

Trabajo de Fin de Grado
Grado en Ingeniería de las Tecnologías
Industriales

Implementación y banco de pruebas de un modelo
dinámico tridimensional de una bicicleta en Matlab.

Autor: Juan Antonio Clavijo González

Tutor: Joaquín Ojeda Granja

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

Sevilla, 2019



Trabajo de Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

Implementación y banco de pruebas de un modelo dinámico tridimensional de una bicicleta en Matlab.

Autor:

Juan Antonio Clavijo González

Tutor:

Joaquín Ojeda Granja

Profesor contratado doctor interino

Dpto. Ingeniería Mecánica y Fabricación

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2019

Trabajo de Fin de Grado: Implementación y banco de pruebas de un modelo dinámico tridimensional de una bicicleta en Matlab.

Autor: Juan Antonio Clavijo González

Tutor: Joaquín Ojeda Granja

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2019

El Secretario del Tribunal

A mi familia

A mis maestros

Agradecimientos

A mi familia, que me han dado todo su apoyo durante el desarrollo del este trabajo.

A mis amigos, que me han motivado y animado cuando más lo necesitaba.

A mi tutor, por su interés y dedicación en la realización de este trabajo.

En este trabajo presentamos la implementación y banco de pruebas de un modelo dinámico tridimensional de una bicicleta en Matlab.

Para realizar la implementación definimos el modelo teórico del sistema multicuerpo desarrollando el análisis de posición, el análisis cinemático y el análisis dinámico directo.

Se ha empleado el cálculo simbólico en Matlab como herramienta para la realizar las simulaciones del modelo tridimensional de la bicicleta.

Al final del trabajo se presentan una serie de simulaciones del sistema multicuerpo en ciertos casos de interés y una descripción de cómo se está comportando la bicicleta.

Abstract

In this paper we present the implementation and test bench of a three-dimensional dynamic model of a bicycle in Matlab.

To make the final implementation of the theoretical model of the multibody system, position analysis, kinematic analysis and direct dynamic analysis.

The symbolic calculation in Matlab has been used as a tool to perform the simulations of the three-dimensional model of the bicycle.

At the end of the work a series of simulations of the multibody system is presented in certain cases of interest and a description of how the bicycle is behaving.

... -translation by Google-

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Índice	xv
Índice de Figuras	xvii
Notación	xix
1 Introducción	21
2 Descripción del problema	23
2.1 <i>Coordenadas</i>	23
2.2 <i>Orientación y posición de los sólidos</i>	24
2.3 <i>Velocidades angulares y de traslación de los sólidos</i>	26
2.4 <i>Cálculo de aceleraciones</i>	27
2.5 <i>Restricciones</i>	28
2.5.1 <i>Restricciones de contacto</i>	28
2.5.2 <i>Restricciones de rodadura sin deslizamiento</i>	29
2.5.3 <i>Grados de libertad</i>	30
3 Dinámica directa	31
3.1 <i>Ecuación de movimiento</i>	31
3.1.1 <i>Matriz de masa y fuerza centrífuga</i>	31
3.1.2 <i>Fuerzas generalizadas</i>	32
3.1.3 <i>Formulación en coordenadas independientes</i>	34
3.1.4 <i>Estabilización de Baumgarte</i>	36
4 Implementación en Matlab	39
4.1 <i>Bloque simbólico</i>	39
4.2 <i>Bloque de la dinámica directa</i>	40
4.2.1 <i>Análisis de posición</i>	40
4.2.2 <i>Análisis cinemático</i>	41
4.2.3 <i>Análisis dinámico</i>	41
4.2.4 <i>Controlador de la bicicleta</i>	41
4.3 <i>Bloque de simulación</i>	41
4.4 <i>Conclusión</i>	<i>¡Error! Marcador no definido.</i>
5 Simulaciones	11
5.1 <i>Caso 1: Trayectoria recta</i>	11
5.1.1 <i>Caso 1.2: Trayectoria recta con velocidad inicial</i>	12
5.2 <i>Caso 2: Rampa</i>	13
5.2.1 <i>Caso 2.1: Bicicleta parada en una rampa</i>	13
5.2.2 <i>Caso 2.1: Bicicleta circulando en una rampa</i>	13
5.3 <i>Caso 3: Trayectoria curva</i>	15
6 Matlab	21
6.1 <i>Bicicleta_simb</i>	21

6.2	<i>dinBicicleta</i>	26
6.3	<i>simula_Bicicleta</i>	29
7	Bibliografía	35

ÍNDICE DE FIGURAS

Figura 2-1. Modelo teórico de la bicicleta.	23
Figura 2-2. Esquema de dos sólidos en contacto.	25
Figura 3-1. Pares ejercidos por el ciclista.	33
Figura 5-1. Bicicleta circulando en línea recta.	11
Figura 5-2. Comprobación de las ecuaciones de restricción.	12
Figura 5-3. Evolución de las velocidades.	12
Figura 5-4. Evolución de la coordenada x_c .	13
Figura 5-5. Esto es el pie de la figura en llano.	14
Figura 5-5. Evolución de las velocidades en rampa.	14
Figura 5-6. Bicicleta describiendo una trayectoria curva.	15
Figura 5-8. Evolución del ángulo de vuelco en una curva.	16

\sin	Función seno
\cos	Función coseno
q	Coordenadas generalizadas
\dot{q}	Velocidades generalizadas
\ddot{q}	Aceleraciones generalizadas
r	Vector posición
v	Vector velocidad
a	Vector aceleración
$\bar{\omega}$	Velocidad angular en coordenadas angulares
α	Aceleración angular
$\bar{\alpha}$	Aceleración angular en locales
A	Matriz de giro
\dot{x}	Derivada de x respecto al tiempo
\bar{r}	Vector posición en locales
A^T	Matriz de giro traspuesta

1 INTRODUCCIÓN

Un sistema multicuerpo es un conjunto de sólidos rígidos y flexibles interconectados por un conjunto de uniones. La dinámica de los sistemas multicuerpos (DSM) busca desarrollar técnicas que nos permitan estudiar el comportamiento de la amplia variedad de sistemas mecánicos con los que convivimos y empleamos día a día.

Las técnicas analíticas de Newton, D'Alembert y Lagrange, antiguamente servían para plantear problemas dinámicos de una cantidad de cuerpos no muy grande, ya que el cálculo se hacía muy complejo o imposible. Con el paso del tiempo, el desarrollo de las tecnologías abrió un camino para poder plantear estos problemas de manera que, haciendo uso de métodos numéricos y matriciales, permitían analizar sistemas que antes eran inabarcables.

La eficiencia de los programas informáticos desarrollados para el análisis DSM depende de varios factores, como de la selección de las coordenadas o del planteamiento numérico que se elige para resolver las ecuaciones de la dinámica. El número de coordenadas influye directamente en el grado de no linealidad de las ecuaciones de movimiento. Un número reducido de coordenadas para describir nuestro sistema nos lleva a que a las ecuaciones que lo describen, posean un alto grado de no linealidad.

Para afrontar el estudio de la dinámica multicuerpo, hay dos formulaciones, desarrolladas en el documento [2], de gran importancia que se emplean en este tipo de análisis. La primera de ellas, y la que se ha empleado en este trabajo, permite obtener las ecuaciones de movimiento del sistema en función de los grados de libertad y de manera que las fuerzas de reacción no aparezcan. Esta formulación se llama *Embedding Technique*. Otra formulación, muy usada también, es *Augmented Formulation* (Formulación Aumentada). En esta formulación, las ecuaciones de movimiento están expresadas en función de un conjunto de coordenadas que no son independientes. Debido a esta dependencia entre las variables, tenemos que plantear las ecuaciones de restricción que describen las relaciones entre los sólidos del sistema. Esto hace que en nuestras ecuaciones de la dinámica aparezcan las fuerzas de reacción.

Ambas técnicas son válidas para plantear cualquier sistema multicuerpo que nos interese analizar, pero evidentemente, cada una de ellas tienen sus ventajas y desventajas. El *Embedding Technique* nos conduce a un conjunto de ecuaciones reducidas que están en función únicamente de los grados de libertad del sistema y en las que no aparecen las fuerzas de reacción. Debido a esta reducción de las ecuaciones, el grado de no linealidad de estas es bastante alto, lo que hace que el coste computacional a la hora de resolver las ecuaciones de movimiento, también sea alto. En cambio, la Formulación Aumentada, el número de ecuaciones aumenta, pero la no linealidad de estas es mucho menor, lo que hace que el coste computacional también sea menor. A pesar de que el coste computacional se reduzca, la complejidad a la hora de desarrollar un algoritmo que nos permita llegar a estas ecuaciones es bastante alta, además de que, en las ecuaciones de movimiento, el número de variables es mayor y aparecen las fuerzas de reacción que hay entre los sólidos del sistema multicuerpo.

En este trabajo empleamos el *Embedding Technique*, ya que, a pesar de su alto coste computacional, el planteamiento de las ecuaciones es más sencillo, además de que las fuerzas de reacción del sistema no intervienen en la resolución de las ecuaciones de movimiento. También, parece interesante ver, cómo influyen de manera directa las acciones del ciclista sobre la bicicleta. Todas las técnicas de formulación de las ecuaciones de movimiento se describen en el documento [2].

En este trabajo se presenta un simulador tridimensional de una bicicleta programado en Matlab. Matlab es un programa de cálculo numérico. Dentro de sus muchas herramientas nos permite usar variables simbólicas para el desarrollo de algoritmos. El cálculo simbólico ofrece una gran eficiencia a nivel de planteamientos de algoritmos que con el cálculo numérico no se pueden realizar. Pero el coste computacional de este último es mucho menor que el del cálculo simbólico.

En este trabajo empleamos el cálculo simbólico para definir la geometría, la cinemática y dinámica del problema y, seguidamente, recurrimos a cálculo numérico para resolver el conjunto de ecuaciones diferenciales de segundo orden, obteniendo como resultado el movimiento de la bicicleta en distintos casos de interés.

Este texto está redactado de la siguiente manera. Primero definiremos nuestro modelo geométrico y cinemático. Después veremos cuáles son las fuerzas que intervienen durante el movimiento de la bicicleta. A continuación, se desarrollarán las ecuaciones de movimiento empleando el *Embedding Technique*, como adelantamos anteriormente. Por último, analizaremos el movimiento de la bicicleta en función de las acciones del ciclista y de las condiciones iniciales que impongamos.

2 DESCRIPCIÓN DEL PROBLEMA

El primer paso a la hora de estudiar cualquier sistema es la descripción del modelo. El sistema multicuerpo que aquí se presenta es una bicicleta. Es necesario definir su geometría y cinemática antes de abordar el análisis dinámico directo. En este capítulo se detalla el planteamiento para la obtención de las posiciones, velocidades, aceleraciones y grados de libertad del sistema. Toda la descripción del problema viene descrita en el documento [4].

2.1 Coordenadas

La Figura 2-1 nos muestra un esquema de la bicicleta en una posición arbitraria en el espacio. La bicicleta se ha desglosado en 4 sólidos móviles que hemos considerado rígidos, ya que las deformaciones de estos no son objeto de estudio en este trabajo. Los sólidos en los que se ha dividido la bicicleta son: la rueda trasera (sólido 2), el cuadro de la bicicleta y el cuerpo del ciclista (sólido 3), el manillar (sólido 4) y la rueda delantera (sólido 5).

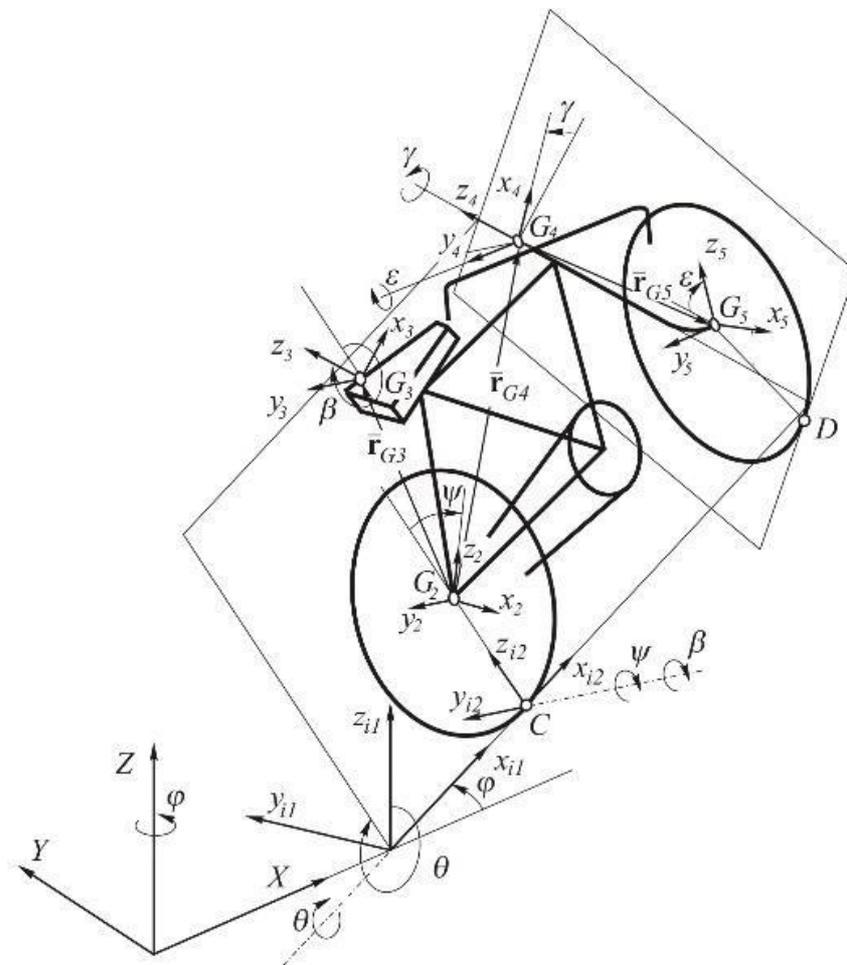


Figura 2-1. Modelo teórico de la bicicleta.

El sistema al que se refieren el resto de sólidos es el sistema de referencia global e inercial $\langle X Y Z \rangle$. Los sistemas $\langle x_{i1} y_{i1} z_{i1} \rangle$ y $\langle x_{i2} y_{i2} z_{i2} \rangle$ sistemas de referencia intermedios que definen la posición y orientación del plano que contiene al sólido 3, el cuadro de la bicicleta. El resto de sistemas de referencia están rígidamente unidos al centro de gravedad de cada sólido, $\langle x_i y_i z_i \rangle$, siendo $i = 2, 3, 4$ y 5 , como se

muestra en el esquema de la Figura 2-1.

El conjunto de coordenadas que se ha utilizado para describir la posición y orientación de la bicicleta es el siguiente:

- Las coordenadas x_C e y_C determinan la posición del punto C , que es punto de contacto de la rueda trasera de la bicicleta con el suelo, es decir, con el plano $\langle X Y \rangle$ del sistema de referencia global.
- El ángulo de guiñada (yaw) φ que forma el eje x_{i1} del sistema intermedio de referencia con el eje X del sistema global de referencia. Se trata de un giro alrededor del eje Z .
- El ángulo de vuelco (roll) θ que forma el eje z_{i2} del sistema intermedio de referencia con el eje z_{i2} del sistema global de referencia. Se trata de un giro alrededor del eje x_{i1} .
- Los ángulos φ y θ determinan la orientación del plano que contiene al cuadro de la bicicleta.
- El ángulo de rodadura (pitch) Ψ que forma el eje z_2 del sistema de referencia asociado a la rueda trasera con el eje z_{i2} del segundo sistema intermedio de referencia. Se trata de un giro alrededor del eje y_{i2} .
- El ángulo β que forma el eje z_3 del sistema de referencia del cuadro con el eje z_{i2} del segundo sistema intermedio de referencia. Se trata de un giro alrededor del eje y_2 .
- El ángulo de dirección γ que forma el eje x_4 del sistema de referencia del manillar con el eje x_3 del sistema de referencia del cuadro. Se trata de un giro alrededor del eje z_3 .
- El ángulo de rodadura (pitch) ε que forma el eje z_5 del sistema de referencia asociado a la rueda delantera con el eje z_4 del sistema de referencia del manillar. Se trata de un giro alrededor del eje y_4 .

Finalmente, el conjunto de coordenadas que se ha usado para la resolución del análisis se agrupa en el vector:

$$q = [x_c \quad y_c \quad \varphi \quad \theta \quad \Psi \quad \beta \quad \gamma \quad \varepsilon]^T \quad (2-1)$$

Este modelo está basado en los documentos [3] y [4].

2.2 Orientación y posición de los sólidos

En los sistemas multicuerpo, para poder definir la posición de cada uno de los puntos de los sólidos que componen el sistema en el espacio, debemos poder determinar la posición de cada punto respecto al sistema global de referencia. Para ello, como ya se ha adelantado antes, es conveniente que definamos un sistema de referencia para cada uno de los sólidos de los que está formado nuestro sistema multicuerpo. De este modo la posición de cada punto de los sólidos, se definirá de manera local, y después se referenciará al sistema global definiendo la posición y orientación relativa del sistema de referencia local del sólido respecto del resto [6].

Primero, debemos definir las matrices de giro de los distintos sistemas de referencia con respecto al sistema global de coordenadas:

$$A^{i1} = A_\varphi \quad (2-2)$$

$$A^{i2} = A_\varphi A_\theta \quad (2-3)$$

$$A^2 = A_\varphi A_\theta A_\Psi \quad (2-4)$$

$$A^3 = A_\varphi A_\theta A_\beta \quad (2-5)$$

$$A^4 = A_\varphi A_\theta A_\beta A_\gamma \quad (2-6)$$

$$A^5 = A_\varphi A_\theta A_\beta A_\gamma A_\varepsilon \quad (2-7)$$

Donde cada una de las matrices se de definen como:

$$A_\varphi = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2-8)$$

$$A_\varphi = \begin{bmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2-9)$$

$$A_\theta = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \quad (2-10)$$

$$A_\Psi = \begin{bmatrix} \cos \Psi & 0 & \sin \Psi \\ 0 & 1 & 0 \\ -\sin \Psi & 0 & \cos \Psi \end{bmatrix} \quad (2-11)$$

$$A_\beta = \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \quad (2-12)$$

$$A_\gamma = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2-13)$$

$$A_\varepsilon = \begin{bmatrix} \cos \varepsilon & 0 & \sin \varepsilon \\ 0 & 1 & 0 \\ -\sin \varepsilon & 0 & \cos \varepsilon \end{bmatrix} \quad (2-14)$$

Como podemos observar en la Figura 2-2, para definir la posición de un punto de uno de los sólidos respecto del sistema global de referencia, necesitamos: la posición del punto en el sistema de referencia local del sólido, la orientación del sólido respecto del sistema de referencia al que está definido, y la posición del origen del sistema de referencia local respecto del sistema de referencia global.

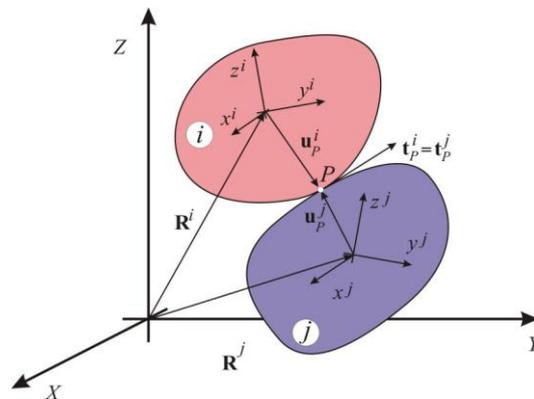


Figura 2-2. Esquema de dos sólidos en contacto.

Tal y como hemos descrito en el apartado 2.1, los sistemas de referencia locales estarán rígidamente unidos al centro de gravedad de cada uno de los sólidos. La posición de los centros de gravedad de los sólidos quedan reflejadas en las siguientes ecuaciones:

$$r_{G2} = r_C + A^{i2} \bar{r}_{G2}^{i2} = \begin{bmatrix} x_c \\ y_c \\ 0 \end{bmatrix} + A^{i2} \begin{bmatrix} 0 \\ 0 \\ R_t \end{bmatrix} \quad (2-15)$$

$$r_{G3} = r_{G2} + A^3 \bar{r}_{G3}^2 = r_{G2} + A^3 \begin{bmatrix} x_3 \\ 0 \\ z_3 \end{bmatrix} \quad (2-16)$$

$$r_{G4} = r_{G3} + A^3 \bar{r}_{G4}^2 = r_{G3} + A^3 \begin{bmatrix} x_4 \\ 0 \\ z_4 \end{bmatrix} \quad (2-17)$$

$$r_{G5} = r_{G4} + A^4 \bar{r}_{G5}^4 = r_{G4} + A^4 \begin{bmatrix} x_5 \\ 0 \\ z_5 \end{bmatrix} \quad (2-18)$$

Los valores de R y x_3, y_3, x_4, y_4, x_5 e y_5 son constantes porque están en coordenadas locales, como consecuencia sus derivadas son nulas y no afectan en el cálculo de las velocidades. El radio de la bicicleta es la constante R y x_3, y_3, x_4, y_4, x_5 e y_5 son las componentes de los vectores que indican la posición del centro de gravedad en coordenadas locales.

Con lo anteriormente expuesto, quedan definidas las orientaciones y posiciones de los sólidos de nuestro sistema multicuerpo.

Ahora pasamos a calcular las velocidades angulares y las velocidades de traslación de cada uno de los sólidos.

2.3 Velocidades angulares y de traslación de los sólidos

Para el cálculo de las velocidades angulares de los sólidos, primero hay que calcular la derivada temporal de la matriz de giro:

$$\dot{A}^i = \frac{dA^i}{dt} = \frac{\partial A^i}{\partial q} \frac{\partial q}{\partial t} = \frac{\partial A^i}{\partial q} \dot{q} \quad (2-19)$$

Con la derivada temporal y sabiendo que $\dot{A}^i A^{iT} = \tilde{w}$, siendo \tilde{w} una matriz antisimétrica, podemos calcular la velocidad angular del sólido i . Reordenando las componentes de la matriz antisimétrica obtenemos el vector de velocidades angulares.

Existe una manera alternativa de representar el vector de velocidad angular. Definimos la matriz de transformación de velocidad angular del sólido i , como:

$$G^i(q) = \frac{\partial w^i}{\partial \dot{q}} \quad (2-20)$$

Con esta matriz obtenemos una expresión más compacta de la velocidad angular del sólido i .

$$w^i = G^i(q) \dot{q} \quad (2-21)$$

Para expresar el vector en coordenadas locales premultiplicamos el vector de velocidad angular del sólido i por la matriz de giro del mismo sólido.

$$\bar{w}^i = A^{iT} w^i \quad (2-22)$$

Haciendo uso de la definición (2-20), la matriz de transformación de velocidades angulares en coordenadas locales se calcula de la siguiente manera:

$$\bar{G}^i = \frac{\partial \bar{w}^i}{\partial \dot{q}} \quad (2-23)$$

La expresión compacta quedaría como sigue:

$$\bar{w}^i = \bar{G}^i(q)\dot{q} \quad (2-24)$$

Con esto hemos calculado las velocidades angulares de cada uno sólidos de la bicicleta.

Las velocidades de traslación de los sólidos se calculan a partir de la derivada temporal de la posición absoluta de los centro de gravedad de los sólidos tal y como se refleja en la siguiente ecuación:

$$\frac{dr_{Gi}}{dt} = \frac{\partial r_{Gi}}{\partial q} \frac{\partial q}{\partial t} \quad (2-25)$$

Como el objetivo de este trabajo es la implementación del desarrollo del problema en Matlab, definimos la matriz de transformación de velocidades H^i como:

$$H^i = \frac{\partial r_{Gi}}{\partial q} \quad (2-26)$$

Lo que permite obtener una expresión más compacta de la velocidad de traslación del sólido i .

$$v_{Gi} = H^i \dot{q} \quad (2-27)$$

Con esto queda definida la posición de cada uno de los puntos del sistema, la velocidad de traslación y angular de la bicicleta.

2.4 Cálculo de aceleraciones

Para calcular las aceleraciones angulares y de traslación derivamos las velocidades angulares y de traslación respecto del tiempo:

$$\frac{dv_{Gi}}{dt} = \frac{dH^i \dot{q}}{dt} = \frac{\partial H^i \dot{q}}{\partial q} \frac{\partial q}{\partial t} + \frac{d\dot{q}}{dt} H^i \quad (2-28)$$

$$\frac{d\bar{w}^i}{dt} = \frac{d\bar{G}^i \dot{q}}{dt} = \frac{\partial \bar{G}^i \dot{q}}{\partial q} \frac{\partial q}{\partial t} + \frac{d\dot{q}}{dt} \bar{G}^i = \frac{\partial \bar{w}^i}{\partial q} \frac{\partial q}{\partial t} + \frac{d\dot{q}}{dt} \bar{G}^i \quad (2-29)$$

Al igual que hicimos con las velocidades, definimos las matrices h y g para que las ecuaciones se puedan expresar de una manera más reducida.

La matriz h es el jacobiano de la matriz de transformación de velocidades respecto del vector de coordenadas generalizadas.

$$h^i = \frac{\partial H^i \dot{q}}{\partial q} \quad (2-30)$$

La matriz g es el jacobiano de la matriz de transformación de velocidades angulares respecto del vector de coordenadas generalizadas

$$\bar{g}^i = \frac{\partial \bar{w}^i}{\partial q} \quad (2-31)$$

Finalmente, las aceleraciones en función de las coordenadas generalizadas queda como sigue,

$$\begin{aligned} a_G^i &= H^i \ddot{q} + h^i \dot{q} \\ \bar{\alpha}^i &= \bar{G}^i \ddot{q} + \bar{g}^i \dot{q} \end{aligned} \quad (2-32)$$

Pasamos a definir las restricciones que dominan nuestro sistema multicuerpo.

2.5 Restricciones

Las restricciones son condiciones que ha de cumplir el sistema multicuerpo en todo momento. Estas condiciones son ecuaciones algebraicas, en general no lineales, que deben de satisfacer las coordenadas en todo instante.

En la bicicleta hay dos tipos de restricciones cinemáticas: de contacto y de rodadura sin deslizamiento.

2.5.1 Restricciones de contacto

Las restricciones de contacto, son aquellas que garantizan que la bicicleta mantiene el contacto de ambas ruedas siempre con el suelo.

En la definición de nuestro modelo, el contacto con el suelo de la rueda trasera ya está garantizado, ya que no tiene componente z , por lo tanto no requiere de restricciones.

En el caso de la rueda delantera la situación es distinta.

Para un caso general de dos sólidos i y j en contacto, Figura 2-2, la restricciones de contacto deben cumplir dos condiciones:

1. Que en el punto de contacto P , un punto de la superficie i y otro punto de la superficie j , ocupen la misma posición.
2. Que el plano tangente del sólido i en el punto P , sea paralelo al plano tangente del sólido j en el mismo punto.

La posición de un punto cualquiera de la rueda delantera, en el sistema de referencia local, se escribe como:

$$\bar{r}_P^5 = \begin{bmatrix} R \cos \xi \\ 0 \\ -R \sin \xi \end{bmatrix}, \xi \in [0 \quad 2\pi] \quad (2-33)$$

donde ξ es un parámetro, angular en este caso, como se observa en la figura 2-1, que determina la orientación del radio que contiene al punto P con respecto al eje x^5 . Las componentes de un vector tangente a la rueda en el punto P en el sistema de referencia local, se calculan como:

$$\bar{t}_P^5 = \frac{\partial \bar{r}_P^5}{\partial \xi} \begin{bmatrix} -R \sin \xi \\ 0 \\ -R \cos \xi \end{bmatrix} \quad (2-34)$$

La posición absoluta de un punto cualquiera de la rueda delantera y las componentes del vector tangente en el sistema de referencia global se calculan como:

$$r_p(\xi) = r_{G5} + A^5 \bar{r}_P^5 \quad (2-35)$$

$$t_p(\xi) = A^5 \bar{t}_P^5 \quad (2-36)$$

Ya que la restricción holonoma obliga a que la rueda delantera del sólido esté siempre en contacto con el suelo, es decir, con el plano $X - Y$, la componente Z del vector posición del punto de contacto debe ser nula y

la componente Z del vector tangente a la rueda en el punto de contacto sea también nula. Estas restricciones se escriben algebraicamente de la siguiente forma:

$$\begin{aligned} [r_p(q, \xi_D)]_Z &= 0 \\ [t_p(q, \xi_D)]_Z &= 0 \end{aligned} \quad (2-37)$$

D representa el punto de contacto de la rueda delantera con el suelo, ξ_D el parámetro angular asociado a dicho punto de contacto.

Las dos ecuaciones algebraicas dadas pueden ser desarrolladas en términos de las coordenadas del sistema q y ξ_D . Es importante observar que aunque el contacto de la rueda delantera introduce las dos restricciones, no se eliminan dos grados de libertad del sistema, puesto que también se ha introducido la incógnita adicional ξ_D . Ya que al ser ambas ecuaciones no lineales, no hay posibilidad de manipularlas algebraicamente para eliminar la coordenada ξ_D . Por tanto se opta por introducir una nueva coordenada en el sistema, que denominamos ξ , que representa el parámetro angular asociado al punto de contacto de la rueda delantera.

Finalmente el vector de restricciones holónomas y el vector coordenadas generalizadas se reescriben como sigue:

$$C^{con}(p) = \begin{bmatrix} [r_p(q, \xi_D)]_Z \\ [t_p(q, \xi_D)]_Z \end{bmatrix} = 0 \quad (2-38)$$

Redefinimos entonces el vector de coordenadas generalizadas como:

$$p = [x_c \quad y_c \quad \varphi \quad \theta \quad \Psi \quad \beta \quad \gamma \quad \varepsilon \quad \xi]^T \quad (2-39)$$

2.5.2 Restricciones de rodadura sin deslizamiento

Las restricciones no holónomas nos imponen que la velocidad del punto de contacto de la rueda con el suelo debe de ser, en todo instante, nula. Estas restricciones se deben de cumplir para las dos ruedas. Para el caso de la rueda trasera, la velocidad del punto C viene dada por:

$$v_C = v_{G_2} + \omega^2 \times r_{G_2C} \quad (2-40)$$

El vector r_{G_2C} contiene las componentes globales del vector que va de G_2 a C . Desarrollando la ecuación resulta:

$$v_C = \begin{bmatrix} \dot{x}_C - R \cos \varphi \dot{\Psi} \\ \dot{y}_C - R \sin \varphi \dot{\Psi} \\ 0 \end{bmatrix} \quad (2-41)$$

Como la componente Z del vector v_C es idénticamente nula, para imponer la restricción de rodadura sin deslizamiento basta con garantizar que las componentes X e Y de dicho vector sean nulas:

$$C^{rod,2}(p, \dot{p}) = \begin{bmatrix} \dot{x}_C - R \cos \varphi \dot{\Psi} \\ \dot{y}_C - R \sin \varphi \dot{\Psi} \end{bmatrix} = 0 \quad (2-42)$$

$C^{rod,2}(p, \dot{p})$ es el nombre genérico que se asigna a las restricciones de rodadura sin deslizamiento de la rueda trasera. Esta ecuación pueden ser escrita de la siguiente forma matricial:

$$B^2(p)\dot{p} = 0 \quad (2-43)$$

Donde la matriz B^2 , que depende de las coordenadas p , se puede obtener como la matriz jacobiana de las

restricciones con respecto a las velocidades del sistema, esto es:

$$B^2 = \frac{\partial C^{rod,2}}{\partial \dot{p}} \quad (2-44)$$

Para la rueda delantera, la velocidad del punto de contacto D se calcula como:

$$v_D = v_{G5} + w^5 \times r_{G5D} \quad (2-45)$$

Si se desarrolla la ecuación se observa que, en este caso, la tercera componente de la velocidad del punto D no es idénticamente nula. Sin embargo, se puede demostrar que si las coordenadas del sistema satisfacen las restricciones de contacto $C(q) = 0$ esta tercera componente si resulta ser nula. Por tanto las restricciones de rodadura sin deslizamiento de la rueda delantera deben garantizar que las componentes X e Y de la velocidad del punto de contacto D sean nulas:

$$C^{rod,5}(p, \dot{p}) = \begin{bmatrix} [v_D]_X \\ [v_D]_Y \end{bmatrix} = 0 \quad (2-46)$$

Esta ecuación puede ser escrita de la siguiente forma matricial:

$$B^5(p)\dot{p} = 0 \quad (2-47)$$

$$B^5 = \frac{\partial C^{rod,5}}{\partial \dot{p}} \quad (2-48)$$

Las restricciones de rodadura sin deslizamiento de las dos ruedas y sus matrices jacobianas se agrupan de la forma:

$$C^{rod}(p, \dot{p}) = \begin{bmatrix} C^{rod,2}(p, \dot{p}) \\ C^{rod,5}(p, \dot{p}) \end{bmatrix} = 0 \quad (2-49)$$

$$B(p) = \begin{bmatrix} B^2(p) \\ B^5(p) \end{bmatrix} \quad (2-50)$$

Finalmente las restricciones de rodadura sin deslizamiento son un conjunto de 4 ecuaciones algebraicas que dependen de las coordenadas y velocidades del sistema. Por tanto, la condición de rodadura sin deslizamiento elimina 4 grados de libertad del sistema.

2.5.3 Grados de libertad

El conjunto de restricciones de la bicicleta se agrupa de la forma:

$$C(p, \dot{p}, t) = \begin{bmatrix} C^{con}(p) \\ C^{rod}(p, \dot{p}) \end{bmatrix} = 0 \quad (2-51)$$

La descripción cinemática de la bicicleta requiere 9 coordenadas (las 8 del vector q más el parámetro del punto de contacto de la rueda delantera ξ) y está sujeta a un conjunto de 6 restricciones (2 de contacto y 4 de rodadura sin deslizamiento). Por tanto, la bicicleta así descrita tiene 3 grados de libertad.

3 INÁMICA DIRECTA

3.1 Ecuación de movimiento

En el estudio de los sistemas multicuerpo hay diferentes técnicas para plantear la ecuación de movimiento que describe el comportamiento de nuestro sistema. En el texto de referencia se plantea la ecuación de movimiento a partir de las ecuaciones de Newton-Euler y se desarrollan las todas las fuerzas generalizadas [3]. Pero en este documento se pretende que la ecuación que defina el movimiento de nuestro sistema no dependa de las fuerzas de restricción, ni las coordenadas dependientes. Con este objetivo en mente, vamos emplear una técnica matricial llamada *Embedding Technique* [2], con la que eliminaremos la dependencia de las reacciones y de las coordenadas dependientes y dejaremos la ecuación de movimiento únicamente en función de las aceleraciones independientes.

3.1.1 Matriz de masa y fuerza centrífuga

Para el cálculo de las matrices de masa y de los vectores de fuerza generalizada centrífuga emplearemos la formulación de Newton-Euler [2]. En esta formulación se asume que el origen de cada uno de los sistemas de coordenadas de cada uno de los sólidos, es el centro de masa del mismo. De esta manera las ecuaciones de Newton-Euler para cada uno de los 4 sólidos rígidos que se supone que forman la bicicleta son:

$$\begin{bmatrix} m^i & 0 \\ 0 & \bar{I}^i \end{bmatrix} \begin{bmatrix} a_G^i \\ \bar{\alpha}^i \end{bmatrix} = \begin{bmatrix} F^i \\ \bar{M}^i \end{bmatrix} \begin{bmatrix} 0 \\ -\bar{\omega}^i \times \bar{I}^i \bar{\omega}^i \end{bmatrix} \quad (3-1)$$

El primer término de la izquierda es la matriz de masa del sólido, que la definimos como matriz \hat{M} . El segundo término a la izquierda de la igualdad son las aceleraciones angulares y de traslación. Los términos a la derecha de la igualdad son las fuerzas externas y centrífugas, que son las matrices \hat{Q} y \hat{Q}_v respectivamente.

Como ya definimos en el subcapítulo del cálculo de aceleraciones, las aceleraciones angulares y de traslación se escriben según las ecuaciones (2-32). Vamos a ponerlas de manera matricial.

$$\begin{bmatrix} a_G^2 \\ \vdots \\ a_G^5 \\ \bar{\alpha}^2 \\ \vdots \\ \bar{\alpha}^5 \end{bmatrix} = \begin{bmatrix} H^2 \\ \vdots \\ H^5 \\ \bar{G}^2 \\ \vdots \\ \bar{G}^5 \end{bmatrix} \ddot{p} + \begin{bmatrix} h^2 \\ \vdots \\ h^5 \\ \bar{g}^2 \\ \vdots \\ \bar{g}^5 \end{bmatrix} \dot{p} \quad (3-2)$$

De esta manera tenemos el vector aceleraciones representado de forma matricial y en función de las coordenadas del problema. Ahora vamos a sustituir el vector matricial dentro de las ecuaciones de Newton-Euler.

Siendo,

$$L = \begin{bmatrix} H^2 \\ \vdots \\ H^5 \\ \bar{G}^2 \\ \vdots \\ \bar{G}^5 \end{bmatrix} \text{ y } l = \begin{bmatrix} h^2 \\ \vdots \\ h^5 \\ \bar{g}^2 \\ \vdots \\ \bar{g}^5 \end{bmatrix} \quad (3-3)$$

y sustituyendo

$$\hat{M}(L\ddot{p} + l\dot{p}) = \hat{Q} + \hat{Q}_v \quad (3-4)$$

Premultiplicando la ecuación por la matriz L^T y reagrupando términos resultan las ecuaciones de Newton-Euler para la bicicleta en términos de las coordenadas p , como sigue:

$$M\ddot{p} = Q + Q_v \quad (3-5)$$

donde:

$$M = L^T \hat{M} L, Q = L^T \hat{Q}, Q_v = L^T (\hat{Q} - \hat{M} l \dot{p}) \quad (3-6)$$

El vector de fuerzas generalizadas que aparece en la ecuación contiene las fuerzas aplicadas y las fuerzas de reacción generalizadas de tal forma que:

$$Q = Q_{ap} + Q_{reac} \quad (3-7)$$

Como vemos, nos aparecen las fuerzas de reacción generalizadas dentro de la ecuación de movimiento de la bicicleta. Veremos después que estas fuerzas desaparecerán de manera que, no intervendrán en el cálculo de las aceleraciones independientes de nuestro problema.

Pasamos ahora a desarrollar cada una de las fuerzas generalizadas que actúan sobre el sistema multicuerpo.

3.1.2 Fuerzas generalizadas

El vector Q_{ap} contiene las fuerzas generalizadas que están actuando sobre el sistema y se obtiene como:

$$Q_{ap} = Q_{grav} + Q_{areo} + Q_{accion} \quad (3-8)$$

donde Q_{grav} es la fuerza generalizada debida a la gravedad, Q_{areo} es la fuerza generalizada debida a la resistencia aerodinámica y Q_{accion} es la fuerza generalizada debida a la acción del ciclista en la bicicleta durante su conducción. Para calcular la expresión de los vectores de fuerza generalizada se va a utilizar el principio de las potencias virtuales. Según este principio, el vector de fuerza generalizada es el vector que multiplicado escalarmente por el vector de velocidades virtuales (\dot{p}^*) da lugar a la potencia virtual que desarrolla la fuerza (o par). A continuación se muestra el cálculo de estos vectores.

3.1.2.1 Fuerza generalizada de gravedad

La fuerza de gravedad generalizada es debida a la fuerza de gravedad que se aplica a cada uno de los sólidos. Los pesos de los sólidos móviles que forman la bicicleta son vectores aplicados en los centros de gravedad y que llevan dirección opuesta al eje vertical del sistema global de referencia.

$$F_{grav}^i = \begin{bmatrix} 0 \\ 0 \\ -m^i g \end{bmatrix}, i = 2, 3, 4 \text{ y } 5 \quad (3-9)$$

Por tanto, la potencia virtual que desarrollan los pesos se obtiene como:

$$\dot{W}_{grav}^* = \sum_{i=2}^5 (v_G^{i*})^T F_{grav}^i = (\dot{p}^*)^T \sum_{i=2}^5 H^{iT} F_{grav}^i \quad (3-10)$$

Donde, como podemos ver, se ha sustituido el vector velocidad por su identidad en función de las coordenadas generalizadas. El vector Q_{grav} queda entonces:

$$Q_{grav} = \sum_{i=2}^5 H^{iT} F_{grav}^i \quad (3-11)$$

3.1.2.2 Fuerza generalizada de resistencia aerodinámica

Para el cálculo de la fuerza generalizada de resistencia aerodinámica se va a suponer que la resistencia aerodinámica es proporcional al cuadrado de la velocidad de avance de la bicicleta y que está aplicada exclusivamente en el centro de gravedad del sólido 3, que forman el cuadro de la bicicleta y el ciclista. Esto es razonable por ser éste el sólido más voluminoso que ofrece la mayoría de la resistencia al aire. En ese caso, el módulo del vector de resistencia aerodinámica resulta:

$$|F_{areo}^3| = c_v (v_G^3)^T v_G^3 = c_v |v_G^3|^2 \quad (3-12)$$

y su dirección será la contraria a la de la velocidad del centro de gravedad del sólido 3, por tanto:

$$F_{areo}^3 = -|F_{areo}^3| \frac{v_G^3}{|v_G^3|} = -c_v |v_G^3| v_G^3 \quad (3-13)$$

La potencia virtual que desarrolla la fuerza aerodinámica resulta:

$$\dot{W}_{areo}^* = (v_G^{3*})^T F_{areo}^3 = (\dot{p}^*)^T H^{3T} F_{areo}^3 \quad (3-14)$$

El vector Q_{areo} queda entonces:

$$Q_{areo} = H^{3T} F_{areo}^3 = -c_v |v_G^3| H^{3T} v_G^3 \quad (3-15)$$

3.1.2.3 Fuerza generalizada de la acción del ciclista

En el desarrollo de este trabajo se ha supuesto que para manejar la bicicleta el ciclista ejerce tres pares de control, Figura 3-1. El par de dirección M_{dir} aplicado en dirección z_4 , el par de pedaleo M_{ped} aplicado en dirección y_2 y el par de vuelco M_{vue} aplicado en dirección x_3 . El valor de estos pares es en principio desconocido.

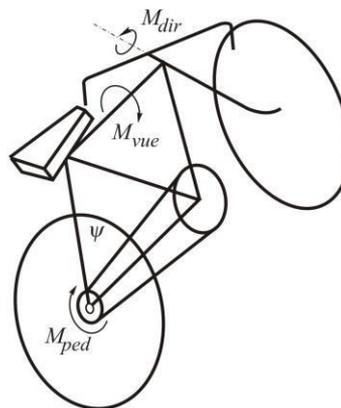


Figura 3-1. Pares ejercidos por el ciclista.

La potencia virtual de estos pares se calcula de la forma:

$$\begin{aligned} \dot{W}_{aero}^* &= (\bar{w}_4^*)^T \begin{bmatrix} 0 \\ 0 \\ M_{dir} \end{bmatrix} + (\bar{w}_2^*)^T \begin{bmatrix} 0 \\ M_{ped} \\ 0 \end{bmatrix} + (\bar{w}_3^*)^T \begin{bmatrix} M_{vue} \\ 0 \\ 0 \end{bmatrix} \\ &= (\dot{p}^*)^T \left(\bar{G}^4 \begin{bmatrix} 0 \\ 0 \\ M_{dir} \end{bmatrix} + \bar{G}^2 \begin{bmatrix} 0 \\ M_{ped} \\ 0 \end{bmatrix} + \bar{G}^3 \begin{bmatrix} M_{vue} \\ 0 \\ 0 \end{bmatrix} \right) \end{aligned} \quad (3-16)$$

El vector Q_{accion} se puede calcular como:

$$Q_{accion} = [\bar{G}^4 \quad \bar{G}^2 \quad \bar{G}^3]^T \begin{bmatrix} M_{vue} \\ M_{ped} \\ M_{dir} \end{bmatrix} = ST \quad (3-17)$$

donde la matriz S contiene las filas de las matrices \bar{G}^i y el vector T contiene los tres pares de acción del ciclista.

3.1.2.4 Fuerza de reacción generalizada

En un sistema mecánico sujeto a restricciones, la fuerza generalizada asociada a las restricciones de cada sólido, si únicamente está sujeto a restricciones holónomas, se puede obtener del siguiente modo:

$$Q_{reac} = -C_p^T \lambda \quad (3-18)$$

Donde C_p es la matriz jacobiana de las restricciones con respecto al vector coordenadas generalizadas y λ es el vector de multiplicadores de Lagrange. Ya que nuestro caso incluye tanto restricciones holónomas como no holónomas, las fuerzas de reacción se calculan como:

$$Q_{reac} = -D^T \lambda \quad (3-19)$$

donde D es la matriz jacobiana de las restricciones holónomas con respecto al vector coordenadas, y no holónomas con respecto al vector de velocidades generalizadas. En el caso de la bicicleta se trata de una matriz de coeficiente de las ecuaciones de restricción a nivel de velocidades.

$$D = \begin{bmatrix} C_p^{con} \\ B \end{bmatrix} \quad (3-20)$$

Finalmente, las ecuaciones de Newton-Euler de la bicicleta se pueden escribir de la forma:

$$M\ddot{p} + D^T \lambda = Q_{grav} + Q_{aero} + Q_{accion} \quad (3-21)$$

3.1.3 Formulación en coordenadas independientes

Ya hemos definido cada uno de los elementos que componen nuestra ecuación de movimiento. Pero nuestro objetivo es que la ecuación de movimiento no dependa ni de las fuerzas de reacción generalizadas, ni de las aceleraciones dependientes.

El primer paso que vamos a realizar es una separación de las coordenadas generalizadas del problema, entre dependientes e independientes. A las coordenadas independientes las llamaremos p_{din} , que son las que están asociadas a cada uno de los momentos que aplica el ciclista a la bicicleta. Luego, siguiendo la descripción del modelo expuesto en el documento de dinámica inversa, las dependientes las separaremos a su vez en dos grupos, las p_{kin} y las p_{dep} . Esta separación de las variables está basada en el documento [5]. Las coordenadas

p_{kin} son denominadas las coordenadas cinemáticas independientes. Con su desglose respecto de las coordenadas p_{dep} , dentro de las dependientes, podremos resolver el problema cinemático del sistema. Posteriormente, cuando analicemos los casos de interés en el apartado de Simulaciones, veremos que estas velocidades independientes serán datos de entrada.

La selección de las variables p_{kin} y p_{dep} se ha tomado como en el documento de referencia.

$$p_{din} = \begin{bmatrix} \theta \\ \Psi \\ \gamma \end{bmatrix}, p_{kin} = \begin{bmatrix} x_c \\ y_c \\ \varphi \\ \varepsilon \end{bmatrix}, p_{dep} = \begin{bmatrix} \beta \\ \xi \end{bmatrix} \quad (3-22)$$

Una vez que separamos las variables, pasamos a representar el vector D (3-23), como el jacobiano de cada grupo de coordenadas que se ha tomado:

$$D\dot{p} = 0 \quad (3-23)$$

Este sistema de ecuaciones pasa a representarse de la siguiente manera:

$$D_{din}\dot{p}_{din} + D_{kin}\dot{p}_{kin} + D_{dep}\dot{p}_{dep} = 0 \quad (3-24)$$

Donde D_{din} representa el jacobiano respecto del vector de posiciones p_{din} y el vector de velocidades \dot{p}_{din} , D_{kin} representa el jacobiano respecto del vector de posiciones p_{kin} y el vector de velocidades \dot{p}_{kin} , D_{dep} representa el jacobiano respecto del vector de posiciones p_{dep} y el vector de velocidades \dot{p}_{dep} . Ahora solo nos queda despejar las coordenadas dependientes en función de las coordenadas independientes.

$$-D_{din}\dot{p}_{din} = D_{kin}\dot{p}_{kin} + D_{dep}\dot{p}_{dep} \quad (3-25)$$

Agrupamos ahora las coordenadas \dot{p}_{kin} y \dot{p}_{dep} de manera que el sistema que nos queda es, por un lado las coordenadas independientes multiplicadas por el jacobiano de las restricciones respecto de las coordenadas independientes D_{din} y por otro lado una matriz cuadrada singular (ya que es las restricciones son linealmente independientes) formada por el jacobiano de las restricciones con respecto de las p_{kin} , \dot{p}_{kin} , p_{dep} y \dot{p}_{dep} . A esa matriz cuadrada se le va a llamar D_{kindep} , y tiene la siguiente estructura:

$$D_{kindep}\dot{p}_{kindep} = \begin{bmatrix} C_{pkin}^{con} & C_{pdep}^{con} \\ B_{\dot{p}kin} & B_{\dot{p}dep} \end{bmatrix} \begin{bmatrix} \dot{p}_{kin} \\ \dot{p}_{dep} \end{bmatrix} \quad (3-26)$$

Despejando las \dot{p}_{kindep} tenemos la ecuación que expresa las velocidades dependientes de nuestro sistema en función de las independientes. Esta relación nos permite resolver el problema cinemático de la bicicleta. Es decir,

$$\dot{p}_{kindep} = -D_{kindep}^{-1}D_{din}\dot{p}_{din} \quad (3-27)$$

Una vez resuelto el problema cinemático, continuamos con el desarrollo de las ecuaciones de restricción.

Volvemos a derivar respecto del tiempo, ahora queremos conseguir una expresión que nos relacione las aceleraciones dependientes con las independientes. Sabiendo que:

$$D\ddot{p} + \dot{D}\dot{p} = 0 \quad (3-28)$$

Entonces,

$$D_{din}\ddot{p}_{din} + D_{kin}\ddot{p}_{kin} + D_{dep}\ddot{p}_{dep} + \dot{D}_{din}\dot{p}_{din} + \dot{D}_{kin}\dot{p}_{kin} + \dot{D}_{dep}\dot{p}_{dep} = 0$$

donde

$$\dot{D}_i = \frac{dD_i \dot{p}_i}{dp_i} \quad (3-29)$$

Ahora operando como hicimos antes nos queda,

$$D_{kindep} \ddot{p}_{kindep} = -(D_{dep} \ddot{p}_{dep} + \dot{D}_{din} \dot{p}_{din} + \dot{D}_{kin} \dot{p}_{kin} + \dot{D}_{dep} \dot{p}_{dep} + D_{kin} \ddot{p}_{kin}) \quad (3-30)$$

Finalmente nos queda una ecuación que nos da relación de las aceleraciones dependientes en función de las independientes. Tan solo nos queda plasmar esa relación en la ecuación de movimiento de tal manera que esta quede en función de las aceleraciones independientes.

El vector de coordenadas generalizadas se puede expresar de la siguiente manera:

$$p = \begin{bmatrix} p_{din} \\ p_{kindep} \end{bmatrix} \quad (3-31)$$

Derivando,

$$\dot{p} = \begin{bmatrix} \dot{p}_{din} \\ \dot{p}_{kindep} \end{bmatrix} \quad (3-32)$$

Sustituimos la ecuación (3-33) en la ecuación (3-35)

$$\ddot{p} = \begin{bmatrix} \ddot{p}_{din} \\ \ddot{p}_{kindep} \end{bmatrix} = \begin{bmatrix} I_{3 \times 3} \\ -D_{kindep}^{-1} D_{kin} \end{bmatrix} \ddot{p}_{din} + \begin{bmatrix} 0_{3 \times 1} \\ -D_{kindep}^{-1} (\dot{D}_{din} \dot{p}_{din} + \dot{D}_{kin} \dot{p}_{kin} + \dot{D}_{dep} \dot{p}_{dep}) \end{bmatrix} \quad (3-33)$$

Donde $I_{3 \times 3}$ es una matriz identidad de 3 filas por 3 columnas y $0_{3 \times 1}$ es un vector columna nulo de 3 filas. Para mayor simplicidad vamos a representar el vector de coordenadas de la siguiente manera.

$$\ddot{p} = E \ddot{p}_{din} + \dot{E} \quad (3-34)$$

Entonces ya solo nos queda sustituir la ecuación (3-25) en la ecuación de movimiento del sistema y así expresamos la ecuación de movimiento en función de las aceleraciones independientes.

$$E^T M E \ddot{p}_{din} + E^T M \dot{E} + E^T D^T \lambda = E^T Q_{grav} + E^T Q_{aero} + E^T Q_{accion} \quad (3-35)$$

Donde se comprueba que (shabana computational dynamics):

$$E^T D^T = 0 \quad (3-36)$$

Finalmente, despejando las aceleraciones independientes en (3-38), obtenemos la ecuación de movimiento del sistema en función de las aceleraciones independientes y sin que aparezcan las fuerzas generalizadas de reacción.

$$\ddot{p}_{din} = (E^T M E)^{-1} \{ E^T (Q_{grav} + Q_{aero} + Q_{accion} - M \dot{E}) \} \quad (3-37)$$

Ya tenemos nuestras ecuaciones de movimiento de la bicicleta, ahora vamos a probar nuestro simulador. Vamos a ver a la bicicleta en varias situaciones, aplicaremos los momentos y cambiaremos las condiciones iniciales para comprobar que la bicicleta se mueve acorde a nuestras demandas.

3.1.4 Estabilización de Baumgarte

En el análisis dinámico directo, debido a que debemos integrar dos veces las ecuaciones de movimiento, pueden aparecer valores distintos de cero cuando evaluamos las ecuaciones de restricción del sistema. Para solventar este problema vamos a utilizar el método de estabilización de *Baumgarte*. Este método se explica en el documento [1]. Consiste en incluir en la ecuación de las restricciones los siguientes parámetros, de manera

que la ecuación final queda de la forma:

$$D\ddot{p} + \dot{D}\dot{p} + 2\alpha(\dot{D}\dot{p}) + \beta^2 C = 0 \quad (3-38)$$

Redefiniendo la matriz \dot{E} , conseguimos que la respuesta se estabilice y se aproxime más a la solución real. El valor de los parámetros se determina a través de la experiencia. En este trabajo se han tomado $\alpha = \beta = 10$.

4 IMPLEMENTACIÓN EN MATLAB

Todo el desarrollo teórico que hemos planteado en este trabajo ha sido con el objetivo de implementarlo en un programa de cálculo numérico. Vamos a describir el procedimiento que se ha seguido para plantear todas las ecuaciones y los métodos para resolverlas. El programa que se ha usado es el Matlab. Matlab es un programa de cálculo numérico que ofrece la posibilidad de trabajar con variables simbólicas.

El programa de la resolución de la dinámica directa de la bicicleta se resume en tres grandes bloques. El bloque simbólico, donde se han declarado las variables y matrices de manera simbólica. El contenido de este bloque está en el fichero *Bicicleta_simb*. El bloque de simulación, donde incluimos los parámetros de la bicicleta e introducimos los valores de entrada de las ecuaciones de movimiento. El contenido de este bloque está en el fichero *simula_Bicicleta*. Y, por último, el bloque de la dinámica directa, donde se han resuelto las ecuaciones diferenciales de segundo orden obteniendo como resultado los valores de coordenadas independientes de el sistema multicuerpo. Este último bloque está contenido en el fichero *dinBicicleta*.

4.1 Bloque simbólico

El cálculo simbólico nos permite manipular las variables de manera simbólica, permitiéndonos realizar operaciones con ellas que no serían posibles si trabajáramos de manera numérica.

El fichero *Bicicleta_simb* sigue prácticamente, el mismo orden en el que se ha desarrollado la teoría del modelo.

Primero se han declarado de forma simbólica todas las variables y parámetros que intervienen en la definición de las ecuaciones del problema. Para declarar en Matlab una variable en forma simbólica usamos la función *sym* y a continuación escribimos todas las variables que queremos que sean simbólicas.

A continuación se han definido los vectores de las coordenadas generalizadas tanto en posición, como en velocidad y aceleración. Estos vectores son p , dp y ddp , respectivamente.

Después hemos definido las matrices de giro de cada uno de los ángulos de giro de los distintos sólidos del problema. La estructura que tienen es la misma que se ha desarrollado en las ecuaciones (2-8) a (2-14). Haciendo uso de estas matrices se han definido las matrices de giro de los sólidos respecto del sistema global de referencia, (2-2) a (2-7).

Para definir las velocidades angulares de los sólidos hemos empleado la función *diff* a cada una de las matrices de giro de los sólidos. Esta función nos devuelve la derivada de la matriz respecto de la coordenada que le indiquemos. Así, como se ve en la ecuación (2-19), definimos la derivada temporal de cada una de las matrices de giro de los sólidos. Como describimos en el apartado de velocidades angulares, la matriz antisimétrica de la velocidad angular de un sólido se puede calcular según la ecuación $\dot{A}^i A^{iT} = \tilde{\omega}$. Reordenando las componentes, se obtienen los vectores velocidad angular de los sólidos. Para ponerlos en coordenadas locales premultiplicamos por la matriz de giro traspuesta, (2-19).

Con los vectores velocidad angular en coordenadas locales, pasamos a definir las matrices de transformación de velocidades angulares en coordenadas locales como se explica en la ecuación (2-23). En Matlab, hacemos uso de la función *jacobian*. Esta función realiza el jacobiano de la variable que nos interese respecto del vector de coordenadas que le indiquemos.

De la misma manera, calculamos la matriz g , que es la matriz jacobiana de la matriz de transformación de velocidades angulares.

Pasamos a definir los vectores posición y velocidad de los centros de gravedad de cada uno de los sólidos del sistema. Como vimos el apartado de posición de los sólidos, la posición de cada sólido se define según las ecuaciones (2-15) a (2-18). Una vez calculadas las posiciones, vamos a calcular las matrices de transformación de velocidades y las velocidades de los centros de gravedad de los sólidos, (2-26) y (2-27).

De la misma manera, calculamos la matriz h , que es la matriz jacobiana de la matriz de transformación de velocidades.

Ya disponemos de todas las matrices necesarias para terminar de definir las restricciones y las matrices que aparecen en la ecuación de movimiento de nuestro sistema multicuerpo.

Para definir las restricciones del sistema, primero debemos calcular la posición del punto de contacto de la rueda delantera con el suelo, el vector tangente en ese punto, su velocidad y la velocidad del punto de contacto de la rueda trasera con el suelo. La posición del punto de contacto de la rueda delantera y el vector tangente a ese punto, vienen definidos en las ecuaciones (2-33) y (2-34). Las velocidades del punto C y del punto D se calculan como definimos en las ecuaciones (2-41) y (2-45). Ahora ya podemos definir el vector de reacciones del sistema, al que hemos llamado C , (2-51).

Las fuerzas generalizadas de gravedad, de resistencia aerodinámica y de la acción del ciclista se declaran de la misma manera que fueron definidas en las ecuaciones (3-11), (3-15) y (3-17). La matriz de masas y de fuerzas centrífugas se calculan como definimos en las ecuaciones (3-6).

Finalmente, lo único que nos queda por definir serían las matrices E y \dot{E} . Debido al que la manipulación de las variables simbólicas supone un coste computacional muy alto, en vez de definir las dos matrices, vamos a definir las matrices de las que están compuestas.

Primero definimos la matriz D , (3-23). Con esta matriz calculamos D_{kindep} y D_{din} , (3-26). Después calculamos las derivadas temporales de cada una de las dos matrices como se define en la ecuación (3-29).

La última parte de este bloque consiste en transformar todas las matrices que vamos a necesitar para la resolución del problema dinámico directo, en ficheros de Matlab. La función *matlabFunction* crea un fichero de Matlab de la matriz de variables simbólicas. Esto nos permitirá evaluar numéricamente las matrices.

4.2 Bloque de la dinámica directa

En este bloque se van a resolver la dinámica directa de la bicicleta.

El análisis dinámico de un sistema multicuerpo consta de tres partes: análisis de posición, análisis cinemático y el análisis dinámico.

4.2.1 Análisis de posición

El análisis de posición consiste en calcular el valor de las coordenadas generalizadas a nivel de posición. El método que se ha empleado en este trabajo es el algoritmo de Newton-Raphson. Este método iterativo consiste en plantear el problema de posición a partir del vector de restricciones holónomas. Primero debemos dar un estimado inicial. Este estimado inicial debe de estar lo más cerca posible de la solución para que el método converga. Si q_i es el estimado inicial, la solución exacta es $q_i + \Delta q$. Aplicando un desarrollo de Taylor y asumiendo que Δq es muy pequeño, llegamos a la siguiente expresión:

$$C_{q_i} \Delta q_i = -C(q_i, t) \quad (4-1)$$

Donde C_q es el jacobiano de las restricciones respecto del vector q_i .

Una vez hemos calculado Δq , actualizamos el estimado inicial y volvemos a realizar el mismo procedimiento. Si el algoritmo converge a la solución exacta, Δq será cada vez más pequeño. El método finaliza cuando Δq es más pequeño que la tolerancia que nosotros le imponemos.

En nuestro caso, el problema de posición a resolver consiste en realizar el algoritmo de Newton-Raphson con las coordenadas p_{dep} . Se ha definido una variable global, llamada *estimadoinicial*, que desde el bloque de *simula_Bici* da un estimado inicial para resolver el análisis de posición. La variable *estadoinicial* permite que el valor de q_i se vaya actualizando al mismo tiempo que el programa resuelve el problema dinámico directo. Se decidió realizar una sola iteración a la hora de aplicar el Newton-Raphson.

Una vez calculemos el vector p_{dep} para el instante t , el análisis de posición estará completado.

4.2.2 Análisis cinemático

Para el análisis cinemático hay que resolver la ecuación (3-27), que nos relaciona el vector de velocidades de las coordenadas independientes con el vector de velocidades de las coordenadas dependientes.

4.2.3 Análisis dinámico

Para terminar con el análisis dinámico directo unicamente nos falta resolver la ecuación de movimiento que define el movimiento de la bicicleta.

La ecuación de movimiento viene dada por:

$$\ddot{p}_{din} = (E^T M E)^{-1} \{E^T (Q_{grav} + Q_{aero} + Q_{accion} - M \dot{E})\} \quad (4-2)$$

Dentro de este bloque, los valores de los momentos que aplica el ciclista a la bicicleta los imponemos nosotros.

4.2.4 Controlador de la bicicleta

En las simulaciones de la bicicleta, en el momento en el que el ciclista comienza a pedalear, la bicicleta se vuelve inestable. Para solventar esta inestabilidad, en el bloque de *dinBicicleta* se ha incluido un controlador que hace que durante el arranque de la bicicleta, esta se mantenga recta. Este controlador actúa de la misma manera en la que el ciclista trata de estabilizar la bicicleta en los primeros instantes.

El controlador que se ha implementado actúa cuando las aceleraciones en el manillar y en el cuadro de la bicicleta superan cierto valor. Cuando esto ocurre, hacemos nulas estas aceleraciones, impidiendo que la bicicleta se incline y termine cayendo.

Una vez calculemos las aceleraciones independientes, solo nos queda integrarlas.

4.3 Bloque de simulación

En este bloque vamos a simular nuestro sistema. Para realizar la simulación de la bicicleta, debemos de resolver la dinámica directa del sistema. En el bloque *dinBicicleta* se calculan las aceleraciones independientes del sistema, únicamente hay que integrarlas.

Un problema dinámico directo, consiste en la resolución de una ecuación diferencial de segundo orden. Matlab no es capaz de resolver este tipo de ecuaciones. El planteamiento que se ha utilizado para resolver este problema, consiste en convertir el problema de segundo orden en un problema de primer orden.

Nuestro sistema de ecuaciones diferenciales de segundo orden tiene la siguiente estructura:

$$\ddot{q}_{din} = M^{-1} Q \quad (4-3)$$

Para transformar el sistema de ecuaciones diferenciales de segundo orden en uno de primer orden, definimos el vector de estado y :

$$y = [\dot{p}_{din} \quad p_{din} \quad p_{kin}]^T \quad (4-4)$$

Derivamos respecto al tiempo.

$$\dot{y} = [\ddot{p}_{din} \quad \dot{p}_{din} \quad \dot{p}_{kin}]^T \quad (4-5)$$

Sustituyendo las ecuaciones nos queda:

$$\dot{y} = \begin{bmatrix} \ddot{p}_{din} \\ \dot{p}_{din} \\ \dot{p}_{kin} \end{bmatrix} = \begin{bmatrix} M^{-1} Q \\ p_{din} \\ p_{kin} \end{bmatrix} \quad (4-6)$$

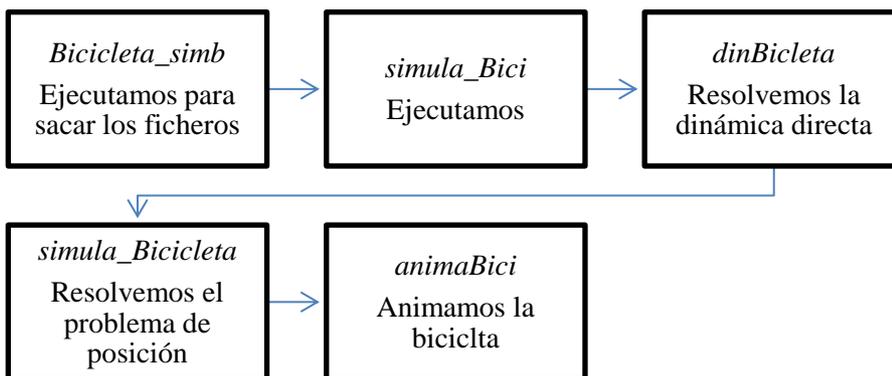
Donde vemos que el problema de segundo orden a pasado a ser un problema de primer orden.

Para la resolución de ecuaciones diferenciales, Matlab tiene varias funciones que son capaces de resolverlo. En este trabajo se ha usado la función *ode45*. Los parámetros de entrada son: el tiempo que se quiere iterar y las condiciones iniciales de las incógnitas. La función devuelve un vector de tiempo y el vector solución para cada instante de tiempo.

Una vez que se ha resuelto la ecuación diferencial ya tenemos el problema dinámico directo completamente resuelto. Tan solo nos queda ver como se mueve la bicicleta. El bloque de simulación incluye un modulo de animación, *animaBici*, y otro modulo que resuelve de nuevo el problema de posición con las soluciones que se han hallado después del *ode45*. Esto es necesario porque el vector solución no nos da información sobre las coordenadas p_{dep} . Para poder animar la bicicleta, temenemos que volver a resolver el problema de posición.

En el siguiente capítulo veremos varias simulaciones de la bicicleta en distintos casos de interés.

4.4 Diagrama de flujo



5 SIMULACIONES

5.1 Caso 1: Trayectoria recta

En este caso se simula la bicicleta partiendo del reposo y recorriendo una trayectoria recta. Las condiciones iniciales son nulas y los momentos aplicados en la bicicleta son: un momento constante en la rueda trasera y el resto de momentos nulos. El valor del momento de pedaleo es de $10Nm$. Los vectores de condiciones iniciales y de momentos serían los siguientes:

$$y_0 = [\dot{p}_{dino} \quad p_{dino} \quad p_{kino}]^T \quad (5-1)$$

$$y_0 = [0 \quad 0 \quad 0]^T \quad (5-2)$$

$$T = \begin{bmatrix} M_{vue} \\ M_{ped} \\ M_{dir} \end{bmatrix} = \begin{bmatrix} 0 \\ 10 \\ 0 \end{bmatrix} \quad (5-3)$$

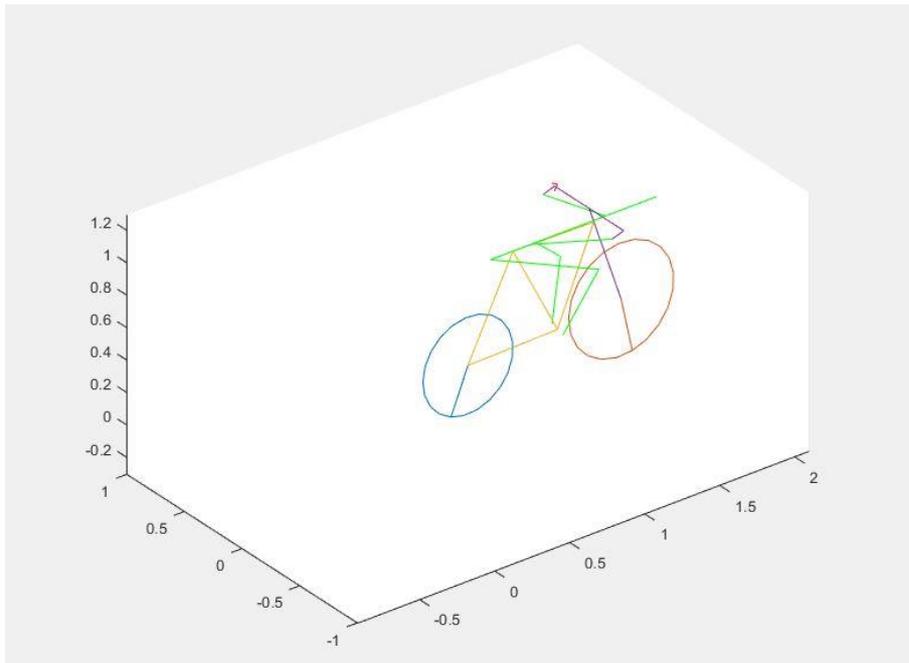


Figura 5-1. Bicicleta circulando en línea recta.

La Figura 5-1 representa a la bicicleta durante la simulación.

La Figura 5-2 refleja la variación de las ecuaciones de restricción a lo largo de toda la simulación, se puede ver que los valores son cercanos a cero. Esto quiere decir que el sistema está cumpliendo las restricciones y que la solución que hemos hallado es una buena aproximación al comportamiento real de la bicicleta.

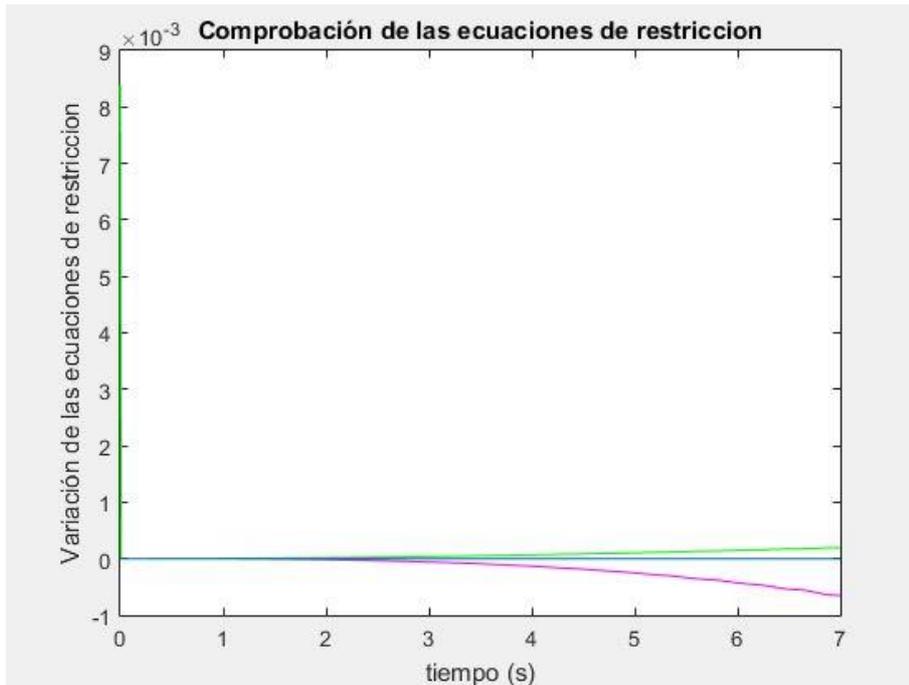


Figura 5-2. Comprobación de las ecuaciones de restricción.

5.1.1 Caso 1.2: Trayectoria recta con velocidad inicial

Vamos a estudiar el caso en el que la bicicleta parte con velocidad inicial y el vector de momentos es nulo. El objetivo es ver si verdaderamente la bicicleta se va parando conforme pasa el tiempo.

La velocidad inicial va a ser de 5 rad/s en la coordenada angular de la rueda trasera.

$$y_0 = [0 \ 5 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (5-4)$$

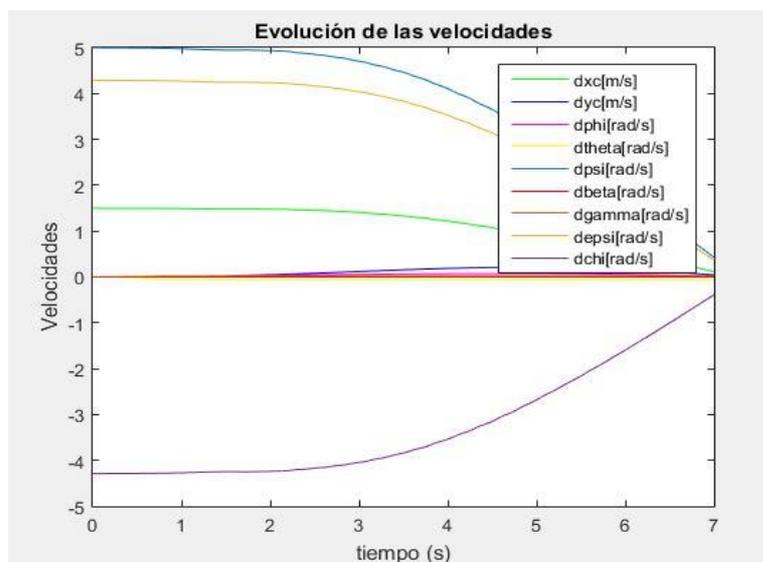


Figura 5-3. Evolución de las velocidades.

Se observa en la Figura 5-3 que la velocidad se reduce debido a que no se está aplicando ningún momento en el pedal.

5.2 Caso 2: Rampa

Ahora queremos ver como se comporta la bicicleta cuando se encuentra en un plano inclinado. Para este caso vamos a comprobar si la bicicleta se mueve cuando la dejamos parada y como evolucionan las velocidades en comparación con el caso del movimiento recto.

5.2.1 Caso 2.1: Bicicleta parada en una rampa

La pendiente de la rampa se ha establecido en un 5%.

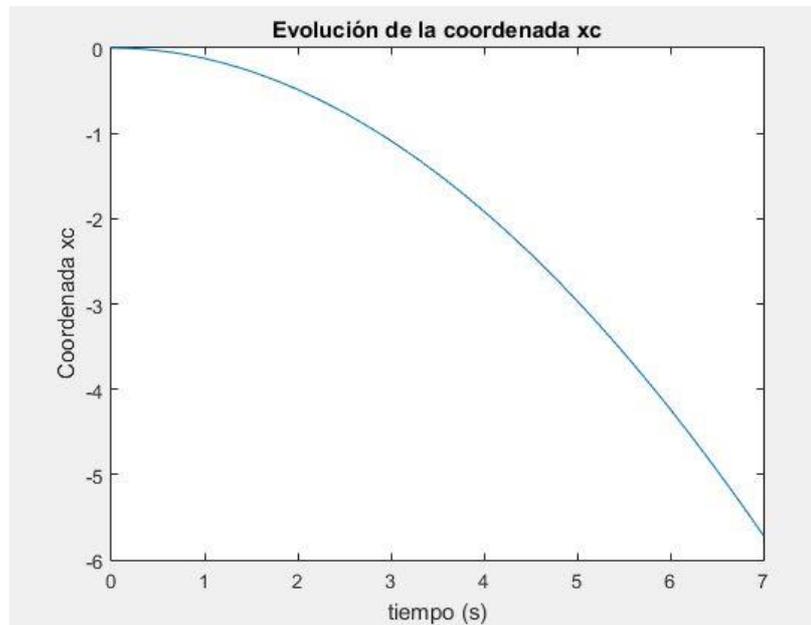


Figura 5-4. Evolución de la coordenada x_c .

Se comprueba que efectivamente la bicicleta está cayendo hacia atrás por el plano inclinado.

5.2.2 Caso 2.1: Bicicleta circulando en una rampa

En este caso vamos a imponer un mismo momento de valor 20Nm en el caso de la bicicleta circulando en llano y subiendo la rampa. Es interesante comparar las velocidades de ambos casos.

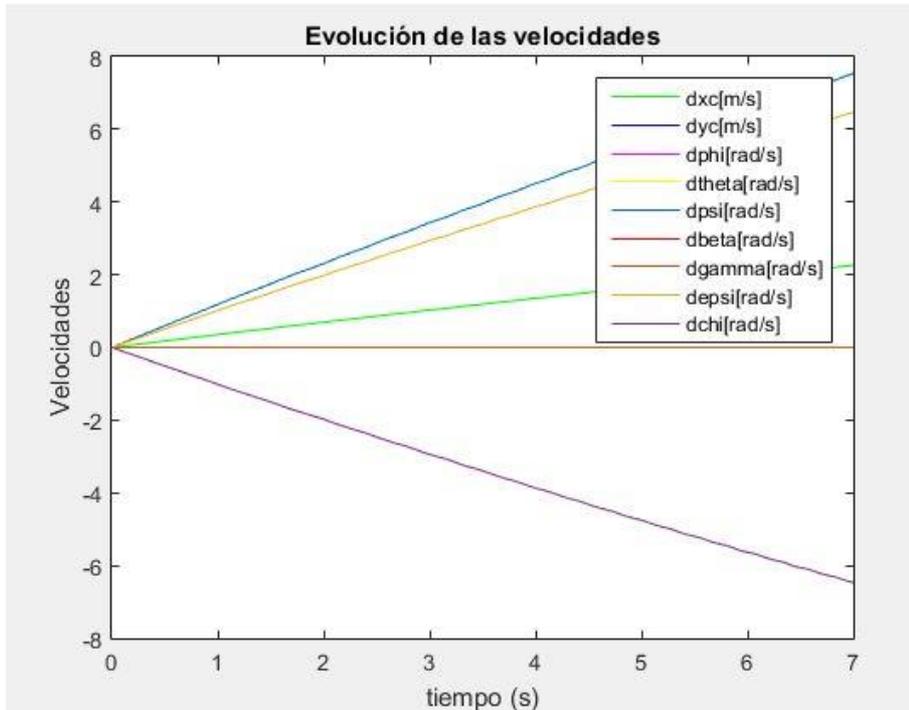


Figura 5-5. Esto es el pie de la figura en llano.

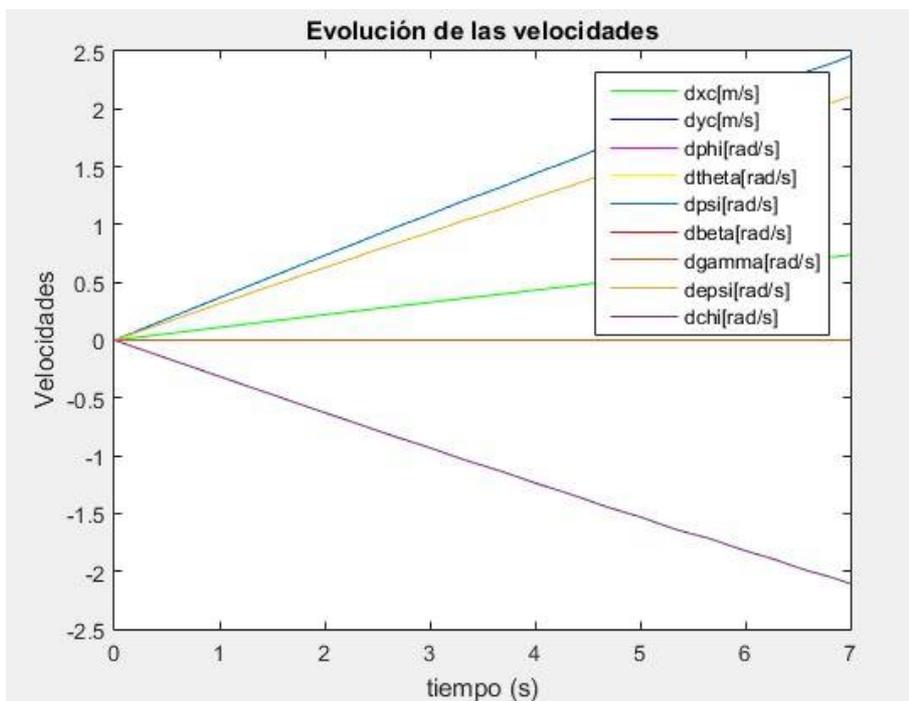


Figura 5-6. Evolución de las velocidades en rampa.

Es evidente la diferencia de magnitudes a nivel de velocidades. Siendo el momento de pedaleo 2 veces mayor las velocidades son significativamente más pequeñas.

5.3 Caso 3: Trayectoria curva

En este caso vamos a tratar de conseguir que la bicicleta describa una trayectoria curva.

Las condiciones que tenemos que definir para que esto ocurra son las siguientes: debemos partir con velocidad inicial no nula y hallar la combinación de momentos ideal para que la bicicleta gire.

Cuando montamos en bicicleta es evidente que no ejercemos la misma fuerza en el pedal que en el manillar. También es cierto que no podemos girar de manera brusca cuando vamos a velocidades altas. Por todo esto es necesario buscar la combinación de valores que permitan a la bicicleta describir una trayectoria curva.

Encontrar la combinación de valores es una cuestión experimental.

Tras varios intentos, en las condiciones en la que la bicicleta describe una trayectoria curva son las siguientes:

$$y_0 = [0 \ 10 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (5-5)$$

$$T = \begin{bmatrix} 0 \\ 10 \\ 5 \end{bmatrix} \quad (5-6)$$

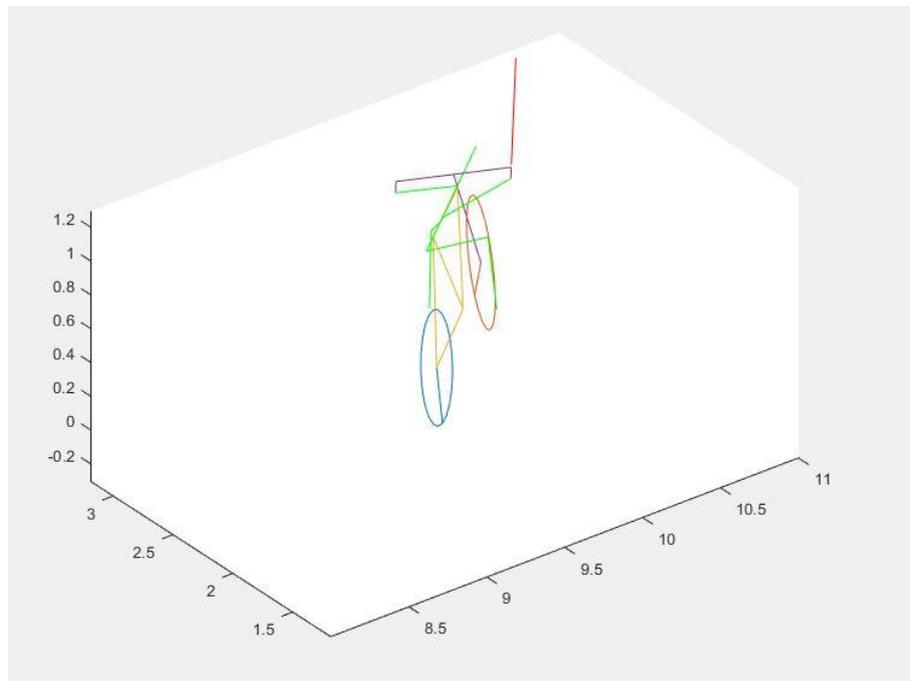


Figura 5-7. Bicicleta describiendo una trayectoria curva.

Como se ve en la Figura 5-6 la bicicleta está describiendo una trayectoria curva. Se puede ver como el cuadro y el ciclista se inclinan hacia el centro de la trayectoria.

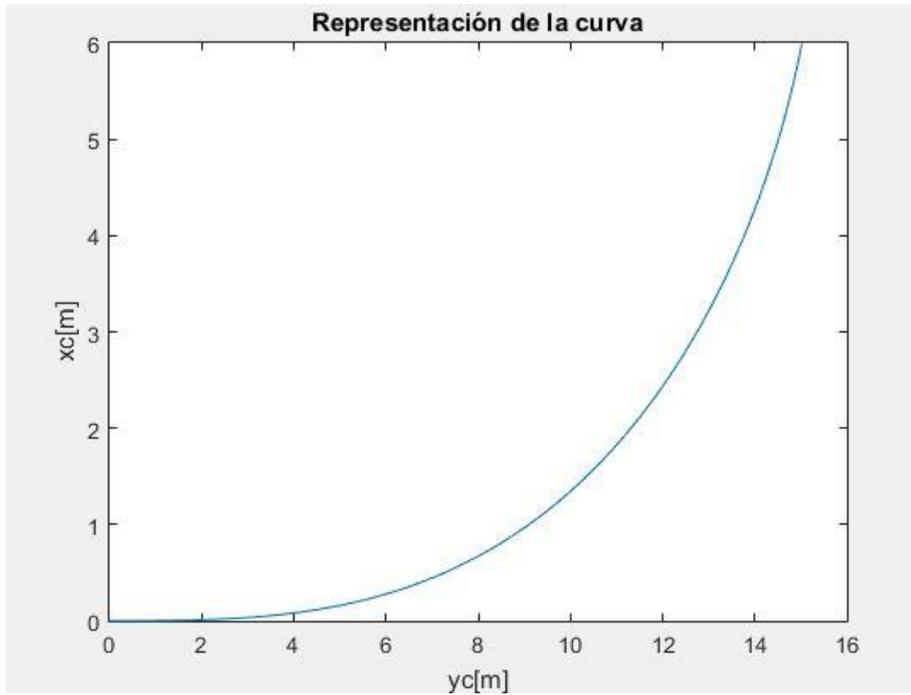


Figura 5-8. Trayectoria curva que describe la bicicleta.

En la Figura 5-8 observamos la trayectoria curva que describe la bicicleta.

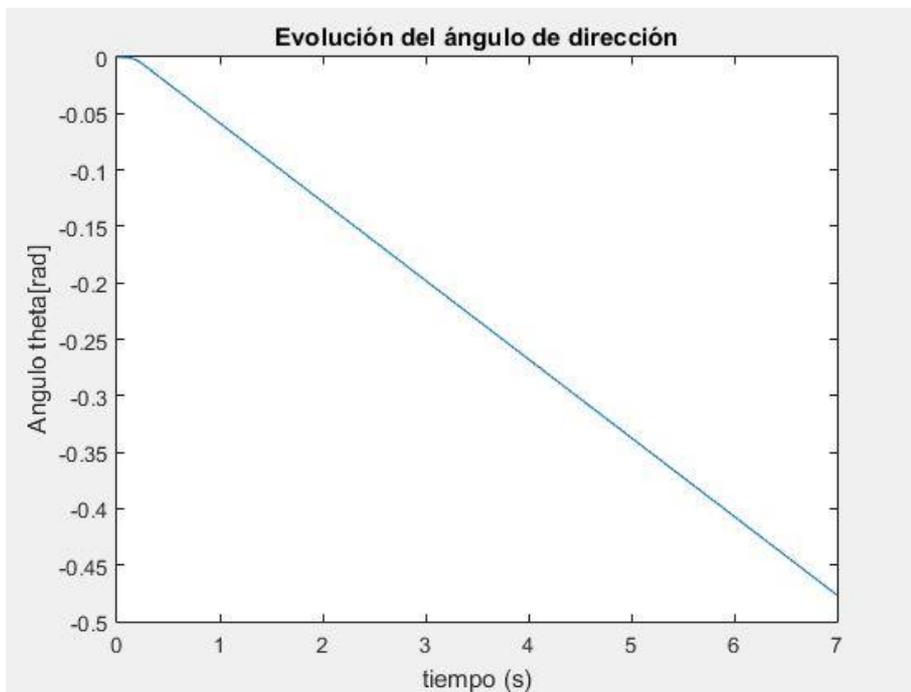


Figura 5-9. Evolución del ángulo de vuelco en una curva.

Se ve en la Figura 5-9 como el ángulo de vuelco va aumentando para realizar la trayectoria curva. Esto siempre pasa cuando queremos girar la bicicleta cuando estamos en movimiento, el cuadro y nuestro cuerpo giran alrededor del centro de curvatura de la trayectoria.

6 CONCLUSIÓN

Se ha realizado el análisis dinámico directo de una bicicleta. Se han implantado las ecuaciones en Matlab y se han realizado una serie de simulaciones de ciertos casos de interés.

En este trabajo se ha expuesto el desarrollo detallado para plantear las ecuaciones en función de las aceleraciones independientes y se han eliminado las fuerzas de restricción de la ecuación de movimiento. En las simulaciones se ha podido apreciar como la bicicleta se comporta de manera parecida a la real.

Un punto de mejora al planteamiento presentado en Matlab, sería la implementación de un controlador que aplicara un vector de momentos que estabilizara la bicicleta mientras está circulando.

7 MATLAB

7.1 Bicicleta_simb

```

% ANÁLISIS DINÁMICO DIRECTO DE UNA BICICLETA %

%% DECLARACIÓN DE VARIABLES %%
% Coordinadas generalizadas %
syms xc yc phi theta psi beta gamma epsi chi real
syms dxc dyc dphi dtheta dpsi dbeta dgamma depsi dchi real
syms ddx ddy ddp ddt dpsi ddbeta ddgamma ddepsi ddchi real

p = [xc yc phi theta psi beta gamma epsi chi]';
dp = [dxc dyc dphi dtheta dpsi dbeta dgamma depsi dchi]';
ddp = [ddxc ddy ddp ddt dpsi ddbeta ddgamma ddepsi ddchi]';

% División de coordenadas %

pdin=[theta psi gamma]';
pkin=[xc yc phi epsi]';
pdep=[beta chi]';

dpdin=[dtheta dpsi dgamma]';
dpkin=[dxc dyc dphi depsi]';
dpdep=[dbeta dchi]';

% Parámetros geométricos %
syms Rt Rd real
syms x3 z3 real
syms x4 z4 real
syms x5 z5 real

%% CINEMÁTICA %%

% MATRICES DE GIRO %

A_phi(1,:) = [cos(phi) -sin(phi) 0];
A_phi(2,:) = [sin(phi) cos(phi) 0];
A_phi(3,:) = [0 0 1];

A_theta(2,:) = [0 cos(theta) -sin(theta)];
A_theta(3,:) = [0 sin(theta) cos(theta)];
A_theta(1,:) = [1 0 0];

A_psi(1,:) = [cos(psi) 0 sin(psi)];
A_psi(2,:) = [0 1 0];
A_psi(3,:) = [-sin(psi) 0 cos(psi)];

A_beta(1,:) = [cos(beta) 0 sin(beta)];
A_beta(2,:) = [0 1 0];
A_beta(3,:) = [-sin(beta) 0 cos(beta)];

A_gamma(1,:) = [cos(gamma) -sin(gamma) 0];
A_gamma(2,:) = [sin(gamma) cos(gamma) 0];
A_gamma(3,:) = [0 0 1];

```

```

A_epsi(1,:) = [ cos(epsi) 0 sin(epsi)];
A_epsi(2,:) = [0 1 0];
A_epsi(3,:) = [-sin(epsi) 0 cos(epsi)];

Ai1 = A_phi;
Ai2 = A_phi*A_theta;
A2 = Ai2*A_psi;
A3 = Ai2*A_beta;
A4 = A3*A_gamma;
A5 = A4*A_epsi;

%Derivadas matrices de giro%
dAi2=diff(Ai2,phi)*dphi+diff(Ai2,theta)*dtheta;
dA2=diff(A2,phi)*dphi+diff(A2,theta)*dtheta+diff(A2,psi)*dpsi;
dA3=diff(A3,phi)*dphi+diff(A3,theta)*dtheta+diff(A3,beta)*dbeta;
dA4=diff(A4,phi)*dphi+diff(A4,theta)*dtheta+diff(A4,beta)*dbeta...
+diff(A4,gamma)*dgamma;
dA5=diff(A5,phi)*dphi+diff(A5,theta)*dtheta+diff(A5,beta)*dbeta...
+diff(A5,gamma)*dgamma+diff(A5,epsi)*depsi;

dAi2=simplify(dAi2);
dA2=simplify(dA2);
dA3=simplify(dA3);
dA4=simplify(dA4);
dA5=simplify(dA5);

%Velocidades angulares%

w2_sk=simplify(dA2*A2');
w3_sk=simplify(dA3*A3');
w4_sk=simplify(dA4*A4');
w5_sk=simplify(dA5*A5');

w2=[w2_sk(3,2) w2_sk(1,3) w2_sk(2,1)]';
w3=[w3_sk(3,2) w3_sk(1,3) w3_sk(2,1)]';
w4=[w4_sk(3,2) w4_sk(1,3) w4_sk(2,1)]';
w5=[w5_sk(3,2) w5_sk(1,3) w5_sk(2,1)]';

w2_lc=simplify(A2'*w2);
w3_lc=simplify(A3'*w3);
w4_lc=simplify(A4'*w4);
w5_lc=simplify(A5'*w5);

G2_lc=jacobian(w2_lc,dp);
G3_lc=jacobian(w3_lc,dp);
G4_lc=jacobian(w4_lc,dp);
G5_lc=jacobian(w5_lc,dp);

g2_lc=jacobian(w2_lc,p);
g3_lc=jacobian(w3_lc,p);
g4_lc=jacobian(w4_lc,p);
g5_lc=jacobian(w5_lc,p);

% VELOCIDADES CDG %

rG2=[xc yc 0]'+Ai2*[0 0 Rt]';
H2=jacobian(rG2,p);
vG2=H2*dp;

```

```

rG3=rG2+A3*[x3 0 z3]';
H3=jacobian(rG3,p);
vG3=H3*dp;

rG4=rG2+A3*[x4 0 z4]';
H4=jacobian(rG4,p);
vG4=H4*dp;

rG5=rG4+A4*[x5 0 z5]';
H5=jacobian(rG5,p);
vG5=H5*dp;

% MATRICES h Y g

h2=jacobian(H2*dp,p);
h3=jacobian(H3*dp,p);
h4=jacobian(H4*dp,p);
h5=jacobian(H5*dp,p);

% POSICIÓN DEL PUNTO D Y TANGENTE AL PLANO XY EN EL PUNTO D %

rD_lc=[Rd*cos(2*pi-chi) 0 Rd*sin(2*pi-chi)]';
tD_lc=diff(rD_lc,chi);

rD=rG5+A5*rD_lc;
tD=A5*tD_lc;

% VELOCIDAD DE LOS PUNTOS C Y D %

rG2_C=[xc yc 0]'-rG2;
vC=vG2+w2_sk*rG2_C;
vC=simplify(vC);

vD=vG5+cross(w5,A5*rD_lc);
vD=simplify(vD);

%% RESTRICCIONES %%

% HOLÓNOMAS %

C_h=[rD(3) tD(3)]';
C_ph=jacobian(C_h,p);

% NO HOLÓNOMAS %

C_nh=[vC(1) vC(2) vD(1) vD(2)]';
B=jacobian(C_nh,p);
C_dpnh=jacobian(C_nh,dp);

% MATRIZ DE RESTRICCIONES %

C=[C_h' C_nh]';
C=simplify(C);
matlabFunction(C,'file','C_Bicicleta');

%% DINÁMICA %%

```

```

syms m2 m3 m4 m5 g cv real
syms I2xx I2yy I2zz real
syms I3xx I3yy I3zz I3zx real
syms I4xx I4yy I4zz I4xy I4xz I4yz real
syms I5xx I5yy I5zz real

% MATRICES DE INERCIA

I2(1,:) = [I2xx 0 0];
I2(2,:) = [0 I2yy 0];
I2(3,:) = [0 0 0];

I3(1,:) = [I3xx 0 I3zx];
I3(2,:) = [0 I3yy 0];
I3(3,:) = [I3zx 0 I3zz];

I4(1,:) = [I4xx I4xy I4xz];
I4(2,:) = [I4xy I4yy I4yz];
I4(3,:) = [I4xz I4yz I4zz];

I5(1,:) = [I5xx 0 0];
I5(2,:) = [0 I5yy 0];
I5(3,:) = [0 0 0];

% MATRIZ DE MASA %

M2=m2*eye(3,3);
M3=m3*eye(3,3);
M4=m4*eye(3,3);
M5=m5*eye(3,3);

MM= [M2 zeros(1,9);zeros(1,3) M3 zeros(1,6);zeros(1,6) M4
zeros(1,3);zeros(1,9) M5];

H= [H2 ;H3 ;H4 ;H5];
Ht=[H2' H3' H4' H5'];
simplify(Ht);
simplify(H);
G= [G2_lc ; G3_lc ; G4_lc ; G5_lc];
Gt=[G2_lc' G3_lc' G4_lc' G5_lc'];
simplify(Gt);
simplify(G);
I= [I2 zeros(1,9);zeros(1,3) I3 zeros(1,6);zeros(1,6) I4
zeros(1,3);zeros(1,9) I5];

M=Ht*MM*H+Gt*I*G;

% FUERZA GRAVITATORIA %

F2=[0 0 -m2*g]';
F3=[0 0 -m3*g]';
F4=[0 0 -m4*g]';
F5=[0 0 -m5*g]';
Fg=[F2;F3;F4;F5];

Qg=Ht*Fg;
matlabFunction(Qg,'file','gravityForceBici');

```

```

% FUERZA AERODINÁMICA %

Qareo=-cv*norm(vG3)*H3'*vG3;

% CENTRIFUGAL FORCES %

g= [g2_lc ; g3_lc ; g4_lc ; g5_lc];
h= [h2 ; h3 ; h4 ; h5];
simplify(g);
simplify(h);

QQv2=-cross(w2_lc,I2*w2_lc);
QQv3=-cross(w3_lc,I3*w3_lc);
QQv4=-cross(w4_lc,I4*w4_lc);
QQv5=-cross(w5_lc,I5*w5_lc);

QQv=[zeros(3*4,1); QQv2 ; QQv3 ; QQv4 ; QQv5];
simplify(QQv);

Lt=[H2' H3' H4' H5' G2_lc' G3_lc' G4_lc' G5_lc'];
simplify(Lt);

Qv=-(Ht*MM*h+Gt*I*g)*dp+Lt*QQv;

% FUERZAS DE ACCION %

syms Mdir Mped Mvue real

T=[Mdir Mped Mvue]';

F=[G4_lc(3,:) ' G2_lc(2,:) ' G3_lc(1,:)'];

%% separacion de variables para sistema no holonomo %%

Cdin=jacobian(C_h,pdin);
Ckin=jacobian(C_h,pkin);
Cdep=jacobian(C_h,pdep);

Bdin=jacobian(C_nh,dpdin);
Bkin=jacobian(C_nh,dpkin);
Bdep=jacobian(C_nh,dpdep);

Dkindep=[Ckin Cdep; Bkin Bdep];
Dkindep=simplify(Dkindep);

Ddin=[Cdin; Bdin];
Ddin=simplify(Ddin);

dDkindep=jacobian(Dkindep*[dpkin' dpdep']',[pkin' pdep]');
dDdin=jacobian(Ddin*dpdin,pdin);

D=[C_ph ; C_dph ];
dD=jacobian(D*dp,p);

matlabFunction(T,'file','Momentos');
matlabFunction(F,'file','Giros');
matlabFunction(dD,'file','Dpunto');

```

```

matlabFunction(C_h, 'file', 'Rest_holonoma');
matlabFunction(M, 'file', 'massMatrixBici');
matlabFunction(Qareo, 'file', 'areoForceBici');
matlabFunction(Qg, 'file', 'gravityForceBici');
matlabFunction(Qv, 'file', 'centrifugalForceBici');
matlabFunction(Cdep, 'file', 'Jacobian_rest_holonoma');
matlabFunction(Dkindep, 'file', 'Dkindep_matrix');
matlabFunction(Ddin, 'file', 'Ddin_matrix');
matlabFunction(dDkindep, 'file', 'dDkindep_matrix');
matlabFunction(dDdin, 'file', 'dDdin_matrix');

```

7.2 dinBicicleta

```
function dy=dinBicicleta(t,y)
```

```

global Param
global estimadoinicial
global M_dir

```

```

Rt=Param.Rt;
Rd=Param.Rd;
x3=Param.x3;
z3=Param.z3;
x4=Param.x4;
z4=Param.z4;
x5=Param.x5;
z5=Param.z5;

```

```

m2=Param.m2;
m3=Param.m3;
m4=Param.m4;
m5=Param.m5;

```

```

g=Param.g;
cv=Param.cv;

```

```

I2xx=Param.I2xx;
I2yy=Param.I2yy;
I3xx=Param.I3xx;
I3yy=Param.I3yy;
I3zz=Param.I3zz;
I3zx=Param.I3zx;
I4xx=Param.I4xx;
I4yy=Param.I4yy;
I4zz=Param.I4zz;
I4xy=Param.I4xy;
I4xz=Param.I4xz;
I4yz=Param.I4yz;
I5xx=Param.I5xx;
I5yy=Param.I5yy;

```

```

alf=Param.alf;
bet=Param.bet;

```

```

%estimacion inicial para las coordenadas dependientes
pdep0=estimadoinicial;

```

```

beta=pdep0(1);
chi=pdep0(2);
pkin=y(7:10);
pdin=y(4:6);

% variables
xc=pkin(1);
yc=pkin(2);
phi=pkin(3);
theta=pdin(1);
psi=pdin(2);
gamma=pdin(3);
epsi=pkin(4);

C_h = Rest_holonoma(Rd,Rt,beta,chi,epsi,gamma,theta,x4,x5,z4,z5);
Cdep = Jacobian_rest_holonoma(Rd,beta,chi,epsi,gamma,theta,x4,x5,z4,z5);

%algoritmo de newton-raphson con las restricciones holonomas
varpdep=-(Cdep^-1)*C_h;
pdep=pdep0+varpdep;
estimadoinicial=pdep;

beta=pdep(1);
chi=pdep(2);

Dkindep = Dkindep_matrix(Rd,Rt,beta,chi,epsi,gamma,phi,theta,x4,x5,z4,z5);
Ddin =Ddin_matrix(Rd,Rt,beta,chi,epsi,gamma,phi,theta,x4,x5,z4,z5);

D=-(Dkindep^-1)*Ddin;
E=[eye(3,3);D];

%Calculo las derivadas de las coordenadas dependientes e independientes
dpkindep=D*y(1:3);
dpdin=y(1:3);
dpkin=dpkindep(1:4);
dpdep=dpkindep(5:6);

dxc=dpkin(1);
dyc=dpkin(2);
dphi=dpkin(3);
dtheta=dpdin(1);
dpsi=dpdin(2);
dgamma=dpdin(3);
depsi=dpkin(4);
dbeta=dpdep(1);
dchi=dpdep(2);

dp=[dxc dyc dphi dtheta dpsi dbeta dgamma depsi dchi]';

C = C_Bicicleta(Rd,Rt,beta,chi,dbeta,depsi,...
    dgamma,dphi,dpsi,dtheta,dxc,dyc,epsi,gamma,phi,theta,x4,x5,z4,z5);

dDkindep = dDkindep_matrix(Rd,Rt,beta,chi,dbeta,...
    dchi,depsi,dphi,epsi,gamma,phi,theta,x4,x5,z4,z5);

```

```

dDdin = dDdin_matrix(Rd,Rt,beta,chi,dgamma,...
    dtheta,epsi,gamma,phi,theta,x4,x5,z4,z5);

dE=[zeros(3,1);-(Dkindep^-1)*(dDdin*dpdin+dDkindep*dpkindep)];

dEbaumgarte=[zeros(3,1);-(Dkindep^-1)*(dDdin*dpdin+dDkindep*dpkindep...
    +2*alf*dDdin*dpdin+2*alf*dDkindep*dpkindep+(bet^2)*C)];

%fuerzas
M_dir=0;
Mvue=0;
Mped=10;

% Fuerzas generalizadas y matriz de masas
M = massMatrixBici(I2xx,I3xx,I4xx,I4xy,I5xx,I4xz,I2yy,I3yy,...
    I4yy,I4yz,I5yy,I3zx,I3zz,I4zz,Rt,beta,epsi,gamma,m2,m3,...
    m4,m5,phi,psi,theta,x3,x4,x5,z3,z4,z5);

Qv = centrifugalForceBici(I2xx,I3xx,I4xx,I4xy,I5xx,I4xz,I2yy,...
    I3yy,I4yy,I4yz,I5yy,I3zx,I3zz,I4zz,Rt,beta,dbeta,depsi,dgamma,...
    dphi,dpsi,dtheta,epsi,gamma,m2,m3,m4,m5,phi,psi,theta,x3,x4,x5,z3,z4,z5);

Qg = gravityForceBici(Rt,beta,g,gamma,m2,m3,m4,m5,theta,x3,x4,x5,z3,z4,z5);

Qareo = areoForceBici(Rt,beta,cv,dbeta,dphi,dtheta,dxc,dyc,phi,theta,x3,z3);

T = Momentos(M_dir,Mped,Mvue);
F = Giros(beta,gamma,phi,theta);
Q=F*T;

Q_h(1,1)=Q(4);
Q_h(2,1)=Q(5);
Q_h(3,1)=Q(7);
Q_h(4,1)=Q(1);
Q_h(5,1)=Q(2);
Q_h(6,1)=Q(3);
Q_h(7,1)=Q(8);
Q_h(8,1)=Q(6);
Q_h(9,1)=Q(9);

Qv_h(1,1)=Qv(4);
Qv_h(2,1)=Qv(5);
Qv_h(3,1)=Qv(7);
Qv_h(4,1)=Qv(1);
Qv_h(5,1)=Qv(2);
Qv_h(6,1)=Qv(3);
Qv_h(7,1)=Qv(8);
Qv_h(8,1)=Qv(6);
Qv_h(9,1)=Qv(9);

Qg_h(1,1)=Qg(4);
Qg_h(2,1)=Qg(5);
Qg_h(3,1)=Qg(7);
Qg_h(4,1)=Qg(1);
Qg_h(5,1)=Qg(2);
Qg_h(6,1)=Qg(3);
Qg_h(7,1)=Qg(8);

```

```

Qg_h(8,1)=Qg(6);
Qg_h(9,1)=Qg(9);

Qareo_h(1,1)=Qareo(4);
Qareo_h(2,1)=Qareo(5);
Qareo_h(3,1)=Qareo(7);
Qareo_h(4,1)=Qareo(1);
Qareo_h(5,1)=Qareo(2);
Qareo_h(6,1)=Qareo(3);
Qareo_h(7,1)=Qareo(8);
Qareo_h(8,1)=Qareo(6);
Qareo_h(9,1)=Qareo(9);

% cálculo de las aceleraciones independientes
total=Qg_h+Qv_h-M*dEbaumgarte+Qareo_h;
ddpdin=(E'*M*E)\E'*(Qv_h+Qg_h+Qareo_h+Q_h-M*dE);
ddgamma=ddpdin(3);
ddtheta=ddpdin(1);

%% CONTROLADOR
if abs(ddgamma)>=(2*pi-0.95*2*pi) || abs(ddtheta)>=(2*pi-0.95*2*pi)

dy=[ddpdin;dpdin;dpkin];
dy(1)=0;
dy(3)=0;

else

dy=[ddpdin;dpdin;dpkin];

end

```

7.3 simula_Bicicleta

```

global Param
global estimadoinicial

%Parámetros de la bicicleta

Param=Parametros;

Rt=Param(3);
Rd=Param(4);
x3=Param(5);
z3=Param(6);
x4=Param(7);
z4=Param(8);
x5=Param(9);
z5=Param(10);

m2=Param(11);
m3=Param(12);
m4=Param(13);
m5=Param(14);

```

```
g=Param(29);  
cv=Param(30);
```

```
I2xx=Param(15);  
I2yy=Param(16);  
I3xx=Param(17);  
I3yy=Param(18);  
I3zz=Param(19);  
I3zx=Param(20);  
I4xx=Param(21);  
I4yy=Param(22);  
I4zz=Param(23);  
I4xy=Param(24);  
I4xz=Param(25);  
I4yz=Param(26);  
I5xx=Param(27);  
I5yy=Param(28);
```

```
alf=Param(40);  
bet=Param(41);
```

```
Param.alf=alf;  
Param.bet=bet;
```

```
Param.Rt=Rt;  
Param.Rd=Rd;  
Param.x3=x3;  
Param.z3=z3;  
Param.x4=x4;  
Param.z4=z4;  
Param.x5=x5;  
Param.z5=z5;
```

```
Param.m2=m2;  
Param.m3=m3;  
Param.m4=m4;  
Param.m5=m5;
```

```
Param.g=g;  
Param.cv=cv;
```

```
Param.I2xx=I2xx;  
Param.I2yy=I2yy;  
Param.I3xx=I3xx;  
Param.I3yy=I3yy;  
Param.I3zz=I3zz;  
Param.I3zx=I3zx;  
Param.I4xx=I4xx;  
Param.I4yy=I4yy;  
Param.I4zz=I4zz;  
Param.I4xy=I4xy;  
Param.I4xz=I4xz;  
Param.I4yz=I4yz;  
Param.I5xx=I5xx;  
Param.I5yy=I5yy;
```

```
%Condiciones iniciales de las coordenadas generalizadas del sistema  
dpdin0=[0 10 0]';
```

```

pdin0=[0 0 0]';
pkin0=[0 0 0 0]';
estimadoinicial=[0 pi/2]';

y0=[dpdin0' pdin0' pkin0']';

t=[0:0.01:7]';

[t,y]=ode45(@dinBicicleta,t,y0);

%% postproceso
pkin=y(:,7:10);
pdin=y(:,4:6);
pdep=zeros(length(t),2);
estimadoinicial=[0 pi/2];

for i=1:length(t)
pdep0=estimadoinicial;
beta=pdep0(1);
chi=pdep0(2);

% variables
xc=pkin(i,1);
yc=pkin(i,2);
phi=pkin(i,3);
theta=pdin(i,1);
psi=pdin(i,2);
gamma=pdin(i,3);
epsi=pkin(i,4);

C_h = Rest_holonoma(Rd,Rt,beta,chi,epsi,gamma,theta,x4,x5,z4,z5);
Cdep = Jacobian_rest_holonoma(Rd,beta,chi,epsi,gamma,theta,x4,x5,z4,z5);

%algoritmo de newton-raphson con las restricciones holonomas
varpdep=- (Cdep^-1)*C_h;
pdep(i,:)=pdep0'+varpdep;
estimadoinicial=pdep(i,:);

Dkindep = Dkindep_matrix(Rd,Rt,beta,chi,epsi,gamma,phi,theta,x4,x5,z4,z5);
Ddin =Ddin_matrix(Rd,Rt,beta,chi,epsi,gamma,phi,theta,x4,x5,z4,z5);

D=- (Dkindep^-1)*Ddin;

dpkindep(:,i)=D*y(i,1:3)';
dpdin(:,i)=y(i,1:3)';
dpkin(:,i)=dpkindep(1:4,i);
dpdep(:,i)=dpkindep(5:6,i);

dbeta=dpdep(1,i);
dchi=dpdep(2,i);
dxc=dpkin(1,i);
dyc=dpkin(2,i);
dphi=dpkin(3,i);
dtheta=dpdin(1,i);
dpsi=dpdin(2,i);
dgamma=dpdin(3,i);
depsi=dpkin(4,i);

```

```

C = C_Bicicleta(Rd,Rt,beta,chi,dbeta,depsi,dgamma,...
    dphi,dpsi,dtheta,dxc,dyc,epsi,gamma,phi,theta,x4,x5,z4,z5);

end
beta=pdep(:,1);
chi=pdep(:,2);
xc=pkin(:,1);
yc=pkin(:,2);
phi=pkin(:,3);
theta=pdin(:,1);
psi=pdin(:,2);
gamma=pdin(:,3);
epsi=pkin(:,4);

dbeta=dpdep(1,:);
dchi=dpdep(2,:);
dxc=dpkin(1,:);
dyc=dpkin(2,:);
dphi=dpkin(3,:);
dtheta=dpdin(1,:);
dpsi=dpdin(2,:);
dgamma=dpdin(3,:);
depsi=dpkin(4,:);

y_sol(:,1)=xc;
y_sol(:,2)=yc;
y_sol(:,3)=phi;
y_sol(:,4)=theta;
y_sol(:,5)=psi;
y_sol(:,6)=beta;
y_sol(:,7)=gamma;
y_sol(:,8)=epsi;
y_sol(:,9)=chi;

for i=1:length(t)
ecuac(:,i)=C_Bicicleta(Rd,Rt,beta(i),chi(i),dbeta(i),depsi(i),dgamma(i),...
dphi(i),dpsi(i),dtheta(i),dxc(i),dyc(i),epsi(i),gamma(i),...
phi(i),theta(i),x4,x5,z4,z5);
end

figure(1)
plot(t,ecuac(1,:), 'g',t,ecuac(2,:), 'y',t,ecuac(3,:), 'r',t,ecuac(4,:), 'b',t,ec
uac(5,:), 'm',t,ecuac(6,:))
title('Comprobación de las ecuaciones de restriccion')
xlabel('tiempo (s)')
ylabel('Variación de las ecuaciones de restriccion')

figure(2)
plot(t,dxc, 'g',t,dyc, 'b',t,dphi, 'm',t,dtheta, 'y',t,dpsi,t,dbeta, 'r',t,dgamma,
t,depsi,t,dchi)
title('Evolución de las velocidades')
xlabel('tiempo (s)')
ylabel('Velocidades')

figure(3)
plot(t,dpsi)

```

```
title('Evolución de la velocidad de la rueda trasera')
xlabel('tiempo (s)')
ylabel('Velocidad de la rueda trasera')

figure(4)
plot(t,theta)
title('Evolución del ángulo de vuelco')
xlabel('tiempo (s)')
ylabel('Angulo theta')

figure(5)
plot(t,gamma)
title('Evolución del ángulo de dirección')
xlabel('tiempo (s)')
ylabel('Angulo gamma')

for i = 1:length(t)
animaBici(t(i),y_sol(i,:));
end
```

8 BIBLIOGRAFÍA

- [1] ROBERSON, Robert E. y SCHWERTASSEK, Richard. *Dynamics of Multibody Systems*. Springer – Verlag, 1988
- [2] SHABANA, Ahmed A. *Computational Dynamics*. 1989
- [3] ESCALONA, Jose Luis. Dinámica inversa de una bicicleta a partir de datos experimentales. Material de la asignatura de Cinemática y dinámica de máquinas.
- [4] ESCALONA, Jose Luis. Cinemática de la bicicleta. Material de la asignatura de Cinemática y dinámica de máquinas.
- [5] A.L. Schwab; J.P. Meijaard. Dynamics of Flexible Multibody Systems with Non – Holonomic Constraints: A Finite Element Approach, *Multibody Systems Dynamics* 10: 2003, p. 107 – 123
- [6] SHABANA, Ahmed A. *Dynamics of Multibody Systems*. New York: Wiley, 1989