6 serán las estructuras giratorias de cada rueda y los sólidos 4 y 7 serán cada una de las ruedas.

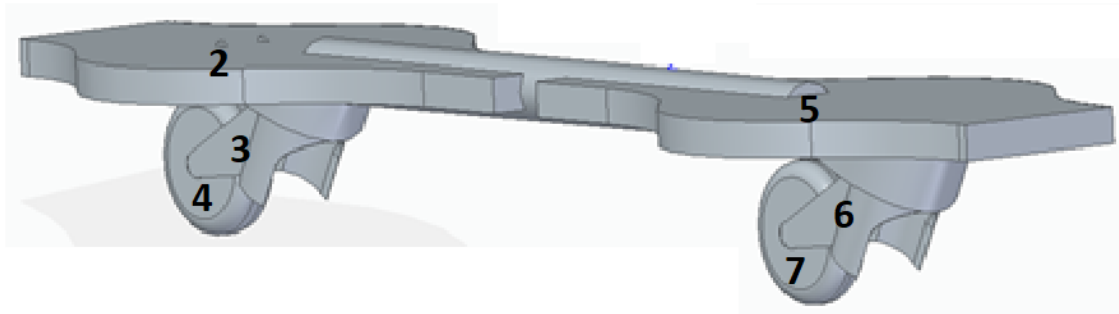


Figura 3-1. El waveboard y sus sólidos

Para orientar el mecanismo en el espacio, se requerirán las 3 coordenadas del centro de gravedad del sólido 2 (x_{21} , y_{21} y z_{21}) y 3 giros (α , β y γ) con los que orientaremos el sólido 2 en el aire. A partir de la posición del sólido 2 se obtendrán las posiciones del centro de gravedad del resto de sólidos mediante giros de los sistemas de referencia (θ_{32} , θ_{43} , θ_{52} , θ_{65} y θ_{76}). Todos los ángulos

Los 3 giros de orientación del sólido 2 serán ángulos de Euler, es decir, el giro α será respecto al eje z, el giro β será respecto al eje x y el giro γ será también respecto al eje z:

$$A_\alpha = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\text{sen}(\alpha) \\ 0 & \text{sen}(\alpha) & \cos(\alpha) \end{bmatrix} \quad A_\beta = \begin{bmatrix} \cos(\beta) & 0 & \text{sen}(\beta) \\ 0 & 1 & 0 \\ -\text{sen}(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad A_\gamma = \begin{bmatrix} \cos(\gamma) & -\text{sen}(\gamma) & 0 \\ \text{sen}(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

La matriz de giro del sólido 2 sería $A_{21} = A_\alpha A_\beta A_\gamma$.

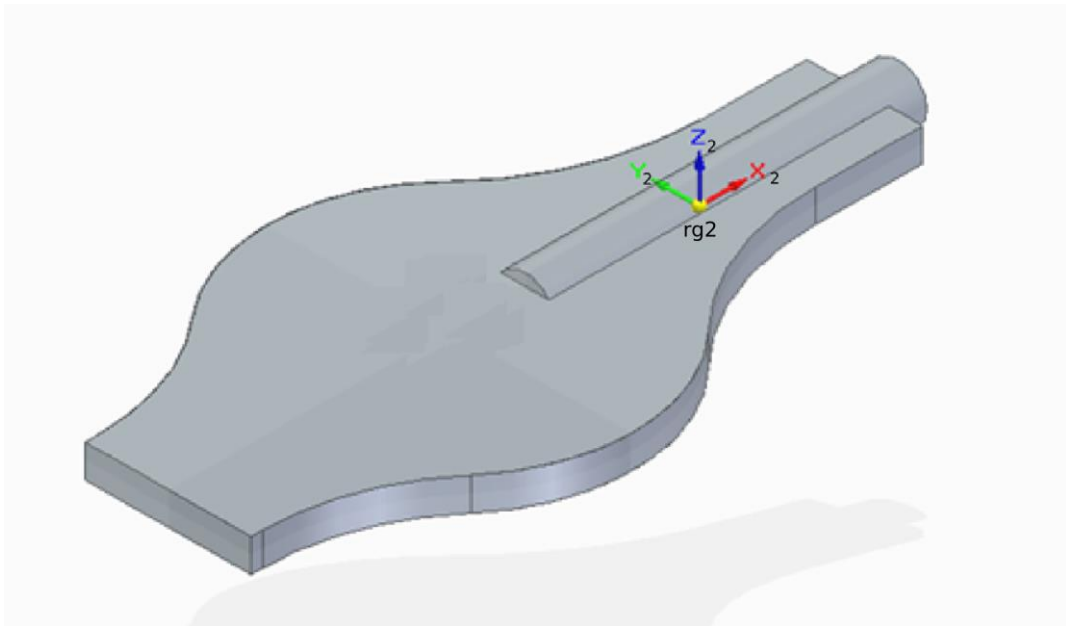


Figura 3-2.Sistema de Referencia Local del Sólido 2

A continuación, para llegar al sistema de referencia local del sólido 3 tendremos la matriz de giro del θ_{32} que es el que otorga al mecanismo el sistema de “rueda loca”, pero antes se necesitará una matriz de giro constante para tener en cuenta los 30° de inclinación de la estructura giratoria respecto a la plataforma (sólido 2). La matriz de giro constante será respecto al eje y_2 y la matriz del giro θ_{32} respecto al eje z_{int} .

$$A_{int} = \begin{bmatrix} \cos\left(\frac{\pi}{3}\right) & 0 & \text{sen}\left(\frac{\pi}{3}\right) \\ 0 & 1 & 0 \\ -\text{sen}\left(\frac{\pi}{3}\right) & 0 & \cos\left(\frac{\pi}{3}\right) \end{bmatrix}$$

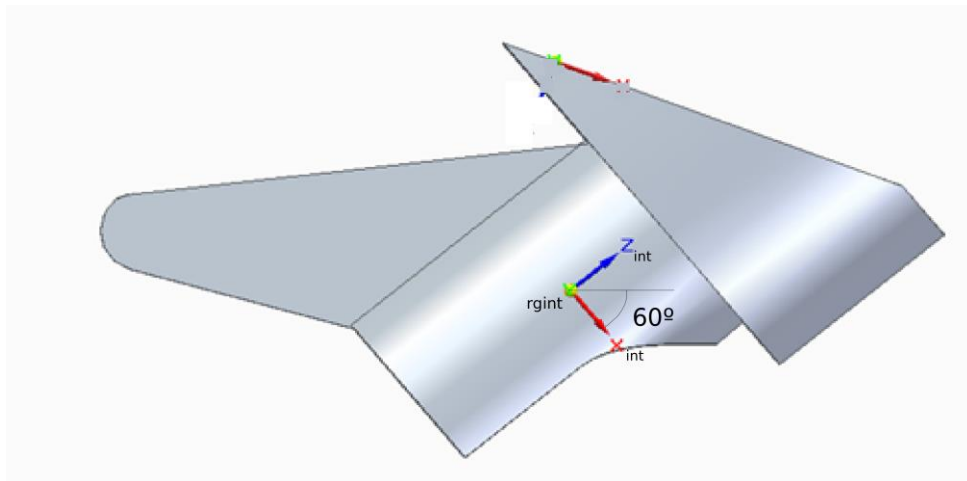


Figura 3-3.Representación del giro θ_{int}

$$A_{\theta_{32}} = \begin{bmatrix} \cos(\theta_{32}) & -\text{sen}(\theta_{32}) & 0 \\ \text{sen}(\theta_{32}) & \cos(\theta_{32}) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

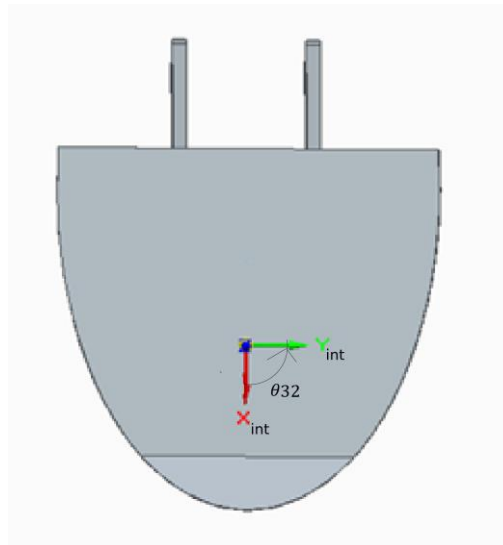


Figura 3-4.Representación del giro θ_{32}

La matriz de giro del sólido 3 será $A_{31} = A_{21}A_{int}A_{\theta_{32}}$.

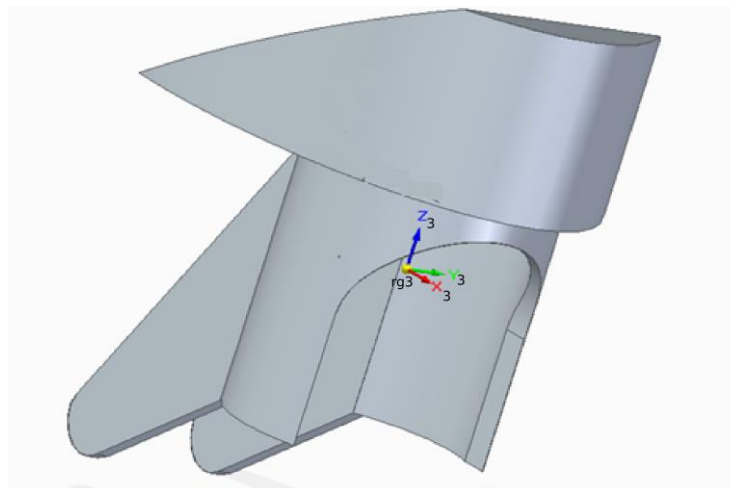


Figura 3-5.Sistema de Referencia Local del Sólido 3

Por último para obtener el sistema de referencia local del sólido 4 se realizará el giro θ_{43} respecto al eje y_3 que determinará el ángulo girado por la rueda.

$$A_{\theta_{43}} = \begin{bmatrix} \cos(\theta_{43}) & 0 & \text{sen}(\theta_{43}) \\ 0 & 1 & 0 \\ -\text{sen}(\theta_{43}) & 0 & \cos(\theta_{43}) \end{bmatrix}$$

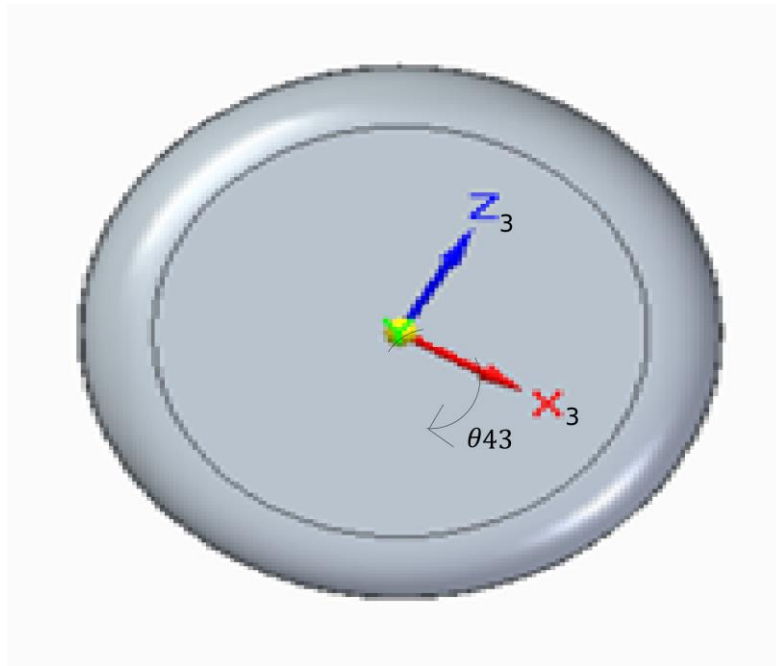


Figura 3-6.Representación del giro θ_{43}

La matriz de giro del sólido 4 será $A_{41} = A_{31}A_{\theta_{43}}$.

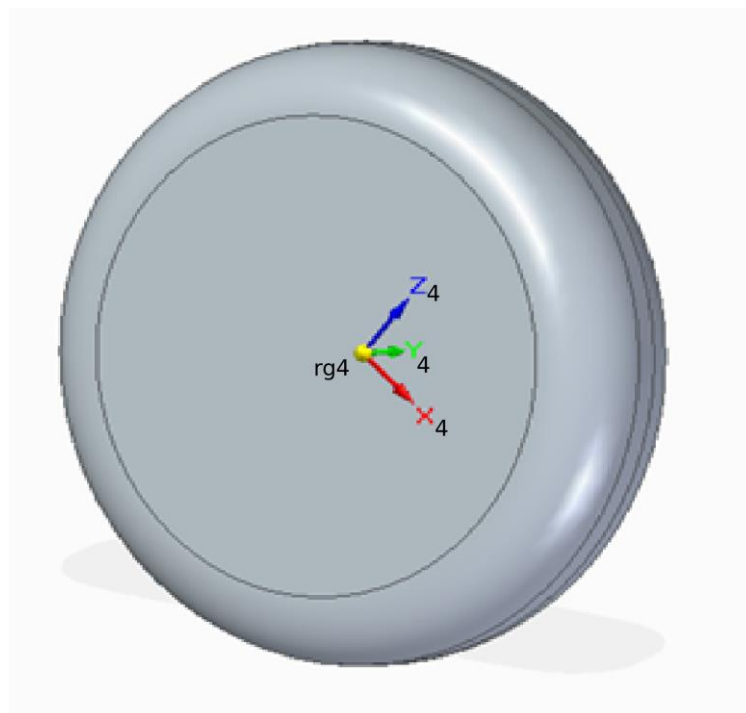


Figura 3-7.Sistema de Referencia Local del Sólido 4

Para pasar a la otra plataforma partiremos desde el sólido 2 teniendo en cuenta el movimiento torsor entre ambas θ_{52} , giro respecto al eje x_2 . Los sólidos 6 y 7 conllevan el mismo procedimiento que los sólidos 3 y 4 pero empezando desde su plataforma, el sólido 5.

$$A_{\theta_{52}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_{52}) & -\text{sen}(\theta_{52}) \\ 0 & \text{sen}(\theta_{52}) & \cos(\theta_{52}) \end{bmatrix}$$

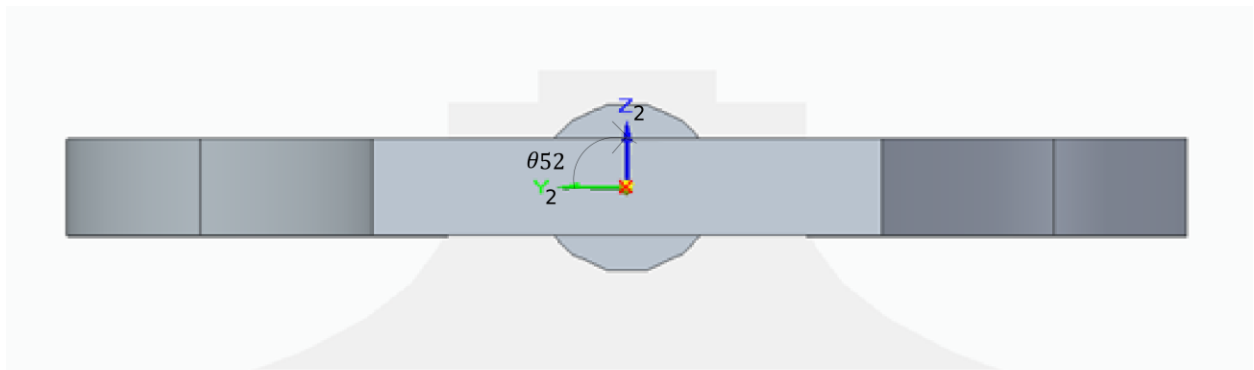


Figura 3-8.Representación del giro θ_{52}

$$A_{51} = A_{21}A_{\theta_{52}} ; \quad A_{61} = A_{51}A_{int}A_{\theta_{65}} ; \quad A_{71} = A_{61}A_{\theta_{76}}$$

Una vez tenemos las matrices de giro de cada uno de los sólidos se calcularán las velocidades angulares locales, y a través de estas las matrices \bar{G} y \bar{g} de cada sólido mediante las herramientas matemáticas vistas en el apartado anterior.

Como se quiere obtener las matrices L y l del mecanismo para llegar a las ecuaciones de movimiento, se calcularán los vectores posición de cada uno de los sólidos, para posteriormente mediante jacobianos obtener las matrices H y h de estos.

$$r_{21} = \begin{bmatrix} x_{21} \\ y_{21} \\ z_{21} \end{bmatrix} ; \quad r_{31} = r_{21} + A_{21} \begin{bmatrix} -0,07 \\ 0 \\ -0,05 \end{bmatrix} ; \quad r_{51} = r_{21} + A_{21} \begin{bmatrix} 0,37 \\ 0 \\ 0 \end{bmatrix} ; \quad r_{61} = r_{51} + A_{51} \begin{bmatrix} 0,1 \\ 0 \\ -0,05 \end{bmatrix}$$

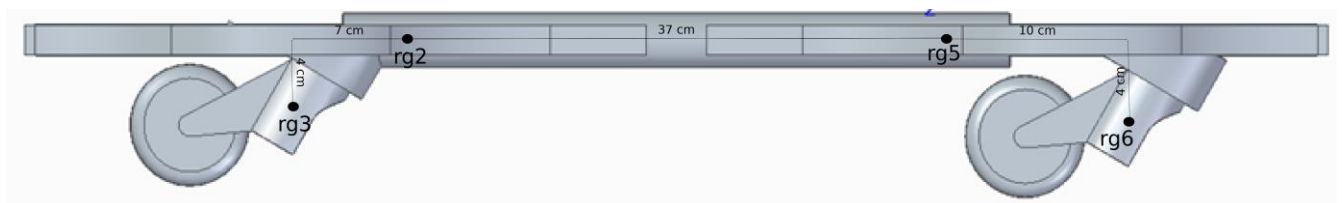


Figura 3-9.Distancia entre los sólidos del waveboard 1

$$r_{41} = r_{31} + A_{31} \begin{bmatrix} -0,04 \\ 0 \\ -0,05 \end{bmatrix} ; \quad r_{71} = r_{61} + A_{61} \begin{bmatrix} -0,04 \\ 0 \\ -0,05 \end{bmatrix}$$

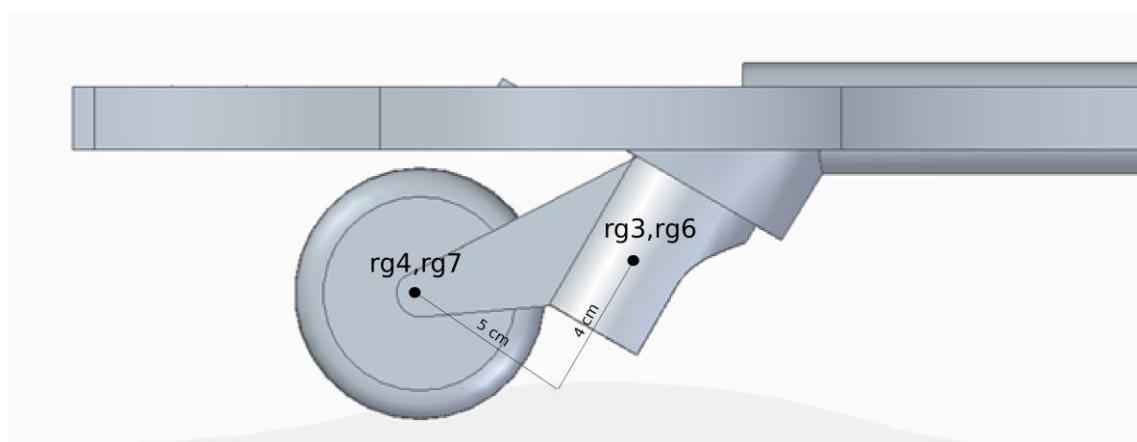


Figura 3-10.Distancia entre los sólidos del waveboard 2

3.2 Ecuaciones de movimiento del waveboard

La matriz de masa de los sólidos \hat{M} será de dimensiones 36x36 ya que tenemos 6 sólidos. Para la formación de esta se necesitarán las masas de cada uno de los cuerpos ($m_2 = m_5 = 1,143 \text{ Kg}$; $m_3 = m_6 = 0,252 \text{ Kg}$; $m_4 = m_7 = 0,105 \text{ Kg}$) así como sus tensores de inercia locales ($\bar{I}_2 = \bar{I}_5 = \begin{bmatrix} 37 & 0 & 0 \\ 0 & 195,5 & 0 \\ 0 & 0 & 231,2 \end{bmatrix} \cdot 10^{-4}$; $\bar{I}_3 = \bar{I}_6 = \begin{bmatrix} 3,3 & 0 & 0 \\ 0 & 2,8 & 0 \\ 0 & 0 & 1,6 \end{bmatrix} \cdot 10^{-4}$; $\bar{I}_4 = \bar{I}_7 = \begin{bmatrix} 0,5 & 0 & 0 \\ 0 & 0,8 & 0 \\ 0 & 0 & 0,5 \end{bmatrix} \cdot 10^{-4}$).

Se situará el waveboard en el suelo mediante la restricción de igualar a cero la coordenada z del vector posición del que será el punto de contacto de cada una de las ruedas con el suelo. Se corroborará que el waveboard siempre se apoya sobre el suelo diciendo que las tangentes a las ruedas en el punto de contacto con

el suelo y el vector normal $n = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ siempre forman 90° , es decir, que la multiplicación escalar de ambos

vectores sea igual a cero. Las cuatro últimas ecuaciones de restricción son no-holonomas y serían las de rodadura sin deslizamiento de cada una de las ruedas. La componente x e y del vector velocidad del punto de contacto con el suelo de cada una de las ruedas serían iguales a cero.

Se obtendrá el vector posición del punto de contacto con el suelo mediante una coordenada no generalizada, realizando un giro ξ de los sistemas de referencia de las ruedas (ξ_1 para el sólido 4 y ξ_2 para el sólido 7) respecto a sus ejes y (y_4 y y_7 , respectivamente) de tal forma que sus ejes x (x_4 y x_7) siempre apunten al punto de contacto con el suelo (x_{ξ_1} y x_{ξ_2}).

$$A_{\xi_i} = \begin{bmatrix} \cos(\xi_i) & 0 & \text{sen}(\xi_i) \\ 0 & 1 & 0 \\ -\text{sen}(\xi_i) & 0 & \cos(\xi_i) \end{bmatrix} \quad i = 1,2$$

$$r_{P1} = r_{41} + A_{41}A_{\xi_1} \begin{bmatrix} R = 0,04 \\ 0 \\ 0 \end{bmatrix} ; \quad r_{P2} = r_{71} + A_{71}A_{\xi_2} \begin{bmatrix} 0,04 \\ 0 \\ 0 \end{bmatrix}$$

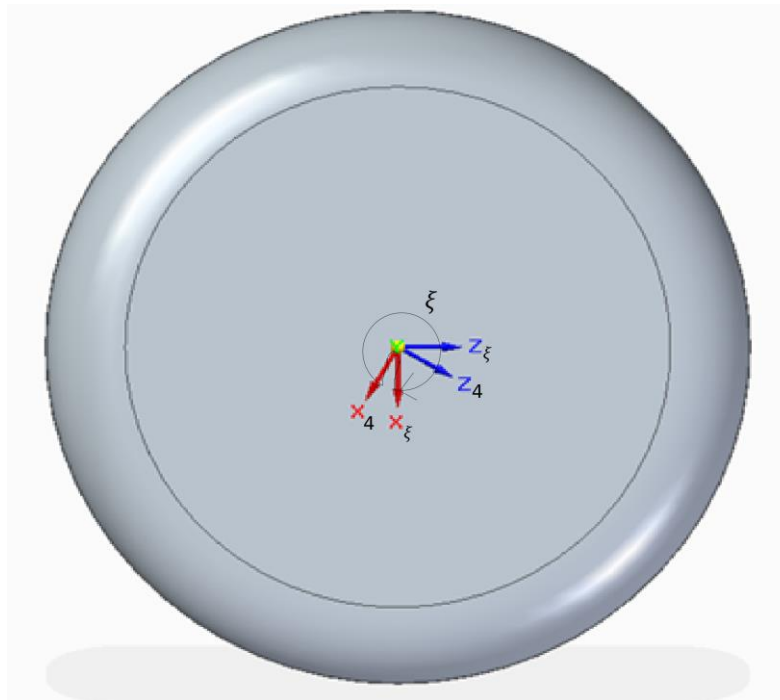


Figura 3-11. Representación del giro ξ

Una vez obtenido el vector posición del punto de contacto se puede calcular la tangente a la rueda en ese punto. La velocidad del punto de contacto se calculará a través de la velocidad del centro de gravedad de las ruedas (obtenidas derivando el vector posición de los sólidos 4 y 7) mediante la ley de composición de velocidades y la coordenada no generalizada.

$$\tan_{P1} = A_{41} \begin{bmatrix} -\text{sen}(\xi_1) \\ 0 \\ -\text{cos}(\xi_1) \end{bmatrix} ; \quad \tan_{P2} = A_{71} \begin{bmatrix} -\text{sen}(\xi_2) \\ 0 \\ -\text{cos}(\xi_2) \end{bmatrix}$$

$$V_{41} = \dot{r}_{41} \quad ; \quad V_{P1} = V_{41} + (A_{41}\bar{w}_{41}) \times \left(A_{41}A_{\xi_1} \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} \right)$$

$$V_{71} = \dot{r}_{71} \quad ; \quad V_{P2} = V_{71} + (A_{71}\bar{w}_{71}) \times \left(A_{71}A_{\xi_2} \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} \right)$$

Finalmente hay 4 ecuaciones de restricción holónomas $C(q, t) = \begin{pmatrix} r_{P1}^z = 0 \\ r_{P2}^z = 0 \\ n \cdot \tan_{P1} = 0 \\ n \cdot \tan_{P2} = 0 \end{pmatrix}$ y 4 ecuaciones de

$$\text{restricción no-holónomas } C(q, \dot{q}, t) = \begin{pmatrix} V_{P1}^x = 0 \\ V_{P1}^y = 0 \\ V_{P2}^x = 0 \\ V_{P2}^y = 0 \end{pmatrix}.$$

Al añadirse estas dos nuevas coordenadas tenemos 13 coordenadas siendo el vector de coordenadas $q = [x_{21} \ y_{21} \ z_{21} \ \alpha \ \beta \ \gamma \ \theta_{52} \ \theta_{32} \ \theta_{65} \ \theta_{43} \ \theta_{76} \ \xi_1 \ \xi_2]$. Por lo que las matrices D y \hat{D} del mecanismo tendrán unas dimensiones de 8×13 , y la matriz E sería nula ya que ninguna de nuestras ecuaciones de restricción son de movilidad.

Respecto a las fuerzas aplicadas a nuestro mecanismo sólo vamos a considerar el peso de cada uno de los cuerpos y la fuerza elástica de un resorte torsional que es el que permite la pivotación de las dos plataformas. Este resorte lo vamos a considerar con una rigidez de k y su elongación la determinará la coordenada θ_{52} . La matriz \hat{Q}_{apl} tendrá unas dimensiones de 36×1 en la que las fuerzas aplicadas estarán en coordenadas globales y los momentos aplicados en coordenadas locales:

$$F_i = \begin{bmatrix} 0 \\ 0 \\ -m_i g \end{bmatrix} \quad i = 2, 3, 4, 5, 6$$

$$M_3 = M_4 = M_6 = M_7 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} ; \quad M_2 = \begin{bmatrix} k\theta_{52} \\ 0 \\ 0 \end{bmatrix} ; \quad M_5 = \begin{bmatrix} -k\theta_{52} \\ 0 \\ 0 \end{bmatrix}$$

$$\hat{Q}_{apl} = [F_2 \ F_3 \ F_4 \ F_5 \ F_6 \ F_7 \ M_2 \ M_3 \ M_4 \ M_5 \ M_6 \ M_7]$$

Para finalizar, el término de Coriolis \hat{Q}_v se calculará mediante los tensores de inercia y la velocidad angular de cada uno de los cuerpos en coordenadas locales teniendo unas dimensiones de 36×1 :

$$F_{vi} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} ; \quad M_{vi} = \begin{bmatrix} 0 \\ 0 \\ -\bar{w}_i x \bar{I}_i \bar{w}_i \end{bmatrix} \quad i = 2, 3, 4, 5, 6, 7$$

$$\hat{Q}_v = [F_{v2} \ F_{v3} \ F_{v4} \ F_{v5} \ F_{v6} \ F_{v7} \ M_{v2} \ M_{v3} \ M_{v4} \ M_{v5} \ M_{v6} \ M_{v7}]$$

Para obtener las ecuaciones en función de las coordenadas del mecanismo se utilizarán las matrices L y l . Como el número de coordenadas es 13 estas matrices tendrán una dimensión de 36×13 .

$$M_{13X13} = L_{13X36}^T \widehat{M}_{36X36} L_{36X13} \quad ; \quad Q_{apl_{13X1}} = L_{13X36}^T \widehat{Q}_{apl_{36X1}}$$

$$Q_{v_{13X1}} = L_{13X36}^T (\widehat{Q}_{v_{36X1}} - \widehat{M}_{36X36} L_{36X13} \dot{q}_{13X1})$$

$$\begin{bmatrix} M_{13X13} & D_{13X8}^T \\ D_{8X13} & 0_{8X8} \end{bmatrix}_{21X21} \begin{bmatrix} \ddot{q}_{13X1} \\ \lambda_{8X1} \end{bmatrix}_{21X1} = Z_{21X1} \quad ; \quad Z_{21X1} = \begin{bmatrix} Q_{apl_{13X1}} + Q_{v_{13X1}} \\ -\dot{D}_{8X13} \dot{q}_{13X1} - 2\alpha \begin{bmatrix} \dot{C}_{h_{4X1}} \\ C_{nh_{4X1}} \end{bmatrix} - \beta^2 \begin{bmatrix} C_{h_{4X1}} \\ 0_{4X1} \end{bmatrix} \end{bmatrix}$$

Una vez obtenidas las ecuaciones de Newton-Euler se procederá a integrarlas para obtener tanto las coordenadas del Sistema como los multiplicadores de Lagrange a lo largo del tiempo.

4 SIMULACIÓN CINEMÁTICA Y DINÁMICA

4.1 Simulación Cinemática y Dinámica Inversa

Se ha comprobado que intentar controlar el waveboard en dinámica directa (ecuaciones de Newton-Euler) mediante unas fuerzas y momentos externos adicionales es imposible, ya que desde que se le aplica cualquier acción al cuerpo este se cae. Lo que se va a hacer es una dinámica inversa generando un movimiento parecido al que se hace cuando se usa el waveboard en la realidad, para poder obtener las acciones exteriores necesarias que permitan controlarlo en lazo abierto. Para esto último es necesario imponer un movimiento realista que permita calcular las acciones que lo producen de manera que estas se metan posteriormente como entrada al problema de dinámica directa para calcular nuevamente nuestras coordenadas a lo largo del tiempo. Se puede decir que lo que se hace con la dinámica inversa es aprender a controlar el waveboard.

Para realizar la dinámica inversa, hay que imponer cómo varían en el tiempo una serie de coordenadas pero no todas, el resto se calculan mediante la simulación cinemática haciendo que se cumplan las ecuaciones de restricción. Para la imposición del movimiento del waveboard se ha tomado como referencia un estudio sobre los diferentes modos de andar del waveboard[5].

En nuestro sistema hay 4 restricciones holónomas y 4 no holónomas que denotamos como sigue

$$C(q) = 0 \xrightarrow{d(\cdot)/dt} C_q(q)\dot{q} = 0 \quad (1)$$

$$C_{nh}(q, \dot{q}) = B(q)\dot{q} = 0 \quad (2)$$

La derivada con respecto al tiempo de las restricciones holónomas puede escribirse también en términos de las coordenadas dependientes e independientes junto con las restricciones no holónomas para poder resolver el problema de velocidad.

$$C_{q_d}(q)\dot{q}_d + C_{q_i}(q)\dot{q}_i = 0 \quad (3)$$

$$B_d(q)\dot{q}_d + B_i(q)\dot{q}_i = 0 \quad (4)$$

Lo que da lugar al siguiente problema

$$\begin{bmatrix} C_{q_d}(q) \\ B_d(q) \end{bmatrix} \dot{q}_d = - \begin{bmatrix} C_{q_i}(q) \\ B_i(q) \end{bmatrix} \dot{q}_i \quad \rightarrow \quad D_d(q)\dot{q}_d = -D_i(q)\dot{q}_i \quad (5)$$

Teniendo en cuenta que el mecanismo tiene 13 coordenadas, para poder hacer una simulación dinámica inversa es necesario imponer el valor de 5 coordenadas. En el instante inicial es necesario resolver el problema de posición a partir de 9 valores de coordenadas, las 5 impuestas y 4 más. Por tanto, conocido $q_0 = q(t_0)$ y las velocidades independientes, se podría resolver el problema de velocidad en el instante inicial usando la ecuación (5). Sustituyendo los valores de las 5 coordenadas impuestas en la ecuación (5) se llega al siguiente sistema de 8 ecuaciones diferenciales de primer orden:

$$\dot{q}_d = -(D_d(q_d, t))^{-1} D_i(q_d, t)\dot{q}_i(t) \quad (6)$$

Este sistema de ecuaciones diferenciales se puede integrar con cualquier algoritmo numérico para sistemas ODE de primer orden. Sin embargo, debido a que las restricciones holónomas se han derivado con respecto al tiempo, es posible que la integración lleve aparejados errores en la satisfacción de las restricciones a nivel de posición. Para evitar esto, puede recurrirse nuevamente al método de estabilización de las restricciones de Baumgarte. Esto consiste en imponer la relación:

$$\dot{C}(q) + \alpha C(q) = 0 \quad (7)$$

donde α es una constante positiva, en lugar de usar solo $\dot{C}(q) = 0$ como se indica en (1). De esta forma el sistema de ecuaciones a integrar sería el siguiente:

$$\dot{q}_d = - \begin{bmatrix} C_{q_d}(q_d, t) \\ B_d(q_d, t) \end{bmatrix}^{-1} \begin{bmatrix} C_{q_i}(q_d, t)\dot{q}_i(t) + \alpha C(q_d, t) \\ B_i(q_d, t)\dot{q}_i(t) \end{bmatrix} \quad (8)$$

con la condición inicial $q_{d0} = q_d(t_0)$.

Para la dinámica inversa la ecuación de movimiento del sistema con restricciones holónomas y no holónomas puede escribirse de forma resumida como sigue

$$M(q)\ddot{q} + D^T(q)\lambda = Q_v(q, \dot{q}) + Q_{apl}(q) + A_\tau(q)\tau \quad (9)$$

donde M es la matriz de masa, Q_v es el vector de fuerzas generalizadas cuadráticas en velocidades (término de Coriolis), Q_{apl} es el vector de fuerzas generalizadas debida al peso de los sólidos y al resorte torsional, y el producto $A_\tau(q)\tau$ es el vector de fuerzas generalizadas resultado de las acciones que actúan sobre el mecanismo, τ . Se incluyeron 5 acciones, tantas como grados de libertad tiene el sistema (13 coordenadas, 8 restricciones). Es importante recalcar que $A_\tau(q)$ puede obtenerse a partir de la matriz jacobiana $L(q)$ para los momentos aplicados, y a partir de la matriz jacobiana $L(q)$ y las matrices de giro correspondientes para las fuerzas aplicadas, ya que en las ecuaciones de movimiento del sistema los momentos se aplican localmente y las fuerzas globalmente.

Una vez realizada la simulación cinemática, integrando el sistema (8), es posible conocer los valores de todas las coordenadas, velocidades y aceleraciones generalizadas en cualquier instante de tiempo $q(t)$, $\dot{q}(t)$ y $\ddot{q}(t)$. Por tanto las 13 ecuaciones del sistema (9) pueden usarse para despejar los valores de 8 multiplicadores y 5 acciones externas como sigue

$$[-D^T(q), A_\tau(q)]_{13 \times 13} \begin{bmatrix} \lambda \\ \tau \end{bmatrix}_{13 \times 1} = [M(q)\ddot{q} - Q_v(q, \dot{q}) - Q_{apl}(q)]_{13 \times 1}$$

Obsérvese que el cálculo de los multiplicadores y de las acciones externas se hace a partir de la resolución de un sistema de ecuaciones lineales del tipo $Ax = b$. Otro detalle importante reside en el signo menos que afecta al término $D^T(q)$. La presencia de este signo hace que los multiplicadores sean iguales a las fuerzas de reacción cambiadas de signo.

A continuación, se realizarán varios casos de simulación variando dos parámetros. Los parámetros que serán variables en nuestra simulación serán la rigidez del muelle torsional (k) y la inclinación de la estructura giratoria, que aunque su valor por defecto son 30° sería interesante probar otros ángulos. Obsérvese que variar la rigidez solo afectará a la dinámica, en cambio variar el ángulo de inclinación afectará tanto a la dinámica como a la cinemática.

Para todos los casos vamos a imponer el movimiento de las coordenadas y_{21} , α , γ , θ_{52} y θ_{43} . Se impondrá movimiento senoidal a todas las coordenadas excepto a θ_{43} que se le impondrá un movimiento a velocidad constante.

De manera que las coordenadas independientes queden $q_i(t) = \begin{bmatrix} y_{21}(t) = 0,01\text{sen}(2\pi t) \\ \alpha(t) = 0,3\text{sen}(2\pi t) \\ \gamma(t) = 0,2\text{sen}(2\pi t) \\ \theta_{52}(t) = 0,4\text{sen}(2\pi t - 1,85) \\ \theta_{43}(t) = 12,5t \end{bmatrix}$.

Para la dinámica inversa se tomarán 3 fuerzas y 2 momentos como fuerzas externas que actúan sobre el mecanismo:

$$\tau = [F2x \quad F2y \quad M2x \quad F5y \quad M5x]$$

Para crear la matriz A_τ , por ejemplo su primera columna corresponde a la fuerza $F2x$ y se calcularía multiplicando las tres primeras filas de la matriz L por la primera columna de la matriz A_{21} , en cambio para la tercera columna correspondiente al momento $M2x$ se obtiene directamente de la fila 19 de la matriz L (primera fila de la matriz \bar{G}_{21}).

4.1.1. Caso 1 de Dinámica Inversa

En los tres primeros casos solo se variará la rigidez, por lo que la cinemática de los tres primeros casos será igual, es decir, el movimiento de los tres primeros casos será el mismo, sólo se diferenciarán en las acciones que hay que hacer para que se produzca ese movimiento

Para este primer caso simularemos con una rigidez de $k = 100 \text{ N/m}$ y con 30° de inclinación de la estructura giratoria. El problema de posición en el instante inicial se resuelve partiendo del valor de las coordenadas impuestas en $t = 0$ más el valor de otras cuatro coordenadas que van a ser $x_{21}(t = 0) = \theta_{32}(t = 0) = \theta_{65}(t = 0) = \theta_{76}(t = 0) = 0$. Las 4 variables restantes (z_{21} , β , ξ_1 y ξ_2) se calculan mediante las 4 ecuaciones de restricción en posición. Con esto ya tenemos nuestras condiciones iniciales para comenzar la integración:

$$q_{a_0} = [x_{21_0} = 0 \quad z_{21_0} = 0,0789 \quad \beta_0 = 0,0109 \quad \theta_{32_0} = 0 \quad \theta_{65_0} = 0 \quad \theta_{76_0} = 0 \quad \xi_{1_0} = 0,5127 \quad \xi_{2_0} = 0,5119]$$

Una vez realizada la integración se obtiene $q(t)$ y mediante la expresión (6) y la derivada de esta expresión se puede calcular $\dot{q}(t)$ y $\ddot{q}(t)$, respectivamente. A continuación se muestran algunas gráficas de varias coordenadas a lo largo del tiempo resultado de la simulación cinemática:

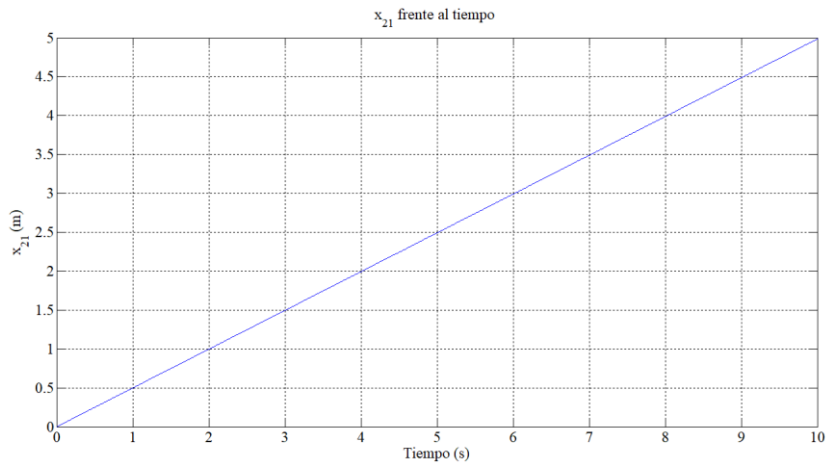


Figura 4-1. x_{21} en Caso 1,2 y 3

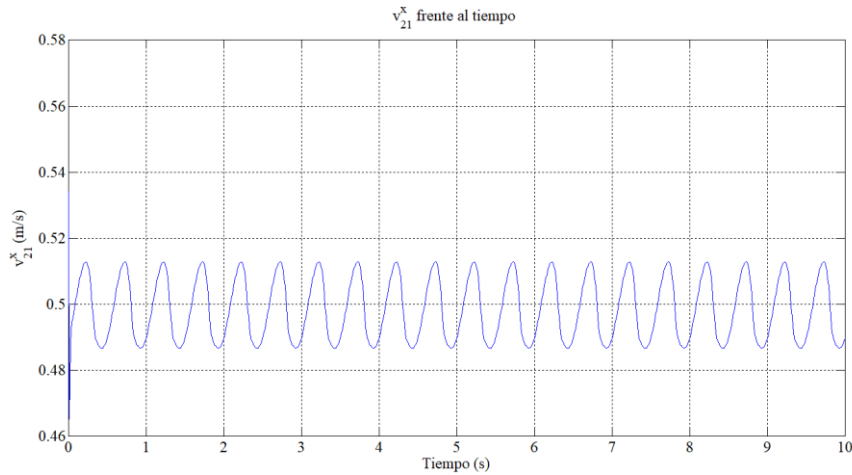


Figura 4-2. \dot{x}_{21} en Caso 1,2 y 3

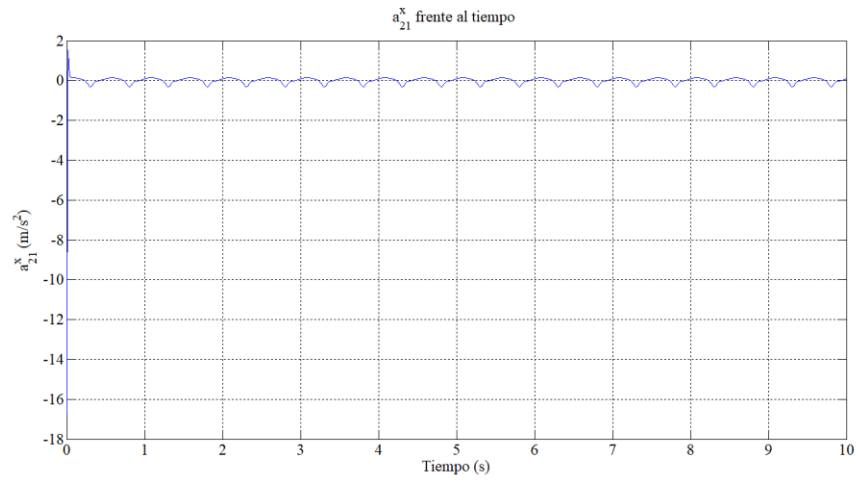


Figura 4-3. \ddot{x}_{21} en Caso 1,2 y 3

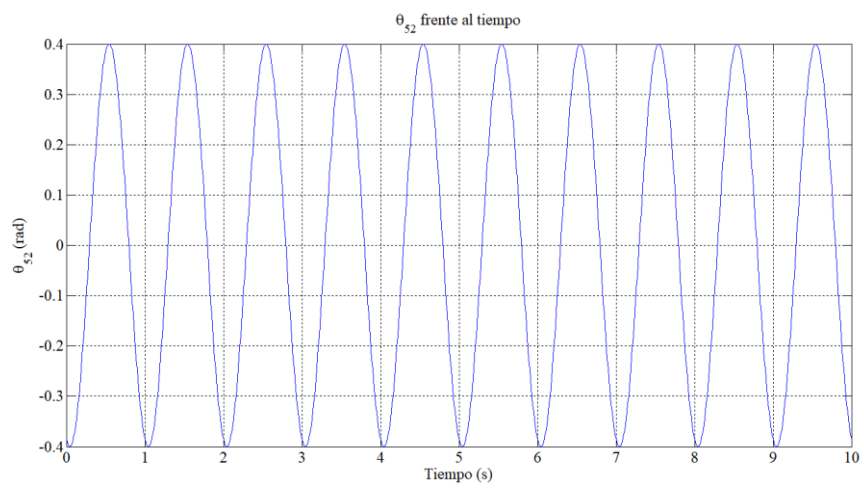


Figura 4-4. θ_{52} en Caso 1,2 y 3

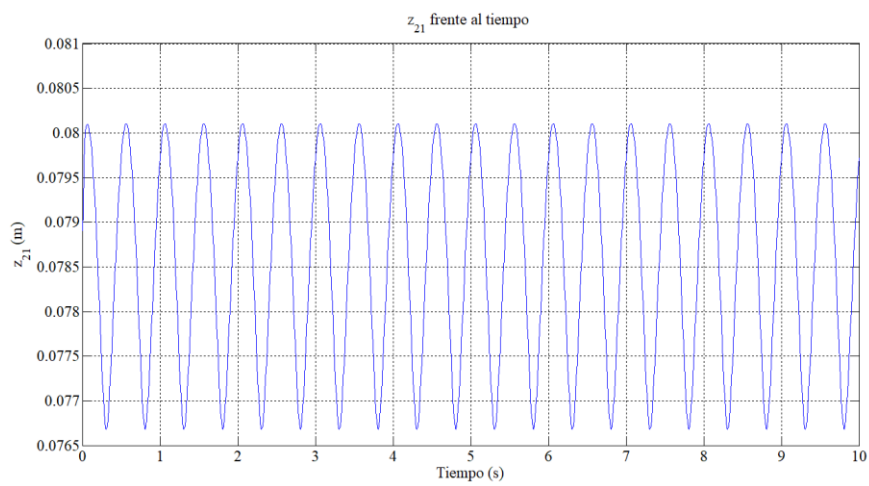


Figura 4-5. z_{21} en Caso 1,2 y 3

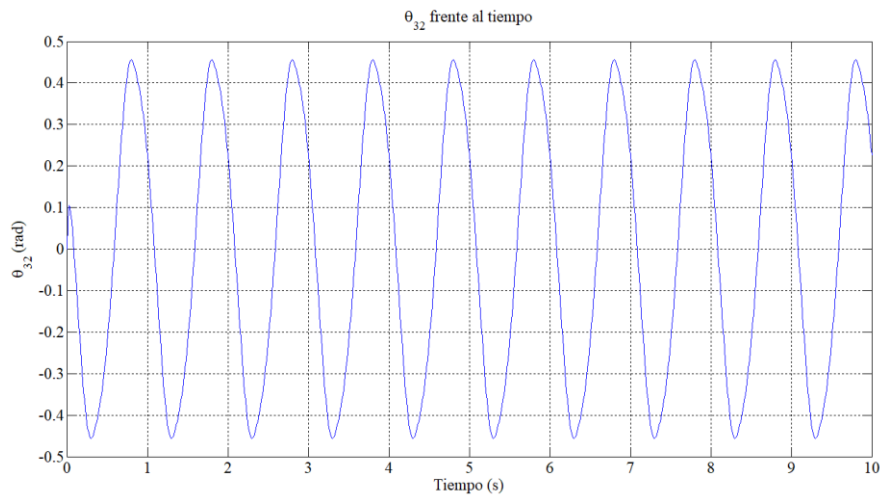


Figura 4-6. θ_{32} en Caso 1,2 y 3

Una vez realizada la dinámica inversa, se puede comprobar si se ha realizado correctamente, por ejemplo comprobando que la suma de los multiplicadores en cualquier instante de tiempo correspondientes a $r_z = 0$ tiene que ser igual al peso del waveboard pero cambiado de signo, ya que como se ha dicho antes los multiplicadores son las fuerzas de reacción cambiadas de signo:

$$\begin{aligned} \lambda_1(t = 7,25) = \lambda_{r_4^z}(t = 7,25) &= -12,1100 & \lambda_2(t = 7,25) = \lambda_{r_7^z}(t = 7,25) &= -17,1746 \\ & & -(m_2 + m_3 + m_4 + m_5 + m_6 + m_7) \cdot 9,81 &= -29,43 \\ & & \lambda_1(t = 7,25) + \lambda_2(t = 7,25) &\sim -29,43 \end{aligned}$$

A continuación se muestran las fuerzas externas a lo largo del tiempo calculadas mediante la dinámica inversa, las cuales son las necesarias para que el movimiento impuesto se realice:

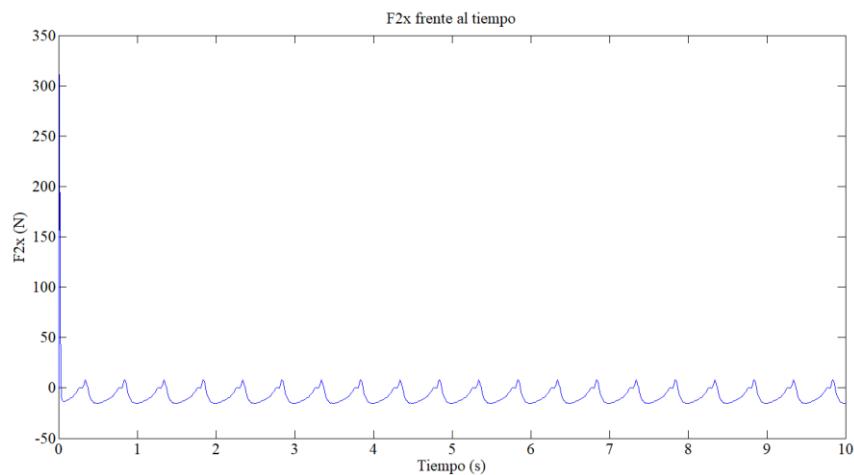


Figura 4-7. F_{2x} en Caso 1

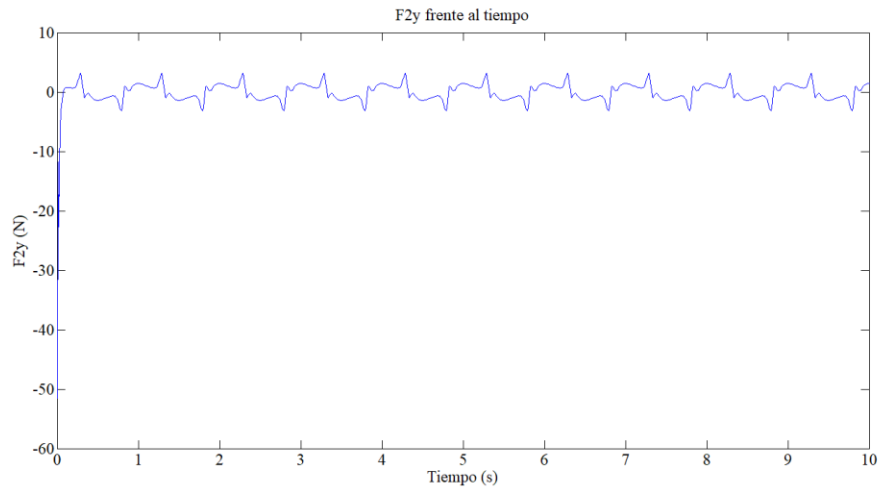


Figura 4-8. F2y en Caso 1

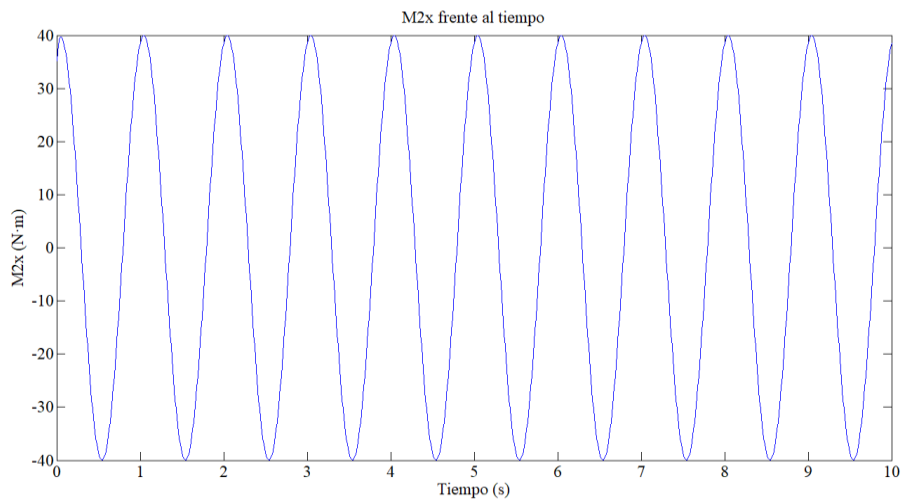


Figura 4-9. M2x en Caso 1

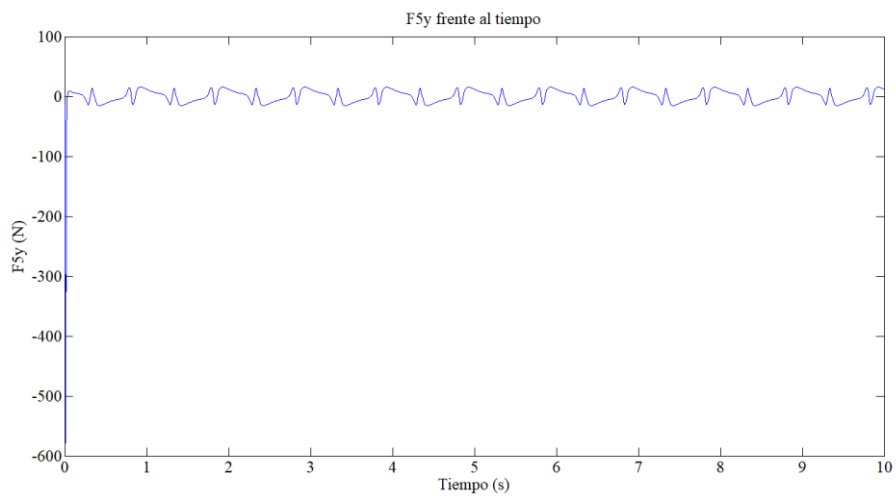


Figura 4-10. F5y en Caso 1

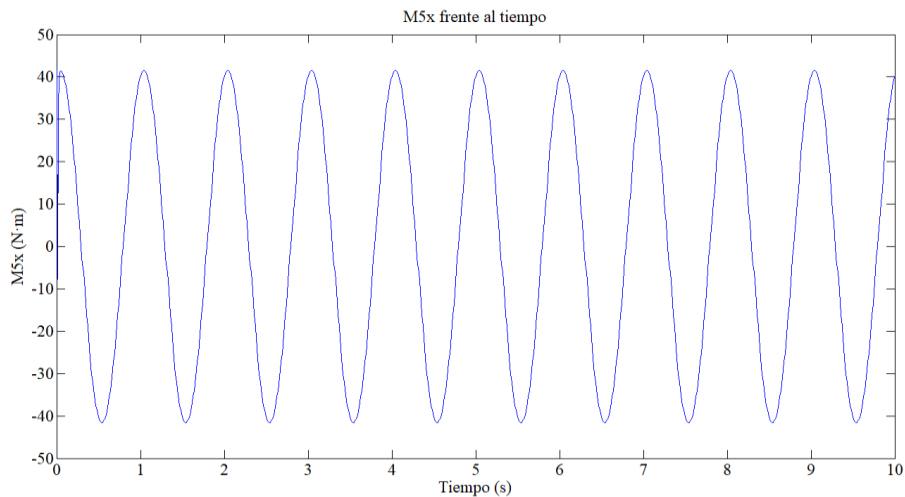


Figura 4-11. M5x en Caso 1

4.1.2. Caso 2 de Dinámica Inversa

Para este segundo caso simularemos con una rigidez de $k = 200 \text{ N/m}$ y con 30° de inclinación de la estructura giratoria. Como solo hemos variado la rigidez del muelle solo se verá afectada la dinámica, por lo que la cinemática de este caso será igual a la del caso anterior. En la dinámica de este caso se aprecian ciertas diferencias con el caso anterior, siendo la amplitud de los momentos $M2x$ y $M5x$ el doble.

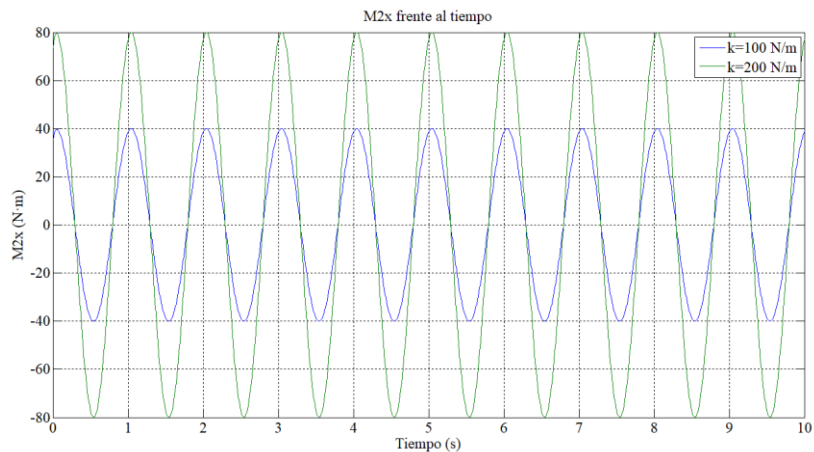


Figura 4-12. M2x en Caso 1 y 2

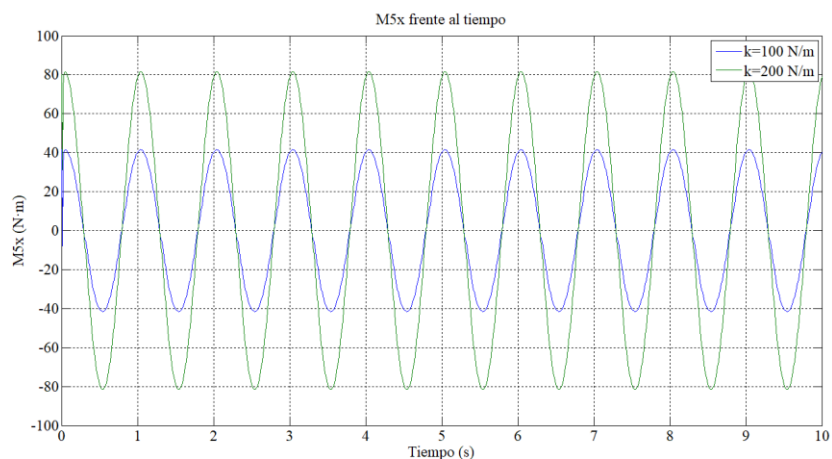


Figura 4-13. M5x en Caso 1 y 2

4.1.3. Caso 3 de Dinámica Inversa

Para este segundo caso simularemos con una rigidez de $k = 400 \text{ N/m}$ y con 30° de inclinación de la estructura giratoria. Como en el caso 2 solo se ven afectadas las amplitudes de los momentos M_{2x} y M_{5x} de la dinámica, siendo la cinemática de este caso también igual a los dos casos anteriores.

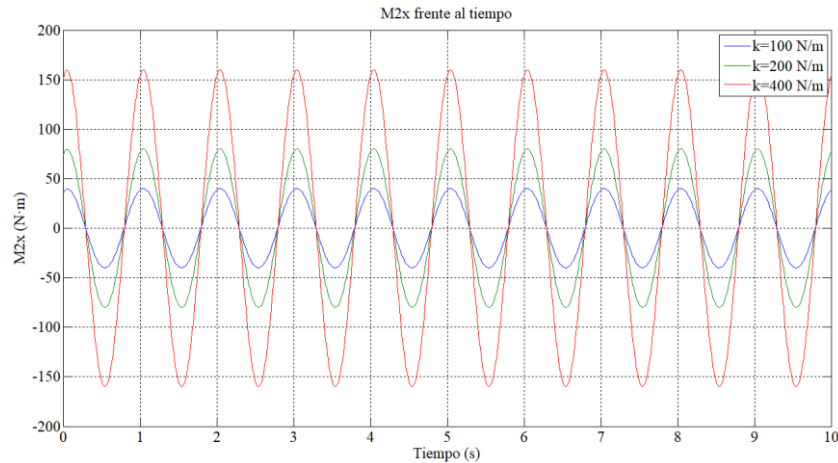


Figura 4-14. M_{2x} en Caso 1,2 y 3

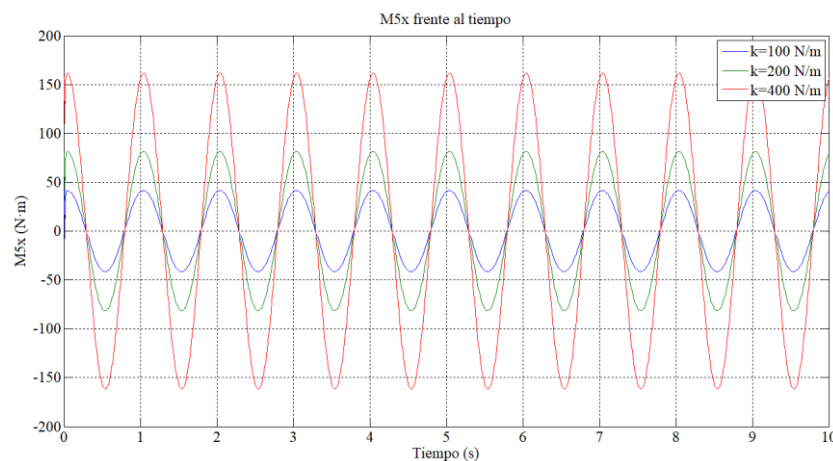


Figura 4-15. M_{5x} en Caso 1,2 y 3

4.1.4. Caso 4 de Dinámica Inversa

En los siguientes dos casos el parámetro que se variará será el ángulo de inclinación, para que se vean afectadas tanto la cinemática como la dinámica. Para este cuarto caso simularemos con una rigidez de $k = 100 \text{ N/m}$ y con 20° de inclinación de la estructura giratoria. Las condiciones iniciales para la integración en este caso serán.

$$q_{d_0} = [x_{21_0} = 0 \quad z_{21_0} = 0,0683 \quad \beta_0 = 0,0094 \quad \theta_{32_0} = 0 \quad \theta_{65_0} = 0 \quad \theta_{76_0} = 0 \quad \xi_{1_0} = 0,3397 \quad \xi_{2_0} = 0,3389]$$

Una vez hecha la simulación con estos parámetros se observan variaciones considerables respecto a los casos anteriores en ciertas coordenadas cinemáticas como por ejemplo en z_{21} y θ_{32} . Podemos observar cómo el punto de equilibrio de z_{21} se hace más pequeño y cómo la amplitud de θ_{32} aumenta disminuyendo el ángulo de inclinación de la estructura.

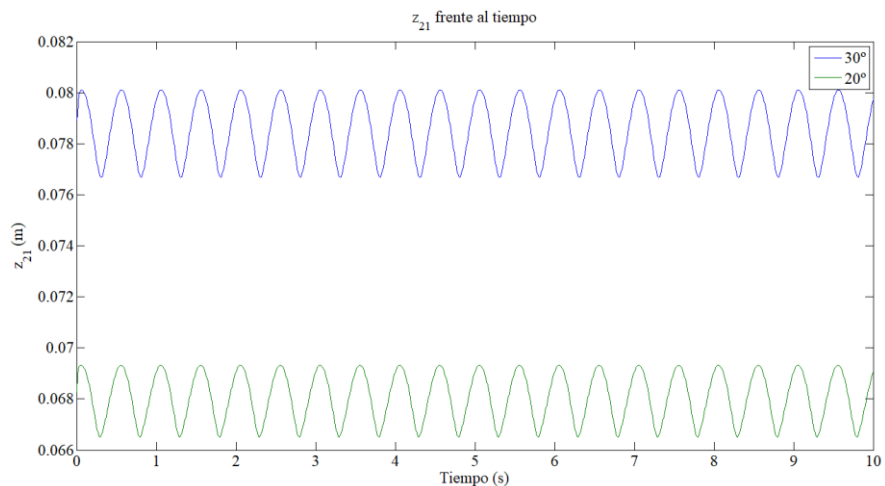


Figura 4-16. z_{21} en Caso 1,2,3 y 4

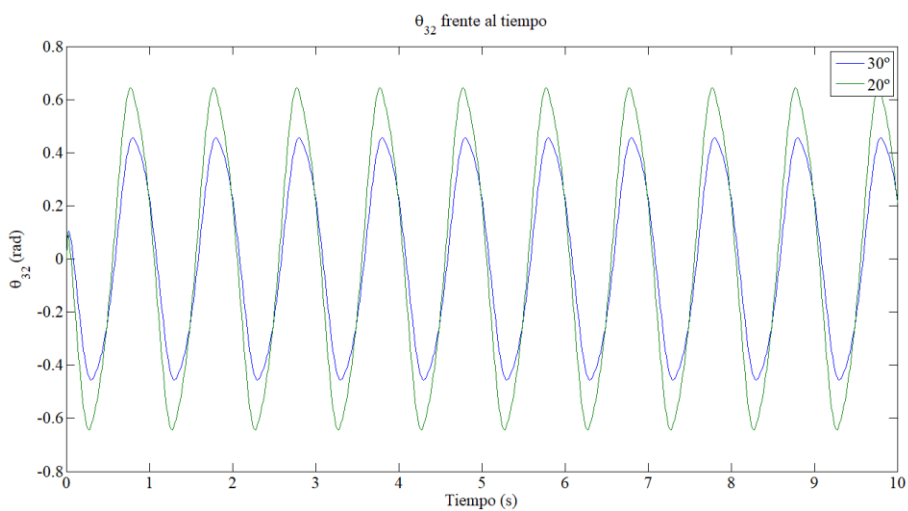


Figura 4-17. θ_{32} en Caso 1,2,3 y 4

Una vez analizada la cinemática, se verá que también la dinámica sufrirá cambios respecto a la del caso 1 a pesar de que la rigidez del muelle es la misma.

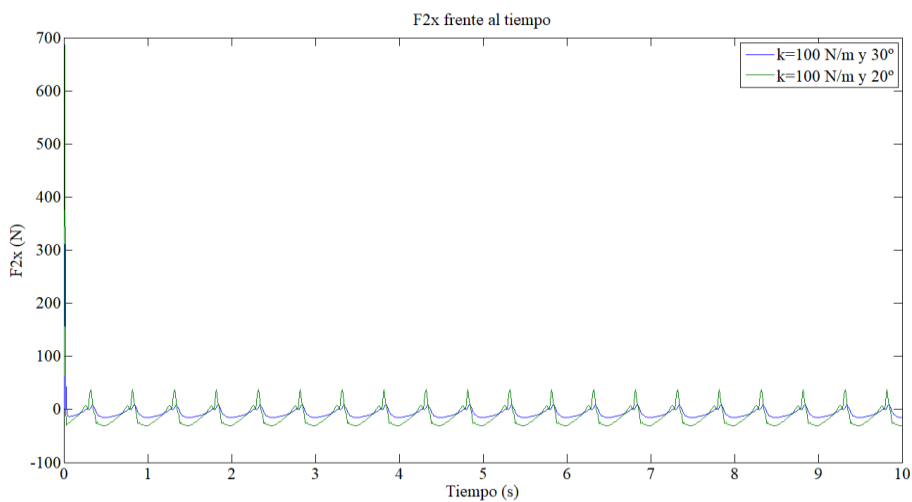


Figura 4-18. F_{2x} en Caso 1 y 4

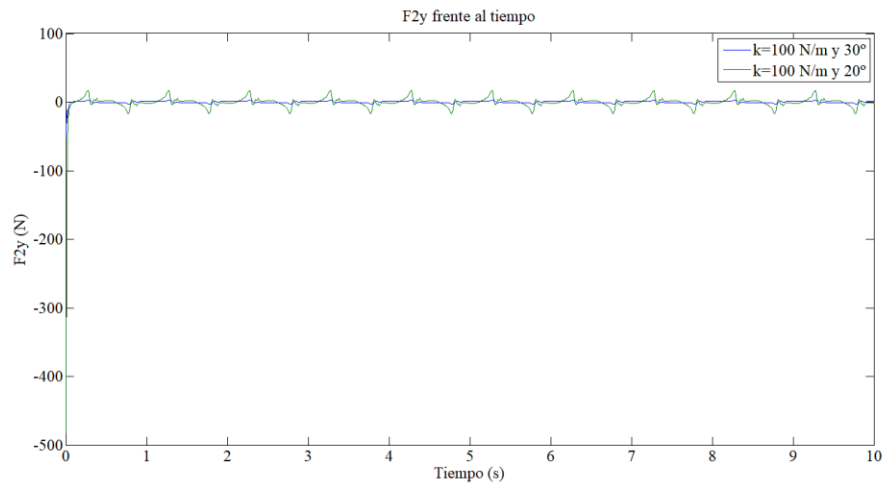


Figura 4-19. F_{2y} en Caso 1 y 4

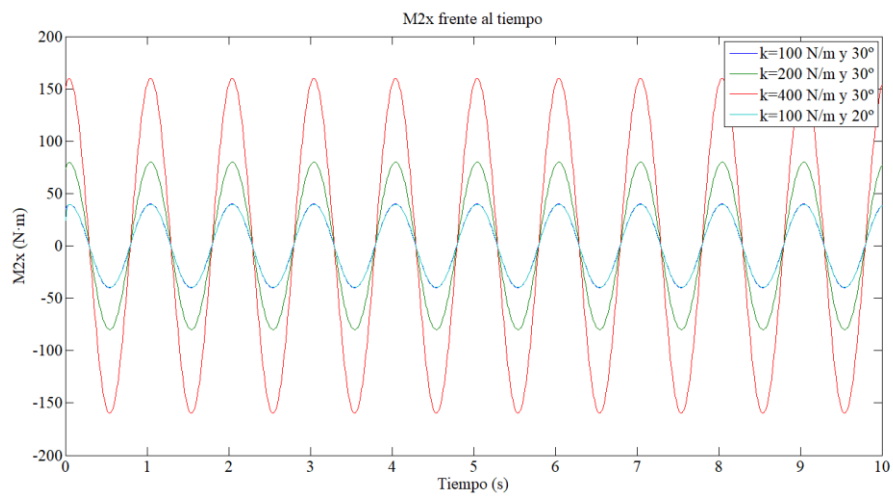


Figura 4-20. M_{2x} en Caso 1,2,3 y 4

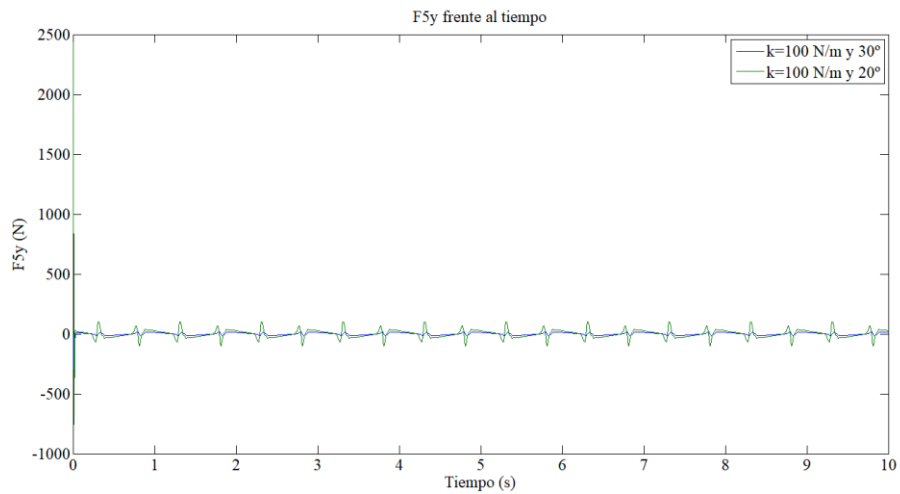


Figura 4-21. F_{5y} en Caso 1 y 4

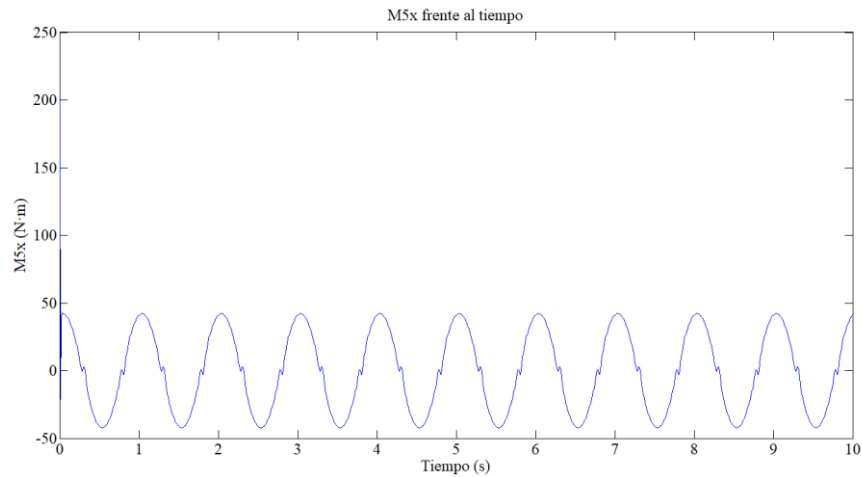


Figura 4-22. M5x en Caso 4

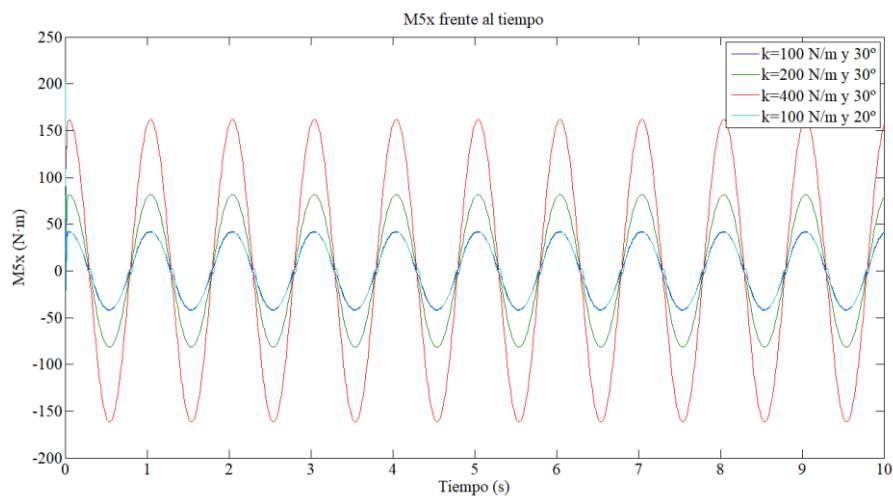


Figura 4-23. M5x en Caso 1,2,3 y 4

4.1.5. Caso 5 de Dinámica Inversa

En los siguientes dos casos el parámetro que se variará será el ángulo de inclinación, para que se vean afectadas tanto la cinemática como la dinámica. Para este cuarto caso simularemos con una rigidez de $k = 100 \text{ N/m}$ y con 40° de inclinación de la estructura giratoria. Las condiciones iniciales para la integración en este caso serán.

$$q_{d_0} = [x_{21_0} = 0 \quad z_{21_0} = 0,0898 \quad \beta_0 = 0,0124 \quad \theta_{32_0} = 0 \quad \theta_{65_0} = 0 \quad \theta_{76_0} = 0 \quad \xi_{1_0} = 0,6858 \quad \xi_{2_0} = 0,6848]$$

Una vez hecha la simulación con estos parámetros se observan variaciones considerables respecto a los casos anteriores en ciertas coordenadas cinemáticas como por ejemplo en z_{21} y θ_{32} . Podemos observar cómo el punto de equilibrio de z_{21} se hace más grande y cómo la amplitud de θ_{32} disminuye aumentando el ángulo de inclinación de la estructura.

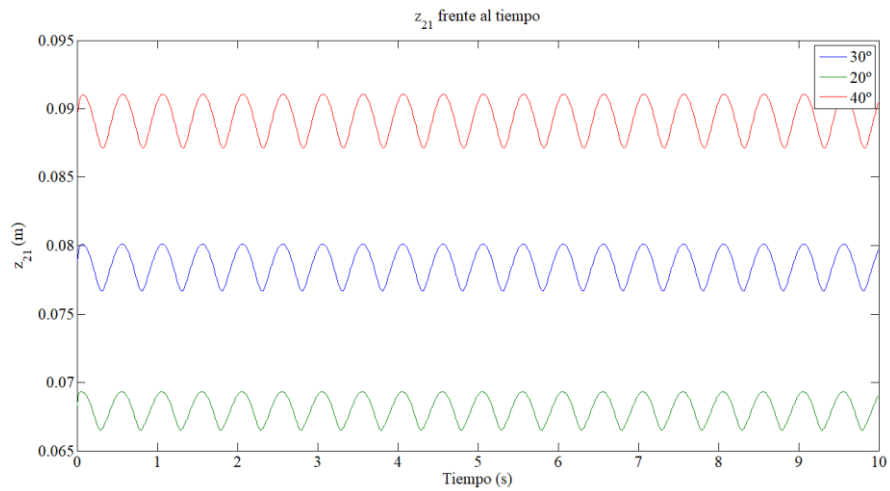


Figura 4-24. z_{21} en Caso 1,2,3,4 y 5

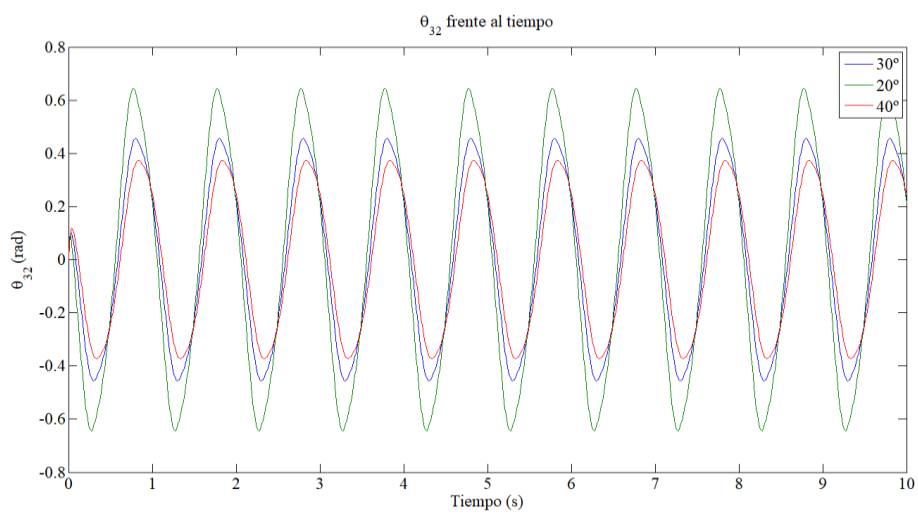


Figura 4-25. θ_{32} en Caso 1,2,3,4 y 5

Una vez analizada la cinemática, se verá que también la dinámica sufrirá cambios respecto a la del caso 1 a pesar de que la rigidez del muelle es la misma.

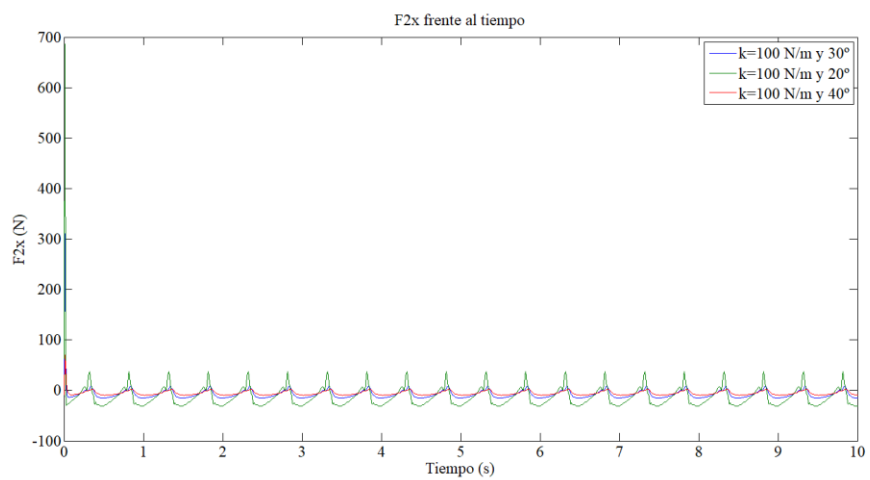


Figura 4-26. F_{2x} en Caso 1,4 y 5

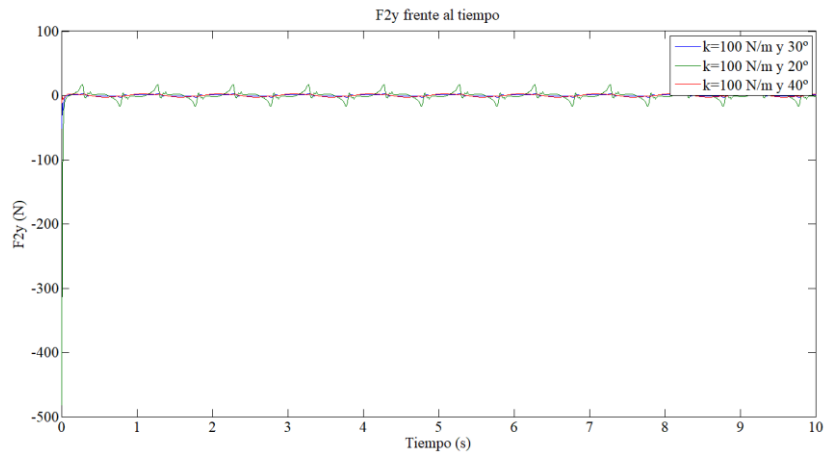


Figura 4-27. F_{2y} en Caso 1,4 y 5

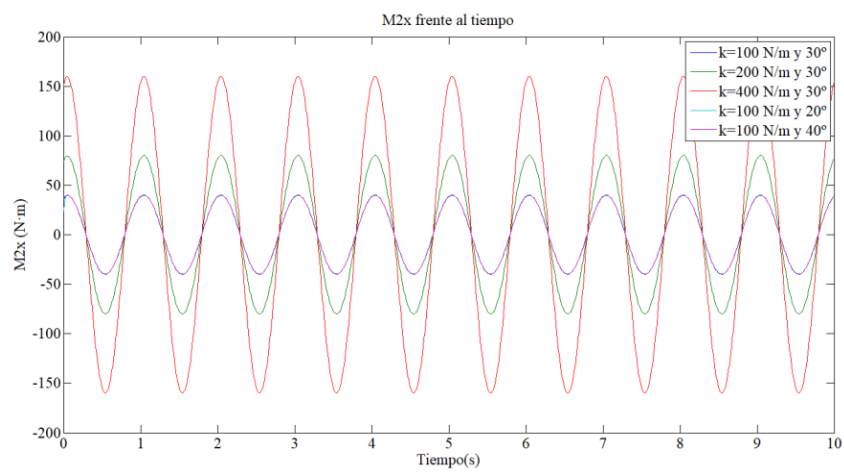


Figura 4-28. M_{2x} en Caso 1,2,3,4 y 5

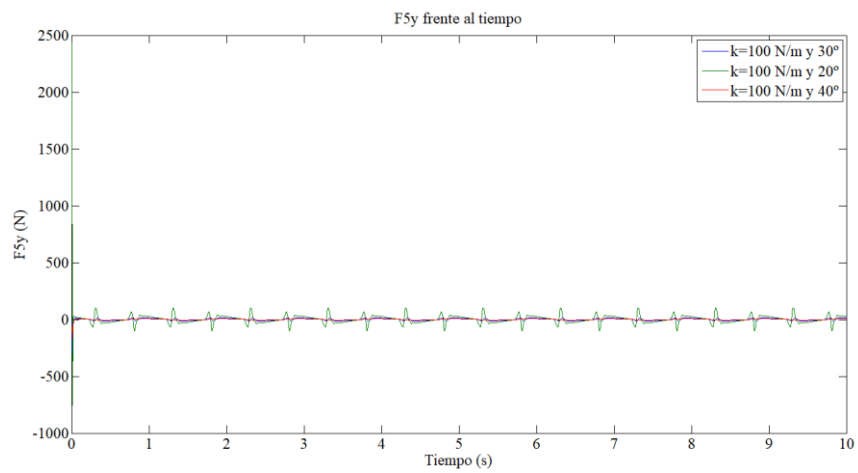


Figura 4-29. F_{5y} en Caso 1,4 y 5

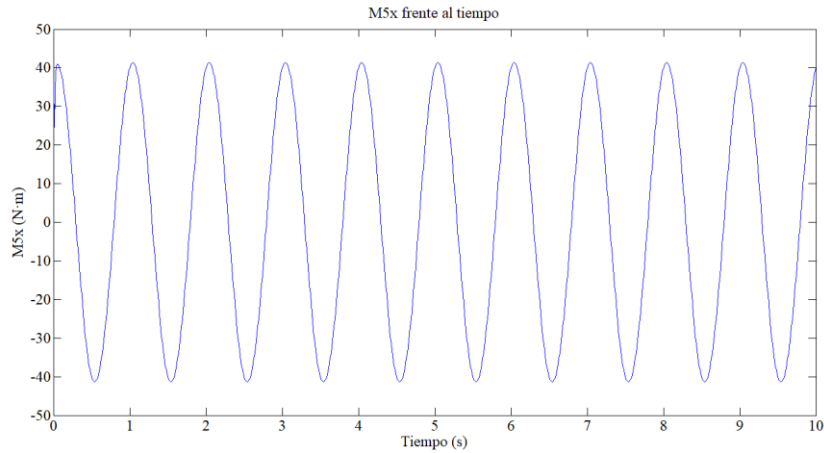


Figura 4-30. M5x en Caso 5

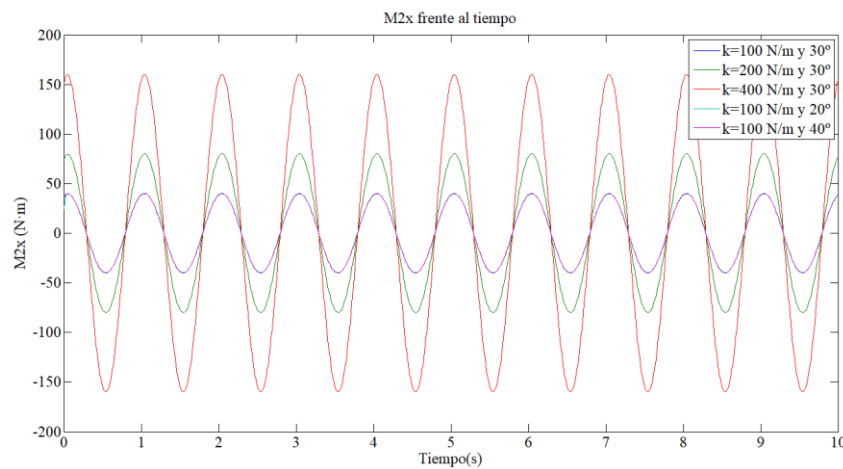


Figura 4-31. M5x en Caso 1,2,3,4 y 5

4.2 Dinámica Directa

Para la simulación del waveboard en dinámica directa se integrarán las ecuaciones vistas en el apartado 3. Se realizarán también varios casos experimentales en los que los parámetros de rigidez e inclinación se mantendrán constantes ($k = 100 \text{ N/m}$ y 30° de inclinación de la estructura giratoria), lo que se irá variando son las 9 coordenadas iniciales de posición calculando las 4 restantes mediante las ecuaciones de restricción en posición y las 5 coordenadas iniciales de velocidad calculando las 8 restantes mediante las ecuaciones de restricción en velocidad.

4.2.1. Caso 1 en Dinámica Directa

En este primer caso vamos a “lanzar” el waveboard en posición vertical a una velocidad constante. Por lo que el valor de las 9 coordenadas de posición serán $x_{21}(t = 0) = y_{21}(t = 0) = \alpha(t = 0) = \gamma(t = 0) = \theta_{52}(t = 0) = \theta_{32}(t = 0) = \theta_{65}(t = 0) = \theta_{43}(t = 0) = \theta_{76}(t = 0) = 0$ calculando las otras cuatro mediante las ecuaciones de restricción en posición: $z_{21}(t = 0) = 0,0804$, $\beta(t = 0) = 0$, $\xi_1(t = 0) = 0,5236$ y $\xi_2(t = 0) = 0,5236$.

En este caso el valor de las 5 coordenadas iniciales de velocidad serán $\dot{y}_{21}(t = 0) = \dot{\alpha}(t = 0) = \dot{\gamma}(t = 0) = \dot{\theta}_{52}(t = 0) = 0$ y $\dot{\theta}_{43}(t = 0) = 125$ calculando las ocho restantes mediante las ecuaciones de restricción en velocidad: $\dot{x}_{21}(t = 0) = 5$, $\dot{z}_{21}(t = 0) = \dot{\beta}(t = 0) = \dot{\theta}_{32}(t = 0) = \dot{\theta}_{65}(t = 0) = 0$, $\dot{\theta}_{76}(t = 0) = 125$ y $\dot{\xi}_1(t = 0) = \dot{\xi}_2(t = 0) = -125$.

Una vez realizada la integración de las ecuaciones de Newton-Euler se obtiene las 13 coordenadas a lo largo del tiempo, cómo los 8 multiplicadores de Lagrange a lo largo del tiempo. A continuación se muestran algunos ejemplos de los resultados de la integración:

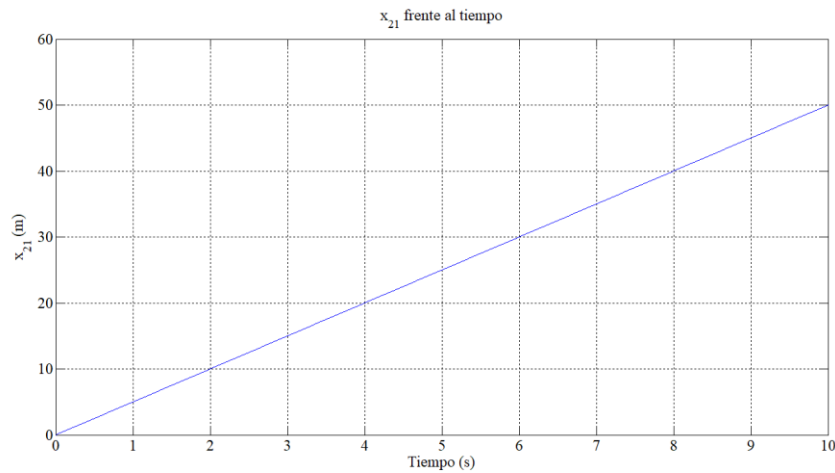


Figura 4-32. x_{21} en Caso 1 de Dinamica Directa

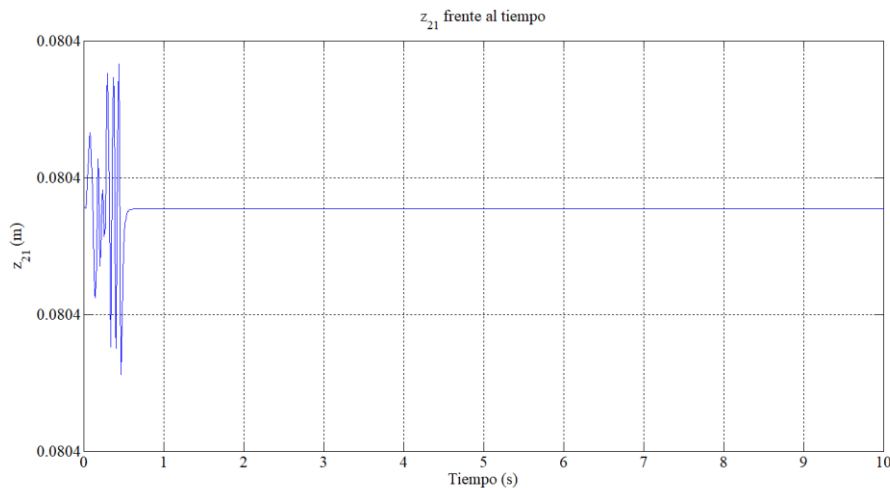


Figura 4-33. z_{21} en Caso 1 de Dinamica Directa

4.2.2. Caso 2 en Dinámica Directa

En este segundo caso vamos a “lanzar” el waveboard una velocidad constant pero con los ángulos de ambas estructuras ligeramente modificados y se podrá apreciar como la altura y la trayectoria del waveboard decrecen.

Por lo que el valor de las 9 coordenadas de posición serán $x_{21}(t = 0) = y_{21}(t = 0) = \alpha(t = 0) = \gamma(t = 0) = \theta_{52}(t = 0) = \theta_{43}(t = 0) = \theta_{76}(t = 0) = 0$, $\theta_{32}(t = 0) = 0,01$ y $\theta_{65}(t = 0) = -0,01$ calculando las otras cuatro mediante las ecuaciones de restricción en posición: $z_{21}(t = 0) = 0,0804$, $\beta(t = 0) = 0$, $\xi_1(t = 0) = 0,5236$ y $\xi_2(t = 0) = 0,5236$.

En este caso el valor de las 5 coordenadas iniciales de velocidad serán $\dot{y}_{21}(t = 0) = \dot{\alpha}(t = 0) = \dot{\gamma}(t = 0) = \dot{\theta}_{52}(t = 0) = 0$ y $\dot{\theta}_{43}(t = 0) = 125$ calculando las ocho restantes mediante las ecuaciones de restricción en velocidad: $\dot{x}_{21}(t = 0) = 5,0001$, $\dot{z}_{21}(t = 0) = -0,0002$, $\dot{\beta}(t = 0) = 0$, $\dot{\theta}_{32}(t = 0) = -4,6650$, $\dot{\theta}_{65}(t = 0) = 4,6650$, $\dot{\theta}_{76}(t = 0) = 125$ y $\dot{\xi}_1(t = 0) = \dot{\xi}_2(t = 0) = -125,0202$.

Una vez realizada la integración de las ecuaciones de Newton-Euler se obtiene las 13 coordenadas a lo largo del tiempo, cómo los 8 multiplicadores de Lagrange a lo largo del tiempo. A continuación se muestran las diferencias de trayectoria y altura con el caso en el que se “lanza” el waveboard en posición vertical. Se muestra tan poco tiempo de este caso, ya que al waveboard caerse las ecuaciones de restricción dejan de

cumplirse y la integración falla.

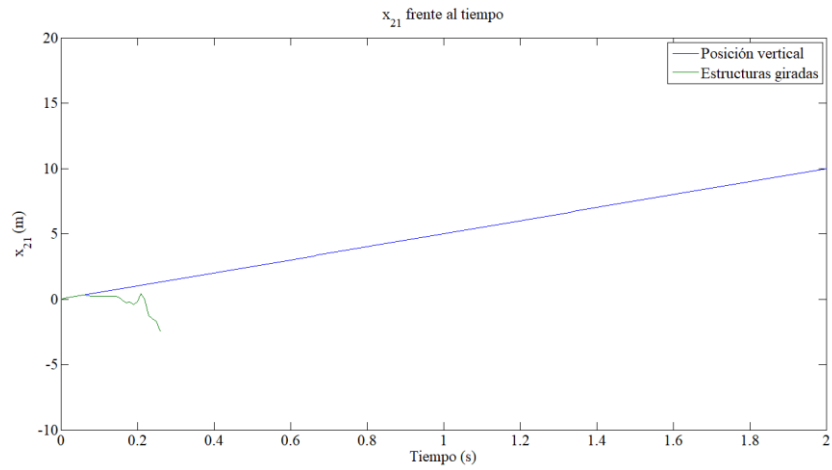


Figura 4-34. x_{21} en Caso 1 y 2 de Dinamica Directa

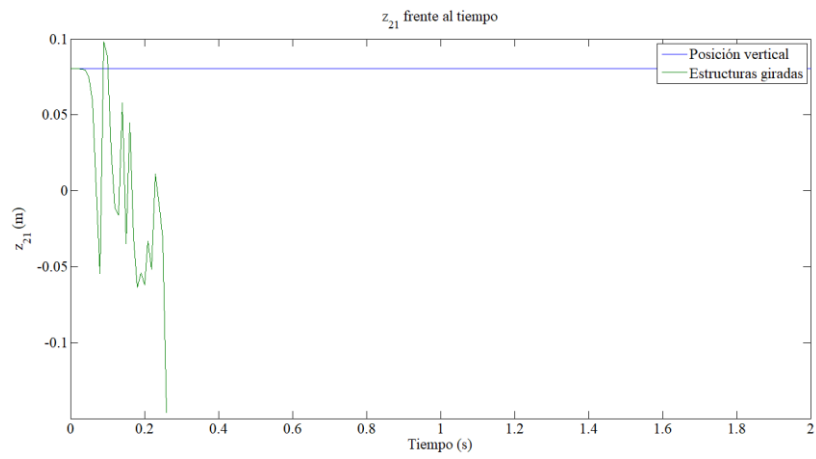


Figura 4-35. z_{21} en Caso 1 y 2 de Dinamica Directa

5 CONCLUSIONES

Este estudio del waveboard permite que se entienda mejor su funcionamiento, permite conocer cómo funcionan entre sí todos sus sólidos y cómo se desarrollan todas sus coordenadas a lo largo del tiempo. Se han realizado ambos procesos de simulación, calcular las fuerzas necesarias para generar un movimiento impuesto (Dinámica Inversa) y calcular las coordenadas del sistema a partir de aplicarle una serie de acciones (Dinámica Directa). Estos estudios han generado un mayor conocimiento de cómo se puede controlar el waveboard, por ejemplo se ha observado que cuando aumentamos la rigidez del muelle torsional, los momentos que hay que aplicarle a las plataformas deben ser mayores para generar el mismo movimiento.

Una continuación a estos estudios podría ser meter cómo acciones exteriores al problema de dinámica directa las fuerzas calculadas en el problema de dinámica inversa y así calcular las coordenadas a lo largo del tiempo de un movimiento realista del waveboard. También se podría tener en cuenta los rozamientos y las fuerzas de resistencia aerodinámica correspondientes., que aunque no modificarán mucho el funcionamiento del waveboard, nos puede permitir un movimiento aún más realista.

BIBLIOGRAFÍA

- [1] José Luis Escalona y Manuel Cornax Atienza, "Simulación y Análisis de la Dinámica de una bicicleta".
- [2] www.decathlon.es
- [3] José Luis Escalona, Apuntes de la asignatura Cinemática y Dinámica de Máquinas.
- [4] J. Baumgarte, "Stabilization of constraints and integrals of motion in dynamical systems".
- [5] Ayush Agrawal, "Experimental gait analysis of waveboard locomotion".

ANEXO I. CÓDIGOS DE MATLAB

Archivo de Generación de Funciones

```
fprintf('Defining symbolic variables...\n')
syms x21 y21 z21 alfa beta gamm theta52 theta32 theta65
theta43 theta76 chil chi2 real
syms x21p y21p z21p alfap betap gammp theta52p theta32p theta65p
theta43p theta76p chilp chi2p real
syms x21pp y21pp z21pp alfapp betapp gammpp theta52pp theta32pp theta65pp
theta43pp theta76pp chilpp chi2pp real

fprintf('Defining constants...\n')
syms grav theta_int u a s e b d R k real % he cambiado "j" por "e" y "L" por "u"
syms m2 m3 m4 m5 m6 m7 real
syms Ix2 Iy2 Iz2 Ix3 Iy3 Iz3 Ix4 Iy4 Iz4 Ix5 Iy5 Iz5 Ix6 Iy6 Iz6 Ix7 Iy7 Iz7 real

fprintf('Building transformation matrices...\n')
q = [x21, y21, z21, alfa, beta, gamm, theta52, theta32, theta65,
theta43, theta76, chil, chi2];
qp = [x21p, y21p, z21p, alfap, betap, gammp, theta52p, theta32p, theta65p,
theta43p, theta76p, chilp, chi2p];
par = [grav theta_int u a s e b d R k m2 m3 m4 m5 m6 m7 Ix2 Iy2 Iz2 Ix3 Iy3 Iz3 Ix4
Iy4 Ix5 Iy5 Iz5 Ix6 Iy6 Iz6 Ix7 Iy7 Iz7];

wvar = pi/2-theta_int;
A_int = [ cos(wvar), 0, sin(wvar);
          0, 1, 0;
          -sin(wvar), 0, cos(wvar)];

A_gamm = [cos(gamm), -sin(gamm), 0;
          sin(gamm), cos(gamm), 0;
          0, 0, 1];

A_beta = [cos(beta), 0, sin(beta);
          0, 1, 0;
          -sin(beta), 0, cos(beta)];

A_alfa = [1, 0, 0;
          0 cos(alfa), -sin(alfa);
          0 sin(alfa), cos(alfa)];

A_theta52 = [1 0 0;
            0 cos(theta52) -sin(theta52);
            0 sin(theta52) cos(theta52)];

A_theta32 = [cos(theta32) -sin(theta32) 0;
            sin(theta32) cos(theta32) 0;
            0 0 1];

A_theta65 = [cos(theta65) -sin(theta65) 0;
            sin(theta65) cos(theta65) 0;
            0 0 1];

A_theta43 = [cos(theta43) 0 sin(theta43);
            0 1 0;
            -sin(theta43) 0 cos(theta43)];

A_theta76 = [cos(theta76) 0 sin(theta76);
            0 1 0];
```

```

        -sin(theta76) 0 cos(theta76)];

A_chi1 = [cos(chi1) 0 sin(chi1);
          0 1 0;
          -sin(chi1) 0 cos(chi1)];

A_chi2 = [cos(chi2) 0 sin(chi2);
          0 1 0;
          -sin(chi2) 0 cos(chi2)];

A21 = A_alfa*A_beta*A_gamm;
if gen, matlabFunction(A21, 'file', 'A21', 'Vars', {q,par}); end
A51=A21*A_theta52;
if gen, matlabFunction(A51, 'file', 'A51', 'Vars', {q,par}); end
A31=A21*A_int*A_theta32;
if gen, matlabFunction(A31, 'file', 'A31', 'Vars', {q,par}); end
A41=A31*A_theta43;
if gen, matlabFunction(A41, 'file', 'A41', 'Vars', {q,par}); end
A61=A51*A_int*A_theta65;
if gen, matlabFunction(A61, 'file', 'A61', 'Vars', {q,par}); end
A71=A61*A_theta76;
if gen, matlabFunction(A71, 'file', 'A71', 'Vars', {q,par}); end

% Body-2
fprintf('Creating G2, G2p, H21 and H21p...\n')

syms A21p real; A21p = 0;
for i=1:length(q)
    A21p = A21p + diff(A21,q(i))*qp(i);
end

Wskew21 =(A21')*A21p ;

w1_21 = Wskew21(3,2);
w2_21 = Wskew21(1,3);
w3_21 = Wskew21(2,1);

w21 = [w1_21,w2_21,w3_21];
w21 = simplify(w21);
G21 = jacobian(w21,qp);
syms g21 real; g21 = 0;
for i=1:length(q)
    g21 = g21 + diff(G21,q(i))*qp(i);
end
if gen, matlabFunction(G21, 'file', 'G2', 'Vars', {q,par}); end
if gen, matlabFunction(g21, 'file', 'G2p', 'Vars', {q,qp,par}); end

r2 = [x21;y21;z21];
H2 = jacobian(r2,q);
syms h2 real; h2 = 0;
for i=1:length(q)
    h2 = h2 + diff(H2,q(i))*qp(i);
end
if gen, matlabFunction(H2, 'file', 'H21', 'Vars', {q,par}); end
if gen, matlabFunction(h2, 'file', 'H21p', 'Vars', {q,qp,par}); end

% Body-3
fprintf('Creating G3, G3p, H31 and H31p...\n')
syms A31p real; A31p = 0;
for i=1:length(q)
    A31p = A31p + diff(A31,q(i))*qp(i);
end

Wskew31 =(A31')*A31p ;

w1_31 = Wskew31(3,2);
w2_31 = Wskew31(1,3);
w3_31 = Wskew31(2,1);

w31 = [w1_31,w2_31,w3_31];

```

```

w31 = simplify(w31);
G31 = jacobian(w31,qp);
syms g31 real; g31 = 0;
for i=1:length(q)
    g31 = g31 + diff(G31,q(i))*qp(i);
end
if gen, matlabFunction(G31,'file','G3','Vars',{q,par}); end
if gen, matlabFunction(g31,'file','G3p','Vars',{q,qp,par}); end

r3 = r2 + A21*[-s;0;-a];
if gen, matlabFunction(r3,'file','r3','Vars',{q,par}); end
H3 = jacobian(r3,q);
syms h3 real; h3 = 0;
for i=1:length(q)
    h3 = h3 + diff(H3,q(i))*qp(i);
end
if gen, matlabFunction(H3,'file','H31','Vars',{q,par}); end
if gen, matlabFunction(h3,'file','H31p','Vars',{q,qp,par}); end

% Body-4
fprintf('Creating G4, G4p, H41 and H41p...\n')
syms A41p real; A41p = 0;
for i=1:length(q)
    A41p = A41p + diff(A41,q(i))*qp(i);
end

Wskew41 = (A41')*A41p ;

w1_41 = Wskew41(3,2);
w2_41 = Wskew41(1,3);
w3_41 = Wskew41(2,1);

w41 = [w1_41,w2_41,w3_41];
w41 = simplify(w41);
G41 = jacobian(w41,qp);
syms g41 real; g41 = 0;
for i=1:length(q)
    g41 = g41 + diff(G41,q(i))*qp(i);
end
if gen, matlabFunction(G41,'file','G4','Vars',{q,par}); end
if gen, matlabFunction(g41,'file','G4p','Vars',{q,qp,par}); end

r4 = r3 + A31*[-d;0;-b];
if gen, matlabFunction(r4,'file','r4','Vars',{q,par}); end
H4 = jacobian(r4,q);
syms h4 real; h4 = 0;
for i=1:length(q)
    h4 = h4 + diff(H4,q(i))*qp(i);
end
if gen, matlabFunction(H4,'file','H41','Vars',{q,par}); end
if gen, matlabFunction(h4,'file','H41p','Vars',{q,qp,par}); end

% Body-5
fprintf('Creating G5, G5p, H51 and H51p...\n')
syms A51p real; A51p = 0;
for i=1:length(q)
    A51p = A51p + diff(A51,q(i))*qp(i);
end

Wskew51 = (A51')*A51p ;

w1_51 = Wskew51(3,2);
w2_51 = Wskew51(1,3);
w3_51 = Wskew51(2,1);

w51 = [w1_51,w2_51,w3_51];
w51 = simplify(w51);
G51 = jacobian(w51,qp);
syms g51 real; g51 = 0;
for i=1:length(q)

```

```

    g51 = g51 + diff(G51,q(i))*qp(i);
end
if gen, matlabFunction(G51,'file','G5','Vars',{q,par}); end
if gen, matlabFunction(g51,'file','G5p','Vars',{q,qp,par}); end

r5 = r2 + A21*[u;0;0];
if gen, matlabFunction(r5,'file','r5','Vars',{q,par}); end
H5 = jacobian(r5,q);
syms h5 real; h5 = 0;
for i=1:length(q)
    h5 = h5 + diff(H5,q(i))*qp(i);
end
if gen, matlabFunction(H5,'file','H51','Vars',{q,par}); end
if gen, matlabFunction(h5,'file','H51p','Vars',{q,qp,par}); end

% Body-6
fprintf('Creating G6, G6p, H61 and H61p...\n')
syms A61p real; A61p = 0;
for i=1:length(q)
    A61p = A61p + diff(A61,q(i))*qp(i);
end

Wskew61 =(A61')*A61p ;

w1_61 = Wskew61(3,2);
w2_61 = Wskew61(1,3);
w3_61 = Wskew61(2,1);

w61 = [w1_61,w2_61,w3_61];
w61 = simplify(w61);
G61 = jacobian(w61,qp);
syms g61 real; g61 = 0;
for i=1:length(q)
    g61 = g61 + diff(G61,q(i))*qp(i);
end
if gen, matlabFunction(G61,'file','G6','Vars',{q,par}); end
if gen, matlabFunction(g61,'file','G6p','Vars',{q,qp,par}); end

r6 = r5 + A51*[e;0;-a];
if gen, matlabFunction(r6,'file','r6','Vars',{q,par}); end
H6 = jacobian(r6,q);
syms h6 real; h6 = 0;
for i=1:length(q)
    h6 = h6 + diff(H6,q(i))*qp(i);
end
if gen, matlabFunction(H6,'file','H61','Vars',{q,par}); end
if gen, matlabFunction(h6,'file','H61p','Vars',{q,qp,par}); end

% Body-7
fprintf('Creating G7, G7p, H71 and H71p...\n')
syms A71p real; A71p = 0;
for i=1:length(q)
    A71p = A71p + diff(A71,q(i))*qp(i);
end

Wskew71 =(A71')*A71p ;

w1_71 = Wskew71(3,2);
w2_71 = Wskew71(1,3);
w3_71 = Wskew71(2,1);

w71 = [w1_71,w2_71,w3_71];
w71 = simplify(w71);
G71 = jacobian(w71,qp);
syms g71 real; g71 = 0;
for i=1:length(q)
    g71 = g71 + diff(G71,q(i))*qp(i);
end
if gen, matlabFunction(G71,'file','G7','Vars',{q,par}); end
if gen, matlabFunction(g71,'file','G7p','Vars',{q,qp,par}); end

```

```

r7 = r6 + A61*[-d;0;-b];
if gen, matlabFunction(r7, 'file', 'r7', 'Vars', {q,par}); end
H7 = jacobian(r7,q);
syms h7 real; h7 = 0;
for i=1:length(q)
    h7 = h7 + diff(H7,q(i))*qp(i);
end
if gen, matlabFunction(H7, 'file', 'H71', 'Vars', {q,par}); end
if gen, matlabFunction(h7, 'file', 'H71p', 'Vars', {q,qp,par}); end

% Constraints
fprintf('Computing holonomic constraints...\n')

rp1 = r4 + A41*(A_chi1*[R;0;0]);
rp2 = r7 + A71*(A_chi2*[R;0;0]);

t4 = A41*[-sin(chi1);0;-cos(chi1)];
t7 = A71*[-sin(chi2);0;-cos(chi2)];

C1 = rp1(3);
C2 = rp2(3);
C3 = t4(3);
C4 = t7(3);
C = [C1;C2;C3;C4];

fprintf('Creating function C...\n')
if gen, matlabFunction(C, 'file', 'C', 'Vars', {q,par}); end

syms Cpunto real; Cpunto = 0;
for i=1:length(q)
    Cpunto = Cpunto + diff(C,q(i))*qp(i);
end
if gen, matlabFunction(Cpunto, 'file', 'Cpunto', 'Vars', {q,qp,par}); end

Cq = jacobian(C,q);
Cq = simplify(Cq);
syms Cqpunto real; Cqpunto = 0;
for i=1:length(q)
    Cqpunto = Cqpunto + diff(Cq,q(i))*qp(i);
end
if gen, matlabFunction(Cqpunto, 'file', 'Cqpunto', 'Vars', {q,qp,par}); end

fprintf('Computing non-holonomic constraints...\n')
syms v41C4 real; v41C4 = 0;
for i=1:length(q)
    v41C4 = v41C4 + diff(r4,q(i))*qp(i);
end
vp1 = v41C4 + A41*cross(w41', [R*cos(chi1);0;-R*sin(chi1)]);

syms v71C7 real; v71C7 = 0;
for i=1:length(q)
    v71C7 = v71C7 + diff(r7,q(i))*qp(i);
end
vp2 = v71C7 + A71*cross(w71', [R*cos(chi2);0;-R*sin(chi2)]);

vp = [vp1(1);vp1(2);vp2(1);vp2(2)];

fprintf('Creating function vp...\n')
if gen, matlabFunction(vp, 'file', 'vp', 'Vars', {q,qp,par}); end

B = jacobian(vp,qp);
B = simplify(B);

D = [Cq;B];
fprintf('Creating function D...\n')
if gen, matlabFunction(D, 'file', 'D', 'Vars', {q,par}); end

Dpunto = [
    (jacobian(Cq*(qp'), q));

```

```

    (jacobian(B*(qp'),q));
fprintf('Creating function Dpunto...\n')
if gen, matlabFunction(Dpunto,'file','Dpunto','Vars',{q,qp,par}); end

DpuntoXqp = [
    (jacobian(Cq*(qp'),q))*(qp');
    (jacobian(B*(qp'),q))*(qp')];
fprintf('Creating function DpuntoXqp...\n')
if gen, matlabFunction(DpuntoXqp,'file','DpuntoXqp','Vars',{q,qp,par}); end

velocidades = [Cpunto;vp];
fprintf('Creating function velocidades...\n')
if gen, matlabFunction(velocidades,'file','velocidades','Vars',{q,qp,par}); end

%Ecuaciones Newton-Euler
fprintf('Assembling equations...\n')

m(1,1:3) = [m2, 0, 0];
m(2,1:3) = [0, m2, 0];
m(3,1:3) = [0, 0, m2];

m(4,4:6) = [m3, 0, 0];
m(5,4:6) = [0, m3, 0];
m(6,4:6) = [0, 0, m3];

m(7,7:9) = [m4, 0, 0];
m(8,7:9) = [0, m4, 0];
m(9,7:9) = [0, 0, m4];

m(10,10:12) = [m5, 0, 0];
m(11,10:12) = [0, m5, 0];
m(12,10:12) = [0, 0, m5];

m(13,13:15) = [m6, 0, 0];
m(14,13:15) = [0, m6, 0];
m(15,13:15) = [0, 0, m6];

m(16,16:18) = [m7, 0, 0];
m(17,16:18) = [0, m7, 0];
m(18,16:18) = [0, 0, m7];

I(1,1:3) = [Ix2, 0, 0];
I(2,1:3) = [0, Iy2, 0];
I(3,1:3) = [0, 0, Iz2];

I(4,4:6) = [Ix3, 0, 0];
I(5,4:6) = [0, Iy3, 0];
I(6,4:6) = [0, 0, Iz3];

I(7,7:9) = [Ix4, 0, 0];
I(8,7:9) = [0, Iy4, 0];
I(9,7:9) = [0, 0, Iz4];

I(10,10:12) = [Ix5, 0, 0];
I(11,10:12) = [0, Iy5, 0];
I(12,10:12) = [0, 0, Iz5];

I(13,13:15) = [Ix6, 0, 0];
I(14,13:15) = [0, Iy6, 0];
I(15,13:15) = [0, 0, Iz6];

I(16,16:18) = [Ix7, 0, 0];
I(17,16:18) = [0, Iy7, 0];
I(18,16:18) = [0, 0, Iz7];

Mgorro = [m zeros(18);
          zeros(18) I];

```

```

fprintf('Creating function L...\n')
L = [H2;H3;H4;H5;H6;H7;G21;G31;G41;G51;G61;G71];
if gen, matlabFunction(L,'file','L','Vars',{q,par}); end
fprintf('Creating function Lpunto...\n')
l=[h2;h3;h4;h5;h6;h7;g21;g31;g41;g51;g61;g71];
if gen, matlabFunction(l,'file','Lpunto','Vars',{q,qp,par}); end

F2=[0;0;-m2*grav];
F3=[0;0;-m3*grav];
F4=[0;0;-m4*grav];
F5=[0;0;-m5*grav];
F6=[0;0;-m6*grav];
F7=[0;0;-m7*grav];

M2=[k*theta52;0;0];
M3=[0;0;0];
M4=[0;0;0];
M5=[k*theta52;0;0];
M6=[0;0;0];
M7=[0;0;0];
Qaplgorro=[F2;F3;F4;F5;F6;F7;M2;M3;M4;M5;M6;M7];

Fv2=[0;0;0];
Fv3=[0;0;0];
Fv4=[0;0;0];
Fv5=[0;0;0];
Fv6=[0;0;0];
Fv7=[0;0;0];

v2 = [Ix2 0 0;
      0 Iy2 0;
      0 0 Iz2]*(w21');
Mv2 = cross(-w21',v2);

v3 = [Ix3 0 0;
      0 Iy3 0;
      0 0 Iz3]*(w31');
Mv3 = cross(-w31',v3);

v4 = [Ix4 0 0;
      0 Iy4 0;
      0 0 Iz4]*(w41');
Mv4 = cross(-w41',v4);

v5 = [Ix5 0 0;
      0 Iy5 0;
      0 0 Iz5]*(w51');
Mv5 = cross(-w51',v5);

v6 = [Ix6 0 0;
      0 Iy6 0;
      0 0 Iz6]*(w61');
Mv6 = cross(-w61',v6);

v7 = [Ix7 0 0;
      0 Iy7 0;
      0 0 Iz7]*(w71');
Mv7 = cross(-w71',v7);

fprintf('Creating function Qvgorro...\n')
Qvgorro = [Fv2;Fv3;Fv4;Fv5;Fv6;Fv7;Mv2;Mv3;Mv4;Mv5;Mv6;Mv7];
if gen, matlabFunction(Qvgorro,'file','Qvgorro','Vars',{q,qp,par}); end

fprintf('Creating function M...\n')
M = L'*Mgorro*L;
if gen, matlabFunction(M,'file','M','Vars',{q,par}); end

fprintf('Creating function Qapl...\n')
Qapl = L'*Qaplgorro;

```



```
matlabFunction(Qapl, 'file', 'Qapl', 'Vars', {q, qp, par});

fprintf('Creating function A...\n')
Qv = L'* (Qvgorro-Mgorro*1*(qp));
A = [M D'; D zeros(8)];
if gen, matlabFunction(A, 'file', 'A', 'Vars', {q, par}); end
```

Archivo de Dinámica Inversa

```
function f=simulacioncinematica(t,X,par)
q=[X(1);0.01*sin(2*pi*t);X(2);0.3*sin(2*pi*t);X(3);0.2*sin(2*pi*t);0.4*sin(2*pi*t-
1.85);X(4:5);12.5*t;X(6:8)];
P=D(q',par);
alfa=50;
qindeppunto=[0.02*pi*cos(2*pi*t);0.6*pi*cos(2*pi*t);0.4*pi*cos(2*pi*t);0.8*pi*cos(2*pi
*t-1.85);12.5];
Ddep=[P(:,1),P(:,3),P(:,5),P(:,8:9),P(:,11:13)];
Dindep=[P(:,2),P(:,4),P(:,6:7),P(:,10)];
f=-Ddep\ (Dindep*qindeppunto+alfa*[C(q',par);0;0;0;0]);
end

grav = 9.81;
theta_int = pi/6;
u = 0.37;
a = 0.05;
s = 0.07;
e = 0.1;
b = 0.05;
d = 0.04;
R = 0.04;

m2=1.143;
Ix2=37e-4;
Iy2=195.5e-4;
Iz2=231.2e-4;

m3=0.252;
Ix3=3.3e-4;
Iy3=2.8e-4;
Iz3=1.6e-4;

m4=0.105;
Ix4=0.5e-4;
Iy4=0.8e-4;
Iz4=0.5e-4;

m5=1.143;
Ix5=37e-4;
Iy5=195.5e-4;
Iz5=231.2e-4;

m6=0.252;
Ix6=3.3e-4;
Iy6=2.8e-4;
Iz6=1.6e-4;

m7=0.105;
Ix7=0.5e-4;
Iy7=0.8e-4;
Iz7=0.5e-4;

k=100;
par = [grav theta_int u a s e b d R k m2 m3 m4 m5 m6 m7 Ix2 Iy2 Iz2 Ix3 Iy3 Iz3 Ix4
Iy4 Iz4 Ix5 Iy5 Iz5 Ix6 Iy6 Iz6 Ix7 Iy7 Iz7];

m = zeros(18,18);

m(1,1) = m2;
m(2,2) = m2;
```

```

m(3,3) = m2;

m(4,4) = m3;
m(5,5) = m3;
m(6,6) = m3;

m(7,7) = m4;
m(8,8) = m4;
m(9,9) = m4;

m(10,10) = m5;
m(11,11) = m5;
m(12,12) = m5;

m(13,13) = m6;
m(14,14) = m6;
m(15,15) = m6;

m(16,16) = m7;
m(17,17) = m7;
m(18,18) = m7;

I=zeros(18,18);

I(1,1) = Ix2;
I(2,2) = Iy2;
I(3,3) = Iz2;

I(4,4) = Ix3;
I(5,5) = Iy3;
I(6,6) = Iz3;

I(7,7) = Ix4;
I(8,8) = Iy4;
I(9,9) = Iz4;

I(10,10) = Ix5;
I(11,11) = Iy5;
I(12,12) = Iz5;

I(13,13) = Ix6;
I(14,14) = Iy6;
I(15,15) = Iz6;

I(16,16) = Ix7;
I(17,17) = Iy7;
I(18,18) = Iz7;

Mgorro=[m zeros(18);zeros(18) I];
ts=0:0.01:10;
%Definicon de variables independientes
y2l=0.01*sin(2*pi*ts);
y2lp=0.02*pi*cos(2*pi*ts);
y2lpp=-0.04*pi*pi*sin(2*pi*ts);
alfa2l=0.3*sin(2*pi*ts);
alfa2lp=0.6*pi*cos(2*pi*ts);
alfa2lpp=-1.2*pi*pi*sin(2*pi*ts);
gamm2l=0.2*sin(2*pi*ts);
gamm2lp=0.4*pi*cos(2*pi*ts);
gamm2lpp=-0.8*pi*pi*sin(2*pi*ts);
theta52=0.4*sin(2*pi*ts-1.85);
theta52p=0.8*pi*cos(2*pi*ts-1.85);
theta52pp=-1.6*pi*pi*sin(2*pi*ts-1.85);
theta43=12.5*ts;
theta43p=12.5;
theta43pp=0;

%obtencion de q0dep

```

```

q0=[0;y21(1);0.16;alfa21(1);0;gamm21(1);theta52(1);0.00;0.0;theta43(1);0;0;0];

options = optimset('Display','iter');

X0 = fsolve(@(X)
C([q0(1:2);X(1);q0(4);X(2);q0(6:11);X(3);X(4)]',par), [q0(3);q0(5);q0(12:13,1)], options
)
q0(3) = X0(1);
q0(5) = X0(2);
q0(12) = X0(3);
q0(13) = X0(4);

q0dep=[q0(1);q0(3);q0(5);q0(8:9);q0(11:13)];

%obtencion qdep
[t,z]=ode45(@(t,X) simulacioncinematica(t,X,par),ts,q0dep);

for k=1:length(t)
    %obtencion q,qp y qpp
    y(k,:)= [z(k,1),y21(k),z(k,2),alfa21(k),z(k,3),gamm21(k),theta52(k),z(k,4:5),theta43(k)
,z(k,6:8)];
    q=y(k,:);
    E=D(q,par);
    Ddep=[E(:,1),E(:,3),E(:,5),E(:,8:9),E(:,11:13)];
    Dindep=[E(:,2),E(:,4),E(:,6:7),E(:,10)];
    qpuntoindep=[y21p(k);alfa21p(k);gamm21p(k);theta52p(k);theta43p];
    qpuntodep=Ddep\(-Dindep*qpuntoindep);
    ypunto(k,:)= [
    qpuntodep(1);
    qpuntoindep(1);
    qpuntodep(2);
    qpuntoindep(2);
    qpuntodep(3);
    qpuntoindep(3:4);
    qpuntodep(4:5);
    qpuntoindep(5);
    qpuntodep(6:8)];
    F=Dpunto(q,ypunto(k,:),par);
    Dpuntodep=[F(:,1),F(:,3),F(:,5),F(:,8:9),F(:,11:13)];
    Dpuntoindep=[F(:,2),F(:,4),F(:,6:7),F(:,10)];
    qdospuntoindep=[y21pp(k);alfa21pp(k);gamm21pp(k);theta52pp(k);theta43pp];
    qdospuntodep=Ddep\(-Dpuntoindep*qdospuntoindep-Dindep*qdospuntoindep-
Dpuntodep*qpuntodep);
    ydospunto(k,:)= [
    qdospuntodep(1);
    qdospuntoindep(1);
    qdospuntodep(2);
    qdospuntoindep(2);
    qdospuntodep(3);
    qdospuntoindep(3:4);
    qdospuntodep(4:5);
    qdospuntoindep(5);
    qdospuntodep(6:8)];
    %Dinamica inversa
    P=L(q,par);
    A2=A21(q,par);
    A5=A51(q,par);

    Atau=[(P(1:3,:))'*A2(:,1),(P(1:3,:))'*A2(:,2),(P(19,:))',(P(10:12,:))'*A5(:,2),(P(28,:
))');
    N=[-E',Atau]\(M(q,par)*(ydospunto(k,:))'-(L(q,par)*(Qvgorro(q,ypunto(k,:),par))-
Mgorro*lpunto(q,ypunto(k,:),par)*(ypunto(k,:))'-Qapl(q,ypunto(k,:),par)));
    multiplicadores(k,:)=(N(1:8))';
    tau(k,:)=(N(9:13))';
end

```

Archivo de Dinámica Directa

```

function f=ecs(t,X,Mgorro,par)
x=X(1:13,1)';
v=X(14:26,1)';
alfa=50;
beta=50;
Z=[Qapl(x,v,par)+L(x,par)'*(Qvgorro(x,v,par)-Mgorro*Lpunto(x,v,par)*v');
   -DpuntoXqp(x,v,par)-2*alfa*velocidades(x,v,par)-beta*beta*[C(x,par);0;0;0;0]];
N=(A(x,par))\Z;
f=[v';N(1:13,1)];
end

function f=multiplicadores(X,Mgorro,par)
x=X(1,1:13);
v=X(1,14:26);
alfa=50;
beta=50;
Z=[Qapl(x,v,par)+L(x,par)'*(Qvgorro(x,v,par)-Mgorro*Lpunto(x,v,par)*v');
   -DpuntoXqp(x,v,par)-2*alfa*velocidades(x,v,par)-beta*beta*[C(x,par);0;0;0;0]];
N=(A(x,par))\Z;
f=N(14:21);
end

grav = 9.81;
theta_int = pi/6;
u = 0.37;
a = 0.05;
s = 0.07;
e = 0.1;
b = 0.05;
d = 0.04;
R = 0.04;

m2=1.143;
Ix2=37e-4;
Iy2=195.5e-4;
Iz2=231.2e-4;

m3=0.252;
Ix3=3.3e-4;
Iy3=2.8e-4;
Iz3=1.6e-4;

m4=0.105;
Ix4=0.5e-4;
Iy4=0.8e-4;
Iz4=0.5e-4;

m5=1.143;
Ix5=37e-4;
Iy5=195.5e-4;
Iz5=231.2e-4;

m6=0.252;
Ix6=3.3e-4;
Iy6=2.8e-4;
Iz6=1.6e-4;

m7=0.105;
Ix7=0.5e-4;
Iy7=0.8e-4;
Iz7=0.5e-4;

k=100;

par = [grav theta_int u a s e b d R k m2 m3 m4 m5 m6 m7 Ix2 Iy2 Iz2 Ix3 Iy3 Iz3 Ix4
Iy4 Iz4 Ix5 Iy5 Iz5 Ix6 Iy6 Iz6 Ix7 Iy7 Iz7];

m = zeros(18,18);

```

```

m(1,1) = m2;
m(2,2) = m2;
m(3,3) = m2;

m(4,4) = m3;
m(5,5) = m3;
m(6,6) = m3;

m(7,7) = m4;
m(8,8) = m4;
m(9,9) = m4;

m(10,10) = m5;
m(11,11) = m5;
m(12,12) = m5;

m(13,13) = m6;
m(14,14) = m6;
m(15,15) = m6;

m(16,16) = m7;
m(17,17) = m7;
m(18,18) = m7;

I=zeros(18,18);

I(1,1) = Ix2;
I(2,2) = Iy2;
I(3,3) = Iz2;

I(4,4) = Ix3;
I(5,5) = Iy3;
I(6,6) = Iz3;

I(7,7) = Ix4;
I(8,8) = Iy4;
I(9,9) = Iz4;

I(10,10) = Ix5;
I(11,11) = Iy5;
I(12,12) = Iz5;

I(13,13) = Ix6;
I(14,14) = Iy6;
I(15,15) = Iz6;

I(16,16) = Ix7;
I(17,17) = Iy7;
I(18,18) = Iz7;

Mgorro=[m zeros(18);zeros(18) I];

ts=0:0.01:10;
q0=[0;0;0;0;0;0;0.0;0;0;0;0;0;0;0];
v = 5;

options = optimset('Display','iter');

X0 = fsolve(@X)
C([q0(1:2);X(1);q0(4);X(2);q0(6:11);X(3);X(4)],par),[q0(3);q0(5);q0(12:13,1)],options
)
q0(3) = X0(1);
q0(5) = X0(2);
q0(12) = X0(3);
q0(13) = X0(4);

```

```
P=D(q0',par);
Ddep=[P(:,1),P(:,3),P(:,5),P(:,8:9),P(:,11:13)];
Dindep=[P(:,2),P(:,4),P(:,6:7),P(:,10)];
qpunto0indep=[0;0;0;0;v/R];
qpunto0dep=Ddep\(-Dindep*qpunto0indep);
qpunto0=[
    qpunto0dep(1);
    qpunto0indep(1);
    qpunto0dep(2);
    qpunto0indep(2);
    qpunto0dep(3);
    qpunto0indep(3:4);
    qpunto0dep(4:5);
    qpunto0indep(5);
    qpunto0dep(6:8)]
rank(Ddep)

[t,y]=ode45(@ (t,X) ecs(t,X,Mgorro,par),ts,[q0;qpunto0]);

for j=1:length(t)
    z(j,:)=(multiplicadores(y(j,:),Mgorro,par))';
end
```

