



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

ESTUDIO NUMÉRICO DEL COMPORTAMIENTO DE UN VEHÍCULO MEDIANTE ANÁLISIS DE SISTEMAS MULTICUERPO

Alumno: Álvaro Brazales Ruiz

Tutor: Prof. D. Javier Fernández Aceituno

Depto.: Ingeniería Mecánica y Minera

Septiembre, 2019

DEPARTAMENTO DE INGENIERÍA MECÁNICA Y
MINERA

ESCUELA POLITÉCNICA SUPERIOR DE
LINARES

**ESTUDIO NUMÉRICO DEL
COMPORTAMIENTO DE UN VEHÍCULO
MEDIANTE ANÁLISIS DE SISTEMAS
MULTICUERPO**

Grado en Ingeniería Mecánica

Autor

Álvaro Brazales Ruiz



Tutor

Javier Fernández Aceituno

Linares, Septiembre de 2019

ÍNDICE DE LA MEMORIA

1-INTRODUCCIÓN	8
2-RESUMEN	10
3-ABSTRACT	12
4-MARCO TEÓRICO	14
4.1-Ecuaciones de posición	14
4.1.1-Posición global o absoluta de un punto cualquiera del sistema multicuerpo	14
4.1.2-Coordenadas generalizadas de un sólido.....	16
4.1.3-Vector de coordenadas generalizadas del sistema multicuerpo.....	17
4.2-Ecuaciones de velocidad	17
4.2.1-Velocidad global o absoluta de un punto cualquiera del sistema multicuerpo.....	17
4.2.2-Velocidades generalizadas de un sólido	18
4.2.3-Vector de velocidades generalizadas del sistema multicuerpo.....	19
4.3-Ecuaciones de aceleración.....	19
4.3.1-Aceleración global o absoluta de un punto cualquiera del sistema multicuerpo	19
4.3.2-Vector de aceleraciones generalizadas de un sólido	20
4.3.3-Vector de aceleraciones generalizadas del sistema multicuerpo	20
4.4-Restricciones del sistema multicuerpo	21
4.4.1-Restricción de eslabón fijo	21
4.4.2-Restricción de unión articulada	22
4.4.3-Restricción de velocidad de giro constante	23
4.4.4-Vector de restricciones del sistema multicuerpo	24
4.4.5-Matriz Jacobiana de las restricciones.....	25
4.4.6-Otros términos relacionados con el vector de restricciones	26
4.5-Concepto de fuerza generalizada	27
4.5.1-Concepto de desplazamiento virtual	27
4.5.2-Concepto de trabajo virtual	29
4.5.3-Vector de fuerza generalizada	30
4.6-Generalización de las fuerzas presentes en el modelo	31
4.6.1-Generalización de la gravedad.....	31
4.6.2-Generalización de la fuerza en un muelle-amortiguador	32

4.6.3-Generalización de un par motor.....	35
4.6.4-Generalización de las fuerzas de contacto.....	36
4.7-Ecuaciones dinámicas empleadas en el modelo	37
4.7.1-Formulación de las ecuaciones de Newton y la ecuación de Euler	37
4.7.2-Ecuaciones adicionales para poder resolver las ecuaciones de movimiento del modelo multicuerpo.....	39
4.7.3-Multiplicadores de Lagrange	40
4.7.4-Método de estabilización de las restricciones de Baumgarte	41
4.7.5-Augmented formulation y embedding technique	42
4.8-Proceso de cálculo en las simulaciones dinámicas	43
4.8.1-Vectores de coordenadas y velocidades generalizadas iniciales	43
4.8.2-Concepto de vector de estado	43
4.9-Contacto elástico rueda-terreno	45
4.9.1-Parametrización del terreno y las ruedas.....	45
Parametrización de las ruedas.....	46
Parametrización del terreno.....	47
4.9.2-Formulación del contacto elástico. Búsqueda de puntos de contacto.....	48
4.9.3-Fuerza normal de contacto	52
4.9.4-Fuerza tangencial de contacto.....	53
4.10-Modelado del terreno	55
4.10.1-Inclusión de pendientes en el terreno	55
Transición modelada con una función polinómica de grado dos	56
Transición modelada con una función polinómica de grado tres	58
4.10.2-Modelado de irregularidades repentinas o baches en el terreno	60
4.10.3-Modelado de irregularidades continuas.....	66
5-DESCRIPCIÓN DEL SISTEMA MULTICUERPO MODELADO	68
5.1-Presentación y enumeración de los sólidos	68
5.2-Uniones entre sólidos del sistema multicuerpo	68
5.3- Parámetros que definen al modelo y justificación de los valores asignados.....	69
5.4- Vectores de coordenadas, velocidades y aceleraciones generalizadas del sistema multicuerpo.....	73
5.5- Matriz de masa del sistema multicuerpo	73
5.6- Restricciones del sistema multicuerpo	74
5.7- Vector de fuerzas generalizadas externas	75

5.8- Configuración inicial del sistema multicuerpo	75
6-IMPLEMENTACIÓN DEL MODELO EN EL SOFTWARE <i>MATLAB</i>.....	77
6.1-Función que define todas las características y parámetros del sistema multicuerpo	77
6.2-Función que obtiene la derivada respecto al tiempo del vector de estado	79
6.3-Funciones auxiliares.....	81
6.4-Programa que realiza las simulaciones dinámicas.....	81
6.4.1-Integración numérica.....	81
6.4.2-Integrador numérico	82
6.5- Función para el posprocesado de resultados	83
6.6- Funciones para la representación de resultados.....	85
6.7- Función para crear una animación de las simulaciones dinámicas.....	86
7-ENSAYOS	88
7.1- Ensayo estático	88
7.2- Simulación del comportamiento del remolque ante una irregularidad repentina	94
7.3- Comparativa del comportamiento del modelo multicuerpo en un terreno perfectamente plano y en un terreno con irregularidades	98
7.4- Simulación de la influencia del coeficiente de fricción en la tracción del vehículo cuando asciende por una pendiente.....	104
8-CONCLUSIONES	112
9-REFERENCIAS BIBLIOGRÁFICAS	115
10-ANEXOS	116

1-INTRODUCCIÓN

El objeto del presente *TFG* será crear un modelo bidimensional numérico, empleando la teoría de sistemas multicuerpo, que permita modelar y analizar la dinámica vertical de un vehículo con remolque. Aunque con menos detalle, la dinámica longitudinal del modelo también ha sido modelada.

La teoría de sistemas multicuerpo ha sido clave y fundamental para la realización del *TFG*. Como consecuencia, se introducirán a continuación de manera escueta los conceptos teóricos de la teoría de sistemas multicuerpo empleados para simular el comportamiento del vehículo y del remolque. Los conceptos teóricos utilizados se clasifican claramente en conceptos cinemáticos y dinámicos. Toda la formulación teórica implementada en el modelo supondrá que los sólidos del sistema son perfectamente rígidos.

La cinemática del modelo definirá los conceptos de coordenadas, velocidades y aceleraciones generalizadas de un sólido cualquiera del sistema multicuerpo. Dichos conceptos, expresados mediante vectores columna de tres dimensiones, resultan de desarrollar las ecuaciones de posición, velocidad y aceleración en forma vectorial y definirán unívocamente la posición, velocidad y aceleración absolutas de cualquier partícula que pertenezca a un sólido del sistema multicuerpo.

Por otro lado, la cinemática definirá las uniones o restricciones existentes entre los diferentes sólidos del sistema multicuerpo. Estas restricciones, las cuales también pueden fijar los valores en el tiempo de alguna coordenada generalizada de un sólido del sistema, restringirán el modo en que se moverán los distintos sólidos rígidos del modelo multicuerpo.

La dinámica del modelo proporcionará las ecuaciones del movimiento del sistema multicuerpo, en las cuales aparecerán las distintas fuerzas actuantes en los sólidos del sistema multicuerpo y las propiedades inerciales de los mismos. Las fuerzas externas y las fuerzas reacción aparecerán como vectores de fuerzas generalizadas. Para introducir el concepto de fuerza generalizada, será preciso desarrollar previamente el concepto de desplazamiento virtual y el principio de los trabajos virtuales.

Las ecuaciones dinámicas del modelo tendrán por objeto el cálculo, en ciertos instantes del intervalo de simulación, de las aceleraciones generalizadas del sistema multicuerpo y las fuerzas de reacción, resultantes de las restricciones del sistema multicuerpo. La cinemática del modelo se acoplará a las ecuaciones del movimiento del modelo multicuerpo mediante las restricciones del sistema multicuerpo.

Como resultado de obtener las aceleraciones generalizadas del sistema, mediante las ecuaciones dinámicas, se podrá integrar numéricamente para obtener las coordenadas generalizadas del sistema multicuerpo en un instante posterior de cualquier simulación dinámica realizada con el modelo.

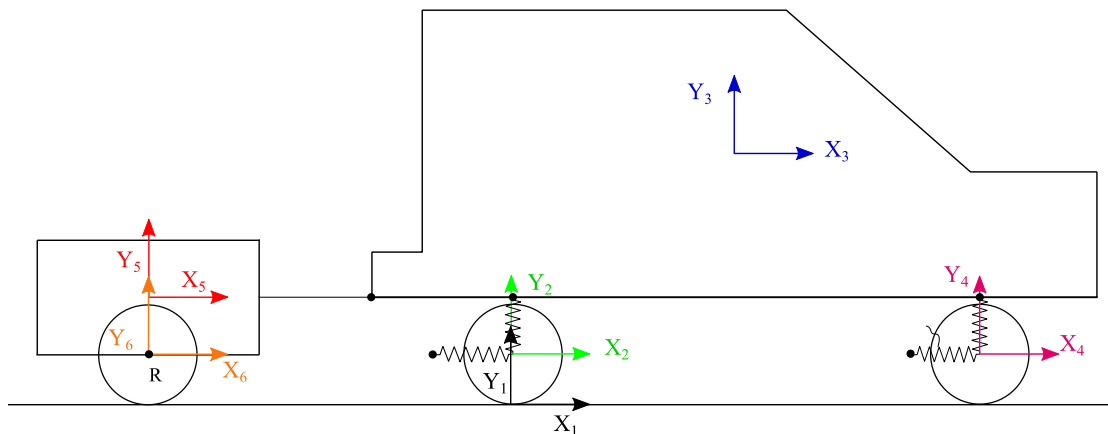
La determinación de las coordenadas generalizadas del sistema multicuerpo en ciertos instantes del intervalo de simulación dará pie a poder obtener animaciones del movimiento de los sólidos del modelo multicuerpo.

El trabajo ha sido implementado íntegramente en el software *Matlab*. El potencial de *Matlab* se adapta perfectamente a las necesidades del modelo en cuanto a programación y cálculo numérico. De este modo, todas las funciones programadas en *Matlab* tendrán por objeto aplicar la teoría de sistemas multicuerpo para modelar y simular el comportamiento del vehículo con remolque.

Sin embargo, las funciones y programas desarrollados no se ciñen únicamente a la teoría de sistemas multicuerpo. Por ejemplo, se han incluido formulaciones para modelar el contacto elástico rueda-terreno y poder modelar al terreno con diversas geometrías (pendientes, baches, irregularidades...), lo cual enriquece notablemente el TFG.

Toda la formulación bidimensional implementada se encuentra formulada de manera matricial. Por ende, la teoría de sistemas multicuerpo es la respuesta a cómo y qué se precisa para modelar teóricamente el comportamiento de un sistema mecánico compuesto de varios sólidos rígidos pero la programación y el cálculo numérico, de la mano del software *Matlab*, son los que han posibilitado que se haya podido aplicar en este TFG de una manera práctica y eficiente la teoría de sistemas multicuerpo poder simular el comportamiento de un vehículo con remolque.

El objetivo del trabajo será comprobar si el modelo implementado es capaz de modelar, dentro de unos límites coherentes y acordes a las exigencias de un trabajo de fin de grado, el comportamiento que se espera que tuviera el vehículo ante las condiciones fijadas e impuestas en cada simulación dinámica. Otros objetivos adicionales del trabajo serán conocer las limitaciones del modelo, cómo se podría mejorar en un futuro el modelo multicuerpo y analizar el potencial del modelo para simular la realidad.



2-RESUMEN

El sistema multicuerpo que pretende simular el comportamiento del vehículo y del remolque está integrado por un total de seis sólidos rígidos:

- El eslabón fijo o de referencia, modelado por el terreno.
- La rueda trasera del vehículo
- La rueda delantera del vehículo
- El chasis
- El remolque
- La rueda del remolque

Los sólidos del sistema multicuerpo estarán conectados entre sí mediante resortes-amortiguadores y mediante uniones articuladas.

Se ha modelado un contacto de tipo elástico entre las ruedas y el terreno, se admitirá la deformación de los mismos en la interfaz de contacto rueda-terreno. La formulación implementada en el modelo para modelar el contacto elástico tendrá como propósito determinar en cada instante los puntos de contacto y calcular las fuerzas normales y tangenciales de contacto en cada rueda del sistema multicuerpo.

Se formularán mediante ecuaciones no lineales las condiciones físicas que deben de satisfacer los puntos de contacto en cada instante. La fuerza normal de contacto será crucial en la dinámica vertical del modelo y estará relacionada directamente con la penetración o indentación de las ruedas del modelo multicuerpo en el terreno. El principal propósito de la formulación de la dinámica longitudinal será calcular el valor de la fuerza tangencial de contacto en cada rueda del sistema multicuerpo, la cual determinará la fuerza de impulsión o tracción desarrollada por el vehículo en cada instante.

Se ha modelado el terreno de modo que puedan incluirse tramos perfectamente planos, tramos con pendiente, irregularidades repentinas o baches e irregularidades en toda la extensión del terreno. Para ello, será necesario definir la función matemática que defina la altura del terreno en cualquier punto del mismo y que modele la geometría deseada.

El modelo se ha programado y parametrizado en el software *Matlab* de modo que se pueda cambiar rápidamente, solamente modificando los valores de ciertas variables de entrada, cualquier propiedad del modelo y del entorno de trabajo (dimensiones características y propiedades mecánicas de cada sólido, parámetros de la suspensión, parámetro del contacto elástico rueda-terreno, geometría del terreno, par motor actuante en las ruedas del vehículo, coeficiente de fricción del terreno, velocidad límite de deslizamiento...). Esto permite estudiar el comportamiento del sistema multicuerpo en una infinidad de casos de estudio de manera rápida y eficiente.

Para que el análisis y estudio de las simulaciones dinámicas realizadas con el modelo sea inmediato, se han programado un conjunto de funciones que tienen por objeto el posprocesado de resultados, representación de cualquier conjunto de variables consideradas de interés y animaciones de las simulaciones dinámicas.

El trabajo se encuentra estructurado en cuatro partes:

1. La primera parte tiene como objeto el desarrollo del marco teórico necesario para la realización del trabajo.
2. La segunda parte del trabajo definirá de manera exhaustiva las características y parámetros que definen al sistema multicuerpo modelado
3. La tercera parte pretende explicar cómo se ha implementado el modelo multicuerpo en el software *Matlab*.
4. Finalmente, la cuarta parte del trabajo presentará un conjunto de simulaciones dinámicas realizadas con el sistema multicuerpo para el *TFG*.

Se presentarán y analizarán los resultados de un total de cuatro ensayos:

- El primer ensayo consiste en una simulación estática. Se determinará la configuración de equilibrio estático del sistema multicuerpo.
- El segundo ensayo tiene por objeto analizar el comportamiento del remolque cuando impacta con una irregularidad repentina o bache.
- El tercer ensayo pretende comparar el comportamiento del vehículo cuando avanza por un terreno perfectamente plano y cuando avanza por un terreno irregular.
- Finalmente, el cuarto ensayo pretende simular la influencia del coeficiente de fricción en la capacidad de impulsión del vehículo cuando se dispone a superar una pendiente.

3-ABSTRACT

The aim of this final grade work is to analyze the vertical dinámica of a vehicle with a tráiler. For this purpose the multibody systems theory has been used. On the other hand, the longitudinal dynamics of the model has been modeled with less detail.

The multibody system is composed of six rigid bodies:

- The fixed link
- The rear wheel of the vehicle
- The front wheel of the vehicle
- The chassis
- The trailer
- Trailer Wheel

The solids of the multibody system will be connected each other by means of springs-dampers elements and revolute joints.

An elastic type contact between the wheels and the ground has been modeled, the deformation of the wheel and the ground will be admitted in the contact surface. The formulation of the elastic contact will aim to determine the contact points at each moment and calculate the normal and tangential forces at each wheel of the multibody system.

The physical conditions that the contact points must satisfy will be formulated through non-linear equations. Normal contact force will be crucial in the vertical dynamics of the multibody model and will be directly related to the penetration of the wheels in the ground. The main purpose of longitudinal dynamics will be to calculate the value of the tangential contact force as it will determine the driving force generated by the vehicle.

The terrain has been modeled so that perfectly flat sections, sections with slopes, potholes and irregularities can be included through the land. For this will be necessary to define the mathematical function that defines the height of the terrain at any point and that models the desired geometry.

The model has been programmed and parametrized to quickly modify any property of the multibody system. This allow us to study the behavior of the multibody system in many situations.

The model has been fully implemented in *Matlab*. *Matlab* software perfectly satisfies the needs of the work in terms of programming and numerical calculation. All the functions programmed in *Matlab* are intended to apply the theory of multibody systems to model the behavior of the vehicle. The functions programmed for the postprocessing of results, representation of variables and animations will make the simulations analysis immediate.

The work is structured in four parts. The first part will develop the theoretical concepts necessary for the development of work. The second part will define in detail the properties

and characteristics of the multibody system. The third part aims to explain how the model has been implemented in *Matlab*. Finally, the fourth part will present the dynamic simulations performed with the multibody model.

The dynamic simulations performed with the model are structured in a total of four study cases:

- In the first one a static simulation will be carried out with the multibody system. The purpose of the simulation will be to determine the static equilibrium configuration of the multibody system.
- The second one will analyze the behavior of the trailer when it hits a pothole.
- The third one aims to analyze the behavior of the vehicle when it advances through a perfectly flat ground and when it advances through uneven terrain.
- To conclude, the fourth one will analyze the influence of the friction coefficient on the traction capacity of the vehicle to overcome a slope.

The main purpose of the final degree project will be to check if the model is capable of reproducing the expected behavior of the vehicle under various conditions. Other additional objectives will be know the limitations of the model, the potential of simulating reality and possible future improvements.

4-MARCO TEÓRICO

4.1-Ecuaciones de posición

4.1.1-Posición global o absoluta de un punto cualquiera del sistema multicuerpo

Todo el marco teórico que se desarrollará parte de las siguientes premisas:

- Se considerará que todos los sólidos del modelo multicuerpo son perfectamente rígidos. Por consiguiente, ningún sólido del sistema multicuerpo sufrirá deformación alguna durante el transcurso de cualquier estudio numérico. Sólo se contemplará un comportamiento elástico para modelar el contacto rueda-terreno.
- Los sólidos del modelo se desplazarán de manera bidimensional. Por ende, toda la formulación del marco teórico atenderá a una formulación plana.

Para poder definir en cada instante la posición de cualquier punto que pertenezca a un sólido del sistema multicuerpo será necesario definir en cada sólido un sistema de referencia. Los sistemas de referencia se unirán rígidamente a cada uno de los sólidos en un punto y serán un reflejo de los movimientos que experimenten en cada instante. Si un sólido del sistema multicuerpo se traslada o rota su sistema de coordenadas lo hará con él, manteniéndose constante la distancia de cualquier punto del sólido respecto a su sistema de referencia (consecuencia directa de abordar la dinámica del sólido rígido).

Para simplificar la formulación y modelado del sistema multicuerpo se ha escogido como origen de cada sistema de referencia el centro de gravedad del sólido al que pertenece. Por ello, las coordenadas de cualquier vector de posición local harán referencia a la distancia del punto en estudio respecto al centro de gravedad de dicho sólido.

Parece lógico querer expresar la posición de cualquier punto respecto a una referencia fija. Se necesita de un sistema de referencia que no cambie ni su orientación ni su posición y desde el que poder definir la configuración del sistema multicuerpo en cualquier instante. Dicho sistema de referencia será el sistema de referencia absoluto o global. La elección de la orientación y posición de dicho sistema de referencia será arbitraria.

Las ecuaciones de posición tienen por objeto determinar la posición absoluta de cualquier punto o partícula P que pertenece a un sólido i cualquiera del sistema multicuerpo. Las ecuaciones de posición se expresarán en forma vectorial y estarán definidas por dos ecuaciones escalares, por abordar la formulación plana.

El vector $\bar{\mathbf{u}}_P^i$ define la posición local de una partícula de la partícula P del sólido i:

$$\bar{\mathbf{u}}_P^i = \begin{bmatrix} \bar{x}_P^i \\ \bar{y}_P^i \end{bmatrix} \quad (1)$$

Siendo $\bar{x}_P^i$ y $\bar{y}_P^i$ las coordenadas cartesianas locales del punto P, expresadas en el sistema de referencia del sólido i.

Para poder obtener la posición absoluta del punto P, debido a la diferencia de orientación existente entre el sistema de referencia global y el sistema de referencia local del sólido, será necesario cambiar de base el vector de posición local. En el caso de que las orientaciones del sólido y del sistema de referencia absoluto sean idénticas, el cambio de base no tendrá efecto alguno.

El cambio de base del vector de posición local se efectuará multiplicando el vector $\bar{\mathbf{u}}_P^i$ por una matriz. Dicha matriz será denotada por matriz de rotación y, por tratarse de un estudio bidimensional, será una matriz cuadrada de dos dimensiones. La matriz de rotación $\mathbf{A}^i$ de un sólido i cualquiera del sistema multicuerpo con una orientación θ^i cualquiera se definirá del siguiente modo:

$$\mathbf{A}^i(\theta) = \begin{bmatrix} \cos(\theta^i) & -\sin(\theta^i) \\ \sin(\theta^i) & \cos(\theta^i) \end{bmatrix} \quad (2)$$

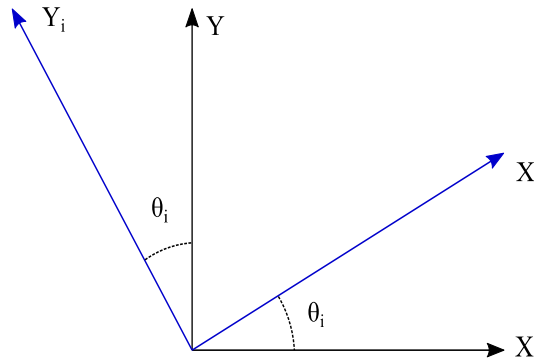


Figura 1. Rotación plana de un sólido cualquiera del sistema multicuerpo

Por ende, multiplicando la matriz $\mathbf{A}^i$ por el vector de posición local $\bar{\mathbf{u}}_P^i$ se obtendrá la posición local de la partícula P en términos del sistema de referencia global o absoluto:

$$\mathbf{u}_P^i = \mathbf{A}^i \bar{\mathbf{u}}_P^i = \begin{bmatrix} \cos(\theta^i) & -\sin(\theta^i) \\ \sin(\theta^i) & \cos(\theta^i) \end{bmatrix} \begin{bmatrix} \bar{x}_P^i \\ \bar{y}_P^i \end{bmatrix} \quad (3)$$

La posición global de cualquier punto del sistema multicuerpo se podrá expresar como la posición absoluta del origen de su sistema de referencia más la posición local de dicho punto expresada en términos del sistema de referencia global.

Será necesario definir la posición absoluta del origen del sistema de referencia de cualquier sólido del sistema multicuerpo. La posición absoluta del origen del sistema de referencia de un sólido i del modelo multicuerpo se definirá mediante el vector $\mathbf{R}^i$:

$$\mathbf{R}^i = \begin{bmatrix} R_x^i \\ R_y^i \end{bmatrix} \quad (4)$$

Por consiguiente, la posición absoluta de cualquier partícula o punto P de un sólido i del sistema multicuerpo se podrá definir mediante la siguiente ecuación vectorial:

$$\mathbf{r}_P^i = \mathbf{R}^i + \mathbf{A}^i \bar{\mathbf{u}}_P^i \quad (5)$$

Desglosando la ecuación vectorial (5) en sus respectivas componentes:

$$\begin{bmatrix} r_{Px}^i \\ r_{Py}^i \end{bmatrix} = \begin{bmatrix} R_x^i \\ R_y^i \end{bmatrix} + \begin{bmatrix} \cos(\theta^i) & -\sin(\theta^i) \\ \sin(\theta^i) & \cos(\theta^i) \end{bmatrix} \begin{bmatrix} \bar{u}_{Px}^i \\ \bar{u}_{Py}^i \end{bmatrix} \quad (6)$$

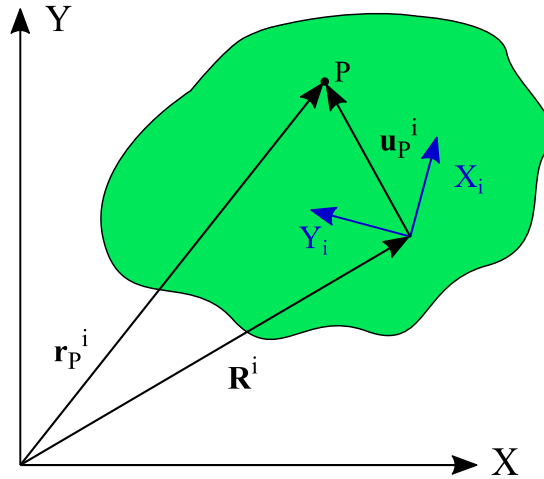


Figura 2. Posición absoluta de una partícula P de un sólido i del sistema multicuerpo

4.1.2-Coordenadas generalizadas de un sólido

Queda constancia de que se puede describir la posición de cualquier partícula P de un sólido i del sistema multicuerpo a partir de tres variables, las componentes del vector $\mathbf{R}^i$ y el escalar θ^i . Estas variables serán las coordenadas generalizadas del sólido i. Se trata de un conjunto de coordenadas que definen de manera unívoca la configuración espacial de un sólido i del modelo multicuerpo.

Las dos primeras coordenadas generalizadas hacen referencia a la posición relativa del sólido respecto al sistema de referencia absoluto mientras que la tercera coordenada generalizada indica la orientación relativa del sólido respecto al sistema de referencia absoluto, el cual no modificará en ningún momento ni de orientación ni de posición.

Se expresarán las coordenadas generalizadas de cualquier sólido del sistema multicuerpo mediante un vector de coordenadas generalizadas $\mathbf{q}^i$:

$$\mathbf{q}^i = \begin{bmatrix} R_x^i \\ R_y^i \\ \theta^i \end{bmatrix} \quad (7)$$

$$\mathbf{q}^i = \begin{bmatrix} \mathbf{R}^i \\ \theta^i \end{bmatrix} \quad (8)$$

4.1.3-Vector de coordenadas generalizadas del sistema multicuerpo

Se puede agrupar los vectores de coordenadas generalizadas $\mathbf{q}^i$ de cada sólido del sistema multicuerpo en un único vector de coordenadas generalizadas del sistema multicuerpo $\mathbf{q}$, que definirá de manera unívoca la configuración espacial de todo el sistema multicuerpo:

$$\mathbf{q} = [R_x^1 \ R_y^1 \ \theta^1 \ R_x^2 \ R_y^2 \ \theta^2 \ R_x^3 \ R_y^3 \ \theta^3 \ R_x^4 \ R_y^4 \ \theta^4 \ \dots \ R_x^{ns} \ R_y^{ns} \ \theta^{ns}]^T \quad (9)$$

$$\mathbf{q} = [\mathbf{R}^1 \ \theta^1 \ \mathbf{R}^2 \ \theta^2 \ \mathbf{R}^3 \ \theta^3 \ \dots \ \mathbf{R}^n \ \theta^{ns}]^T \quad (10)$$

El número de componentes del vector de coordenadas generalizadas del sistema multicuerpo $\mathbf{q}$ se obtendrá multiplicando por tres el número de sólidos ns que conformen el sistema multicuerpo.

4.2-Ecuaciones de velocidad

4.2.1-Velocidad global o absoluta de un punto cualquiera del sistema multicuerpo

El objeto de las ecuaciones de velocidad será determinar la velocidad absoluta $\dot{\mathbf{r}}_P^i$ de cualquier partícula o punto P que pertenezca a un sólido i cualquiera del sistema multicuerpo.

Derivando respecto al tiempo la ecuación (5) se obtendrá las ecuaciones de velocidad en forma vectorial:

$$\dot{\mathbf{r}}_P^i = \dot{\mathbf{R}}^i + \dot{\theta}^i \mathbf{A}_\theta^i \bar{\mathbf{u}}_P^i \quad (11)$$

El primer término del miembro derecho de la ecuación (11) es la velocidad absoluta del origen del sistema de referencia del sólido i , definida por el vector $\dot{\mathbf{R}}^i$.

El segundo término del miembro derecho de la ecuación (11) resulta de aplicar la regla de la cadena al segundo término del miembro derecho de la ecuación (5) y define la velocidad relativa del punto P respecto al origen del sistema de referencia del sólido i . Dicho término está compuesto por la velocidad angular del sólido $\dot{\theta}^i$, la derivada parcial de la matriz $\mathbf{A}^i$ respecto a la orientación θ^i del sólido i y por el vector de posición local de la partícula P analizada $\bar{\mathbf{u}}_P^i$.

La derivada parcial de la matriz de orientación $\mathbf{A}^i$ del sólido i del sistema multicuerpo respecto a su orientación θ^i se define mediante la siguiente matriz:

$$\mathbf{A}_{\theta}^i = \frac{\partial \mathbf{A}^i}{\partial \theta^i} = \begin{bmatrix} -\sin(\theta^i) & -\cos(\theta^i) \\ \cos(\theta^i) & -\sin(\theta^i) \end{bmatrix} \quad (12)$$

La velocidad angular $\dot{\theta}^i$ del sólido i del modelo multicuerpo se puede definir mediante el vector $\boldsymbol{\omega}^i$, perpendicular al plano:

$$\boldsymbol{\omega}^i = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}^i \end{bmatrix} \quad (13)$$

Se puede demostrar que el segundo término del miembro derecho de la ecuación (11) equivale al producto vectorial del vector de velocidad angular del sólido $\boldsymbol{\omega}^i$ y el vector de posición local de la partícula P $\mathbf{u}_P^i$:

$$\dot{\mathbf{r}}_P^i = \dot{\mathbf{R}}^i + \boldsymbol{\omega}^i \times \mathbf{u}_P^i \quad (14)$$

Las ecuaciones de velocidad simplemente expresan que la velocidad absoluta $\dot{\mathbf{r}}_P^i$ de una partícula P de un sólido i del modelo multicuerpo puede ser expresada como la velocidad absoluta del punto escogido como origen de su sistema de referencia $\dot{\mathbf{R}}^i$, velocidad común a todos los puntos del sólido i y ligada al desplazamiento del mismo, más la velocidad relativa del punto P respecto al origen del sistema de referencia del sólido, el valor de la velocidad relativa dependerá de la partícula P analizada y está ligado directamente con la rotación del sólido analizado.

4.2.2-Velocidades generalizadas de un sólido

Las ecuaciones de velocidad sugieren que la velocidad absoluta de cualquier partícula P de un sólido i del sistema multicuerpo queda totalmente definida mediante tres velocidades: la velocidad absoluta del origen de su sistema de referencia, definida por el vector $\dot{\mathbf{R}}^i$, y la velocidad angular $\boldsymbol{\omega}^i$ del sólido en cuestión.

Por ende, definiendo un vector de velocidades generalizada $\dot{\mathbf{q}}^i$ en cada sólido del modelo multicuerpo se podrá definir la velocidad absoluta de cualquier partícula P de un sólido i sistema multicuerpo:

$$\dot{\mathbf{q}}^i = \begin{bmatrix} \dot{R}_x^i \\ \dot{R}_y^i \\ \dot{\theta}^i \end{bmatrix} \quad (15)$$

$$\dot{\mathbf{q}}^i = \begin{bmatrix} \dot{\mathbf{R}}^i \\ \dot{\theta}^i \end{bmatrix} \quad (16)$$

4.2.3-Vector de velocidades generalizadas del sistema multicuerpo

De igual modo que se procedió con el vector de coordenadas generalizadas del sistema multicuerpo $\mathbf{q}$, se puede definir un vector de velocidades generalizadas del sistema multicuerpo $\dot{\mathbf{q}}$ que defina de manera unívoca la velocidad absoluta de cualquier punto P del modelo multicuerpo:

$$\dot{\mathbf{q}} = [\dot{R}_x^1 \dot{R}_y^1 \dot{\theta}^1 \dot{R}_x^2 \dot{R}_y^2 \dot{\theta}^2 \dot{R}_x^3 \dot{R}_y^3 \dot{\theta}^3 \dot{R}_x^4 \dot{R}_y^4 \dot{\theta}^4 \dots \dot{R}_x^{ns} \dot{R}_y^{ns} \dot{\theta}^{ns}]^T \quad (17)$$

$$\dot{\mathbf{q}} = [\dot{R}^1 \dot{\theta}^1 \dot{R}^2 \dot{\theta}^2 \dot{R}^3 \dot{\theta}^3 \dot{R}^4 \dot{\theta}^4 \dots \dot{R}^{ns} \dot{\theta}^{ns}]^T \quad (18)$$

4.3-Ecuaciones de aceleración

4.3.1-Aceleración global o absoluta de un punto cualquiera del sistema multicuerpo

El siguiente paso, una vez definida la posición absoluta y velocidad absoluta de cualquier partícula P del sistema multicuerpo, mediante los vectores $\mathbf{r}_P^i$ y $\dot{\mathbf{r}}_P^i$, será definir la aceleración global $\ddot{\mathbf{r}}_P^i$ de un punto P cualquiera que pertenece a un sólido i cualquiera del sistema multicuerpo mediante las ecuaciones de aceleración.

La aceleración absoluta $\ddot{\mathbf{r}}_P^i$ de la partícula P del sólido i se definirá mediante la suma de la aceleración absoluta del origen del sistema de referencia del sólido i, definida por el vector $\ddot{\mathbf{R}}^i$, y la aceleración relativa de la partícula P respecto al origen del sistema de referencia del sólido i. Por ende, derivando respecto al tiempo la ecuación (14):

$$\ddot{\mathbf{r}}_P^i = \ddot{\mathbf{R}}^i + \dot{\theta}^i \dot{\mathbf{A}}_\theta^i \bar{\mathbf{u}}_P^i + \ddot{\theta}^i \mathbf{A}_\theta^i \bar{\mathbf{u}}_P^i \quad (19)$$

El primer término del miembro derecho de la ecuación (19) es la aceleración absoluta del origen del sistema de referencia del sólido i, aceleración común a todas las partículas del sólido i y relacionada con la aceleración lineal del sólido. El segundo y tercer término del miembro derecho de la ecuación (19) están relacionados con la rotación del sólido i.

Un sólido del sistema multicuerpo que gira podrá tener dos tipos de aceleraciones. Una de ellas, la aceleración normal $(\mathbf{a}_P^i)_n$, está relacionada únicamente con el giro o cambio de dirección del vector velocidad mientras que la segunda, la aceleración tangencial $(\mathbf{a}_P^i)_t$, está ligada al cambio de magnitud de la velocidad angular del sólido $\boldsymbol{\omega}^i$.

Una expresión más intuitiva y conocida de las ecuaciones de aceleración es la siguiente:

$$\ddot{\mathbf{r}}_P^i = \ddot{\mathbf{R}}^i + \boldsymbol{\omega}^i \times (\boldsymbol{\omega}^i \times \mathbf{u}_P^i) + \boldsymbol{\alpha}^i \times \mathbf{u}_P^i \quad (20)$$

Las aceleraciones normal y tangencial pueden definirse del siguiente modo:

$$(\mathbf{a}_P^i)_n = \boldsymbol{\omega}^i \times (\boldsymbol{\omega}^i \times \mathbf{u}_P^i) \quad (21)$$

$$(\mathbf{a}_P^i)_t = \boldsymbol{\alpha}^i \times \mathbf{u}_P^i \quad (22)$$

Como consecuencia de derivar dos veces la ecuación (5) han aparecido nuevos términos. Estos nuevos términos son la aceleración angular $\boldsymbol{\alpha}^i$ del sólido i del modelo multicuerpo, definida por un vector perpendicular al plano, y la derivada respecto al tiempo de la matriz $\mathbf{A}_\theta^i$:

$$\boldsymbol{\alpha}^i = \begin{bmatrix} 0 \\ 0 \\ \ddot{\theta}^i \end{bmatrix} \quad (23)$$

$$\dot{\mathbf{A}}_\theta^i = -\mathbf{A}^i \dot{\theta}^i \quad (24)$$

4.3.2-Vector de aceleraciones generalizadas de un sólido

Como ocurre a nivel de posición y velocidad, la aceleración absoluta $\ddot{\mathbf{r}}_P^i$ de cualquier punto o partícula P de un sólido i del modelo multicuerpo queda totalmente definida mediante tres aceleraciones: la aceleración del origen del sistema de referencia, definida por el vector $\ddot{\mathbf{R}}^i$, y aceleración angular $\boldsymbol{\alpha}^i$ del sólido i .

Consecuentemente, se podrá definir la aceleración absoluta $\ddot{\mathbf{r}}_P^i$ de cualquier partícula P de un sólido i del modelo multicuerpo mediante su vector de aceleraciones generalizadas $\ddot{\mathbf{q}}^i$:

$$\ddot{\mathbf{q}}^i = \begin{bmatrix} \ddot{R}_x^i \\ \ddot{R}_y^i \\ \ddot{\theta}^i \end{bmatrix} \quad (25)$$

$$\ddot{\mathbf{q}}^i = \begin{bmatrix} \ddot{\mathbf{R}}^i \\ \ddot{\theta}^i \end{bmatrix} \quad (26)$$

4.3.3-Vector de aceleraciones generalizadas del sistema multicuerpo

Finalmente, se podrá definir la aceleración absoluta $\ddot{\mathbf{r}}_P^i$ de cualquier partícula P del sistema multicuerpo mediante un vector $\ddot{\mathbf{q}}$ que contenga los vectores de aceleraciones generalizadas $\ddot{\mathbf{q}}^i$ de cada sólido del sistema multicuerpo:

$$\ddot{\mathbf{q}} = \left[\ddot{R}_x^1 \ddot{R}_y^1 \ddot{\theta}^1 \ddot{R}_x^2 \ddot{R}_y^2 \ddot{\theta}^2 \ddot{R}_x^3 \ddot{R}_y^3 \ddot{\theta}^3 \ddot{R}_x^4 \ddot{R}_y^4 \ddot{\theta}^4 \dots \ddot{R}_x^{ns} \ddot{R}_y^{ns} \ddot{\theta}^{ns} \right]^T \quad (27)$$

$$\ddot{\mathbf{q}} = \left[\ddot{\mathbf{R}}^1 \ddot{\theta}^1 \ddot{\mathbf{R}}^2 \ddot{\theta}^2 \ddot{\mathbf{R}}^3 \ddot{\theta}^3 \ddot{\mathbf{R}}^4 \ddot{\theta}^4 \dots \ddot{\mathbf{R}}^{ns} \ddot{\theta}^{ns} \right]^T \quad (28)$$

4.4-Restricciones del sistema multicuerpo

Para un sólido del modelo multicuerpo que no se encuentre sujeto a ningún tipo de restricción, que pueda desplazarse y girar arbitrariamente, será necesario definir sus tres coordenadas generalizadas para poder expresar su configuración en cada instante. Por consiguiente, si ningún sólido del sistema multicuerpo se encontrase sujeto a algún tipo de restricción sería necesario definir el vector de coordenadas generalizadas $\mathbf{q}^i$ de cada sólido del sistema multicuerpo.

Sin embargo, en cualquier sistema multicuerpo hay interacción y conexión entre los sólidos que lo componen. Esta conexión se lleva a cabo mediante uniones que impondrán una serie de restricciones al movimiento relativo entre ellos, anulando ciertos grados de libertad relativos. Por ello, las coordenadas generalizadas de los sólidos pasarán de ser totalmente independientes a existir ciertas dependencias entre ellas, precisamente por las uniones y ligaduras que presentan los sólidos del modelo multicuerpo.

Se pueden definir una gran cantidad de restricciones. Dependiendo del tipo de restricción se anularán más o menos y diferentes grados de libertad en el movimiento relativo de los mismos. En el presente trabajo se describirán únicamente las restricciones empleadas para modelar el comportamiento del vehículo con remolque.

En primera instancia será necesario discernir entre dos tipos de restricciones. En cualquier sistema multicuerpo podrán existir restricciones impuestas por las uniones existentes entre sólidos (unión esférica, unión cilíndrica, unión de rótula, unión prismática...) o restricciones definidas por el usuario.

Esta última puede ser, por ejemplo, fijar la magnitud de una velocidad generalizada de un sólido cualquiera del sistema multicuerpo. Piense, por ejemplo, en fijar la velocidad lineal de un eslabón que desliza por el suelo.

Al introducir restricciones de usuario estamos induciendo a lo que se conoce como dinámica inversa. Aparecerán fuerzas o momentos que serán los responsables de que se cumplan las restricciones de usuario. Aunque se definan las restricciones de usuario a nivel de velocidad, la restricción de usuario fija implícitamente la evolución temporal de la coordenada generalizada asociada a la velocidad generalizada fijada por la restricción.

Se proseguirá describiendo las restricciones empleadas en el trabajo. El número de ecuaciones necesarias para formular cada restricción dependerá del número de grados de libertad fijados por la restricción y estarán contenidas bajo un vector de restricciones $\mathbf{C}^i$.

4.4.1-Restricción de eslabón fijo

La restricción de eslabón fijo asegura que el sólido que modela o representa el sistema de referencia absoluto o global no se desplace ni gire en el plano. El vector de coordenadas

generalizadas del sólido que modela al sistema de referencia $\mathbf{q}^i$ no variará durante las simulaciones dinámicas.

Como consecuencia, se tratará de imponer que las coordenadas generalizadas del eslabón fijo sean constantes en cualquier instante. La orientación θ^i y ubicación del sistema de referencia absoluto, definida por el vector $\mathbf{R}^i$, son plenamente arbitrarias. Se tomarán de modo que se simplifique el modelado del sistema multicuerpo.

La restricción de eslabón fijo se define mediante la siguiente ecuación vectorial:

$$\mathbf{C}^i(\mathbf{q}^i) = \mathbf{0} \quad (29)$$

$$\mathbf{C}^i(\mathbf{q}^i) = \mathbf{q}^i - \mathbf{a} = \mathbf{0} \quad (30)$$

Donde $\mathbf{q}^i$ es el vector de coordenadas generalizadas del sólido que modela al eslabón fijo y $\mathbf{a}$ es un vector de constantes que define la posición y orientación del sistema de referencia absoluto:

$$\mathbf{q}^i = \begin{bmatrix} R_x^i \\ R_y^i \\ \theta^i \end{bmatrix} \quad (31)$$

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad (32)$$

Desglosando la ecuación (30) en sus respectivas componentes:

$$R_x^i = a_1 \quad (33)$$

$$R_y^i = a_2 \quad (34)$$

$$\theta^i = a_3 \quad (35)$$

Para que el sistema de referencia absoluto no cambie de posición ni de orientación deberán de aparecer una fuerza y un momento en su origen que impidan su desplazamiento o rotación.

4.4.2-Restricción de unión articulada

La restricción de unión articulada se trata de una de las restricciones más simples y comunes de cualquier modelo multicuerpo. Dicha restricción impone un desplazamiento relativo nulo entre los puntos P_i y P_j de los sólidos i y j del sistema multicuerpo.

Se trata de imponer que las posiciones absolutas $\mathbf{r}_p^i$ y $\mathbf{r}_p^j$ de los puntos articulados P_i y P_j sean idénticas. Haciendo uso de la ecuación (5) se podrá definir la ecuación vectorial que modele la restricción de una unión articulada:

$$\mathbf{C}^i(\mathbf{q}^i, \mathbf{q}^j) = \mathbf{0} \quad (36)$$

$$\mathbf{C}^i(\mathbf{q}^i, \mathbf{q}^j) = \mathbf{r}_{P_i}^i - \mathbf{r}_{P_j}^j = \mathbf{0} \quad (37)$$

$$\mathbf{r}_{P_i}^i = \mathbf{r}_{P_j}^j \quad (38)$$

Desarrollando la ecuación (38) en sus respectivas componentes:

$$\mathbf{R}^i + \mathbf{A}^i \bar{\mathbf{u}}_{P_i}^i - \mathbf{R}^j + \mathbf{A}^j \bar{\mathbf{u}}_{P_j}^j = \mathbf{0} \quad (39)$$

$$R_x^i + \bar{x}_P^i \cos \theta^i - \bar{y}_P^i \sin \theta^i - R_x^j - \bar{x}_P^j \cos \theta^j + \bar{y}_P^j \sin \theta^j = 0 \quad (40)$$

$$R_y^i + \bar{x}_P^i \sin \theta^i + \bar{y}_P^i \cos \theta^i - R_y^j - \bar{x}_P^j \sin \theta^j - \bar{y}_P^j \cos \theta^j = 0 \quad (41)$$

Se ve claramente que el vector de restricción $\mathbf{C}^i$ depende únicamente de los vectores de coordenadas generalizadas $\mathbf{q}^i$ y $\mathbf{q}^j$ de los sólidos articulados.

La unión articulada se verá cumplida por la aparición de una fuerza de reacción en el punto de articulación P que impedirá un desplazamiento relativo entre los puntos P_i y P_j .

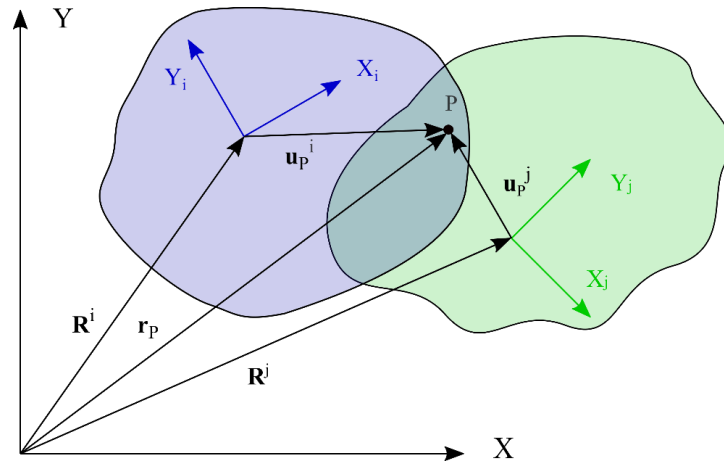


Figura 3. Unión articulada entre los sólidos i y j del sistema multicuerpo en el punto P

4.4.3-Restricción de velocidad de giro constante

La restricción de velocidad de giro constante es una restricción de usuario que se aplicará a las ruedas del vehículo. Aunque la restricción se define a nivel de velocidad, la restricción de usuario se formulará a nivel de posición. Suponga que la restricción se impone al sólido i del sistema multicuerpo.

Por ende, se trata de imponer que la derivada respecto al tiempo de la orientación θ^i del sólido i sea una constante c :

$$\frac{d\theta^i}{dt} = c = \dot{\theta}^i \quad (42)$$

$$d\theta^i = \dot{\theta}^i dt \quad (43)$$

Integrando entre el instante inicial t_0 y un instante t cualquiera del análisis numérico:

$$\int_{t_0}^t d\theta^i = \int_{t_0}^t c dt \quad (44)$$

$$\theta^i(t) - \theta^i(t_0) = \dot{\theta}^i(t - t_0) \quad (45)$$

Suponiendo que el análisis parte desde un tiempo inicial t_0 igual a cero segundos y que la orientación inicial del sólido en cuestión θ_0^i en el instante inicial es idéntica a la orientación del sistema de referencia absoluto, se podrá formular la restricción de velocidad de giro constante como sigue:

$$c^i(q^i, t) = \theta^i(t) - \dot{\theta}^i t = 0 \quad (46)$$

La restricción de velocidad de giro constante será clave en la dinámica longitudinal del modelo. Esta restricción junto a la introducción de un par motor, como una fuerza de usuario, serán las únicas vías de promover el movimiento longitudinal del vehículo modelado.

La restricción de velocidad de giro constante inducirá la aparición de un par o momento M^i que provocará que la velocidad de giro $\dot{\theta}^i$ no cambie, dinámica inversa. Registrando el valor del momento M^i se podrá definir ley de par que provocaría teóricamente que la rueda en todo momento tuviera la misma velocidad de giro $\dot{\theta}^i$.

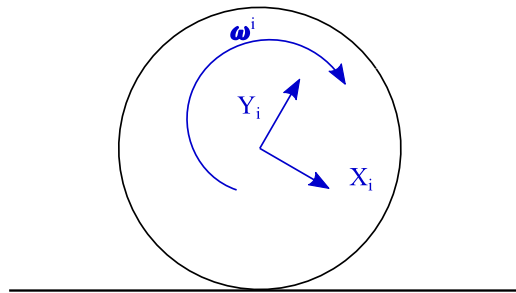


Figura 4. Restricción de velocidad de giro constante impuesta al sólido i del sistema multicuerpo

4.4.4-Vector de restricciones del sistema multicuerpo

Agruparemos la información de todas las restricciones del modelo multicuerpo en un único vector $\mathbf{C}$, el cual se definirá como vector de restricciones del sistema multicuerpo. El vector de restricciones $\mathbf{C}$ será un vector columna que tendrá tantas componentes como la sumatoria de las ecuaciones de todas las restricciones que contenga el sistema multicuerpo en cuestión.

Imagine un sistema multicuerpo que consta de dos uniones articuladas y una restricción de usuario de velocidad de giro constante. El vector de restricciones $\mathbf{C}$ tendrá un total de ocho componentes. Si las restricciones efectivamente se cumplen, el vector $\mathbf{C}$ deberá de ser un vector nulo debido a la forma de formular las ecuaciones de las restricciones.

El vector de restricciones del sistema multicuerpo $\mathbf{C}$, en su expresión más general con nr componentes, tendrá el siguiente aspecto:

$$\mathbf{C}(\mathbf{q}, t) = [C_1(\mathbf{q}, t) \ C_2(\mathbf{q}, t) \ C_3(\mathbf{q}, t) \ ... \ C_{nr}(\mathbf{q}, t)]^T \quad (47)$$

4.4.5-Matriz Jacobiana de las restricciones

Una vez introducido el concepto de restricción del sistema multicuerpo, es preciso y conveniente proseguir con el concepto de la matriz Jacobiana de las restricciones. Una vez presentada, en su forma más general, la matriz Jacobiana de las restricciones, se formulará posteriormente la matriz Jacobiana para las restricciones empleadas en el modelo.

La matriz Jacobiana de las restricciones, definida por la matriz $\mathbf{C}_q$, no es más que la derivada parcial del vector de restricciones $\mathbf{C}$ respecto al vector de coordenadas generalizadas del sistema multicuerpo $\mathbf{q}$. Por ende, al tratarse de la derivada de una matriz respecto a otro vector, la matriz $\mathbf{C}_q$ tendrá tantas filas como componentes tenga el vector de restricciones $\mathbf{C}$ y tantas columnas como componentes tenga el vector de coordenadas generalizadas del sistema multicuerpo $\mathbf{q}$.

Suponga que el vector de coordenadas generalizadas $\mathbf{q}$ tiene n componentes y el vector de restricciones $\mathbf{C}$ tiene m componentes. La matriz Jacobiana de las restricciones $\mathbf{C}_q$ de dicho modelo multicuerpo tendrá una dimensión de $m \times n$ y tomará la siguiente forma:

$$\mathbf{C}_q = \begin{bmatrix} \frac{\partial C_1}{\partial q_1} & \frac{\partial C_1}{\partial q_2} & \frac{\partial C_1}{\partial q_3} & \dots & \frac{\partial C_1}{\partial q_n} \\ \frac{\partial C_2}{\partial q_1} & \frac{\partial C_2}{\partial q_2} & \frac{\partial C_2}{\partial q_3} & \dots & \frac{\partial C_2}{\partial q_n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial C_m}{\partial q_1} & \frac{\partial C_m}{\partial q_2} & \frac{\partial C_m}{\partial q_3} & \dots & \frac{\partial C_m}{\partial q_n} \end{bmatrix} \quad (48)$$

La matriz Jacobiana $\mathbf{C}_q$ de la restricción de eslabón fijo tendrá tres filas, el número de ecuaciones que definen a la restricción, y tantas columnas como coordenadas generalizadas tenga el modelo multicuerpo.

Los únicos términos no nulos de la matriz $\mathbf{C}_q$ serán los asociados a las coordenadas generalizadas del eslabón fijo, definidas por el vector $\mathbf{q}^i$. Las demás componentes de la matriz Jacobiana, asociadas a las coordenadas generalizadas del resto de sólidos del sistema

multicuerpo, serán nulas por no presentar dependencia alguna respecto a dichas coordenadas generalizadas las ecuaciones de la restricción.

Suponga que el eslabón fijo es el primer sólido del sistema multicuerpo, sus coordenadas generalizadas, definidas por el vector $\mathbf{q}^1$, ocupan las tres primeras componentes del vector $\mathbf{q}$. Por ende, la matriz Jacobiana $\mathbf{C}_q$ asociada a la restricción de eslabón fijo tomará el siguiente aspecto

$$\mathbf{C}_q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (49)$$

La matriz Jacobiana asociada a la restricción de unión articulada $\mathbf{C}_q$ tendrá dos filas y tantas columnas como componentes tenga el vector de coordenadas generalizadas del sistema multicuerpo $\mathbf{q}$. Los términos no nulos de $\mathbf{C}_q$ serán los relacionados con las coordenadas generalizadas de los sólidos i y j del modelo multicuerpo articulados, definidas por los vectores $\mathbf{q}^i$ y $\mathbf{q}^j$.

Suponga que los sólidos articulados, sólidos i y j del sistema multicuerpo, ocupan posiciones arbitrarias dentro del modelo multicuerpo. La matriz Jacobiana $\mathbf{C}_q$ asociada a la restricción de una unión articulada presentará el siguiente aspecto:

$$\mathbf{C}_q = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & -\bar{x}_p^i \sin \theta^i - \bar{y}_p^i \cos \theta^i & -1 & 0 & \bar{x}_p^j \sin \theta^j + \bar{y}_p^j \cos \theta^j & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \bar{x}_p^i \cos \theta^i - \bar{y}_p^i \sin \theta^i & \cdots & 0 & -1 & -\bar{x}_p^j \cos \theta^j + \bar{y}_p^j \sin \theta^j & \cdots & 0 & 0 & 0 \end{bmatrix} \quad (50)$$

La matriz Jacobiana asociada a la restricción de velocidad de giro constante $\mathbf{C}_q$ tendrá una única fila, recuerde que una única ecuación define a la restricción, y tantas columnas como componentes presente el vector de coordenadas generalizadas del sistema multicuerpo $\mathbf{q}$. La matriz Jacobiana $\mathbf{C}_q$ tendrá un único término no nulo, el asociado a la orientación del sólido i sobre el cual se ha impuesto la restricción de usuario.

Suponga que el sólido i , cuya velocidad de giro se ha fijado mediante la restricción de usuario, ocupa una posición arbitraria dentro del sistema multicuerpo. La matriz Jacobiana $\mathbf{C}_q$ asociada a la restricción de velocidad de giro constante tomará el siguiente aspecto:

$$\mathbf{C}_q = [0 \quad 0 \quad 0 \cdots 0 \quad 0 \quad 1 \cdots 0 \quad 0 \quad 0] \quad (51)$$

4.4.6-Otros términos relacionados con el vector de restricciones

Es necesario presentar y formular otros términos relacionados con el vector de restricciones del sistema multicuerpo $\mathbf{C}$. Dichos términos harán acto de presencia en las ecuaciones de movimiento del modelo multicuerpo.

En primera instancia se definirá la derivada parcial respecto al tiempo del vector de restricciones $\mathbf{C}$, definido por m componentes:

$$\mathbf{C}_t = \left[\frac{\partial C_1}{\partial t} \frac{\partial C_2}{\partial t} \frac{\partial C_3}{\partial t} \frac{\partial C_4}{\partial t} \dots \frac{\partial C_m}{\partial t} \right]^T \quad (52)$$

Si se deriva respecto al tiempo la expresión (52) se obtendrá el vector $\mathbf{C}_{tt}$ que define la segunda derivada parcial respecto al tiempo del vector de restricciones del sistema multicuerpo $\mathbf{C}$:

$$\mathbf{C}_{tt} = \left[\frac{\partial}{\partial t} \left(\frac{\partial C_2}{\partial t} \right) \frac{\partial}{\partial t} \left(\frac{\partial C_3}{\partial t} \right) \frac{\partial}{\partial t} \left(\frac{\partial C_4}{\partial t} \right) \frac{\partial}{\partial t} \left(\frac{\partial C_5}{\partial t} \right) \dots \frac{\partial}{\partial t} \left(\frac{\partial C_m}{\partial t} \right) \right]^T \quad (53)$$

Tal y como se verá en la formulación dinámica del modelo, en las ecuaciones del movimiento del sistema multicuerpo aparecerá la matriz que resulta de derivar parcialmente la matriz Jacobiana $\mathbf{C}_q$ respecto al tiempo t:

$$\mathbf{C}_{qt} = \begin{bmatrix} \frac{\partial}{\partial t} \left(\frac{\partial C_1}{\partial q_1} \right) & \frac{\partial}{\partial t} \left(\frac{\partial C_1}{\partial q_2} \right) & \frac{\partial}{\partial t} \left(\frac{\partial C_1}{\partial q_3} \right) & \dots & \frac{\partial}{\partial t} \left(\frac{\partial C_1}{\partial q_n} \right) \\ \frac{\partial}{\partial t} \left(\frac{\partial C_2}{\partial q_1} \right) & \frac{\partial}{\partial t} \left(\frac{\partial C_2}{\partial q_2} \right) & \frac{\partial}{\partial t} \left(\frac{\partial C_2}{\partial q_3} \right) & \dots & \frac{\partial}{\partial t} \left(\frac{\partial C_2}{\partial q_n} \right) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial t} \left(\frac{\partial C_m}{\partial q_1} \right) & \frac{\partial}{\partial t} \left(\frac{\partial C_m}{\partial q_2} \right) & \frac{\partial}{\partial t} \left(\frac{\partial C_m}{\partial q_3} \right) & \dots & \frac{\partial}{\partial t} \left(\frac{\partial C_m}{\partial q_n} \right) \end{bmatrix} \quad (54)$$

La matriz $\mathbf{C}_{qt}$ será nula si ninguno de los términos que definen a $\mathbf{C}_q$ presenta una dependencia explícita con el tiempo.

4.5-Concepto de fuerza generalizada

Se necesitará conocer el concepto de fuerza generalizada para poder formular en las ecuaciones dinámicas las fuerzas actuantes en cada sólido del sistema multicuerpo. El concepto de fuerza generalizada tiene por base el principio de los trabajos virtuales y termina relacionándose de manera directa las coordenadas generalizadas de un sólido.

Por consiguiente, será necesario definir primeramente el concepto de desplazamiento virtual y de trabajo virtual.

4.5.1-Concepto de desplazamiento virtual

Un desplazamiento virtual aplicado a una partícula P de un sólido i del sistema multicuerpo es un desplazamiento de orden infinitesimal y que ocurre sin ningún paso de tiempo. Por dicha razón se denota por virtual, ningún desplazamiento puede ocurrir sin un paso de tiempo finito. Además, dicho desplazamiento virtual debe de respetar las restricciones que tenga el sólido en cuestión. Por ende, el desplazamiento virtual debe de respetar las condiciones de contorno del sólido i del sistema multicuerpo.

En primera instancia, la posición absoluta $\mathbf{r}_p^i$ del punto o partícula P que se desplazará virtualmente dependerá de las coordenadas generalizadas del sólido al que pertenece, definidas por el vector $\mathbf{q}^i$, y del tiempo t:

$$\mathbf{r}_p^i = f(\mathbf{q}^i, t) \quad (55)$$

La diferencial total respecto al tiempo del vector $\mathbf{r}_p^i$ será:

$$\frac{d\mathbf{r}_p^i}{dt} = \frac{\partial \mathbf{r}_p^i}{\partial \mathbf{q}^i} \dot{\mathbf{q}}^i + \frac{\partial \mathbf{r}_p^i}{\partial t} \quad (56)$$

Multiplicando por un diferencial de tiempo dt a ambos lados de la igualdad de la expresión (56) :

$$d\mathbf{r}_p^i = \frac{\partial \mathbf{r}_p^i}{\partial \mathbf{q}} d\mathbf{q} + \frac{\partial \mathbf{r}_p^i}{\partial t} dt \quad (57)$$

Si se hace nulo el último término de la expresión (57) el desplazamiento infinitesimal real $d\mathbf{r}_p^i$ se convertirá en un desplazamiento virtual $\delta\mathbf{r}_p^i$, recuerde que un desplazamiento virtual implica que el desplazamiento ocurra sin ningún paso de tiempo. Nótese además que la terminología empleada para expresar un desplazamiento virtual $\delta\mathbf{r}_p^i$ difiere respecto a la usada para un desplazamiento infinitesimal real $d\mathbf{r}_p^i$. Por ello, se podrá definir el desplazamiento virtual del modo que sigue:

$$\delta\mathbf{r}_p^i = \frac{\partial \mathbf{r}_p^i}{\partial \mathbf{q}^i} \delta\mathbf{q}^i = \mathbf{r}_{q^i}^i \delta\mathbf{q}^i \quad (58)$$

Retomando las ecuaciones de posición, se puede expresar el desplazamiento virtual $\delta\mathbf{r}_p^i$ en función de las coordenadas generalizadas del sólido i del sistema multicuerpo:

$$\delta\mathbf{r}_p^i = \delta\mathbf{R}^i + \mathbf{A}_\theta^i \bar{\mathbf{u}}_p^i \delta\theta^i \quad (59)$$

Se puede reescribir la expresión (59) del siguiente modo:

$$\delta\mathbf{r}_p^i = \mathbf{r}_{q^i}^i \delta\mathbf{q}^i \quad (60)$$

Los términos que aparecen en la expresión (60) son el cambio virtual $\delta\mathbf{q}^i$ en las coordenadas generalizadas del sólido i y la derivada parcial del vector $\mathbf{r}_p^i$ respecto al vector de coordenadas generalizadas del sólido i $\mathbf{q}^i$:

$$\mathbf{q}^i = \begin{bmatrix} R_x^i \\ R_y^i \\ \theta^i \end{bmatrix} \quad (61)$$

$$\mathbf{r}_{\mathbf{q}^i}^i = \frac{\partial \mathbf{r}^i}{\partial \mathbf{q}^i} = [\mathbf{I} \ \mathbf{A}_{\theta}^i \bar{\mathbf{u}}_P^i] \quad (62)$$

El término $\mathbf{I}$ de la expresión (62) representa a una matriz identidad de orden 2.

4.5.2-Concepto de trabajo virtual

Una vez descrito el concepto de desplazamiento virtual se desarrollará el concepto de trabajo virtual, el cual es vital para poder entender el concepto de fuerza generalizada.

Suponga que sobre el sólido i del sistema multicuerpo actúa una fuerza neta, definida por el vector de fuerza $\mathbf{F}^i$, y un momento neto M^i debido únicamente a los momentos actuantes sobre el sólido i y no al momento que pueda ejercer la fuerza resultante externa $\mathbf{F}^i$.

Si se aplica un desplazamiento virtual $\delta \mathbf{r}_P^i$ a una partícula P cualquiera del sólido i , lo cual derivará en un cambio virtual en las coordenadas generalizadas del sólido i , el conjunto de fuerzas y momentos actuantes desarrollarán un trabajo virtual δW^i .

El desplazamiento virtual $\delta \mathbf{r}_P^i$ inducirá un cambio virtual en las coordenadas del origen del sistema de referencia del sólido, definido por $\delta \mathbf{R}^i$, y un cambio virtual en la orientación del sólido i del modelo multicuerpo, definido por $\delta \theta^i$.

El momento neto M^i que actúa sobre el sólido i del sistema multicuerpo realizará un trabajo virtual debido únicamente al cambio virtual de la orientación del sólido $\delta \theta^i$. Por otro lado, el vector de fuerza neta $\mathbf{F}^i$ que actúa sobre el sólido en cuestión, al poder estar aplicadas a una distancia finita respecto al origen del sistema de referencia del sólido i , desarrollará un trabajo virtual δW^i tanto por el desplazamiento virtual del sólido $\delta \mathbf{R}^i$ como por el cambio virtual en la orientación del mismo $\delta \theta^i$. El momento ejercido por el vector fuerza neta $\mathbf{F}^i$ será respecto al origen del sistema de referencia del sólido i del modelo multicuerpo.

Como consecuencia de todo lo descrito anteriormente, el trabajo virtual δW^i total desarrollado tanto por $\mathbf{F}^i$ como por M^i se formulará del siguiente modo:

$$\delta W^i = \mathbf{F}^{iT} \delta \mathbf{r}_P^i + M^i \delta \theta^i \quad (63)$$

Sustituyendo el término $\delta \mathbf{r}_P^i$ de la expresión (63) por la expresión (59):

$$\delta W^i = \mathbf{F}^{iT} (\delta \mathbf{R}^i + \mathbf{A}_{\theta}^i \bar{\mathbf{u}}_P^i \delta \theta^i) + M^i \delta \theta^i \quad (64)$$

Agrupando los términos del trabajo virtual δW^i debidos a un cambio virtual de las coordenadas del sistema de referencia del sólido i , denotado por el vector $\delta \mathbf{R}^i$, y los debidos a un cambio virtual de la orientación θ^i del sólido i , denotado por el escalar $\delta \theta^i$:

$$\delta W^i = \mathbf{F}^{iT} \delta \mathbf{R}^i + (\mathbf{F}^{iT} \mathbf{A}_\theta^i \bar{\mathbf{u}}_P^i + M^i) \delta \theta^i \quad (65)$$

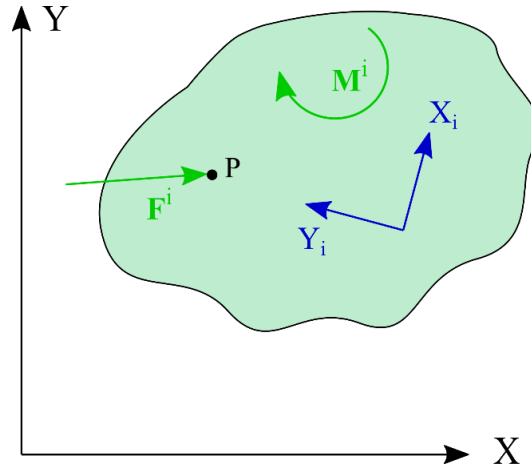


Figura 5. Trabajo virtual desarrollado por las fuerzas y momentos actuantes sobre el sólido i

4.5.3-Vector de fuerzas generalizadas

Una vez descritos y formulados los conceptos de desplazamiento virtual y de trabajo virtual, se podrá introducir el concepto de fuerza generalizada. El vector de fuerza resultante $\mathbf{F}^i$ que acompaña al cambio virtual de las coordenadas absolutas del origen del sistema de referencia del sólido i , definido por el vector $\delta \mathbf{R}^i$, será el vector de fuerza generalizada $\mathbf{Q}_R^i$ asociado al desplazamiento del sólido i del sistema multicuerpo:

$$\mathbf{Q}_R^i = \mathbf{F}^i \quad (66)$$

El escalar que acompaña al cambio virtual de la orientación del sólido $\delta \theta^i$ se trata de una fuerza generalizada asociada con la rotación del sólido i en el plano. Se expresará dicha fuerza generalizada del siguiente modo:

$$Q_\theta^i = M^i + \mathbf{F}^{iT} \mathbf{A}_\theta^i \bar{\mathbf{u}}_P^i \quad (67)$$

Es evidente que el segundo término del miembro derecho de la expresión (67) es el momento que ejerce el vector de fuerzas generalizadas $\mathbf{Q}_R^i$ respecto al origen del sistema de referencia del sólido i del modelo multicuerpo.

Dicho par puede ser igualmente expresado como el producto vectorial del vector de posición local del punto $\bar{\mathbf{u}}_P^i$ y del vector de fuerza resultante $\mathbf{F}^i$:

$$Q_\theta^i = M^i + (\bar{\mathbf{u}}_P^i \times \mathbf{F}^i) \cdot \mathbf{k} \quad (68)$$

Se agruparán el vector y el escalar de fuerzas generalizadas anteriores, que generalizan el conjunto de fuerzas y momentos actuantes sobre el sólido i del sistema multicuerpo, en un vector de fuerzas generalizadas externas $\mathbf{Q}^i$:

$$\mathbf{Q}^i = [\mathbf{Q}_R^i \ Q_\theta^i]^T \quad (69)$$

Será preciso generalizar todas las fuerzas y momentos, mediante vectores de fuerzas generalizadas $\mathbf{Q}^i$, que actúen en cada uno de los sólidos del sistema multicuerpo.

En un problema plano, como el abordado en el presente *TFG*, el proceso de generalización de fuerzas consiste únicamente en determinar la fuerza resultante sobre cada sólido del sistema multicuerpo, que determinará las dos primeras componentes del vector de fuerzas generalizadas externas $\mathbf{Q}^i$, y la suma del momento neto que actúa sobre cada sólido y el momento neto que ejercen las fuerzas respecto al origen del sistema de referencia, que determinará la tercera componente del vector de fuerzas generalizadas externas $\mathbf{Q}^i$.

4.6-Generalización de las fuerzas presentes en el modelo

4.6.1-Generalización de la gravedad

Suponga que se pretende generalizar la gravedad, mediante el vector de fuerzas generalizadas $\mathbf{Q}_g^i$, que actúa sobre el sólido i del sistema multicuerpo.

Tomando como origen del sistema de referencia del sólido i su centro de gravedad, la generalización de la gravedad se simplifica notablemente. Aplicando el principio de los trabajos virtuales se podrá determinar el trabajo virtual δW^i que desarrolla la gravedad:

$$\delta W^i = -m^i g \delta y^i \quad (70)$$

La gravedad desarrollará únicamente un trabajo virtual δW^i por el desplazamiento virtual vertical del centro de gravedad δy^i . Por tener la misma localización el centro de gravedad y el origen del sistema de referencia del sólido i , el desplazamiento vertical virtual del centro de gravedad δy^i será igual al cambio virtual sufrido por la coordenada generalizada R_y^i :

$$\delta y^i = \delta R_y^i \quad (71)$$

Por ende, la expresión que define el trabajo virtual δW^i desarrollado por la gravedad sobre el sólido i será:

$$\delta W^i = -m^i g \delta R_y^i \quad (72)$$

El vector de fuerzas generalizadas $\mathbf{Q}_g^i$ que generaliza a la gravedad que actúa sobre el sólido i del sistema multicuerpo de masa m^i tendrá una única componente no nula, asociada con el vector $\mathbf{Q}_R^i$:

$$\mathbf{Q}_g^i = [0 \quad -m^i g \quad 0]^T \quad (73)$$

4.6.2-Generalización de la fuerza en un muelle-amortiguador

El presente apartado del marco teórico generalizará las fuerzas que aparecerán en los muelles-amortiguadores del modelo y proporcionará las expresiones necesarias para determinar la longitud del muelle l y la velocidad a la que se deforma $\dot{l}$.

Debe de tener claro que los muelles-amortiguadores del sistema multicuerpo no son sólidos del modelo sino que son simplemente elementos de fuerza conectados en sus extremos a sólidos del sistema multicuerpo.

Las fuerzas propiciadas por los muelles-amortiguadores tendrán una componente elástica debida a la variación de longitud del muelle y una componente amortiguadora debida a la velocidad a la que se separan los puntos que definen sus extremos:

$$f_{ma} = k(l - l_0) + c\dot{l} \quad (74)$$

La componente elástica de la fuerza f_{ma} está definida por la rigidez del muelle k . Por otro lado, la componente amortiguadora de la fuerza f_{ma} está modelada por el coeficiente de amortiguación c . La longitud inicial del muelle se denotará por l_0 .

Se aplicará el principio de los trabajos virtuales para generalizar las fuerzas f_{ma} que aparecerán en los muelles-amortiguadores del sistema multicuerpo. El trabajo virtual δW desarrollado por la fuerza f_{ma} al inducir un desplazamiento virtual en la longitud del muelle δl será:

$$\delta W = -f_{ma} \delta l \quad (75)$$

Supóngase que se pretende analizar la fuerza que aparecerá en un determinado muelle-amortiguador del modelo multicuerpo que se encuentra conectado a los sólidos i y j del sistema multicuerpo. El muelle-amortiguador se encuentra conectado al sólido i en el punto P_i mientras que al sólido j en el punto P_j . Lógicamente, las partículas P_i y P_j pertenecen a los sólidos i y j del sistema multicuerpo.

La longitud del muelle l se podrá calcular a partir del vector que conecta los puntos P_i y P_j . Dicho vector se define de manera inmediata restando las posiciones absolutas de los puntos P_i y P_j , definidas por los vectores $\mathbf{r}_{P_i}^i$ y $\mathbf{r}_{P_j}^j$. Dependiendo del orden en el que se resten dichos

vectores, el vector que conecta los puntos P_i y P_j , definido por el vector de posición $\mathbf{r}_p^{ij}$, podrá apuntar al punto P_i o al punto P_j .

En definitiva, el vector $\mathbf{r}_p^{ij}$ define la longitud y orientación del muelle y se puede obtener haciendo uso de la expresión (5):

$$\mathbf{r}_p^{ij} = \mathbf{r}_{p_i}^i - \mathbf{r}_{p_j}^j = \mathbf{R}^i + \mathbf{A}^i \bar{\mathbf{u}}_{p_i}^i - \mathbf{R}^j + \mathbf{A}^j \bar{\mathbf{u}}_{p_j}^j \quad (76)$$

Finalmente, la longitud del muelle l se calculará determinado el módulo del vector $\mathbf{r}_p^{ij}$:

$$l = \left(\mathbf{r}_p^{ijT} \mathbf{r}_p^{ij} \right)^{1/2} \quad (77)$$

Teniendo en cuenta la longitud del muelle l depende únicamente de las coordenadas generalizadas de los sólidos i y j del modelo multicuerpo, definidas por los vectores $\mathbf{q}^i$ y $\mathbf{q}^j$, se puede definir el cambio virtual de la longitud del muelle δl del siguiente modo:

$$\delta l = \frac{\partial l}{\partial \mathbf{q}} \delta \mathbf{q} \quad (78)$$

El vector de coordenadas generalizadas $\mathbf{q}$ de la expresión (78) agrupa los vectores de coordenadas generalizadas $\mathbf{q}^i$ y $\mathbf{q}^j$:

$$\mathbf{q} = [\mathbf{q}^{iT} \mathbf{q}^{jT}]^T = [\mathbf{R}^{iT} \theta^i \mathbf{R}^{jT} \theta^j]^T \quad (79)$$

Tras operar convenientemente, se puede formular el desplazamiento virtual aplicado al muelle δl como sigue:

$$\delta l = \hat{\mathbf{I}}^T \left[\frac{\partial \mathbf{r}_p^{ij}}{\partial \mathbf{q}^i} \delta \mathbf{q}^i + \frac{\partial \mathbf{r}_p^{ij}}{\partial \mathbf{q}^j} \delta \mathbf{q}^j \right] \quad (80)$$

$$\delta l = \hat{\mathbf{I}}^T \left[\frac{\partial \mathbf{r}_p^{ij}}{\partial \mathbf{q}^i} \quad \frac{\partial \mathbf{r}_p^{ij}}{\partial \mathbf{q}^j} \right] [\delta \mathbf{q}^i \delta \mathbf{q}^j]^T \quad (81)$$

El término $\hat{\mathbf{I}}$ resulta de normalizar el vector $\mathbf{r}_p^{ij}$, e indica la dirección en la que actuará la fuerza f_{ma} . Teniendo en cuenta la expresión y dependencia del vector $\mathbf{r}_p^{ij}$, se puede derivar el vector $\mathbf{r}_p^{ij}$ respecto a los vectores $\mathbf{q}^i$ y $\mathbf{q}^j$ del siguiente modo:

$$\frac{\partial \mathbf{r}_p^{ij}}{\partial \mathbf{q}^i} = [\mathbf{I} \quad \mathbf{A}_\theta^i \bar{\mathbf{u}}_p^i] \quad (82)$$

$$\frac{\partial \mathbf{r}_p^{ij}}{\partial \mathbf{q}^j} = [\mathbf{I} \quad \mathbf{A}_\theta^j \bar{\mathbf{u}}_p^j] \quad (83)$$

El término $\mathbf{I}$ que aparece en las expresiones (82) y (83) es una matriz identidad de orden 2.

Tras todas las expresiones desarrolladas, el trabajo virtual δW desarrollado por la fuerza f_{ma} será:

$$\delta W = -f_{ma} \hat{\mathbf{I}}^T \left[\frac{\partial \mathbf{r}_{P^{ij}}}{\partial \mathbf{q}^i} \frac{\partial \mathbf{r}_{P^{ij}}}{\partial \mathbf{q}^j} \right] [\delta \mathbf{q}^i \delta \mathbf{q}^j]^T \quad (84)$$

Se aprecia que en la expresión (84) aparecen explícitamente los cambios virtuales en las coordenadas generalizadas de los sólidos i y j del sistema multicuerpo, definidos por $\delta \mathbf{q}^i$ y $\delta \mathbf{q}^j$. Por ende, se puede reescribir la expresión (84) con los vectores de fuerzas generalizadas $\mathbf{Q}^i$ y $\mathbf{Q}^j$:

$$\delta W = \mathbf{Q}^{iT} \delta \mathbf{q}^i + \mathbf{Q}^{jT} \delta \mathbf{q}^j \quad (85)$$

$$\mathbf{Q}^i = -f_{ma} \left[\bar{\mathbf{u}}_P^i{}^T \mathbf{A}_\theta^i{}^T \right] \hat{\mathbf{I}} \quad (86)$$

$$\mathbf{Q}^j = -f_{ma} \left[\bar{\mathbf{u}}_P^j{}^T \mathbf{A}_\theta^j{}^T \right] \hat{\mathbf{I}} \quad (87)$$

Queda solamente por definir el término $\dot{l}$. La velocidad con la que el muelle cambia de longitud se calculará diferenciando respecto al tiempo la longitud del muelle l , definida por la expresión (77):

$$\dot{l} = \frac{\partial l}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial t} = \frac{\partial l}{\partial \mathbf{q}} \dot{\mathbf{q}} \quad (88)$$

$$\dot{l} = \hat{\mathbf{I}}^T \frac{\partial \mathbf{r}_{P^{ij}}}{\partial \mathbf{q}} \dot{\mathbf{q}} \quad (89)$$

El vector $\dot{\mathbf{q}}$ que aparece en las expresiones (88) y (89) contiene los vectores de velocidades generalizadas de los sólidos i y j del sistema multicuerpo, definidos por $\dot{\mathbf{q}}^i$ y $\dot{\mathbf{q}}^j$:

$$\dot{\mathbf{q}} = [\dot{R}_x^i \dot{R}_y^i \dot{\theta}^i \dot{R}_x^j \dot{R}_y^j \dot{\theta}^j]^T \quad (90)$$

$$\dot{\mathbf{q}} = [\dot{\mathbf{R}}^i \dot{\theta}^i \dot{\mathbf{R}}^j \dot{\theta}^j]^T \quad (91)$$

$$\dot{\mathbf{q}} = [\dot{\mathbf{q}}^i \dot{\mathbf{q}}^j]^T \quad (92)$$

Tras toda la formulación presentada, queda claro que para generalizar la fuerza que aparecerá en un muelle-amortiguador del modelo será necesario disponer de los parámetros que modelan al muelle-amortiguador (k , c , l_0), conocer los puntos de conexión de los muelles-

amortiguadores a cada sólido del modelo multicuerpo (P_i y P_j) y los vectores de coordenadas y velocidades generalizadas de los sólidos que sustentan al muelle-amortiguador ($\mathbf{q}$ y $\dot{\mathbf{q}}$).

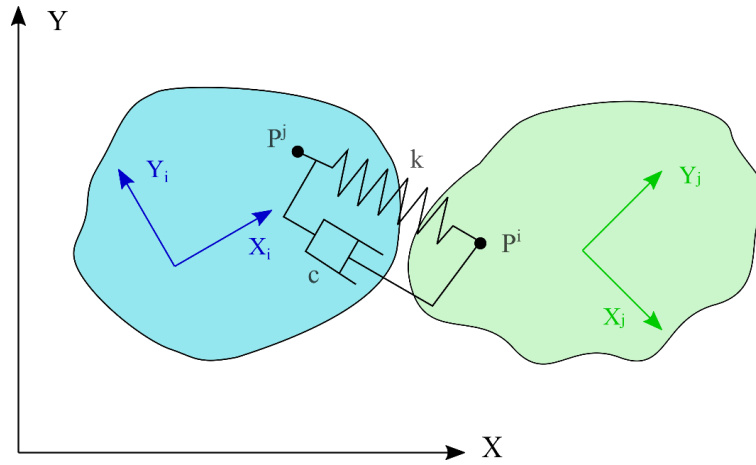


Figura 6. Sólidos rígidos i y j del sistema multicuerpo conectados por un muelle-amortiguador

4.6.3-Generalización de un par motor

Supóngase que se quiere aplicar un par o momento M^i en un sólido i del sistema multicuerpo. Dicho par se debe de generalizar mediante un vector de fuerzas generalizadas externas $\mathbf{Q}^i$. El trabajo virtual δW^i desarrollado por el momento M^i se define mediante la siguiente expresión:

$$\delta W^i = M^i \delta \theta^i \quad (93)$$

El par M^i solamente desarrollará trabajo virtual por un cambio virtual de la orientación del sólido i del modelo multicuerpo, definido por $\delta \theta^i$. Por consiguiente, los términos asociados al vector $\mathbf{Q}_R^i$ serán nulos. Por otro lado, el escalar asociado al término Q_θ^i será igual al momento M^i introducido:

$$Q_\theta^i = M^i \quad (94)$$

Finalmente, el vector de fuerzas generalizadas $\mathbf{Q}^i$ que generaliza al par o momento M^i aplicado sobre el sólido i del sistema multicuerpo se define como sigue:

$$\mathbf{Q}^i = [0 \quad 0 \quad M^i]^T \quad (95)$$

Un modo de activar el movimiento longitudinal del vehículo y del remolque será incluir un par motor M^i a alguna rueda del vehículo (el vehículo tendrá tracción en un solo eje) o a ambas ruedas del vehículo (el vehículo tendría tracción total).

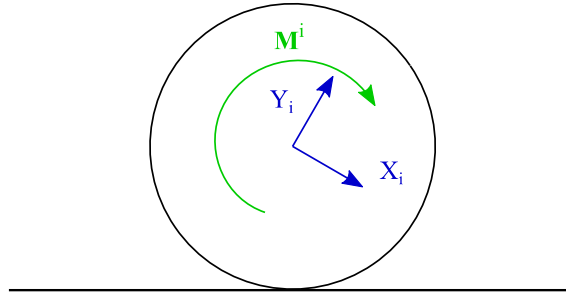


Figura 7. Par motor introducido sobre el sólido i del sistema multicuerpo

4.6.4-Generalización de las fuerzas de contacto

Se pretende ahora generalizar, mediante un vector de fuerzas generalizadas externas $\mathbf{Q}^i$, las fuerzas normales y tangenciales resultantes del contacto elástico rueda-terreno. Suponga que el sólido i del sistema multicuerpo será el que interactuará con el terreno, modelado por el sólido del modelo que representa el sistema de referencia absoluto.

La fuerza normal de contacto actuará en todo momento en la dirección del centro de gravedad de la rueda, el cual coincide con el origen del sistema de referencia del sólido que modele a la rueda. Por ello, la fuerza normal de contacto no desarrollará trabajo virtual δW^i por un cambio virtual en la orientación del sólido $\delta \theta^i$. Sin embargo, la fuerza normal de contacto desarrollará un trabajo virtual δW^i debido al cambio virtual de las dos primeras coordenadas generalizadas del sólido en cuestión, definido por el vector $\delta \mathbf{R}^i$.

Como consecuencia, el vector de fuerzas generalizadas $\mathbf{Q}_{\text{Fnc}}^i$ que generaliza a la fuerza normal de contacto, definida por el vector de fuerza $\mathbf{F}_{\text{nc}}^i$, se define por la expresión que sigue:

$$\mathbf{Q}_{\text{Fnc}}^i = [F_{\text{ncx}}^i \quad F_{\text{n cy}}^i \quad 0]^T \quad (96)$$

Los términos F_{ncx}^i y $F_{\text{n cy}}^i$ son las componentes cartesianas horizontal y vertical del vector de fuerza normal $\mathbf{F}_{\text{nc}}^i$. Para calcular el vector de fuerza $\mathbf{F}_{\text{nc}}^i$ se empleará una simplificación de la teoría de contacto de *Hertz*, desarrollada en el apartado 4.9.3.

Con respecto a la fuerza tangencial de contacto, definida por el vector de fuerza $\mathbf{F}_{\text{tc}}^i$, ésta desarrollará un trabajo virtual δW^i debido a un cambio virtual de todas las coordenadas generalizadas del sólido i , definido por el vector $\delta \mathbf{R}^i$ y por el escalar $\delta \theta^i$.

Por ende, el vector de fuerzas generalizadas $\mathbf{Q}_{\text{Ftc}}^i$ que generaliza la fuerza tangencial de contacto $\mathbf{F}_{\text{tc}}^i$ que actúa sobre el sólido i del sistema multicuerpo se define mediante la siguiente expresión:

$$\mathbf{Q}_{\text{Ftc}}^i = [F_{\text{tcx}}^i \quad F_{\text{tcy}}^i \quad (\mathbf{u}_p^i \times \mathbf{F}_{\text{tc}}^i) \mathbf{k}]^T \quad (97)$$

Los términos F_{tcx}^i y F_{tcy}^i son las componentes longitudinal y transversal de la fuerza tangencial de contacto $\mathbf{F}_{tc}^i$. El vector de posición local $\mathbf{u}_p^i$ define la posición local del punto P donde contactan la rueda y el terreno.

Para determinar el vector de fuerza $\mathbf{F}_{tc}^i$ se hará uso de un modelo lineal con saturación de *Coulomb*, mostrado y desarrollado en el apartado 4.9.4.

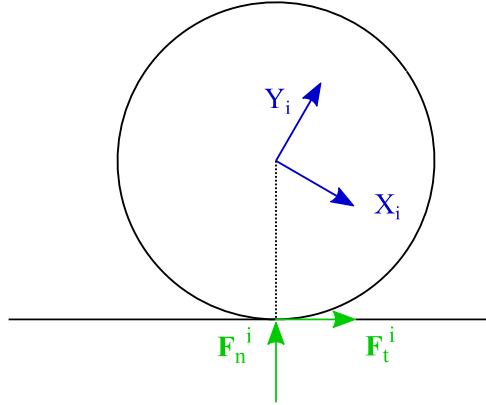


Figura 8. Fuerzas normal y tangencial de contacto actuantes sobre el sólido i del sistema multicuerpo

4.7-Ecuaciones dinámicas empleadas en el modelo

Será necesario presentar un conjunto de ecuaciones dinámicas que permitan obtener el vector $\ddot{\mathbf{q}}$ para integrar numéricamente y obtener la configuración del sistema multicuerpo, definida por el vector $\mathbf{q}$, en un instante posterior de tiempo. Teniendo el vector $\mathbf{q}$ en ciertos instantes de un intervalo temporal se podrá simular el comportamiento del sistema multicuerpo.

El vector $\ddot{\mathbf{q}}$ no será la única incógnita que determinará las ecuaciones del movimiento del modelo. Las restricciones del sistema multicuerpo, definidas por el vector $\mathbf{C}$, propician la aparición de una serie de fuerzas o momentos de reacción, que se deberán de generalizar mediante el vector de fuerzas generalizadas $\mathbf{Q}_{FR}$, que promueven el cumplimiento de las restricciones.

El objeto del presente apartado del marco teórico del trabajo será proporcionar un conjunto de ecuaciones que permitan calcular los vectores $\ddot{\mathbf{q}}$ y $\mathbf{Q}_{FR}$.

4.7.1-Formulación de las ecuaciones de *Newton* y la ecuación de *Euler*

Las ecuaciones de *Newton*, resultado de aplicar un balance de fuerzas en un sólido i del sistema multicuerpo, nos permitirán determinar el vector $\ddot{\mathbf{R}}^i$ en cada sólido del sistema multicuerpo, recuerde que dicho vector definía las dos primeras componentes del vector $\mathbf{q}^i$. Aplicando las ecuaciones de *Newton* a un sólido i del sistema multicuerpo:

$$\mathbf{F}^i = \int_{V^i} \rho^i \ddot{\mathbf{r}}^i dV^i \quad (98)$$

Si el origen del sistema de referencia del sólido i del sistema multicuerpo de masa m^i coincide con su centro de gravedad, se pueden expresar las ecuaciones de *Newton* del siguiente modo:

$$\mathbf{F}^i = m^i \ddot{\mathbf{R}}^i \quad (99)$$

$$F_x^i = m^i \ddot{R}_x^i \quad (100)$$

$$F_y^i = m^i \ddot{R}_y^i \quad (101)$$

Aplicando las ecuaciones de *Newton* a cada sólido del modelo multicuerpo obtendremos un total de $2 \times n_s$ ecuaciones. Si se conociesen los vectores de fuerza resultante $\mathbf{F}^i$ que actúan sobre cada sólido del sistema multicuerpo, las ecuaciones de *Newton* permitirían obtener el vector $\ddot{\mathbf{R}}^i$ de cada sólido del modelo multicuerpo. Las ecuaciones de *Newton* únicamente permitirán obtener el vector $\ddot{\mathbf{R}}^i$ en cada sólido del sistema multicuerpo si ningún sólido del modelo estuviera sujeto a alguna restricción.

Aplicando la ecuación de *Euler* a un sólido i cualquiera del sistema multicuerpo, resultado de aplicar un balance de momentos:

$$\mathbf{M}_R^i = \int_{V^i} \rho^i \mathbf{u}^i \times \ddot{\mathbf{r}}^i dV^i \quad (102)$$

Si el origen del sistema de referencia del sólido i del modelo multicuerpo, con un momento de inercia de J^i , coincide con su centro de gravedad, la ecuación (102) se puede reescribir del siguiente modo:

$$J^i \ddot{\theta}^i = M^i \quad (103)$$

Aplicando la ecuación de *Euler* a cada sólido del sistema multicuerpo, en caso de que ninguno estuviera sujeto a algún tipo de restricción, se obtendría la aceleración angular $\ddot{\theta}^i$ de cada sólido del sistema multicuerpo.

Se puede expresar de manera matricial las expresiones que resultan de aplicar las ecuaciones de *Newton* y la ecuación de *Euler* a cada sólido del sistema multicuerpo:

$$\mathbf{M} \ddot{\mathbf{q}} = \mathbf{Q} \quad (104)$$

Los términos de la expresión (104) son la matriz de masa del sistema multicuerpo $\mathbf{M}$, el vector de aceleraciones generalizadas del sistema multicuerpo $\ddot{\mathbf{q}}$ y el vector de fuerzas

generalizadas $\mathbf{Q}$, que incluye las fuerzas gravitatorias, las fuerzas de contacto, las fuerzas de los muelles-amortiguadores, las fuerzas de reacción y las fuerzas de usuario aplicadas.

La matriz de masa $\mathbf{M}$ de un sistema multicuerpo compuesto por ns sólidos, de masa m^i y momento de inercia J^i cada uno, toma el siguiente aspecto:

$$\mathbf{M} = \begin{bmatrix} m^1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & m^1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & J^1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & m^2 & 0 & 0 & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & m^2 & 0 & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & J^2 & \dots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & m^{ns} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & m^{ns} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & J^{ns} \end{bmatrix} \quad (105)$$

El vector de fuerzas generalizadas $\mathbf{Q}$ se puede descomponer en el vector fuerzas generalizadas externas $\mathbf{Q}_e$ y el vector de fuerzas generalizadas de reacción $\mathbf{Q}_{FR}$. Por ello, la expresión (104) se puede reescribir del siguiente modo:

$$\mathbf{M}\ddot{\mathbf{q}} = \mathbf{Q}_e + \mathbf{Q}_{FR} \quad (106)$$

Si ningún sólido del sistema multicuerpo tuviera alguna restricción, aplicando la expresión (104) se podrá determinar de manera inmediata el vector $\ddot{\mathbf{q}}$.

4.7.2-Ecuaciones adicionales para poder resolver las ecuaciones de movimiento del modelo multicuerpo

Para un sistema multicuerpo con ciertas restricciones, definidas por $\mathbf{C}$, no se puede hacer uso de la expresión (106) para obtener $\ddot{\mathbf{q}}$ y $\mathbf{Q}_{FR}$ ya que el sistema de ecuaciones lineales resultante dispone de más incógnitas que ecuaciones. Por ende, será necesario buscar unas ecuaciones adicionales para poder emplear la expresión (106), reordenada y agrupando en un vector de incógnitas $\mathbf{x}$ los vectores $\ddot{\mathbf{q}}$ y $\mathbf{Q}_{FR}$.

Afortunadamente, los vectores $\mathbf{Q}_{FR}$ y $\mathbf{C}$ tienen el mismo número de componentes. Por consiguiente, las ecuaciones adicionales se obtendrán a partir de las restricciones del modelo multicuerpo.

Concretamente, las ecuaciones adicionales se obtendrán derivando dos veces respecto al tiempo el vector de restricciones del sistema multicuerpo $\mathbf{C}$. Teniendo en cuenta que el vector de restricciones del sistema multicuerpo $\mathbf{C}$ depende del tiempo t y del vector de coordenadas generalizadas del sistema multicuerpo $\mathbf{q}$:

$$\frac{d}{dt} \mathbf{C}(\mathbf{q}, t) = \mathbf{0} \quad (107)$$

$$\mathbf{C}_q \dot{\mathbf{q}} + \mathbf{C}_t = \mathbf{0} \quad (108)$$

Derivando de nuevo respecto al tiempo la expresión (108):

$$\frac{d}{dt} (\mathbf{C}_q \dot{\mathbf{q}} + \mathbf{C}_t) = \mathbf{0} \quad (109)$$

$$\mathbf{C}_q \ddot{\mathbf{q}} + (\mathbf{C}_q \dot{\mathbf{q}})_q \dot{\mathbf{q}} + 2\mathbf{C}_{qt} \dot{\mathbf{q}} + \mathbf{C}_{tt} = \mathbf{0} \quad (110)$$

$$\mathbf{C}_q \ddot{\mathbf{q}} = -(\mathbf{C}_q \dot{\mathbf{q}})_q \dot{\mathbf{q}} - 2\mathbf{C}_{qt} \dot{\mathbf{q}} - \mathbf{C}_{tt} \quad (111)$$

Se agruparán todos los términos del miembro derecho de la expresión (111) en un único término, denotado por el vector $\mathbf{Q}_d$:

$$\mathbf{Q}_d = -(\mathbf{C}_q \dot{\mathbf{q}})_q \dot{\mathbf{q}} - 2\mathbf{C}_{qt} \dot{\mathbf{q}} - \mathbf{C}_{tt} \quad (112)$$

Finalmente, sustituyendo $\mathbf{Q}_d$ en la expresión (111):

$$\mathbf{C}_q \ddot{\mathbf{q}} = \mathbf{Q}_d \quad (113)$$

4.7.3-Multiplicadores de Lagrange

Antes de presentar aspecto final que tomarán las ecuaciones dinámicas empleadas en el modelo se introducirá el concepto de multiplicadores de *Lagrange*, definidos por el vector $\boldsymbol{\lambda}$.

La introducción de los multiplicadores de *Lagrange* en las ecuaciones del movimiento del sistema multicuerpo provoca que el vector $\mathbf{Q}_{FR}$ no aparezca de manera explícita en las ecuaciones dinámicas. Tras introducir los multiplicadores de *Lagrange*, la matriz de coeficientes $\mathbf{A}$ del sistema de ecuaciones lineales será simétrica y el vector de incógnitas $\mathbf{x}$ de las ecuaciones dinámicas estará formado por los vectores $\ddot{\mathbf{q}}$ y $\boldsymbol{\lambda}$:

$$\begin{bmatrix} \mathbf{M} & \mathbf{C}_q^T \\ \mathbf{C}_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_e \\ \mathbf{Q}_d \end{bmatrix} \quad (114)$$

Como consecuencia de acoplar las expresiones (106) y (113) e introducir el vector de multiplicadores de *Lagrange* $\boldsymbol{\lambda}$ las ecuaciones del movimiento del sistema multicuerpo conformarán un sistema de ecuaciones lineales que puede ser resuelto mediante el método de inversión de matrices.

La matriz de coeficientes $\mathbf{A}$ está definida por la matriz Jacobiana de las restricciones $\mathbf{C}_q$ y por la matriz de masa del sistema multicuerpo $\mathbf{M}$:

$$\mathbf{A} = \begin{bmatrix} \mathbf{M} & \mathbf{C}_q^T \\ \mathbf{C}_q & \mathbf{0} \end{bmatrix} \quad (115)$$

El vector de términos independientes $\mathbf{b}$ está definido por el vector de fuerzas generalizadas externas $\mathbf{Q}_e$ y por el término $\mathbf{Q}_d$, resultado de derivar dos veces respecto al tiempo el vector de restricciones $\mathbf{C}$:

$$\mathbf{b} = \begin{bmatrix} \mathbf{Q}_e \\ \mathbf{Q}_d \end{bmatrix} \quad (116)$$

El vector de incógnitas está definido por el vector de multiplicadores de *Lagrange* λ y por el vector de aceleraciones generalizadas del sistema multicuerpo $\ddot{\mathbf{q}}$:

$$\mathbf{x} = \begin{bmatrix} \ddot{\mathbf{q}} \\ \lambda \end{bmatrix} \quad (117)$$

El sistema de ecuaciones lineales de la expresión (114) se resolverá mediante el método de inversión de matrices:

$$\mathbf{Ax} = \mathbf{b} \quad (118)$$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \quad (119)$$

El vector de multiplicadores de *Lagrange* λ se relaciona de manera directa con el vector de fuerzas generalizadas de reacción $\mathbf{Q}_{FR}$ mediante la siguiente expresión:

$$\mathbf{Q}_{FR} = -\mathbf{C}_q^T \lambda \quad (120)$$

4.7.4-Método de estabilización de las restricciones de Baumgarte

Note que para acoplar las aceleraciones generalizadas del sistema multicuerpo, formuladas a partir de las ecuaciones de *Newton* y de la ecuación de *Euler*, y las restricciones del sistema multicuerpo, definidas por el vector de restricciones $\mathbf{C}$, y definir las ecuaciones del movimiento del sistema multicuerpo, definidas por la expresión (114), ha sido necesario derivar dos veces respecto al tiempo el vector $\mathbf{C}$. Como consecuencia, las ecuaciones de la expresión (114) solamente imponen el cumplimiento la segunda derivada respecto al tiempo de las restricciones y no las propias restricciones del sistema multicuerpo.

En cualquier simulación numérica asegurar únicamente el cumplimiento de la segunda derivada respecto al tiempo del vector $\mathbf{C}$ conlleva errores de tipo numérico. Este error, como consecuencia de no estar cumpliendo las restricciones debidamente, se irá acumulando con

el paso de cualquier simulación numérica dando lugar al fenómeno conocido como *drift*. El término *drift* simplemente hace referencia al fenómeno de acumulación de error numérico como consecuencia de no imponer que se cumpla el vector $\mathbf{C}$ en las ecuaciones dinámicas.

Para solventar la problemática del fenómeno *drift* se modificarán ligeramente las ecuaciones del movimiento del sistema multicuerpo mediante el método de estabilización de las restricciones de *Baumgarte*. Dicho método incluye los coeficientes de *Baumgarte* α y β :

$$\begin{bmatrix} \mathbf{M} & \mathbf{C}_q^T \\ \mathbf{C}_q & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \dot{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_e \\ \mathbf{Q}_d - 2\alpha(\mathbf{C}_q \dot{\mathbf{q}} + \mathbf{C}_t) - (\beta)^2 \mathbf{C} \end{bmatrix} \quad (121)$$

Se observa claramente que en la expresión (121), a diferencia de en la expresión (114), aparece el vector de restricciones del sistema multicuerpo $\mathbf{C}$. De este modo, se asegura que se cumplan las restricciones del modelo.

Sin embargo, una elección errónea de los valores de los coeficientes de *Baumgarte* puede conducir a que se obtengan soluciones numéricas erróneas. No se conoce con exactitud qué valores asignar a los coeficientes de *Baumgarte* pues dependen del modelo multicuerpo a simular y de la rigidez equivalente del mismo. Solamente se tiene conocimiento, desde un punto de vista matemático, que el coeficiente α debe de ser mayor que cero y que el coeficiente β debe de ser distinto de cero.

4.7.5-Augmented formulation y embedding technique

En la teoría de sistemas multicuerpo existen dos tipos de formulaciones dinámicas:

- La *augmented formulation* o formulación aumentada.
- La *embeddig technique*.

La diferencia básica entre ambas formulaciones es la partición de las coordenadas generalizadas del sistema multicuerpo en coordenadas independientes y coordenadas dependientes.

La formulación empleada para el desarrollo del modelo ha sido la *augmented formulation*. La *augmented formulation* no hace distinción alguna entre coordenadas generalizadas independientes o dependientes. De este modo, se formularán un mayor número de ecuaciones dinámicas obteniendo sistemas de ecuaciones lineales de mayores dimensiones. Sin embargo, dicha formulación es más general de formular y plantear.

La *embeddig technique* resuelve únicamente, por medio de las ecuaciones del movimiento del sistema multicuerpo, un determinado conjunto de coordenadas generalizadas independientes. De este modo, el número de ecuaciones dinámicas se reducirá y se evitará el fenómeno del *drift* asociado a las restricciones, pero la formulación perderá generalidad.

4.8-Proceso de cálculo en las simulaciones dinámicas

4.8.1-Vectores de coordenadas y velocidades generalizadas iniciales

El objeto de cualquier simulación dinámica realizada con el sistema multicuerpo modelado y que abarque un determinado intervalo de tiempo será determinar en ciertos instantes, por tratarse de una solución numérica o discreta, el vector $\mathbf{q}$.

El vector $\mathbf{q}$ en un cierto instante t se obtendrá integrando numéricamente el vector $\ddot{\mathbf{q}}$ en un instante $t - \Delta t$, siendo Δt el paso de integración numérica.

Para obtener en un cierto instante de la simulación dinámica el vector de aceleraciones generalizadas del sistema multicuerpo $\ddot{\mathbf{q}}$ será necesario calcular la matriz de coeficientes $\mathbf{A}$ y el vector de términos independientes $\mathbf{b}$, definidos por las expresiones (115) y (116). Para poder determinar los términos de la matriz $\mathbf{A}$ y del vector $\mathbf{b}$ bastará con disponer de los vectores de coordenadas y velocidades generalizadas del sistema multicuerpo $\mathbf{q}$ y $\dot{\mathbf{q}}$.

Como consecuencia de todo lo descrito, se puede expresar el vector $\ddot{\mathbf{q}}$ como una función dependiente del instante temporal de la simulación t y de los vectores $\mathbf{q}$ y $\dot{\mathbf{q}}$:

$$\ddot{\mathbf{q}} = \mathbf{G}(\mathbf{q}, \dot{\mathbf{q}}, t) \quad (122)$$

Tras todo lo descrito, queda de manifiesto la necesidad de definir unos vectores de coordenadas y velocidades generalizadas iniciales del sistema multicuerpo $\mathbf{q}_0$ y $\dot{\mathbf{q}}_0$. Los vectores $\mathbf{q}_0$ y $\dot{\mathbf{q}}_0$ establecerán la configuración inicial del sistema multicuerpo, a nivel de posición y velocidad, y permitirán poder calcular en el instante t_0 el vector $\ddot{\mathbf{q}}$.

Integrando numéricamente el vector de aceleraciones generalizadas del modelo multicuerpo $\ddot{\mathbf{q}}$ se podrá calcular los vectores $\mathbf{q}$ y $\dot{\mathbf{q}}$ en un instante posterior $t + \Delta t$. De nuevo, conociendo los vectores $\mathbf{q}$ y $\dot{\mathbf{q}}$ se podrán resolver las ecuaciones de movimiento del sistema multicuerpo para obtener el vector de aceleraciones generalizadas del sistema multicuerpo $\ddot{\mathbf{q}}$.

Si no se dispone de una configuración inicial del sistema multicuerpo a modelar, definida por los vectores $\mathbf{q}_0$ y $\dot{\mathbf{q}}_0$, no se podrá simular el comportamiento del modelo multicuerpo.

4.8.2-Concepto de vector de estado

Para obtener el vector de coordenadas generalizadas del sistema multicuerpo $\mathbf{q}$ a partir del vector de aceleraciones generalizadas del sistema multicuerpo $\ddot{\mathbf{q}}$ se deberán de resolver un conjunto de ecuaciones diferenciales algebraicas ordinarias de orden dos.

El objeto de introducir el vector de estado $\mathbf{y}$ es resolver un conjunto de ecuaciones diferenciales ordinarias algebraicas de orden uno. Para conseguir dicho propósito el vector de estado $\mathbf{y}$ deberá de aunar a los vectores $\mathbf{q}$ y $\dot{\mathbf{q}}$:

$$\mathbf{y} = \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \quad (123)$$

Por ende, integrando numéricamente una única vez la derivada respecto al tiempo del vector de estado $\dot{\mathbf{y}}$ obtendremos el vector $\mathbf{q}$ en un instante posterior. Por consiguiente, la derivada respecto al tiempo del vector de estado $\dot{\mathbf{y}}$ deberá de agrupar los vectores $\dot{\mathbf{q}}$ y $\ddot{\mathbf{q}}$:

$$\dot{\mathbf{y}} = \begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} \quad (124)$$

A costa de transformar las ecuaciones diferenciales ordinarias de orden dos en ecuaciones diferenciales ordinarias de orden uno se ha duplicado el número de ecuaciones diferenciales a resolver, el vector $\dot{\mathbf{y}}$ tiene el doble de componentes que el vector $\ddot{\mathbf{q}}$.

Examinando la dependencia del vector $\dot{\mathbf{y}}$ podemos deducir que se pueden definir cada componente del mismo a partir de los vectores $\mathbf{q}$ y $\dot{\mathbf{q}}$. A su vez, recordando que el vector $\mathbf{y}$ contiene a los vectores $\mathbf{q}$ y $\dot{\mathbf{q}}$, se llega a la conclusión de que solamente se precisa el vector de estado $\mathbf{y}$ y el instante temporal de la simulación numérica t para poder obtener el vector $\dot{\mathbf{y}}$ en el instante t :

$$\dot{\mathbf{y}} = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{G}(\mathbf{q}, \dot{\mathbf{q}}, t) \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{q}} \\ \mathbf{G}(\mathbf{y}, t) \end{bmatrix} \quad (125)$$

$$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, t) \quad (126)$$

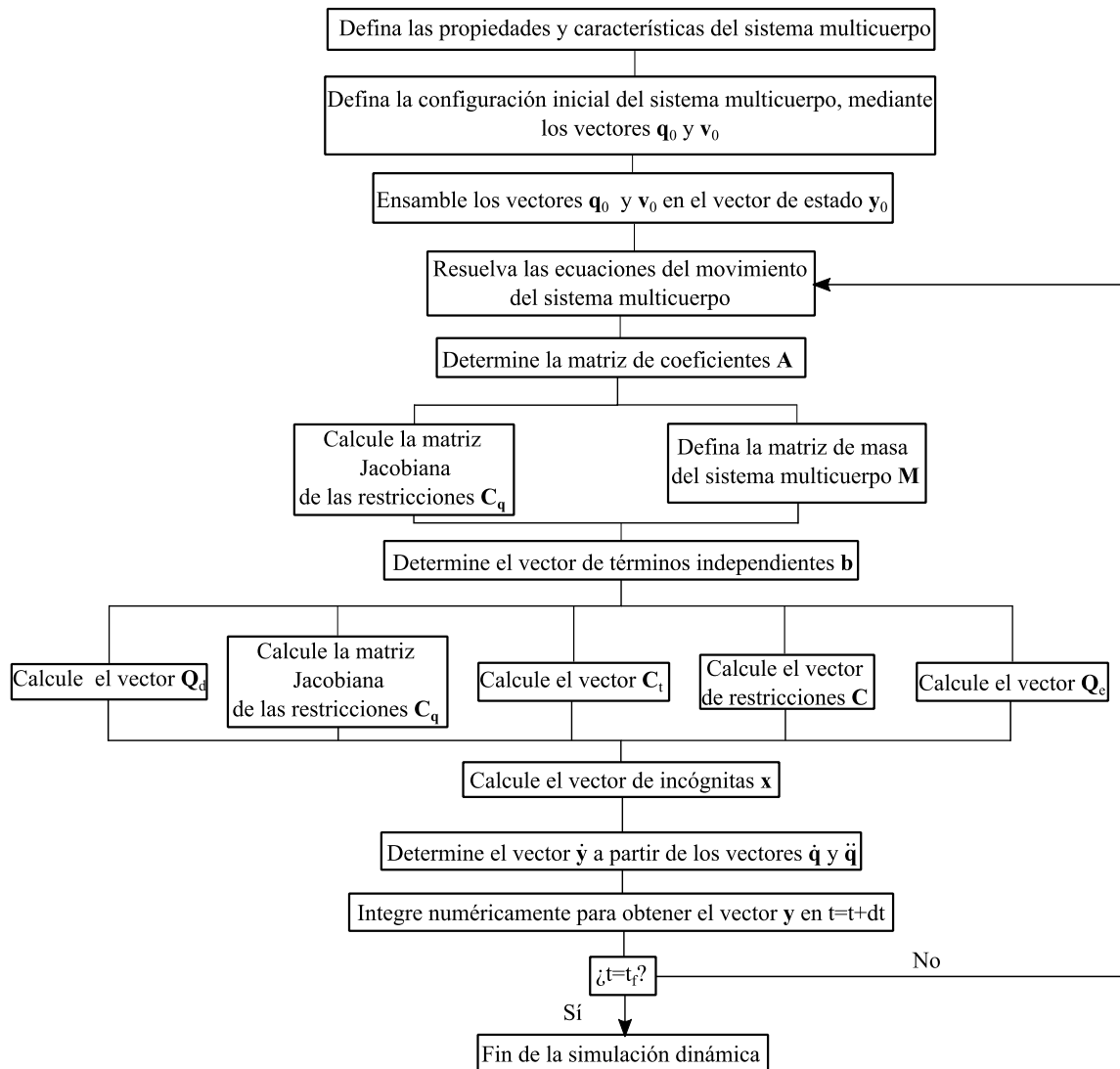


Figura 9. Metodología de cálculo en una simulación dinámica

4.9-Contacto elástico rueda-terreno

El presente apartado describirá cómo se ha modelado el contacto elástico rueda-terreno. La formulación del contacto elástico definirá el valor de la fuerza normal de contacto, modelada con la teoría de contacto de *Hertz* y clave en la dinámica vertical del modelo, y de la fuerza tangencial de contacto, clave en la dinámica longitudinal del modelo y será la que propicie la aparición de una fuerza de impulsión en las ruedas del vehículo y del remolque. Una vez determinadas las magnitudes de las fuerzas normales y tangenciales de contacto se deberán de generalizar, tal y como se detalló en el apartado 4.6.4.

4.9.1-Parametrización del terreno y las ruedas

En primera instancia es necesario parametrizar las curvas que definen a la rueda y al terreno. La parametrización de una curva consistirá en la definición de una función vectorial $\mathbf{f}$,

dependiente de un único parámetro s , que definirá la posición de cualquier punto P de la curva mediante un vector de posición. Se denotarán a las coordenadas s que parametrizan a las curvas del terreno y de las ruedas como coordenadas curvilíneas o de superficie.

Derivando la función $\mathbf{f}$ respecto a la coordenada curvilínea s se definirá la función vectorial $\mathbf{g}$ que definirá el vector tangente en cualquier punto o partícula P de la curva parametrizada.

Derivando la función $\mathbf{g}$ respecto a la coordenada curvilínea s se obtiene la función vectorial $\mathbf{h}$ que define el vector normal en cualquier punto P de la curva parametrizada.

Los vectores de posición, los vectores tangentes y los vectores normales a las curvas parametrizadas serán clave en la formulación del contacto elástico rueda-terreno.

Parametrización de las ruedas

Las ruedas del vehículo serán modeladas como simples circunferencias. Por ello, se deberá de parametrizar una circunferencia para parametrizar a cada rueda del modelo multicuerpo.

La coordenada curvilínea s que parametrizará a cada rueda del modelo será el ángulo que forme cualquier punto P de la misma con el sistema de referencia cartesiano asociado a la rueda.

Por consiguiente, la función $\mathbf{f}$ que define el vector de posición local de cualquier partícula P de una rueda de radio R del modelo multicuerpo tomará la siguiente forma:

$$\mathbf{f}_i(s) = [R \cos(s) \ R \sin(s)]^T \quad (127)$$

Se ha incluido un subíndice a la función $\mathbf{f}$ para dejar constancia a qué rueda del sistema multicuerpo parametriza la función $\mathbf{f}$. El vector de posición que resulta de aplicar la expresión (127) estará expresado en el sistema de referencia de la rueda, que podrá tener una orientación distinta a la que tenga el sistema de referencia global. Por ende, para expresar dicho vector de manera global será necesario multiplicar la matriz de rotación de la rueda $\mathbf{A}^i$ por el vector de posición resultante de emplear la función $\mathbf{f}_i$.

Derivando la función $\mathbf{f}_i$ respecto a la coordenada curvilínea s se obtiene la función $\mathbf{g}_i$ que define el vector tangente a la circunferencia, que modela a una rueda, en cualquier punto P de la misma:

$$\mathbf{f}'_i = \mathbf{g}_i(s) = [-R \sin(s) \ R \cos(s)]^T \quad (128)$$

Si se deriva la función $\mathbf{g}_i$ respecto a la coordenada curvilínea s que parametriza a la rueda, se definirá la función $\mathbf{h}_i$ que proporciona el vector normal en cualquier punto P de la rueda:

$$\mathbf{f}''_i = \mathbf{h}_i(s) = [-R \cos(s) \ -R \sin(s)]^T \quad (129)$$

El vector tangente, definido por la función $\mathbf{f}'_i$, y el vector normal, definido por $\mathbf{f}''_i$, en cualquier punto P de una rueda del modelo multicuerpo serán perpendiculares. Además, note que existen dos posibles orientaciones para definir el vector normal en cualquier punto P de la rueda.

El vector normal en un punto P se puede determinar de manera inmediata conociendo el vector tangente en dicho punto. Si se cambian de orden las componentes del vector tangente en el punto P y se cambia el signo de una de sus componentes, se obtendrá el vector normal a la rueda en el punto P. Dependiendo de qué componente del vector tangente en P se cambie de signo el vector normal en el punto P apuntará hacia la parte cóncava o convexa de la rueda.

Para formular el contacto elástico se ha optado por escoger como criterio que cualquier vector normal definido deberá de apuntar hacia el interior de la curva parametrizada.

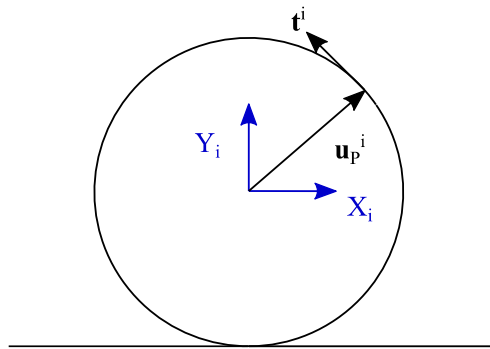


Figura 10. Vectores posición y tangente en un punto P cualquiera de una rueda del modelo multicuerpo

Parametrización del terreno

La parametrización del terreno depende directamente de la función matemática y que defina su altura en cualquier punto P del mismo. La función escalar y definirá la altura y geometría del terreno.

La coordenada de superficie s que parametrizará al terreno será la distancia horizontal existente entre cualquier punto P del terreno y el origen del sistema de referencia global. Recuerde que el sistema de referencia global modelará al terreno.

La función $\mathbf{f}$ que definirá la posición local de cualquier punto P del terreno será:

$$\mathbf{f}(s) = [s \quad y(s)]^T \quad (130)$$

Derivando la función $\mathbf{f}$ respecto a la coordenada curvilínea s se tendrá la función $\mathbf{g}$ que define el vector tangente en cualquier punto P del terreno:

$$\mathbf{f}' = \mathbf{g}(s) = \left[1 \quad \frac{dy}{ds} \right]^T \quad (131)$$

Finalmente, intercambiando las componentes de la función $\mathbf{g}$ y cambiando el signo de la segunda componente se define la función vectorial $\mathbf{h}$:

$$\mathbf{f}''^i = \mathbf{h}(s) = \left[\frac{dy}{ds} \quad -1 \right]^T \quad (132)$$

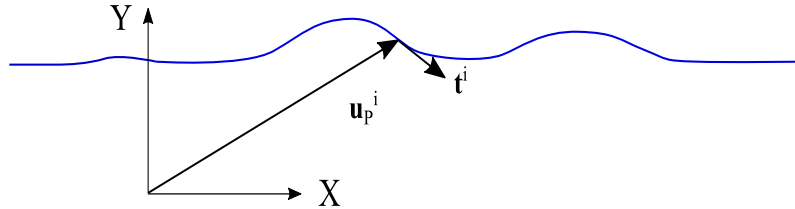


Figura 11. Vectores posición y tangente en un punto P cualquiera del terreno

4.9.2-Formulación del contacto elástico. Búsqueda de puntos de contacto

El modelo de contacto formulado para este TFG permite que las ruedas del modelo multicuerpo y el terreno interactúen en la interfaz de contacto pudiéndose deformar localmente mediante indentación o penetración. Por consiguiente, se trata de un contacto de tipo elástico. Recuerde que, por otro lado, globalmente todos los sólidos del sistema multicuerpo se considerarán perfectamente rígidos, no admitirán deformación alguna.

El modelo necesitará, a partir de la configuración del sistema multicuerpo definida por el vector $\mathbf{y}$, determinar el punto donde el terreno y la rueda han contactado. El contacto elástico ocurrirá en un único punto en cada sólido.

Se deberán de definir las condiciones que satisfacen los puntos que contactan físicamente, mediante un conjunto de ecuaciones, para que el modelo evaluando un punto de la rueda y un punto el terreno sea capaz de determinar si físicamente se ha podido o no producir contacto entre los mismos.

Suponga que el terreno y una rueda del modelo son los sólidos i y j del sistema multicuerpo. Además, las curvas que definen la geometría de ambos sólidos del modelo multicuerpo se encuentran totalmente parametrizadas.

El modelo se dispondrá a evaluar los puntos P_i y P_j pertenecientes al terreno y a la rueda respectivamente. Los puntos P_i y P_j estarán definidos por sus respectivas coordenadas de superficie s_i y s_j .

Se determinará en primera instancia la distancia entre los puntos de contacto mediante el vector que une los puntos evaluados, P_i y P_j . Para ello, haciendo uso de la expresión (5), se determinarán las posiciones absolutas de los puntos P_i y P_j mediante los vectores de posición $\mathbf{r}_{P_i}^i$ y $\mathbf{r}_{P_j}^j$.

Para poder definir los vectores $\mathbf{r}_{P_i}^i$ y $\mathbf{r}_{P_j}^i$ será necesario disponer de las matrices de rotación $\mathbf{A}^i$ y $\mathbf{A}^j$ y de los vectores locales de posición de los puntos P_i y P_j .

Los vectores locales de posición $\bar{\mathbf{u}}_{P_i}^i$ y $\bar{\mathbf{u}}_{P_j}^j$ de los puntos P_i y P_j se calcularán por medio de las expresiones (127) y (130):

$$\bar{\mathbf{u}}_{P_i}^i = [s_i \quad y(s_i)]^T \quad (133)$$

$$\bar{\mathbf{u}}_{P_j}^j = [R \cos(s_j) \quad R \sin(s_j)]^T \quad (134)$$

Haciendo uso de la expresión (5) se obtendrán las posiciones absolutas de los puntos P_i y P_j , definidas por los vectores $\mathbf{r}_{P_i}^i$ y $\mathbf{r}_{P_j}^j$:

$$\mathbf{r}_{P_i}^i = \mathbf{R}^i + \mathbf{A}^i \bar{\mathbf{u}}_{P_i}^i \quad (135)$$

$$\mathbf{r}_{P_j}^j = \mathbf{R}^j + \mathbf{A}^j \bar{\mathbf{u}}_{P_j}^j \quad (136)$$

Restando los vectores $\mathbf{r}_{P_i}^i$ y $\mathbf{r}_{P_j}^j$ se obtendrá la distancia entre los puntos evaluados P_i y P_j , definida por el vector $\mathbf{r}_{ij}$:

$$\mathbf{r}_{ij} = \mathbf{r}_{P_j}^j - \mathbf{r}_{P_i}^i \quad (137)$$

$$\mathbf{r}_{ij} = \mathbf{R}^j + \mathbf{A}^j \bar{\mathbf{u}}_{P_j}^j - \mathbf{R}^i + \mathbf{A}^i \bar{\mathbf{u}}_{P_i}^i \quad (138)$$

Dependiendo el orden en que se resten los vectores $\mathbf{r}_{P_i}^i$ y $\mathbf{r}_{P_j}^j$ el vector $\mathbf{r}_{ij}$ apuntará hacia el punto P_i o hacia el punto P_j . El sentido escogido será determinante para que el modelo sea capaz de detectar puntos que, aun satisfaciendo las ecuaciones del contacto, físicamente no han podido contactar. Proyectando el vector $\mathbf{r}_{ij}$ sobre el vector normal al terreno en P_i el modelo podrá determinar si los puntos P_i y P_j realmente han contactado. El signo del producto escalar dictará si dichos vectores difieren o no de sentido.

Para formular el contacto elástico será necesario determinar los vectores tangentes y normales a las curvas parametrizadas en los puntos P_i y P_j . Haciendo uso de las expresiones (128), (129), (131) y (132)

$$\mathbf{t}^i = \left[1 \quad \frac{dy}{ds} \right]^T \quad (139)$$

$$\mathbf{t}^j = [-R \sin(s_j) \quad R \cos(s_j)]^T \quad (140)$$

$$\mathbf{n}^i = \left[\frac{dy}{ds} - 1 \right]^T \quad (141)$$

$$\mathbf{n}^j = \left[-R \cos(s_j) - R \sin(s_j) \right]^T \quad (142)$$

Se trata ahora de formular, a partir de toda la terminología anterior, las ecuaciones que deben de satisfacer las coordenadas de superficie s_i y s_j de los puntos P_i y P_j que realmente han contactado en el instante t evaluado. Las ecuaciones derivarán de las condiciones físicas necesarias para que se produzca contacto entre la rueda y el terreno:

- Los vectores tangentes $\mathbf{t}^i$ y $\mathbf{t}^j$ en los puntos de contacto P_i y P_j deben de tener la misma dirección. Si los vectores $\mathbf{t}^i$ y $\mathbf{t}^j$ son paralelos, por la perpendicularidad existente entre los vectores tangentes y normales a una curva parametrizada en un punto cualquiera, los vectores $\mathbf{n}^i$ y $\mathbf{n}^j$ también lo serán.

Suponga una rueda que rueda en un plano totalmente horizontal. El punto de contacto siempre se encontrará en la vertical del centro de gravedad de la rueda, de manera ideal. Solamente en dicho punto P_j el vector $\mathbf{t}^j$ tiene una dirección idéntica al vector $\mathbf{t}^i$. Dicha condición la satisfacen en todo momento dos puntos de la rueda, el punto que contacta y su homólogo a 180 °. El modelo deberá de tener en cuenta que físicamente sólo un punto puede haber contactado con el terreno.

- Cuando los puntos P_i y P_j contactan la rueda indentará o penetrará en el terreno en la dirección de los vectores $\mathbf{n}^i$ y $\mathbf{n}^j$. Por ende, para que los puntos P_i y P_j hayan podido físicamente contactar el vector $\mathbf{r}_{ij}$ debe de ser perpendicular a los vectores $\mathbf{t}^i$ y $\mathbf{t}^j$. Y por consecuencia, el vector $\mathbf{r}_{ij}$ deberá de ser paralelo a los vectores $\mathbf{n}^i$ y $\mathbf{n}^j$.

Las condiciones físicas descritas se pueden imponer mediante las siguientes ecuaciones:

$$\mathbf{t}^{iT} \cdot \mathbf{r}_{ij} = 0 \quad (143)$$

$$\mathbf{t}^{iT} \cdot \mathbf{n}^j = 0 \quad (144)$$

Las incógnitas de las ecuaciones (143) y (144) serán las coordenadas de superficie s_i y s_j que definen a los puntos P_i y P_j que han contactado.

Debido a la no linealidad de las ecuaciones que formulan al contacto elástico será preciso emplear el algoritmo de *Newton-Raphson* para determinar las coordenadas curvilíneas s_i y s_j .

Básicamente lo que hará el modelo será, para una configuración del sistema multicuerpo definida por el vector $\mathbf{y}$, empezar a evaluar las expresiones (143) y (144) para diversas parejas de puntos P_i y P_j . Cuando una pareja de puntos P_i y P_j , definida por sus coordenadas

de superficie s_i y s_j , haga cero las dos ecuaciones del contacto elástico el modelo podrá validar que los puntos P_i y P_j han contactado físicamente en el instante t .

Una vez el modelo ha determinado los puntos de contacto P_i y P_j , el modelo procederá seguidamente a calcular las fuerzas normales y tangenciales de contacto.

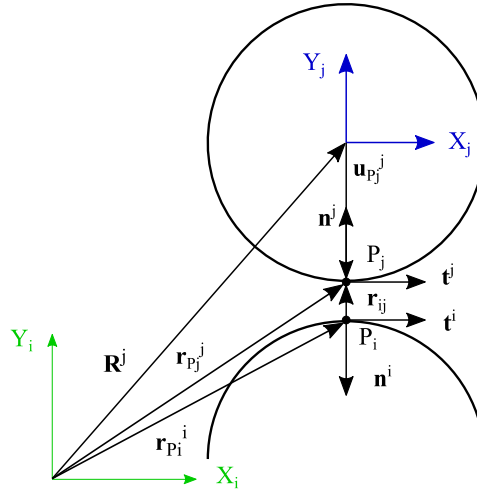


Figura 12. Terminología de los vectores que se precisan para determinar los puntos de contacto P_i y P_j

4.9.4-Fuerza normal de contacto

Cualquier rueda del modelo multicuerpo deberá de indentar una cantidad finita en el terreno para que este pueda ejercer una fuerza normal que la equilibre. Por ello, el primer paso para determinar la fuerza normal de contacto F_{nc} será determinar la indentación δ de la rueda en el terreno.

La indentación tendrá un valor idéntico al módulo del vector que une los puntos de contacto P_i y P_j , definido por $\mathbf{r}_{ij}$. De igual modo, podemos determinar la indentación por medio de la siguiente expresión:

$$\delta = \mathbf{n}^{iT} \cdot \mathbf{r}_{ij} \quad (145)$$

Si el modelo ha determinado de manera correcta los puntos de contacto P_i y P_j , proyectar el vector $\mathbf{r}_{ij}$ sobre el vector $\mathbf{n}^i$ es lo mismo que determinar el módulo de $\mathbf{r}_{ij}$. En este momento el modelo deberá de analizar el signo del producto escalar. Si el producto escalar diese como resultado un valor negativo indicaría que los sólidos i y j del sistema multicuerpo, que modelan al terreno y a la rueda, no han contactado. Por ende, la fuerza normal de contacto F_{nc} deberá de ser nula si δ toma un valor negativo.

Debido a que se modelará a la fuerza normal de contacto F_{nc} con una componente amortiguadora será preciso previamente determinar las velocidades absolutas de los puntos P_i y P_j , definidas por los vectores $\mathbf{v}_{P_i}^i$ y $\mathbf{v}_{P_j}^j$. Haciendo uso de la expresión (14):

$$\mathbf{v}_{P_i}^i = \dot{\mathbf{R}}^i + \dot{\theta}^i \mathbf{A}_{\theta}^i \bar{\mathbf{u}}_{P_i}^i \quad (146)$$

$$\mathbf{v}_{P_j}^j = \dot{\mathbf{R}}^j + \dot{\theta}^j \mathbf{A}_{\theta}^j \bar{\mathbf{u}}_{P_j}^j \quad (147)$$

La velocidad relativa entre los puntos P_i y P_j , definida por el vector $\mathbf{v}_{ij}$, se calculará restando los vectores de velocidad $\mathbf{v}_{P_i}^i$ y $\mathbf{v}_{P_j}^j$:

$$\mathbf{v}_{ij} = \mathbf{v}_{P_i}^i - \mathbf{v}_{P_j}^j = \dot{\mathbf{R}}^i + \dot{\theta}^i \mathbf{A}_{\theta}^i \bar{\mathbf{u}}_{P_i}^i - \dot{\mathbf{R}}^j + \dot{\theta}^j \mathbf{A}_{\theta}^j \bar{\mathbf{u}}_{P_j}^j \quad (148)$$

Del modo en el que se ha efectuado la resta de los vectores $\mathbf{v}_{P_i}^i$ y $\mathbf{v}_{P_j}^j$, el vector $\mathbf{v}_{ij}$ representa la velocidad con la que la rueda ve al terreno aproximarse a ella.

Lo que realmente interesa para formular la fuerza normal de contacto F_{nc} es la velocidad relativa a la que contactan o se aproximan los puntos P_i y P_j , definida por el vector $\dot{\delta}$. La velocidad de indentación $\dot{\delta}$ se determinará proyectando el vector de velocidad relativa $\mathbf{v}_{ij}$ sobre el vector normal al terreno en el punto P_i , definido por $\mathbf{n}^i$. Para que la velocidad de indentación sea positiva, debido al modo de definir el vector $\mathbf{v}_{ij}$, se debe de añadir un signo negativo a la proyección de los vectores $\mathbf{v}_{ij}$ y $\mathbf{n}^i$:

$$\dot{\delta} = -\mathbf{v}_{ij}^T \cdot \mathbf{n}^i \quad (149)$$

Se modelará la fuerza normal de contacto v empleando la teoría de contacto de *Hertz*:

$$F_{nc} = F_e + F_a = K_H \delta^{\frac{3}{2}} + C \dot{\delta} |\delta| \quad (150)$$

La fuerza normal de contacto tendrá una componente elástica (F_e) y una componente amortiguadora (F_a). Donde K_H es la constante Hertziana del contacto elástico y C es la constante de amortiguación de la componente amortiguadora F_a de la fuerza normal de contacto F_{nc} .

Observe que la componente elástica F_e no tiene una dependencia lineal con la indentación δ . Por otra parte, la componente amortiguadora F_a de la fuerza normal de contacto F_{nc} sí depende de manera lineal de la velocidad de indentación $\dot{\delta}$.

La única razón de multiplicar la componente amortiguadora de la fuerza normal F_a de contacto por la indentación δ es para asegurar que la fuerza de contacto F_{nc} sea nula cuando no exista penetración entre la rueda y el terreno

Debido a la forma de definir los vectores $\mathbf{n}^i$ y $\mathbf{n}^j$, el vector de la fuerza normal de contacto $\mathbf{F}_{nc}$ se puede definir mediante la siguiente expresión:

$$\mathbf{F}_{nc} = F_{nc} \cdot \mathbf{n}^j \quad (151)$$

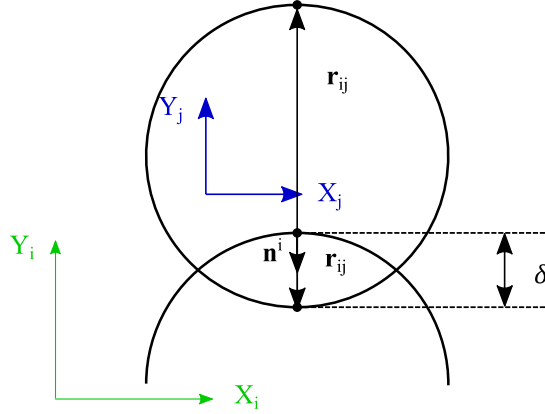


Figura 13. Indentación o penetración del sólido j en el sólido i

4.9.4-Fuerza tangencial de contacto

La fuerza tangencial de contacto F_{tc} aparecerá como consecuencia de la fricción rueda-terreno. La fricción resultado del giro de la rueda, por la introducción de un par motor o restricción de usuario velocidad de giro constante, promoverá la aparición de una fuerza tangencial de contacto que producirá la impulsión de la rueda.

Los parámetros que modelarán la fuerza tangencial de contacto F_{tc} serán el coeficiente de fricción del terreno μ , la velocidad de deslizamiento v_{desl} y la velocidad límite de deslizamiento v_{lim} .

La velocidad de deslizamiento v_{desl} será igual el módulo del vector de velocidad relativa $\mathbf{v}_{ij}$:

$$v_{desl} = |\mathbf{v}_{ij}| \quad (152)$$

La fuerza tangencial ha sido modelada por una función matemática, dependiente de la velocidad de deslizamiento v_{desl} , definida por dos trozos claramente diferenciados. La velocidad de deslizamiento que se halla en la frontera de ambos tramos es la velocidad límite de deslizamiento v_{lim} .

- Cuando la velocidad de deslizamiento v_{desl} es menor que la velocidad límite de deslizamiento v_{lim} la fuerza tangencial de contacto F_{tc} está definida por la siguiente función matemática:

$$F_{tc} = \frac{v_{desl}}{v_{lim}} \mu F_{nc} \quad (153)$$

En el primer tramo de la función F_{tc} , definido por la expresión (153), la fuerza tangencial de contacto F_{tc} presenta dependencia lineal con la velocidad de deslizamiento v_{desl} .

En dicho tramo la rueda no alcanza su capacidad máxima de tracción. Dependiendo de cómo de cerca se encuentre la velocidad de deslizamiento v_{desl} de la velocidad límite de deslizamiento v_{lim} , la rueda desarrollará un mayor o menor porcentaje de su máxima capacidad de tracción o impulsión.

- Cuando la rueda se encuentra en máximo deslizamiento, la velocidad de deslizamiento v_{desl} iguala a la velocidad límite de deslizamiento v_{lim} , la rueda estará desarrollando su máxima capacidad de impulsión. Empleando un modelo de fricción de *Coulomb*:

$$F_{tc} = \mu F_{nc} \quad (154)$$

Por consiguiente, el segundo tramo que define a la función F_{tc} es un tramo constante. Note que μ es el coeficiente de fricción dinámico. Por ende, no se ha modelado la fuerza tangencial de contacto F_{tc} con un coeficiente de fricción estática máximo.

Debido a la forma de definir el vector de velocidad relativa $\mathbf{v}_{ij}$, el vector de la fuerza tangencial de contacto $\mathbf{F}_{tc}$ se define por medio de la siguiente expresión:

$$\mathbf{F}_{tc} = F_{tc} \cdot \frac{\mathbf{v}_{ij}}{|\mathbf{v}_{ij}|} \quad (155)$$

Note que valor de la velocidad límite de deslizamiento v_{lim} no tendrá influencia en la máxima capacidad de impulsión o tracción de las ruedas del sistema multicuerpo pero sí en que se alcance en ellas o no la máxima impulsión. La velocidad límite de deslizamiento marcará el régimen de giro en el que la rueda alcanzará su máxima de tracción, cuanto mayor sea la velocidad límite de deslizamiento v_{lim} mayor régimen de giro deberá de alcanzar la rueda para desarrollar su máxima capacidad tractora.

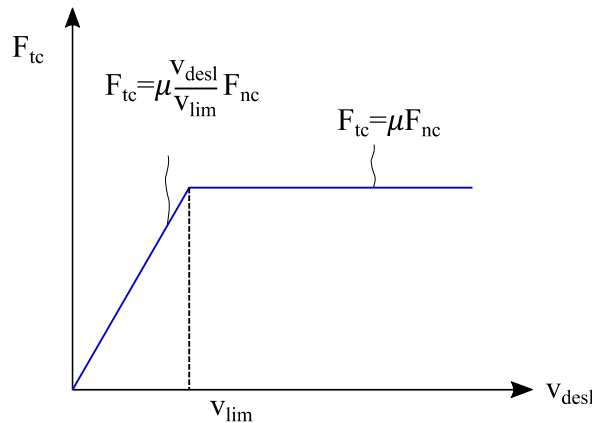


Figura 14. Función matemática que define la fuerza tangencial de contacto

4.10-Modelado del terreno

La geometría del terreno ha sido parametrizada y programada de tal modo que pueda darse diversas formas al terreno. Parametrizar el terreno nos permitirá estudiar y analizar el comportamiento del sistema multicuerpo en una diversidad de situaciones.

El comportamiento del vehículo modelado está directamente influenciado por la forma del terreno por el que avanza. De este modo, se podrá analizar y estudiar el comportamiento del modelo multicuerpo ante una rica variedad de condiciones y entornos.

El modelo se encuentra totalmente programado de modo que el usuario pueda cambiar la forma del terreno sin tener que definir una nueva función matemática o tener que conocer la programación interna del modelo. Modificando el valor de ciertas variables de entrada el terreno se adapta de manera instantánea a las nuevas condiciones de contorno. Se podría haber optado por calcular únicamente las funciones matemáticas que modelan al terreno en las simulaciones dinámicas analizadas para el presente *TFG*.

Se tratará en definitiva de definir la función matemática que defina la altura del terreno en cualquier punto.

El caso más elemental y simple consiste en definir al terreno perfectamente plano. La función matemática que modelaría en este caso al terreno sería una función matemática de valor constante. Si se demanda una forma más compleja la función matemática que modelará al terreno pasará a ser una función definida a trozos. Lógicamente, cada trozo estará formulado por una función matemática distinta.

4.10.1-Inclusión de pendientes en el terreno

Se procede a explicar cómo definir un terreno que conste de un tramo recto seguido de un tramo con pendiente. El poder estudiar el comportamiento del vehículo ante pendientes de diversa naturaleza enriquece sin duda el modelo.

En su versión más simple y cómoda la función matemática que define la altura del terreno consta únicamente de dos trozos. En el primer tramo la función tendrá un valor constante e igual a la altura del terreno mientras que en el segundo trozo la función matemática que modela al terreno consistiría en una recta con una pendiente cualquiera.

Esta definición del terreno puede derivar en problemas numéricos en el proceso de búsqueda de los puntos de contacto en la transición recta-pendiente. Esto podría conllevar a que, cuando el vehículo entra en la transición, el modelo no pudiera encontrar los puntos de contacto, determine que no existe indentación alguna, y prosiguiese con la simulación dinámica.

Por consiguiente, puede ocurrir que cuando el modelo determine de nuevo puntos de contacto entre las ruedas y el terreno sea tal la indentación, debido a que el vehículo se ha

adentrado mucho en la transición sin ascender por la pendiente, que la fuerza normal de contacto sea tan grande que sea capaz de levantar a los sólidos del sistema multicuerpo del terreno.

La manera más sencilla de solventar este problema es reducir el paso de integración numérica. Esto implicaría un mayor tiempo de cálculo en las simulaciones dinámicas.

Este problema se ha solucionado incorporando una transición que precede al tramo con pendiente. La transición se modelará mediante una función polinómica de grado dos o tres, como el usuario desee.

De este modo, la función matemática que define la altura del terreno en cualquier punto pasará será modelada mediante tres trozos y el modelo podrá encontrar satisfactoriamente los puntos de contacto en la transición del tramo recta-pendiente.

Los parámetros que modelarán la altura del terreno en cualquier punto serán el comienzo de la transición x_i , el final de la transición x_f y la pendiente del tramo ascendente m .

El terreno no puede presentar ninguna discontinuidad. Se deberá de imponer, por medio un conjunto de ecuaciones, que exista continuidad entre las funciones matemáticas que definen a cada a trozo. Se obtendrá un sistema de ecuaciones lineales cuyas incógnitas serán los parámetros que definen a las funciones matemáticas de cada trozo.

El tramo recto estará definido por una función y_1 de valor constante, donde C es la altura del terreno:

$$y_1(s) = C \quad -\infty < s < x_i \quad (156)$$

Transición modelada con una función polinómica de grado dos

Se modelará primeramente a la transición mediante una función polinómica de grado dos. Por ende, tres ecuaciones se deben de definir para obtener los parámetros que definen a la función matemática de la transición. La ecuación de la función que modelará a la transición tendrá el siguiente aspecto general:

$$y_2(s) = as^2 + bs + c \quad x_i \leq s < x_f \quad (157)$$

Existen tres requisitos fundamentales para asegurar la continuidad del terreno.

- La función y_2 debe de tener en x_i un valor idéntico a la altura del terreno en el tramo recto C .
- La pendiente de la función y_2 debe ser nula en x_i .
- La pendiente de la transición y_2 en x_f debe de ser idéntica a la pendiente m del tramo ascendente.

Con las dos primeras condiciones descritas se asegura la continuidad del terreno en x_i . Expresando las tres condiciones expuestas en forma de ecuaciones:

$$y_2(s = x_i) = C = ax_i^2 + bx_i + c \quad (158)$$

$$\frac{dy_2}{ds}(s = x_i) = 0 = 2ax_i + b \quad (159)$$

$$\frac{dy_2}{ds}(s = x_f) = m = 2ax_f + b \quad (160)$$

Expresando el sistema de ecuaciones lineales resultante de manera matricial:

$$\mathbf{Ax} = \mathbf{b} \quad (161)$$

$$\mathbf{A} = \begin{bmatrix} x_i^2 & x_i & 1 \\ 2x_i & 1 & 0 \\ 2x_f & 1 & 0 \end{bmatrix} \quad (162)$$

$$\mathbf{b} = \begin{bmatrix} C \\ 0 \\ m \end{bmatrix} \quad (163)$$

$$\mathbf{x} = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad (164)$$

Invirtiendo la matriz de coeficientes $\mathbf{A}$ y multiplicando por el vector de términos independientes $\mathbf{b}$ se obtiene el vector de incógnitas $\mathbf{x}$ que define completamente a la transición y_2 :

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \quad (165)$$

El tramo del terreno con pendiente se modelará mediante una recta y_3 de pendiente m cuya ecuación es la siguiente:

$$y_3(s) = ms + d \quad x_f < s < \infty \quad (166)$$

- Para asegurar la continuidad entre la transición y el tramo con pendiente la función y_3 debe de tener un valor idéntico al que tiene la función y_2 en x_f :

$$y_3(s = x_f) = mx_f + d = ax_f^2 + bx_f + c \quad (167)$$

De la expresión (163) se obtiene el parámetro d que termina de definir a la función y_3 .

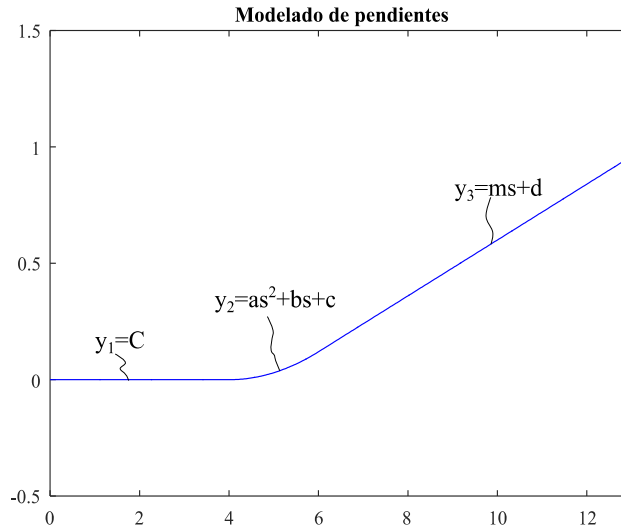


Figura 15. Funciones matemáticas precisadas para modelar un tramo recto y un tramo con pendiente

Transición modelada con una función polinómica de grado tres

A continuación se modelará la transición con una función polinómica de grado tres. La función y_1 , que modela el tramo perfectamente plano del terreno, se expresará de manera idéntica:

$$y_1(s) = C \quad -\infty < s < x_i \quad (168)$$

La función y_2 que modela la transición presenta la siguiente expresión general:

$$y_2(s) = as^3 + bs^2 + cs + d \quad x_i \leq s < x_f \quad (169)$$

Modelando la transición mediante una función polinómica de grado tres se puede imponer una condición adicional. Por consiguiente, aparecerá un parámetro adicional en la formulación. Haciendo que y_2 sea una función polinómica de grado tres se puede fijar la altura del terreno y_f en el punto x_f .

Para asegurar la continuidad del terreno se deben de cumplir cuatro condiciones:

- Las tres primeras condiciones son las mismas que las descritas cuando la función y_2 es modelada con una función polinómica de grado dos.
- La cuarta condición es que la función y_2 tenga un valor de y_f en el punto x_f .

Imponiendo las cuatro condiciones descritas en forma de ecuaciones:

$$y_2(s = x_i) = C = ax_i^3 + bx_i^2 + cx_i + d \quad (170)$$

$$\frac{dy_2}{s} (s = x_i) = 0 = 3ax_i^2 + 2bx_i + c \quad (171)$$

$$\frac{dy_2}{s} (s = x_f) = m = 3ax_f^2 + 2bx_f + c \quad (172)$$

$$y_2 (s = x_f) = y_f = ax_f^3 + bx_f^2 + cx_f + d \quad (173)$$

Expresando de manera matricial el sistema de ecuaciones lineales resultante:

$$\mathbf{Ax} = \mathbf{b} \quad (174)$$

$$\mathbf{A} = \begin{bmatrix} x_i^3 & x_i^2 & x_i & 1 \\ 3x_i^2 & 2x_i & 1 & 0 \\ 3x_f^2 & 2x_f & 1 & 0 \\ x_f^3 & x_f^2 & x_f & 1 \end{bmatrix} \quad (175)$$

$$\mathbf{x} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \quad (176)$$

$$\mathbf{b} = \begin{bmatrix} C \\ 0 \\ m \\ y_f \end{bmatrix} \quad (177)$$

El vector de incógnitas $\mathbf{x}$ define de manera unívoca a la función y_2 , que modelará la transición y cumplirá las condiciones impuestas para asegurar la continuidad de la función y_2 con las funciones y_1 y y_3 .

La función y_3 que define el tramo del terreno con pendiente será modelada y definida de manera idéntica a como se hizo cuando la función y_2 era una función polinómica de grado dos:

$$y_3(s) = ms + e \quad x_f < s < \infty \quad (178)$$

$$y_3(s = x_f) = y_f = mx_f + e \quad (179)$$

El modelo está programado de tal modo que el usuario, cuando define todas las características y parámetros del modelo multicuerpo, deberá de introducir si desea o no incluir una pendiente en el terreno. A continuación, en caso de que sí desee incluir una pendiente en el terreno, deberá de indicar si desea modelar a la transición con una función polinómica de grado dos o de grado tres. Finalmente, deberá de fijar el comienzo de la transición x_i , el punto donde finaliza la transición x_f , magnitud de la pendiente m , y, en caso de que la transición haya sido modelada como una función polinómica de grado tres, la altura del terreno y_f en x_f .

Desarrollando la misma formulación se puede incluir un tramo de terreno perfectamente plano a continuación del tramo con pendiente m . Como consecuencia, se deberá de añadir una nueva transición entre el tramo con pendiente y el nuevo tramo perfectamente plano. La transición podrá ser modelada como una función polinómica de grado dos o de grado tres.

Para dotar al terreno de tales características será necesario incluir dos parámetros adicionales a la formulación. Dichos parámetros son el punto x_{f1} donde comienza la segunda transición y el punto x_{f2} donde finaliza la segunda transición. Si se modelase la segunda transición con una función polinómica de grado tres se podría fijar la altura del terreno y_{f2} en el punto x_{f2} .

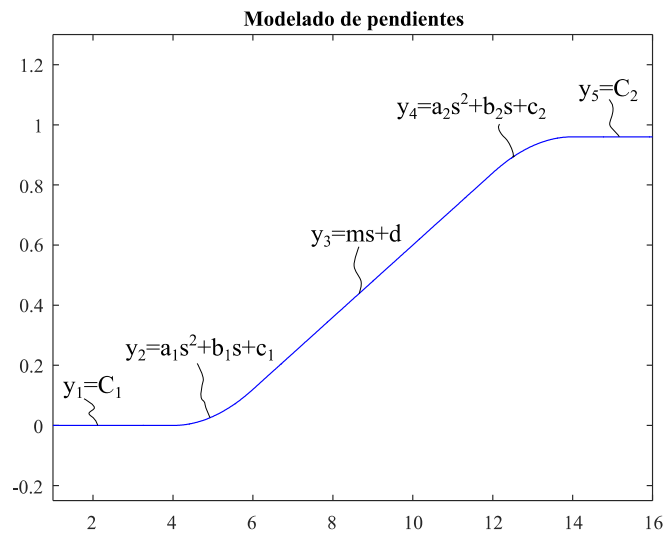


Figura 16. Funciones matemáticas necesarias para modelar dos tramos rectos y un tramo con pendiente

4.10.2-Modelado de irregularidades repentinas o baches en el terreno

El presente apartado describirá cómo se ha modelado una irregularidad repentina o bache en el terreno. La posibilidad de modelar distintos baches permite analizar la respuesta del vehículo y del remolque ante una perturbación de tal tipo. Analizar el comportamiento de un vehículo ante perturbaciones abruptas y repentinas puede tener un gran interés e importancia, más inclusive cuando se pretende modelar la dinámica vertical del vehículo.

El hecho de que el remolque no lleve suspensión, a diferencia del propio vehículo, permitirá ver cómo de eficiente es la suspensión del vehículo y cómo responde el remolque cuando contacta con un bache. Un ensayo dinámico del trabajo consistirá en analizar la influencia de la masa del remolque en su respuesta a un bache. La inclusión de una irregularidad repentina en el terreno para simular dicho ensayo justifica la necesidad de detallar cómo han sido modelados los baches.

El bache se modelará mediante cuatro funciones polinómicas de grado dos. Por ello, la función matemática que definirá la altura del terreno en cualquier punto será una función definida por seis trozos.

El terreno será perfectamente plano antes y después del bache. Además, en su punto medio el bache tendrá pendiente nula.

Se trata de plantear un conjunto de ecuaciones, cuyas incógnitas sean los parámetros que definan a las funciones matemáticas de cada trozo, que aseguren la continuidad del terreno en toda su extensión.

Para formular la irregularidad repentina será necesario definir una serie de parámetros, equivalentes a los parámetros empleados para incluir pendientes en el terreno. Estos parámetros definirán el comienzo y fin de cada trozo de la función matemática del terreno. Además, se definirá la pendiente que define el primer y último tramo de la irregularidad repentina.

La primera parte plana del terreno será modelada por una función y_1 constante de valor C_1 que finaliza en el punto x_1 :

$$y_1 = C_1 \quad -\infty < s < x_1 \quad (180)$$

El primer tramo del bache se definirá por la función polinómica de grado dos y_2 que definirá la altura del terreno entre los puntos x_1 y x_2 :

$$y_2(s) = a_2s^2 + b_2s + c_2 \quad x_1 < s < x_2 \quad (181)$$

Tres requisitos se requieren para asegurar que la función y_2 no presente discontinuidad alguna:

- La función y_2 deberá de tener un valor de C_1 en el punto x_1 .
- La función y_2 debe de tener una pendiente nula en el punto x_1 .
- La pendiente de la función y_2 en el punto x_2 debe de valer m_1 .

Las ecuaciones resultantes de fijar las tres condiciones descritas son:

$$y_2(s = x_1) = C_1 = a_2x_1^2 + b_2x_1 + c_2 \quad (182)$$

$$\frac{dy_2}{ds}(s = x_1) = 0 = 2a_2x_1 + b_2 \quad (183)$$

$$\frac{dy_2}{ds}(s = x_2) = m_1 = 2a_2x_2 + b_2 \quad (184)$$

De nuevo, se puede expresar el sistema de ecuaciones lineales definido por las expresiones anteriores de manera matricial:

$$\mathbf{A}_1 \mathbf{x}_1 = \mathbf{b}_1 \quad (185)$$

$$\mathbf{A}_1 = \begin{bmatrix} x_1^2 & x_1 & 1 \\ 2x_1 & 1 & 0 \\ 2x_2 & 1 & 0 \end{bmatrix} \quad (186)$$

$$\mathbf{x}_1 = \begin{bmatrix} a_2 \\ b_2 \\ c_2 \end{bmatrix} \quad (187)$$

$$\mathbf{b}_1 = \begin{bmatrix} C_1 \\ 0 \\ m_1 \end{bmatrix} \quad (188)$$

El vector de incógnitas $\mathbf{x}_1$ definirá completamente a la función y_2 .

El segundo tramo de la irregularidad repentina está modelado por la función polinómica de grado dos y_3 :

$$y_3(s) = a_3 s^2 + b_3 s + c_3 \quad x_2 < s < x_3 \quad (189)$$

Las condiciones que aseguran la continuidad de la función y_3 con el resto del terreno son las siguientes:

- La función y_3 deberá de tener el mismo valor que tiene la función y_2 en x_2 .
- La pendiente de la función y_2 en x_2 debe de tener un valor igual a m_1 .
- La pendiente de la función y_2 en el punto x_3 debe de ser nula.

La última condición descrita asegura que la irregularidad repentina sea perfectamente plana en su punto más alto.

Desarrollando las ecuaciones que resultan de las condiciones fijadas para y_3 :

$$y_3(s = x_2) = y_2(s = x_2) \quad (190)$$

$$a_3 x_2^2 + b_3 x_2 + c_3 = a_2 x_2^2 + b_2 x_2 + c_2 \quad (191)$$

$$\frac{dy_3}{ds}(s = x_2) = \frac{dy_2}{ds}(s = x_2) = m_1 \quad (192)$$

$$2a_2 x_2 + b_2 = 2a_3 x_2 + b_3 = m_1 \quad (193)$$

$$\frac{dy_3}{ds}(s = x_3) = 0 \quad (194)$$

$$2a_3 x_3 + b_3 = 0 \quad (195)$$

Expresando, de nuevo, el sistema de ecuaciones lineales de manera matricial:

$$\mathbf{A}_2 \mathbf{x}_2 = \mathbf{b}_2 \quad (196)$$

$$\mathbf{A}_2 = \begin{bmatrix} x_2^2 & x_2 & 1 \\ 2x_2 & 1 & 0 \\ 2x_3 & 1 & 0 \end{bmatrix} \quad (197)$$

$$\mathbf{x}_2 = \begin{bmatrix} a_3 \\ b_3 \\ c_3 \end{bmatrix} \quad (198)$$

$$\mathbf{b}_2 = \begin{bmatrix} a_2 x_2^2 + b_2 x_2 + c_2 \\ m_1 \\ 0 \end{bmatrix} \quad (199)$$

La tercera parte del bache será modelado por la función polinómica de grado dos y_4 :

$$y_4(s) = a_4 s^2 + b_4 s + c_4 \quad x_3 < s < x_4 \quad (200)$$

Para que la función y_4 no tenga discontinuidades con las funciones y_3 y y_5 se deben de cumplir las siguientes condiciones:

- La función y_4 deberá de tener el mismo valor que presenta la función y_3 en el punto x_3 .
- La pendiente de la función y_4 deberá de ser nula en el punto x_3 .
- La función y_4 deberá de tener una pendiente de valor m_2 en x_4 .

Expresando dichas condiciones en forma de ecuaciones:

$$y_4(s = x_3) = y_3(s = x_3) \quad (201)$$

$$a_4 x_3^2 + b_4 x_3 + c_4 = a_3 x_3^2 + b_3 x_3 + c_3 \quad (202)$$

$$\frac{dy_4}{ds}(s = x_3) = \frac{dy_3}{ds}(s = x_3) = 0 \quad (203)$$

$$2a_4 x_3 + b_4 = 2a_3 x_3 + b_3 = 0 \quad (204)$$

$$\frac{dy_4}{ds}(s = x_4) = m_2 \quad (205)$$

$$2a_4 x_4 + b_4 = m_2 \quad (206)$$

Como siempre, se resolverá mediante inversión de matrices el sistema de ecuaciones lineales resultante de imponer las condiciones que aseguran la continuidad de la función y_4 :

$$\mathbf{A}_3 \mathbf{x}_3 = \mathbf{b}_3 \quad (207)$$

$$\mathbf{A}_3 = \begin{bmatrix} x_3^2 & x_3 & 1 \\ 2x_3 & 1 & 0 \\ 2x_4 & 1 & 0 \end{bmatrix} \quad (208)$$

$$\mathbf{x}_3 = \begin{bmatrix} a_3 \\ b_3 \\ c_3 \end{bmatrix} \quad (209)$$

$$\mathbf{b}_3 = \begin{bmatrix} a_3 x_3^2 + b_3 x_3 + c_3 \\ 0 \\ m_2 \end{bmatrix} \quad (210)$$

Finalmente, en lo que refiere a funciones polinómicas de grado dos que modelan el bache, se definirá la función polinómica de grado dos y_5 que hará de transición entre la irregularidad repentina y la segunda parte plana del terreno:

$$y_5(s) = a_5 s^2 + b_5 s + c_5 \quad x_4 < s < x_5 \quad (211)$$

Las condiciones que aseguran la continuidad de la función y_5 con la función y_4 y el segundo tramo perfectamente plano del terreno, definido por la función y_6 , son las descritas a continuación:

- La función y_5 debe de tener un valor idéntico al que tiene la función y_4 en x_4 .
- La pendiente de la función y_5 en el punto x_4 debe de tener un valor igual m_2 .
- La función y_5 debe de tener una pendiente nula en el punto x_5

La última condición descrita asegura que el terreno sea nuevamente plano tras la irregularidad repentina.

Expresando las condiciones descritas, que definen en definitiva a la función y_5 , en forma de ecuaciones:

$$y_4(s = x_4) = y_5(s = x_4) \quad (212)$$

$$a_4 x_4^2 + b_4 x_4 + c_4 = a_5 x_4^2 + b_5 x_4 + c_5 \quad (213)$$

$$\frac{dy_4}{ds}(s = x_4) = \frac{dy_5}{ds}(s = x_4) = m_2 \quad (214)$$

$$2a_4 x_4 + b_4 = 2a_5 x_4 + b_5 = m_2 \quad (215)$$

$$\frac{dy_5}{ds}(s = x_5) = 0 \quad (216)$$

$$2a_5 x_5 + b_5 = 0 \quad (217)$$

La resolución del sistema de ecuaciones lineales, definido por las expresiones anteriores, permitirá calcular el vector de incógnitas $\mathbf{x}_4$:

$$\mathbf{A}_4 \mathbf{x}_4 = \mathbf{b}_4 \quad (218)$$

$$\mathbf{A}_3 = \begin{bmatrix} x_4^2 & x_4 & 1 \\ 2x_4 & 1 & 0 \\ 2x_5 & 1 & 0 \end{bmatrix} \quad (219)$$

$$\mathbf{x}_4 = \begin{bmatrix} a_4 \\ b_4 \\ c_4 \end{bmatrix} \quad (220)$$

$$\mathbf{b}_4 = \begin{bmatrix} a_4 x_4^2 + b_4 x_4 + c_4 \\ m_2 \\ 0 \end{bmatrix} \quad (221)$$

Finalmente, la segunda parte plana del terreno será modelada por la función y_6 de valor constante e igual a C_2 :

$$y_6 = C_2 = a_5 x_5^2 + b_5 x_5 + c_5 \quad x_5 < s < \infty \quad (222)$$

Tras todo lo desarrollado queda claro que la geometría del bache queda definida de manera unívoca mediante los valores de los parámetros $x_1, x_2, x_3, x_4, x_5, m_1$ y m_2 .

Por otro lado, tras resolver los sistemas de ecuaciones lineales citados, la función matemática que define la altura del terreno en cualquier parte del mismo, incluido la parte definida por la irregularidad repentina, queda totalmente definida con las magnitudes de los parámetros $C_1, a_2, b_2, c_2, a_3, b_3, c_3, a_4, b_4, c_4, a_5, b_5, c_5$ y C_2 .

El modelo ha sido programado de modo que el usuario pueda modelar fácilmente un bache sobre el terreno sin tener que definir ninguna función matemática o conocer la programación interna del modelo. Simplemente deberá de definir los valores de los parámetros $x_1, x_2, x_3, x_4, x_5, m_1$ y m_2 .

Además, la irregularidad repentina se puede modelar sin dificultad alguna de manera que preceda a un tramo con pendiente, precedido a su vez por la transición asociada, modelada por una función polinómica de grado dos o tres.

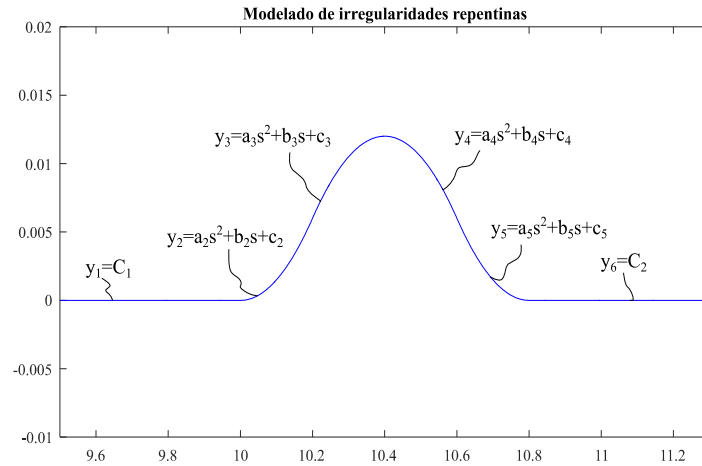


Figura 17. Funciones matemáticas precisadas para modelar un bache en el terreno

4.10.3-Modelado de irregularidades continuas

El presente apartado desarrollará cómo modelar el terreno con irregularidades en toda su extensión. Se trata del modo más realista de definir al terreno pues ningún terreno es perfectamente plano. De este modo, se podrá analizar comportamiento del vehículo y del remolque cuando avanzan por irregularidades diversas.

Se tratará de definir una función matemática que simule, del modo más realista posible, las irregularidades propias de cualquier terreno.

Se debe tener precaución con qué tipo de irregularidades se dota al terreno pues si se definen irregularidades demasiado caóticas o poco realistas el modelo puede tener problemas numéricos a la hora de encontrar el punto de contacto entre las ruedas del modelo multicuerpo y el terreno.

Se ha optado por modelar las irregularidades del terreno mediante la suma de armónicos o funciones sinusoidales del siguiente tipo:

$$y(s) = A \sin(\omega s + \varphi) \quad (223)$$

Donde los términos A , ω y φ son la amplitud, frecuencia angular y desfase de la función sinusoidal y .

Si se modelase el terreno mediante una única función sinusoidal y el terreno presentaría un comportamiento demasiado regular y periódico, siendo una forma poco realista pues cualquier terreno es completamente irregular sin exhibir un comportamiento o geometría perfectamente periódica.

Por ende, la función matemática y que definirá la altura del terreno en cualquier punto resultará de la suma de varios armónicos o funciones sinusoidales:

$$y(s) = \sum_{i=1}^n A_i \sin(\omega_i s + \varphi_i) \quad (224)$$

La función y definida por la expresión (224) dotará al terreno de un aspecto menos periódico y más realista que el definido por la expresión (223).

Para modelar al terreno con irregularidades será preciso que el usuario defina el número de armónicos y la amplitud A_i , la frecuencia angular ω_i y el desfase φ_i de cada armónico. El modelo está programado y parametrizado de modo que no se tenga que definir la función y como en la expresión (224).

De igual modo, se podrán incluir irregularidades a las geometrías desarrolladas y formuladas en los anteriores apartados, es decir, se podrán generar tramos irregulares con pendiente.

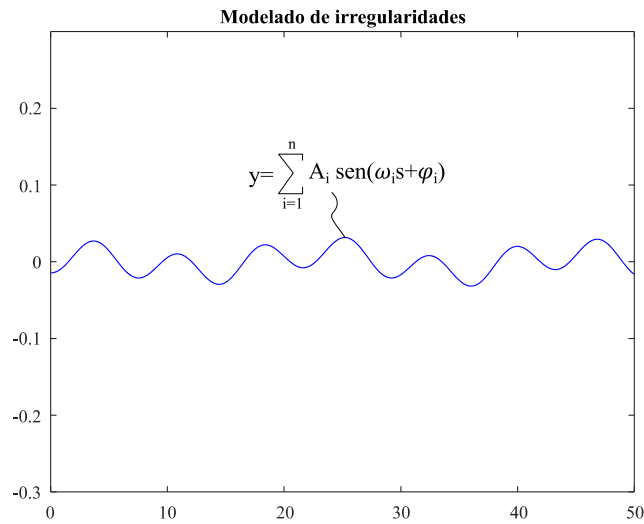


Figura 18. Función matemática empleada para modelar irregularidades en el terreno

5-DESCRIPCIÓN DEL SISTEMA MULTICUERPO MODELADO

5.1-Presentación y enumeración de los sólidos

El sistema multicuerpo modelado está formado por un total de seis sólidos rígidos:

- El terreno, que modela al eslabón fijo. El terreno definirá el sistema de referencia absoluto desde el cual se describirá toda la configuración del modelo multicuerpo.
- La rueda trasera del vehículo.
- La rueda delantera del vehículo.
- El chasis del vehículo.
- El remolque.
- La rueda del remolque.

Será necesario dejar constancia de cómo se ha enumerado a cada sólido del modelo. La enumeración dada al sistema multicuerpo es crucial para tener claro a qué sólido pertenece cada componente de las matrices y vectores empleados en la formulación del modelo.

- Primer sólido: Terreno.
- Segundo sólido: Rueda trasera del vehículo.
- Tercer sólido: Chasis del vehículo.
- Cuarto sólido: Rueda delantera del vehículo.
- Quinto sólido: Remolque.
- Sexto sólido: Rueda del remolque.

Será necesario definir para cada sólido del sistema multicuerpo un sistema de referencia cartesiano. Tal y como se ha comentado a lo largo del trabajo, el origen del sistema de referencia de cada sólido del modelo multicuerpo se situará en su centro de gravedad.

La situación del sistema de referencia absoluto o global se podrá apreciar en la figura del sistema multicuerpo. Los sistemas de referencia restantes, asociados a los restantes sólidos rígidos del sistema, partirán de una orientación inicial idéntica a la que tiene el sistema de referencia absoluto.

5.2-Uniones entre sólidos del sistema multicuerpo

Las ruedas del vehículo se hallan conectadas al chasis mediante muelles o resortes mecánicos con sus correspondientes amortiguadores. Cada rueda del vehículo se encuentra unida al chasis por medio de dos muelles, un muelle vertical y un muelle longitudinal. Por consiguiente, el modelo constará de un total de cuatro muelles-amortiguadores.

Los muelles verticales tendrán la función de sustentar al chasis mientras que los muelles longitudinales tendrán por objeto transmitir el movimiento desde las ruedas al chasis. La definición de los muelles verticales tendrá una gran importancia en la dinámica vertical del

modelo mientras que la definición de los muelles horizontales en la dinámica longitudinal del modelo.

Los muelles verticales y longitudinales se encuentran conectados a las ruedas del vehículo en sus centros de gravedad. Los puntos de conexión de cada muelle-amortiguador al chasis se definirán posteriormente mediante una tabla.

Los puntos de conexión de los muelles longitudinales han sido definidos de tal modo que bajo condiciones estáticas ninguno de ellos modificará su longitud, de este modo se evitará que el modelo multicuerpo oscile horizontalmente en los instantes iniciales

El remolque se encuentra unido al chasis mediante una unión articulada. El chasis arrastrará al remolque en su movimiento por el terreno. El punto de articulación se aclarará más tarde.

La rueda del remolque se encuentra articulada, en su centro de gravedad, al remolque. A diferencia de las ruedas del vehículo, la rueda del remolque no presentará suspensión. Como consecuencia, la rueda del remolque será susceptible de poder perder el contacto con el terreno si se dan unas condiciones que lo propicien.

La siguiente tabla definirá unívocamente los puntos de conexión de los muelles-amortiguadores y de las uniones articuladas. Los puntos de conexión de la suspensión vertical, de la suspensión longitudinal y de las uniones articuladas se definirán mediante las coordenadas locales de los mismos respecto a los sólidos que se encuentran articulados o suspendidos:

Uniones articuladas		
Unión chasis-remolque	$\bar{\mathbf{u}}_p^3$	$\bar{\mathbf{u}}_p^5$
	$\begin{bmatrix} -\frac{1}{2} \cdot L_3 & -\frac{1}{2} \cdot H_3 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} \cdot L_5 + d & 0 \end{bmatrix}$
Unión remolque-rueda del remolque	$\bar{\mathbf{u}}_p^5$	$\bar{\mathbf{u}}_p^6$
	$\begin{bmatrix} 0 & -\frac{1}{2} \cdot H_5 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \end{bmatrix}$

Tabla 5.1. Definición de las articulaciones

Muelles-amortiguadores		
Muelle vertical trasero	$\bar{\mathbf{u}}_p^2$	$\bar{\mathbf{u}}_p^3$
	$\begin{bmatrix} 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -0,2985 \cdot L_3 & -\frac{1}{2} \cdot H_3 \end{bmatrix}$
Muelle vertical delantero	$\bar{\mathbf{u}}_p^3$	$\bar{\mathbf{u}}_p^4$
	$\begin{bmatrix} 0,2985 \cdot L_3 & -\frac{1}{2} \cdot H_3 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \end{bmatrix}$
Muelle longitudinal trasero	$\bar{\mathbf{u}}_p^2$	$\bar{\mathbf{u}}_p^3$
	$\begin{bmatrix} 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -0,2985 \cdot L_3 - l_0 & -l_0 - H_3 + \frac{g \cdot m_3}{k_v} \end{bmatrix}$
Muelle longitudinal delantero	$\bar{\mathbf{u}}_p^3$	$\bar{\mathbf{u}}_p^4$
	$\begin{bmatrix} 0,2985 \cdot L_3 + l_0 & -l_0 - H_3 + \frac{g \cdot m_3}{k_v} \end{bmatrix}$	$\begin{bmatrix} 0 & 0 \end{bmatrix}$

Tabla 5.2. Definición de los muelles-amortiguadores

5.3- Parámetros que definen al modelo y justificación de los valores asignados

Las dimensiones características que caracterizarán a los sólidos del sistema multicuerpo serán los radios de las ruedas del modelo, la longitud y altura del chasis y la longitud y altura del remolque:

- Todas las ruedas del modelo tienen un radio R de 0,4 metros.
- La longitud L_3 y altura H_3 del chasis tienen un valor de 4,95 metros y 1,84 metros respectivamente.
- El remolque tiene una altura H_5 de 0,5 metros y una longitud L_5 de 2 metros.

La distancia, que define la distancia entre el punto de articulación chasis-remolque y el cuerpo del remolque, tiene un valor de 1 metro.

Las propiedades inerciales de cada sólido del modelo multicuerpo serán determinantes para definir de la matriz de masa del sistema multicuerpo $\mathbf{M}$:

- El terreno tendrá una masa m_1 y momento de inercia I_1 nulos.
- Las ruedas del modelo multicuerpo tienen una masa m_2 , m_4 y m_6 de 10 kg.
- El chasis tiene una masa m_3 de 3230 kg.
- El remolque tiene una masa m_5 de 700 kg

Los momentos de inercia de los sólidos rígidos del sistema multicuerpo han sido aproximados a los valores que tomarían sus geometrías elementales.

- Los momentos de inercia de las ruedas del modelo multicuerpo han sido aproximados a los momentos de inercia de dos círculos:

$$I_{2,4,6} = \frac{1}{2} m_{1,2,4,6} R^2 \quad (225)$$

- El momento de inercia del chasis I_3 y el momento de inercia del remolque I_5 han sido aproximados a los momentos de inercia de dos rectángulos:

$$I_3 = \frac{1}{12} m_3 (L_3^2 + H_3^2) \quad (226)$$

$$I_5 = \frac{1}{12} m_5 (L_5^2 + H_5^2) \quad (227)$$

Todos los muelles-amortiguadores del sistema multicuerpo tienen una longitud inicial l_0 de 0,35 m.

Los muelles verticales del modelo han sido modelados con rigideces k_v idénticas. Las rigideces k_v han sido evaluadas de modo que, bajo condiciones estáticas, los muelles verticales del modelo se deformen 10 centímetros. Los puntos de conexión de los muelles

verticales del modelo multicuerpo propiciarán que ambos soporten, cuando el sistema se encuentra en reposo, la mitad del peso del chasis.

Por consiguiente, para que se cumpla lo descrito, las rigideces k_v de los muelles verticales del modelo multicuerpo deberán de valer:

$$k_v = \frac{\frac{3230}{2} \text{ kg} \cdot 9,81 \frac{\text{m}}{\text{s}^2}}{0,1 \text{ m}} = 158431,5 \frac{\text{N}}{\text{m}} \quad (228)$$

Cuando el sistema multicuerpo se encuentre en movimiento, debido a la inercia del chasis, existirá una transferencia de peso a la rueda trasera de modo que cada muelle vertical no soportará el mismo porcentaje de peso del chasis.

Los muelles longitudinales han sido modelados con rigideces k_h idénticas. Sin embargo, la determinación de la rigidez k_h de cada muelle longitudinal no es intuitiva ni se presta a ser calculada mediante cálculos sencillos, su función no es de sustentar sino de impulsar. Como consecuencia, las rigideces k_h de los muelles longitudinales han sido definidas a partir de los resultados obtenidos en diversas simulaciones dinámicas hasta encontrar el valor más óptimo:

$$k_h = 950589 \frac{\text{N}}{\text{m}} \quad (229)$$

Los coeficientes de amortiguación c de los muelles del modelo multicuerpo han sido definidos de manera empírica en base a los análisis de diversas simulaciones dinámicas realizadas con el modelo. Se ha ido variando el valor del coeficiente de amortiguación c de cada muelle-amortiguador hasta conseguir una respuesta del sistema multicuerpo razonable y realista.

El mejor modo de ajustar los valores de c es mediante ensayos estáticos y dejar que el sistema oscile verticalmente hasta que encuentre una configuración estática de reposo. El estudio y análisis de una simulación estática con el modelo multicuerpo puede comprobar si los valores de c son óptimos o no lo son.

Finalmente, los coeficientes de amortiguación de los muelles verticales c_v y los coeficientes de amortiguación de los muelles horizontales c_h se han definido por medio de los siguientes valores:

$$c_v = 4752,95 \frac{\text{Ns}}{\text{m}} \quad (230)$$

$$c_h = 47529,5 \frac{\text{Ns}}{\text{m}} \quad (231)$$

Queda por definir los parámetros que modelarán el contacto elástico de cada rueda con el terreno. Se han definido propiedades idénticas para los tres contactos elásticos del modelo multicuerpo.

El coeficiente de fricción del terreno μ puede ser modificado en función de las condiciones que se deseen simular. En condiciones normales, μ ha tomado un valor de 0,8 en las simulaciones dinámicas realizadas con el modelo. El coeficiente de fricción μ delimitará la capacidad de tracción en cada rueda del modelo multicuerpo. Un ensayo realizado con el modelo y recogido por el presente *TFG* estudiará la influencia de μ en la capacidad de impulsión del vehículo.

La magnitud de la velocidad límite de deslizamiento v_{lim} será crucial en la dinámica longitudinal del modelo. Recuerde que la magnitud de v_{lim} dictará el régimen de giro donde las ruedas alcanzarán el máximo deslizamiento posible. En condiciones normales, se ha modelado v_{lim} con un valor de 0,1 m/s.

El valor de la constante *Hertziana* K_H ha sido determinado de modo que, bajo condiciones estáticas, las ruedas del vehículo indenten o penetren en el terreno una cantidad de 2 centímetros:

$$K_H = \frac{\left(10 \text{ kg} + \frac{3230}{2} \text{ kg}\right) 9,81 \frac{\text{m}}{\text{s}^2}}{(0,02 \text{ m})^{\frac{3}{2}}} = 5,636 \cdot 10^6 \frac{\text{N}}{\text{m}^{\frac{3}{2}}} \quad (232)$$

La simulación estática servirá para corroborar si efectivamente bajo condiciones estáticas las ruedas del vehículo presentan una indentación δ de 2 cm.

La constante *Hertziana* K_H del contacto elástico rueda del remolque-terreno ha sido definida con una magnitud idéntica a las constantes *Hertzianas* de los contactos elásticos ruedas del vehículo-terreno. De este modo, la rueda del remolque necesitará indentar o penetrar en el terreno una cantidad menor que las ruedas del vehículo para equilibrar el peso del conjunto remolque-rueda del remolque. Parece lógico que la rueda del remolque presente una indentación δ menor pues la rueda del remolque soportará un peso menor, recuerde que tendrá que equilibrar su propio peso y el peso del remolque transmitido mediante la unión articulada.

El coeficiente de amortiguación C que define la componente amortiguadora F_a de la fuerza normal de contacto F_{nc} ha sido definido de manera empírica, en base a los resultados obtenidos en numerosas simulaciones dinámicas:

$$C = 112721,66 \frac{\text{Ns}}{\text{m}} \quad (233)$$

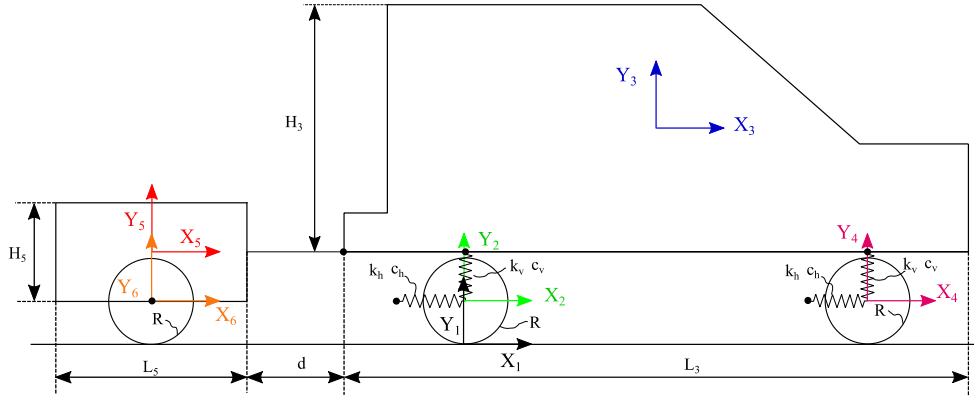


Figura 19. Esquema general del sistema multicuerpo modelado

5.4- Vectores de coordenadas, velocidades y aceleraciones generalizadas del sistema multicuerpo

Debido a que el sistema multicuerpo está formado por un total de seis sólidos, los vectores $\mathbf{q}$, $\dot{\mathbf{q}}$ y $\ddot{\mathbf{q}}$ tendrán un total de 18 componentes:

$$\mathbf{q} = [R_x^1 \ R_y^1 \ \theta^1 \ R_x^2 \ R_y^2 \ \theta^2 \ R_x^3 \ R_y^3 \ \theta^3 \ \dots \ R_x^6 \ R_y^6 \ \theta^6]^T \quad (234)$$

$$\dot{\mathbf{q}} = [\dot{R}_x^1 \ \dot{R}_y^1 \ \dot{\theta}^1 \ \dot{R}_x^2 \ \dot{R}_y^2 \ \dot{\theta}^2 \ \dot{R}_x^3 \ \dot{R}_y^3 \ \dot{\theta}^3 \ \dots \ \dot{R}_x^6 \ \dot{R}_y^6 \ \dot{\theta}^6]^T \quad (235)$$

$$\ddot{\mathbf{q}} = [\ddot{R}_x^1 \ \ddot{R}_y^1 \ \ddot{\theta}^1 \ \ddot{R}_x^2 \ \ddot{R}_y^2 \ \ddot{\theta}^2 \ \ddot{R}_x^3 \ \ddot{R}_y^3 \ \ddot{\theta}^3 \ \dots \ \ddot{R}_x^6 \ \ddot{R}_y^6 \ \ddot{\theta}^6]^T \quad (236)$$

Será muy importante tener claro la enumeración de los sólidos del sistema multicuerpo para discernir a qué sólido del modelo multicuerpo hace referencia cada componente de los vectores $\mathbf{q}$, $\dot{\mathbf{q}}$ y $\ddot{\mathbf{q}}$.

5.5- Matriz de masa del sistema multicuerpo

Para el sistema multicuerpo modelado la matriz de masa del modelo multicuerpo $\mathbf{M}$ será una matriz cuadrada de dimensiones 18x18:

$$\mathbf{M} = \begin{bmatrix} m^1 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & m^1 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & J^1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & m^2 & 0 & 0 & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & m^2 & 0 & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & J^2 & \dots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \ddots & \ddots & \ddots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & m^6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & m^6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & J^6 \end{bmatrix} \quad (237)$$

5.6- Restricciones del sistema multicuerpo

La restricción de eslabón fijo y las restricciones asociadas a las uniones articuladas serán fijas e invariantes en cualquier simulación dinámica. Por ende, el vector de restricciones del sistema multicuerpo $\mathbf{C}$ tendrá como mínimo 7 componentes:

$$\mathbf{C}(\mathbf{q}, t) = \mathbf{0} \quad (238)$$

$$\mathbf{C}(\mathbf{q}, t) = [C_1 \ C_2 \ C_3 \ C_4 \ C_5 \ C_6 \ C_7]^T \quad (239)$$

El vector de restricciones del sistema multicuerpo $\mathbf{C}$ tendrá las dimensiones del vector $\mathbf{C}$ de la expresión (239) cuando se active la dinámica longitudinal del modelo multicuerpo con un par motor, introducido como un vector de fuerzas generalizadas $\mathbf{Q}$, en alguna rueda del vehículo.

Si se introduce una restricción de velocidad de giro constante en una rueda del vehículo se introducirá una nueva componente en el vector $\mathbf{C}$ de la expresión (239). Si se introdujera dos restricciones de velocidad de giro constante, una a cada rueda del vehículo, se añadirían dos componentes al vector $\mathbf{C}$ de la expresión (239).

Por consiguiente, el número máximo de componentes que podrá tener el vector $\mathbf{C}$ en el sistema multicuerpo modelado en este *TFG* serán nueve:

$$\mathbf{C}(\mathbf{q}, t) = [C_1 \ C_2 \ C_3 \ C_4 \ C_5 \ C_6 \ C_7 \ C_8 \ C_9]^T \quad (240)$$

Las dimensiones de la matriz Jacobiana de las restricciones $\mathbf{C}_q$ dependerá de las restricciones fijadas en cada simulación dinámica. El número de columnas de la matriz $\mathbf{C}_q$ será siempre igual a 18, depende únicamente del número de coordenadas generalizadas del sistema multicuerpo.

Por ejemplo, si se añade una restricción de velocidad de giro constante a la rueda trasera del vehículo la matriz Jacobiana $\mathbf{C}_q$ será de 8 x 18:

$$\mathbf{C}_q = \begin{bmatrix} \frac{\partial C_1}{\partial q_1} & \frac{\partial C_1}{\partial q_2} & \frac{\partial C_1}{\partial q_3} & \dots & \frac{\partial C_1}{\partial q_{18}} \\ \frac{\partial C_2}{\partial q_1} & \frac{\partial C_2}{\partial q_2} & \frac{\partial C_2}{\partial q_3} & \dots & \frac{\partial C_2}{\partial q_{18}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial C_8}{\partial q_1} & \frac{\partial C_8}{\partial q_2} & \frac{\partial C_8}{\partial q_3} & \dots & \frac{\partial C_8}{\partial q_{18}} \end{bmatrix} \quad (241)$$

Las dimensiones del resto de vectores y matrices relacionados con el vector de restricciones ($\mathbf{C}_t$, $\mathbf{C}_{tt}$, $\mathbf{C}_{qt}$), y que aparecen en las ecuaciones dinámicas implementadas, dependerán de las restricciones definidas para cada simulación dinámica.

De igual modo, las dimensiones del término $\mathbf{Q}_d$, relacionado con la segunda derivada respecto al tiempo del vector $\mathbf{C}$, dependerán de las restricciones empleadas en la simulación dinámica realizada con el modelo multicuerpo.

5.7- Vector de fuerzas generalizadas externas

Cada vector de fuerzas generalizadas $\mathbf{Q}$ empleado en el modelo multicuerpo será un vector columna de 18 componentes, pues depende de las coordenadas generalizadas del sistema multicuerpo. El vector de fuerzas generalizadas externas del sistema multicuerpo $\mathbf{Q}_e$ estará formado por los vectores de fuerzas generalizadas externas $\mathbf{Q}_e^i$ de cada sólido del modelo multicuerpo:

$$\mathbf{Q}_e = [\mathbf{Q}_e^{1T} \quad \mathbf{Q}_e^{2T} \quad \mathbf{Q}_e^{3T} \quad \mathbf{Q}_e^{4T} \quad \mathbf{Q}_e^{5T} \quad \mathbf{Q}_e^{6T}]^T \quad (242)$$

A su vez, el vector de fuerzas generalizadas externas de cada sólido del sistema multicuerpo $\mathbf{Q}_e^i$ será resultado de sumar los vectores de fuerzas generalizadas externas asociados a cada una de las fuerzas presentes en el modelo (fuerzas gravitatorias, fuerzas de los muelles-amortiguadores, fuerzas de usuario y fuerzas de contacto elástico):

$$\mathbf{Q}_e^i = \mathbf{Q}_g^i + \mathbf{Q}_{ce}^i + \mathbf{Q}_{us}^i + \mathbf{Q}_{ma}^i \quad (243)$$

Ensamblando para el resto de sólidos del modelo multicuerpo:

$$\mathbf{Q}_e = \mathbf{Q}_g + \mathbf{Q}_{ce} + \mathbf{Q}_{us} + \mathbf{Q}_{ma} \quad (244)$$

5.8- Configuración inicial del sistema multicuerpo

Para poder realizar cualquier simulación dinámica será necesario la definición de un vector de estado inicial $\mathbf{y}_0$, lo cual implica definir los vectores $\mathbf{q}_0$ y $\dot{\mathbf{q}}_0$.

El vector de coordenadas generalizadas iniciales del sistema multicuerpo $\mathbf{q}_0$ se definirá en primera instancia a partir de la geometría del modelo.

La configuración establecida por dicho vector $\mathbf{q}_0$ no se corresponde a una configuración de equilibrio estática del sistema multicuerpo pues ninguna rueda del modelo multicuerpo presenta indentación alguna en el terreno. Por ende, se trataría de una configuración de equilibrio estática de sistema multicuerpo si las ruedas no tuvieran que indentar en el terreno para equilibrar las fuerzas que soportan. La simulación estática tendrá por fin obtener un vector $\mathbf{q}_0$ que defina una configuración de equilibrio estática del sistema multicuerpo.

Para definir el vector de coordenadas generalizadas iniciales de cada sólido $\mathbf{q}_0^i$ se deberá de tener en cuenta la orientación inicial θ_0^i de cada sólido del sistema multicuerpo y la posición

inicial del centro de gravedad (origen del sistema de referencia) de cada sólido del modelo multicuerpo, definida por el vector $\mathbf{R}_0^i$:

$$\mathbf{q}_0^1 = [0 \quad 0 \quad 0]^T \quad (245)$$

$$\mathbf{q}_0^2 = [0 \quad R \quad 0]^T \quad (246)$$

$$\mathbf{q}_0^3 = \left[0,2985 \cdot L_3 \quad R + l_0 + \frac{1}{2} H_3 - \frac{0,5gm_3}{k} \quad 0 \right]^T \quad (247)$$

$$\mathbf{q}_0^4 = [2 \cdot 0,2985 \cdot L_3 \quad R \quad 0]^T \quad (248)$$

$$\mathbf{q}_0^5 = \left[-0,2015 \cdot L_3 - d - \frac{1}{2} \cdot L_5 \quad R + \frac{1}{2} \cdot H_5 \quad 0 \right]^T \quad (249)$$

$$\mathbf{q}_0^6 = \left[-0,2015 \cdot L_3 - d - \frac{1}{2} \cdot L_5 \quad R \quad 0 \right]^T \quad (250)$$

Ensamblando los vectores de coordenadas generalizadas iniciales de cada sólido $\mathbf{q}_0^i$ en el vector de coordenadas generalizadas iniciales del sistema multicuerpo $\mathbf{q}_0$:

$$\mathbf{q}_0 = [\mathbf{q}_0^{1T} \quad \mathbf{q}_0^{2T} \quad \mathbf{q}_0^{3T} \quad \mathbf{q}_0^{4T} \quad \mathbf{q}_0^{5T} \quad \mathbf{q}_0^{6T}]^T \quad (251)$$

Por otro lado, la definición del vector de velocidades generalizadas iniciales del sistema multicuerpo $\dot{\mathbf{q}}_0$ presenta dependencia directa con el tipo de simulación dinámica que se realice con el modelo.

Por ejemplo, si se impone una restricción de velocidad de giro constante en alguna rueda del vehículo conviene definir la velocidad de giro $\dot{\theta}^i$ en el vector $\dot{\mathbf{q}}_0^i$ asociado a la rueda del modelo multicuerpo. El resto de vectores de velocidades $\dot{\mathbf{q}}_0^i$ pueden definirse en primera instancia como vectores nulos, si no se precisa de la información necesaria para definir el resto de vectores $\dot{\mathbf{q}}_0^i$.

En su forma más simple y general, el vector de velocidades generalizadas del sistema multicuerpo $\dot{\mathbf{q}}_0$ tomará el siguiente aspecto:

$$\dot{\mathbf{q}}_0 = [\dot{\mathbf{q}}_0^{1T} \quad \dot{\mathbf{q}}_0^{2T} \quad \dot{\mathbf{q}}_0^{3T} \quad \dot{\mathbf{q}}_0^{4T} \quad \dot{\mathbf{q}}_0^{5T} \quad \dot{\mathbf{q}}_0^{6T}]^T \quad (252)$$

6-IMPLEMENTACIÓN DEL MODELO EN EL SOFTWARE *MATLAB*

El presente apartado tiene como fin explicar cómo se ha implementado y modelado el sistema multicuerpo en el software *Matlab*. Se tratará de aplicar todo lo desarrollado y formulado en el marco teórico e implementar todas las características del modelo multicuerpo descritas en la descripción del modelo.

Modelar el sistema multicuerpo en *Matlab* consistirá en la programación de diversas funciones y programas que permitan simular el comportamiento del vehículo con distintas características y en distintas condiciones. En las simulaciones dinámicas se obtendrán soluciones numéricas y no analíticas o continuas. Por consiguiente, se calculará el vector $\mathbf{y}$, que define la configuración del modelo multicuerpo, en ciertos instantes t del intervalo de simulación y no en cada instante t .

Se describirán todas las funciones programadas en el software *Matlab* y se citará la función que desempeña cada una de ellas en el proceso de cálculo numérico. A modo general, se describirán las siguientes funciones y programas:

- La función que define todas las características y parámetros del modelo multicuerpo
- La función que permite obtener el vector $\dot{\mathbf{y}}$, necesario para integrar numéricamente y poder obtener el vector $\mathbf{y}$ en un instante $t + \Delta t$
- El programa que resuelve cualquier simulación dinámica y define el proceso de integración numérica (tipo de integrador numérico a emplear en la simulación, en qué instantes se desea calcular el vector $\mathbf{y}$...)
- Funciones programadas para el posproceso de resultados
- Funciones programadas para representar cualquier conjunto de parámetros o variables del modelo multicuerpo
- Función que permite obtener una animación del movimiento del sistema multicuerpo en cualquier simulación

Si desea profundizar la programación de cualquier función a la cual se haga referencia en los siguientes apartados podrá verlas con todo tipo de detalles en los Anexos del presente TFG.

6.1-Función que define todas las características y parámetros del sistema multicuerpo

La función *Definicion_Modelo.m* tiene por objeto definir todos los parámetros y características que modelan al sistema multicuerpo. Si se quiere variar el valor de algún parámetro del modelo se debe acudir a dicha función pues define los valores de todas las

variables de entrada que necesita el modelo para simular el comportamiento del vehículo con del remolque.

La función *Definicion_Modelo.m* se ha programado de modo que se definen los valores de todas las variables del modelo en las primeras líneas de código. Por ello, en las restantes líneas de código no aparece la magnitud de ninguna variable sino el nombre de la variable que la modela. Por ende, si se desea cambiar alguna característica del modelo multicuerpo bastará con acudir a las primeras líneas de la función.

La función *Definicion_Modelo.m* define los siguientes parámetros del modelo:

- Dimensiones características y propiedades mecánicas de cada sólido.
- Geometría del terreno. En la función *Definicion_Modelo.m* es donde se indicará si se quiere incluir alguna pendiente en el terreno, si se desea modelar un bache o si se desean incluir irregularidades en el terreno. En caso afirmativo, se deberán de definir los valores de los parámetros que definen a cada geometría y que fueron desarrollados en el marco teórico.
- Parámetros que definen a los muelles-amortiguadores del modelo multicuerpo. Se deberán de definir los puntos de conexión, magnitudes de la rigidez y coeficiente de amortiguación y la longitud inicial del muelle.
- Parámetros que modelan el contacto elástico rueda-terreno. Se deberán de incluir los valores del coeficiente de fricción, velocidad límite de deslizamiento, constante *Hertziana*, coeficiente de amortiguación de la componente amortiguadora de la fuerza normal de contacto y una condición inicial de contacto, definida por las coordenadas curvilíneas s_{i0} y s_{j0} de los puntos de contacto iniciales P_{i0} y P_{j0} .
- Restricciones del sistema multicuerpo. Se definirán los puntos de articulación y las restricciones de velocidad de giro constante que se deseen incluir, las restricciones de eslabón fijo y uniones articuladas deberán de estar siempre presentes.
- Configuración inicial del sistema multicuerpo. Se deberá de definir el vector $\mathbf{y}_0$, definido por los vectores $\mathbf{q}_0$ y $\dot{\mathbf{q}}_0$.
- Fuerzas de usuario. Se deberá de definir si se desea incluir un par motor, y su magnitud, sobre alguna rueda del vehículo o sobre ambas.

Los valores de todas las variables citadas se agruparán en una única variable, denominada *SMcuerpo*.

El resto de funciones necesitarán hacer uso de todos los valores definidos en *Definicion_Modelo.m* durante cualquier simulación dinámica. Como consecuencia, la variable *SMcuerpo* deberá de ser declarada como una variable global para que se pueda acceder, desde otras funciones programadas, a los valores definidos en *Definicion_Modelo.m*.

6.2-Función que obtiene la derivada respecto al tiempo del vector de estado

La función *EcMovimiento_Baumgarte.m* tiene por objeto determinar el vector $\dot{\mathbf{y}}$, que contiene los vectores $\dot{\mathbf{q}}$ y $\ddot{\mathbf{q}}$, en un instante t cualquiera de una simulación dinámica.

Las entradas que tomará *EcMovimiento_Baumgarte.m* serán el instante t de simulación y el vector de estado $\mathbf{y}$, que contiene los vectores $\mathbf{q}$ y $\dot{\mathbf{q}}$, en el instante t . La función *EcMovimiento_Baumgarte.m* necesitará hacer uso de los valores de todas las variables definidas en *Definicion_Modelo.m*.

Para poder calcular el vector $\ddot{\mathbf{q}}$ la función *EcMovimiento_Baumgarte.m* deberá de resolver las ecuaciones del movimiento del sistema multicuerpo. Por ello, la función *EcMovimiento_Baumgarte.m* deberá de determinar todos los términos de la matriz de coeficientes $\mathbf{A}$ y del vector de términos independientes $\mathbf{b}$ de la expresión (121). Para ello, *EcMovimiento_Baumgarte.m* llamará a un conjunto de funciones programadas que le proporcionarán los términos que aparecen en la matriz $\mathbf{A}$ y en el vector $\mathbf{b}$.

Para calcular la matriz $\mathbf{A}$ en un instante t la función *EcMovimiento_Baumgarte.m* deberá de determinar la matriz de masa del sistema multicuerpo $\mathbf{M}$ y la matriz Jacobiana de las restricciones $\mathbf{C}_q$. Para definir la matriz $\mathbf{A}$ se han programado las siguientes funciones:

- La función *Mmasa.m* calcula la matriz $\mathbf{M}$ a partir del vector $\mathbf{q}$ en el instante t .
- La función *Jacobiana.m* calcula la matriz $\mathbf{C}_q$ a partir del instante t y el vector $\mathbf{q}$. La función *Jacobiana.m* se apoyará de la función *Jacobianausuario* para determinar las componentes de $\mathbf{C}_q$ asociadas a las restricciones de usuario.

Para calcular $\mathbf{b}$ en un instante t la función *EcMovimiento_Baumgarte.m* deberá de determinar los vectores $\mathbf{C}$, $\mathbf{C}_t$, $\mathbf{C}_{tt}$, $\mathbf{Q}$ y la matriz $\mathbf{C}_{qt}$. Para determinar todos los términos citados se han programado las siguientes funciones:

- La función *Restricciones.m* calcula el vector $\mathbf{C}$ a partir del instante t de la simulación y el vector $\mathbf{q}$. La función *Restricciones.m* llama a la función *Restrus.m* para determinar las componentes de $\mathbf{C}$ asociadas a las restricciones de usuario.
- Las funciones *DtRestricciones.m* y *DtdtRestricciones.m* calculan los vectores $\mathbf{C}_t$ y $\mathbf{C}_{tt}$. La función *DtRestricciones.m* tiene como entradas el instante t de la simulación y el vector $\mathbf{q}$. Por otro lado, la función *DtdtRestricciones.m* recibe como entradas el instante t de la simulación y los vectores $\mathbf{q}$ y $\dot{\mathbf{q}}$.
- Para calcular la matriz $\mathbf{C}_{qt}$ se ha programado la función *DtJacobiana.m*, que recibirá como entradas el instante t de la simulación dinámica y los vectores $\mathbf{q}$ y $\dot{\mathbf{q}}$.

Para calcular el vector $\mathbf{Q}$ se ha programado una función para cada uno de los vectores de fuerzas generalizadas, asociados a las fuerzas presentes en el modelo multicuerpo, que se precisan para obtener $\mathbf{Q}$:

- La función *Qgravedad.m* determina el vector $\mathbf{Q}_g$ a partir del vector $\mathbf{q}$.
- La función *QUusuario.m* calcula el vector $\mathbf{Q}_{us}$ a partir del instante t de la simulación y los vectores $\mathbf{q}$ y $\dot{\mathbf{q}}$.
- La función *Qma.m* calcula el vector $\mathbf{Q}_{ma}$ a partir de los vectores $\mathbf{q}$ y $\dot{\mathbf{q}}$.
- La función *Qcontacto.m* determina el vector $\mathbf{Q}_{ce}$ a partir de los vectores $\mathbf{q}$ y $\dot{\mathbf{q}}$. El vector $\mathbf{Q}_{ce}$ será la suma de los vectores de fuerzas generalizadas asociados a las fuerzas normales y tangenciales de contacto.

La función *Qcontacto.m* tiene implementada toda la formulación del contacto elástico rueda-terreno. Deberá de determinar en cualquier instante t , para poder calcular $\mathbf{Q}_{ce}$, las coordenadas de superficie s_i y s_j que definirán a los puntos de contacto P_i y P_j .

Para resolver las ecuaciones del contacto elástico se ha empleado la función *fsolve.m* de *Matlab*. Dicha función emplea el método de *Newton-Raphson* para proporcionar las coordenadas curvilíneas s_i y s_j . Las entradas que precisa la función *fsolve.m* son unas coordenadas curvilíneas iniciales s_{i0} y s_{j0} , y una función desde la que poder evaluar, a partir de unas coordenadas cualesquiera s_i y s_j , las ecuaciones del contacto elástico.

La función que precisa la función *fsolve.m* para evaluar las ecuaciones del contacto elástico ha sido programada con el nombre *Evalua_sistemanl_contacto*.

La función *Evalua_sistemanl_contacto* recibirá unas coordenadas curvilíneas s_i y s_j y proporcionará un vector. El vector citado proporcionará como salida el resultado de evaluar las ecuaciones definidas por las expresiones (143) y (144) con las coordenadas curvilíneas s_i y s_j que recibió como entrada. Por ende, cuando el vector de salida sea un vector nulo el modelo podrá validar que los puntos de contacto P_i y P_j , definidos por s_i y s_j , han contactado físicamente en el instante t .

Se puede ajustar y configurar el modo en que la función *fsolve.m* resuelve las ecuaciones del contacto elástico mediante opciones de optimización que determinarán aspectos como tolerancias, criterios de parada o número de iteraciones permitidas.

Una vez que la función *EcMovimiento_Baumgarte.m* ha definido todos los términos de la matriz $\mathbf{A}$ y del vector $\mathbf{b}$ determinará, por el método de inversión de matrices, el vector de incógnitas $\mathbf{x}$.

El vector de incógnitas $\mathbf{x}$ definirá los vectores $\ddot{\mathbf{q}}$ y λ .

Habiendo determinado el vector $\ddot{\mathbf{q}}$ y teniendo el vector $\dot{\mathbf{q}}$ del vector de estado $\mathbf{y}$ en el instante t , resultado de integrar numéricamente en el instante $t - \Delta t$ el vector $\dot{\mathbf{y}}$, la función *EcMovimiento_Baumgarte.m* podrá definir el vector $\dot{\mathbf{y}}$ en el instante t .

A partir del vector λ la función *EcMovimiento_Baumgarte.m* podrá calcular el vector de fuerzas generalizadas de reacción $\mathbf{Q}_{FR}$ por medio de la expresión (120)

6.3-Funciones auxiliares

La función *EcMovimiento_Baumgarte.m* hará uso de una serie de funciones auxiliares que aplican conceptos teóricos vistos en el marco teórico del presente TFG. A continuación se describen dichas funciones:

- La función *Coor_global.m* determina el vector $\mathbf{r}_p^i$ de cualquier partícula P de un sólido i del sistema multicuerpo a partir de su vector de coordenadas generalizadas $\mathbf{q}^i$. La función *Coor_global.m* tiene implementadas las ecuaciones de posición.
- Las funciones *Matriz_transformacion.m* y *Deriva_MatRot.m* calculan las matrices $\mathbf{A}^i$ y $\mathbf{A}_0^i$ de un sólido i del modelo multicuerpo a partir de su vector de coordenadas generalizadas $\mathbf{q}^i$.
- La función *Parametriza_terreno.m*, a partir de la geometría definida para el terreno en *Definicion_Modelo.m*, determina la altura del terreno y los vectores $\bar{\mathbf{u}}_{P_i}^i$, $\mathbf{t}^i$ y $\mathbf{n}^i$ en cualquier punto P^i del mismo a partir de la coordenada curvilínea s_i que define al punto P^i . La función *Parametriza_terreno.m* contiene toda la formulación desarrollada en el marco teórico para incluir pendientes, baches e irregulares en el terreno.
- La función *Parametriza_Rueda.m* calcula los vectores $\bar{\mathbf{u}}_{P_j}^j$, $\mathbf{t}^j$ y $\mathbf{n}^j$ en cualquier punto P_j a partir de la coordenada s_j que define al punto P_j .

6.4-Programa que realiza las simulaciones dinámicas

El programa *Simula.m* definirá todas las condiciones del proceso de integración numérica y obtendrá la solución numérica o discreta de la simulación realizada, el vector $\mathbf{y}$ en ciertos instantes t del intervalo de simulación, pudiendo así simular el movimiento del modelo multicuerpo durante la simulación dinámica.

El programa *Simula.m* cargará los valores de todas las variables y parámetros definidos en la función *Definicion_Modelo.m*, definirá las condiciones del proceso de integración numérica y calculará el vector $\mathbf{y}$ en ciertos instantes t del intervalo de la simulación dinámica haciendo uso de la función *EcMovimiento_Baumgarte.m* y de las condiciones iniciales del sistema multicuerpo definidas en *Definicion_Modelo.m*, definidas por los vectores $\mathbf{q}_0$ y $\dot{\mathbf{q}}_0$.

6.4.1-Integración numérica

En primera instancia se deberá de definir en *Simula.m* la duración de la simulación dinámica y establecer en qué instantes, dentro del intervalo de integración numérica, se desea calcular

el vector $\mathbf{y}$. Todo ello se definirá mediante un vector de tiempos $\mathbf{t}$. En este punto existen dos formas de definir el intervalo de integración numérica:

- Si se define el vector de tiempos $\mathbf{t}$ únicamente mediante el instante inicial t_0 y el instante final t_f de la simulación dinámica, $\mathbf{t} = [t_0 \ t_f]$, el integrador numérico proporcionará la solución numérica del vector $\mathbf{y}$ en función del paso de integración numérica Δt que haya empleado a lo largo de la simulación dinámica. Por consiguiente, definiendo así el vector de tiempos $\mathbf{t}$ no se conocerá con antelación en que instantes t se dispondrá del vector $\mathbf{y}$.
- La otra alternativa consiste en definir un vector de tiempos $\mathbf{t}$ definido por diversos instantes de tiempo t contenidos en el intervalo de integración numérica. De este modo, el integrador numérico empleará el paso de integración que tenga que emplear pero evaluará, mediante interpolación, la solución en los tiempos de consulta especificados en el vector $\mathbf{t}$. De esta forma se sabrá en qué instantes de tiempo t se tendrá el vector de estado $\mathbf{y}$.

6.4.2-Integrador numérico

En el programa *Simula.m* se definirá qué integrador numérico emplear en cada simulación. Independientemente del integrador numérico utilizado, se deberá de proporcionar una solución inicial, definida por el vector $\mathbf{y}_0$, el vector de tiempos $\mathbf{t}$ y una función desde la que poder determinar el vector $\dot{\mathbf{y}}$ a partir del instante t y el vector $\mathbf{y}$ en el instante t . La solución inicial $\mathbf{y}_0$ la tomará el modelo de la función *Definicion_Modelo.m* y la función que *Simula.m* usará para calcular el vector $\dot{\mathbf{y}}$ será la función *EcMovimiento_Baumgarte.m*.

Debido a que el vector $\mathbf{y}$ depende únicamente del instante t de simulación, se podrá hacer uso en el presente *TFG* de los *ode solvers* del software *Matlab* (integradores numéricos de ecuaciones diferenciales ordinarias).

En el presente trabajo se ha empleado principalmente el *ode45*, recomendado como primera opción y principalmente enfocado para problemas poco rígidos, y el *ode15s*, recomendado para problemas rígidos. Debido a la rigidez del modelo, el *ode15s* ha proporcionado un mayor rendimiento computacional para simular el comportamiento del vehículo.

Por otro lado, para integrar numéricamente en las simulaciones realizadas con el modelo se ha programado el método de *Runge-Kutta* explícito de cuarto orden y paso de integración constante:

$$\mathbf{y}_{t+\Delta t} = \mathbf{y}_t + \frac{1}{6}(\mathbf{f}_1 + 2\mathbf{f}_2 + 2\mathbf{f}_3 + \mathbf{f}_4) \quad (253)$$

Desarrollando los términos de la expresión (253):

$$\mathbf{f}_1 = \Delta t \dot{\mathbf{y}}(\mathbf{y}_t, t) \quad (254)$$

$$\mathbf{f}_2 = \Delta t \dot{\mathbf{y}}\left(\mathbf{y}_t + \frac{1}{2} \mathbf{f}_1, t + \frac{\Delta t}{2}\right) \quad (255)$$

$$\mathbf{f}_3 = \Delta t \dot{\mathbf{y}}\left(\mathbf{y}_t + \frac{1}{2} \mathbf{f}_2, t + \frac{\Delta t}{2}\right) \quad (256)$$

$$\mathbf{f}_2 = \Delta t \dot{\mathbf{y}}(\mathbf{y}_t + \mathbf{f}_3, t + \Delta t) \quad (257)$$

Para implementar el método de *Runge-Kutta* se ha programado la función *Runge_kutta.m*. Si quisiera ver cómo se ha programado la función *Runge_kutta.m* acuda a los Anexos del trabajo.

6.5- Función para el posproceso de resultados

El programa *Simula.m*, tras completar el proceso cálculo, solamente proporciona los vectores $\mathbf{q}$ y $\dot{\mathbf{q}}$ en ciertos instantes t del intervalo de integración numérica. Por ende, no se tiene registrado el valor de ninguna variable del modelo multicuerpo que puede ser de interés para analizar los resultados obtenidos en cualquier simulación.

Queda de relieve la necesidad de programar una función de posproceso de resultados. La función de posprocesado permitirá, además, poder graficar los valores de las variables posprocesadas y analizar en mayor profundidad las simulaciones.

La función *PProceso.m* tendrá por fin posprocesar los vales de cualquier variable o parámetro del modelo multicuerpo a lo largo de los instantes t definidos por el vector de tiempos $\mathbf{t}$. Las entradas que tomará la función *PProceso.m* serán el vector de tiempos $\mathbf{t}$ y el vector de estado $\mathbf{y}$ en los instantes t definidos en el vector de tiempos $\mathbf{t}$. Las salidas de la función *PProceso.m* serán cualquier conjunto de variables que se deseen posprocesar, obteniéndose las mismas en los instantes t del vector de tiempos $\mathbf{t}$. Parece razonable posprocesar en *PProceso.m* la mayoría de variables del modelo multicuerpo y posteriormente, dependiendo del tipo de simulación dinámica, discernir qué variables, de todas las posprocesadas por *PProceso.m*, se quieren analizar. La función *PProceso.m* llamará a las mismas funciones que llama la función *EcMovimiento_Baumgarte.m*.

Carece de sentido alguno posprocesar en *PProceso.m* las variables del modelo que no cambiarán de valor durante el paso de cualquier simulación dinámica pues se podrán calcular introduciendo el valor de las entradas que precisen en cualquier instante t de la simulación dinámica. Ejemplos de estas variables son el vector $\mathbf{Q}_{us}$, la matriz $\mathbf{M}$ o el vector $\mathbf{Q}_g$.

Las variables posprocesadas en la función *PProceso.m* pueden ser resumidas del siguiente modo:

- Se han posprocesado todas las variables que la función *EcMovimiento_Baumgarte.m* calcula en cada instante t .
- En lo que se refiere a los muelles-amortiguadores del modelo multicuerpo, a diferencia de la función *Obten_derivada_Baumgarte.m* que sólo tiene por variable de salida el vector $\mathbf{Q}_{ma}$, la función *PProceso.m* posprocesa en cada instante t la longitud del muelle l , la velocidad a la que el muelle cambia su longitud $\dot{l}$, componente elástica y amortiguadora de la fuerza f_{ma} , el vector $\mathbf{r}_p^{ij}$ que define la longitud y orientación del muelle, las componentes del vector de la fuerza f_{ma} , definido por $\mathbf{Q}_{ma}$, y el momento que ejerce la fuerza f_{ma} sobre cada sólido que sustenta el muelle-amortiguador.
- En lo referido a los contactos elásticos rueda-terreno del modelo multicuerpo, la función *PProceso.m* posprocesa, para cada contacto elástico, un total de 33 variables en cada instante t . Ejemplos de las variables posprocesadas por *PProceso.m* son las coordenadas curvilíneas s_i y s_j de los puntos P_i y P_j que han contactado en el instante t , la indentación δ , la velocidad de indentación $\dot{\delta}$, la velocidad de deslizamiento v_{desl} , la fuerza normal F_{nc} y tangencial F_{tc} de contacto... Puede ver en profundidad las variables del contacto elástico posprocesadas en *PProceso.m* en los Anexos del trabajo.

Si se define únicamente el vector de tiempos $\mathbf{t}$ mediante el instante inicial t_0 y el instante final t_f puede ser que no se disponga de un posproceso de resultados equiespaciado en el tiempo, como consecuencia de no tener el vector $\mathbf{y}$ equiespaciado en el intervalo de integración numérica (dependerá del paso de integración que haya empleado el integrador numérico). Por consiguiente, puede ser conveniente y necesario realizar un remuestreo o *resampling* de la solución numérica dada por el integrador numérico.

Para realizar el remuestreo será necesario en primera instancia definir un vector de tiempos $\mathbf{t}_r$ que defina los instantes t dentro del intervalo de integración numérica en los que se desea tener los valores de las variables posprocesadas por *PProceso.m*.

Las variables posprocesadas se obtendrán en los instantes t de consulta, definidos por el vector $\mathbf{t}_r$, mediante la interpolación de los valores de las variables posprocesadas por *PProceso.m* en los instantes t definidos por el vector de tiempos $\mathbf{t}$.

Para interpolar los valores de las variables posprocesadas se ha empleado la función del software *Matlab interp1.m*. La función *interp1.m* necesitará tomar como entradas el vector de tiempos original $\mathbf{t}$, el vector $\mathbf{t}_r$ que define los instantes t de la nueva solución, y los vectores solución originales de las variables posprocesadas por la función *PProceso.m*.

6.6- Funciones para la representación de resultados

La representación de las variables posprocesadas ayudará a comprender y analizar qué ha ocurrido a lo largo de cualquier simulación dinámica y determinar si los resultados obtenidos son coherentes o no.

Puede llegar a ser muy tedioso y repetitivo tener que, tras cada simulación, escribir líneas de código para representar cualquier par de variables posprocesadas y adjuntar títulos, leyendas o unidades a dichas representaciones. La programación de funciones para representar cualquier par de variables posprocesadas solventará dicha situación. Además, agilizará considerablemente el estudio y análisis de los resultados obtenidos en cualquier simulación.

Las funciones de representación de resultados tomarán como entradas las variables posprocesadas por la función *PProceso.m*. Por ende, si no se hubiese programado la función *PProceso.m* no se podría representar ninguna variable o parámetro del modelo multicuerpo.

Se citan a continuación las funciones programadas para la representación de resultados:

- Las funciones *plotposiciones.m* y *plotvelocidades.m* representan las componentes de los vectores $\mathbf{q}^i$ y $\dot{\mathbf{q}}^i$ de cada sólido del sistema multicuerpo a partir del vector de tiempos $\mathbf{t}$ de la simulación y los vectores $\mathbf{q}$ y $\dot{\mathbf{q}}$.
- La función *plotmuelles.m* grafican la longitud l de cada muelle del modelo multicuerpo en cada instante t de la simulación a partir del vector de tiempos $\mathbf{t}$ de la simulación y el vector *PPma* posprocesado por la función *PProceso.m*.
- La función *plotindentaciones.m* grafica el valor de la indentación δ en el terreno de cada rueda del modelo multicuerpo a partir del vector de tiempos $\mathbf{t}$, el vector $\mathbf{q}$ y el vector *PPcont* posprocesado por la función *PProceso.m*.
- La función *plotruedas.m* representa los puntos de contacto determinados por el modelo en cada rueda del sistema multicuerpo a partir del vector $\mathbf{q}$ y el vector *PPcont* posprocesado por la función *PProceso.m*.
- La función *plotdeslizamiento.m* analiza al detalle todas las velocidades que definen el proceso de deslizamiento en cada rueda del modelo. La función *plotdeslizamiento.m* ayudará a analizar y comprender la dinámica longitudinal del modelo. Tomará como entradas los vectores *PPcont* y $\dot{\mathbf{q}}$.
- Las funciones *plotfuerzashorizontales.m* , *plotfuerzasverticales.m* tienen por objeto analizar las componentes horizontales y verticales de cada una de las fuerzas que actúan sobre cada sólido del sistema multicuerpo y los momentos que ejercen las mismas respecto al centro de gravedad de cada sólido del modelo multicuerpo. Dichas funciones tomarán como entradas los vectores de fuerzas generalizadas calculados por la función *EcMovimiento_Baumgarte.m*, y posprocesados por la función *PProceso.m*, y el vector de tiempos $\mathbf{t}$.

Lógicamente, las funciones descritas serán un apoyo para el análisis y estudio de los resultados obtenidos en cada simulación dinámica. Sin embargo, no quiere decir que todas las gráficas adjuntadas en el presente *TFG* hayan sido obtenidas por las funciones descritas pues a veces puede ser interesante modificar ligeramente las gráficas resultantes de las funciones citadas.

6.7- Función para crear una animación de las simulaciones dinámicas

La función *Anima.m* tiene por objeto proyectar una animación del movimiento de cada uno de los sólidos del sistema multicuerpo en cada simulación dinámica.

La visualización de la animación es la manera más ilustrativa e inmediata de verificar si ha habido errores durante la simulación. Además, enriquece el trabajo pues se puede apreciar perfectamente el movimiento del modelo multicuerpo y la geometría del terreno.

Las entradas que precisa la función *Anima.m* son un vector de tiempos que defina los instantes t en los que se desea proyectar la configuración del sistema multicuerpo, y el vector $\mathbf{q}$ en cada instante t definido por el vector de tiempos. El vector de tiempos podría ser, por ejemplo, el vector $\mathbf{t}$.

Con el vector $\mathbf{q}$, que define los vectores $\mathbf{q}^i$ de cada sólido del sistema multicuerpo, se podrá proyectar la posición global de cualquier punto P que pertenezca a un sólido del modelo multicuerpo. Para ello, será necesario definir un conjunto finito de puntos P que delimiten la geometría de cada sólido del modelo multicuerpo.

Los puntos P serán definidos mediante sus posiciones locales, definidas por los vectores $\bar{\mathbf{u}}_P^i$. Posteriormente, la función *Coor_global.m* calculará las posiciones globales de los puntos P determinando los vectores de posición $\mathbf{r}_P^i$.

Como es lógico, en el presente documento no se insertarán las animaciones de las simulaciones estudiadas. En la presentación del *TFG* se podrán apreciar al detalle las animaciones de cada simulación.

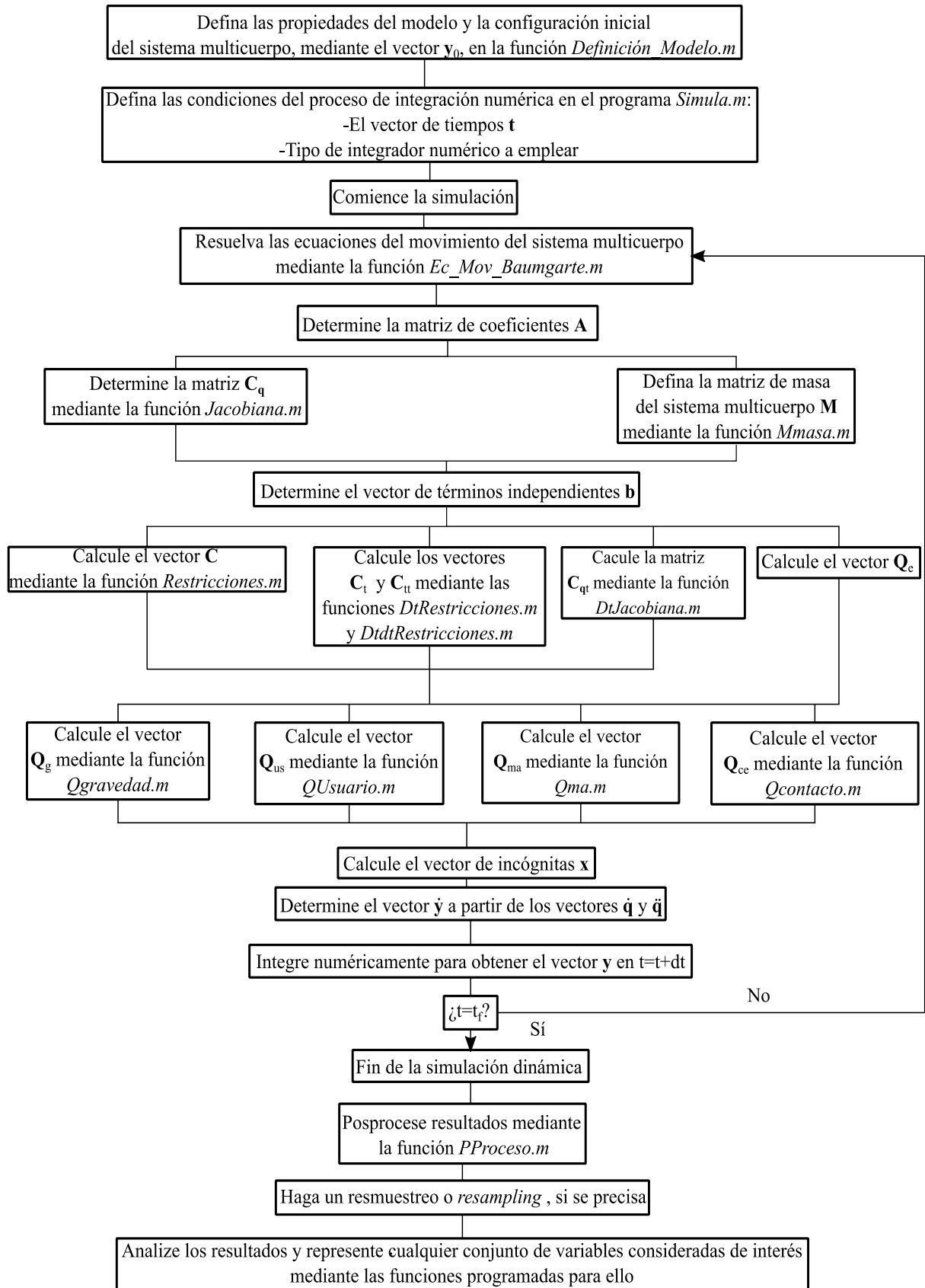


Figura 20. Metodología de cálculo a seguir en el software Matlab en cualquier simulación dinámica realizada con el modelo multicuerpo

7-ENSAYOS

Se han realizado un total de cinco ensayos con el sistema multicuerpo modelado para el presente *TFG*:

- El primer ensayo consiste en una simulación estática donde se pretende que el sistema multicuerpo alcance una configuración de reposo. Dicha configuración de equilibrio estática, definida por el vector $\mathbf{q}_0$ obtenido tras finalizar el ensayo, servirá de condición inicial para las posteriores simulaciones dinámicas, donde el vehículo y el remolque estarán animados de movimiento.
- En el segundo ensayo se ha modelado una irregularidad repentina en el terreno. De este modo, se podrá analizar la respuesta del vehículo y del remolque cuando contactan con el bache. Este ensayo dictará como de eficiente es la suspensión vertical modelada e incluirá la formulación descrita en el apartado 4.10.2.
- En el tercer ensayo se analizará la respuesta del sistema multicuerpo cuando el terreno está definido con irregularidades. En este ensayo se incluirá la formulación desarrollada en el apartado 4.10.3.
- El cuarto y último ensayo pretende estudiar la influencia del coeficiente de fricción μ en la capacidad de tracción o impulsión del vehículo cuando pretende superar un tramo con pendiente, siendo la dinámica longitudinal clave en este ensayo. En este último ensayo se aplicará la formulación del apartado 4.10.1.

7.1- Ensayo estático

Antes de realizar cualquier simulación dinámica donde el sistema multicuerpo esté dotado de movimiento, mediante una restricción de usuario de velocidad de giro o un par motor, se calculará el vector $\mathbf{q}$ cuando el sistema multicuerpo haya alcanzado o encontrado una configuración de equilibrio estática. El vector $\mathbf{q}$ obtenido será el vector $\mathbf{q}_0$ para las restantes simulaciones dinámicas donde el vehículo y el remolque avanzarán por el terreno.

Además, el presente ensayo servirá para comprobar si son óptimos o no los valores de los parámetros que modelan el comportamiento de los muelles-amortiguadores del sistema multicuerpo y de los contactos elásticos rueda-terreno.

La simulación estática tendrá por configuración inicial el vector $\mathbf{q}_0$ definido a partir de la geometría del modelo multicuerpo, el cual fue definido y desarrollado en el apartado 5.8.

Se debe de establecer un tiempo de simulación suficiente para que, tras finalizar la simulación, el sistema multicuerpo no siga oscilando en busca de una posición de reposo. En base a diversas simulaciones estáticas realizadas, se ha establecido un tiempo de 8 segundos para la simulación. El tiempo que tarde el sistema multicuerpo en encontrar una posición de equilibrio está altamente influenciado por los valores de los parámetros que

modelan a los muelles-amortiguadores y a los contactos elásticos rueda-terreno, de aquí el hecho de que este ensayo sirva para analizar las magnitudes de los mismos.

Este ensayo se focalizará en analizar la dinámica vertical del modelo, recuerde que los puntos de unión de los muelles longitudinales del modelo fueron definidos de manera que no indujeran oscilaciones horizontales bajo condiciones estáticas. Por ende, se espera que los muelles-amortiguadores longitudinales no varíen su longitud durante el ensayo estático.

Las oscilaciones verticales del sistema se analizarán estudiando la magnitud de la coordenada generalizada R_y^i de cada sólido del sistema multicuerpo en función del tiempo de simulación t .

En primera instancia se analizará el comportamiento oscilatorio vertical del chasis:

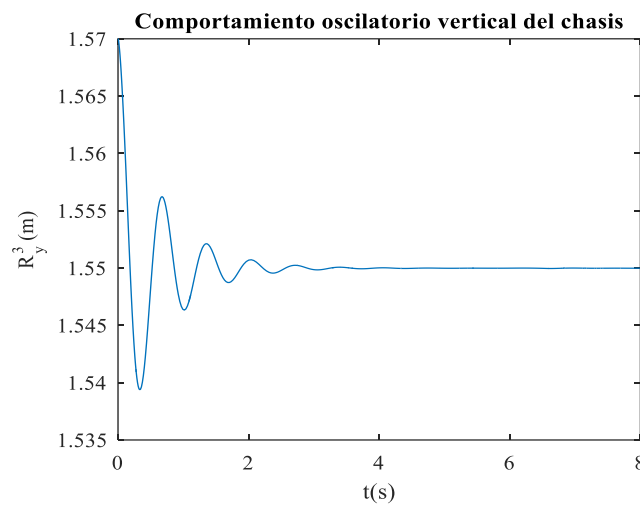


Figura 21. Oscilaciones verticales el chasis

Se corrobora, a la vista de la dependencia que presenta R_y^3 con t , que el chasis encuentra una posición reposo, pues R_y^3 se estabiliza en un valor constante.

El comportamiento oscilatorio vertical exhibido por el chasis es el propio de un sistema mecánico subamortiguado. Por consiguiente, se puede afirmar que los valores asignados a la rigidez vertical k_v y al coeficiente de amortiguación c_v de los muelles-amortiguadores verticales del modelo son óptimos pues el comportamiento oscilatorio del chasis está únicamente influenciado por los parámetros de la suspensión vertical del vehículo.

Es totalmente coherente que la magnitud final de la coordenada generalizada R_y^3 sea menor que la inicial pues las ruedas del modelo deberán de indentar en el terreno una cantidad finita para equilibrar las fuerzas que soportan.

Se prosigue analizando el comportamiento oscilatorio vertical de la rueda trasera y delantera del vehículo:

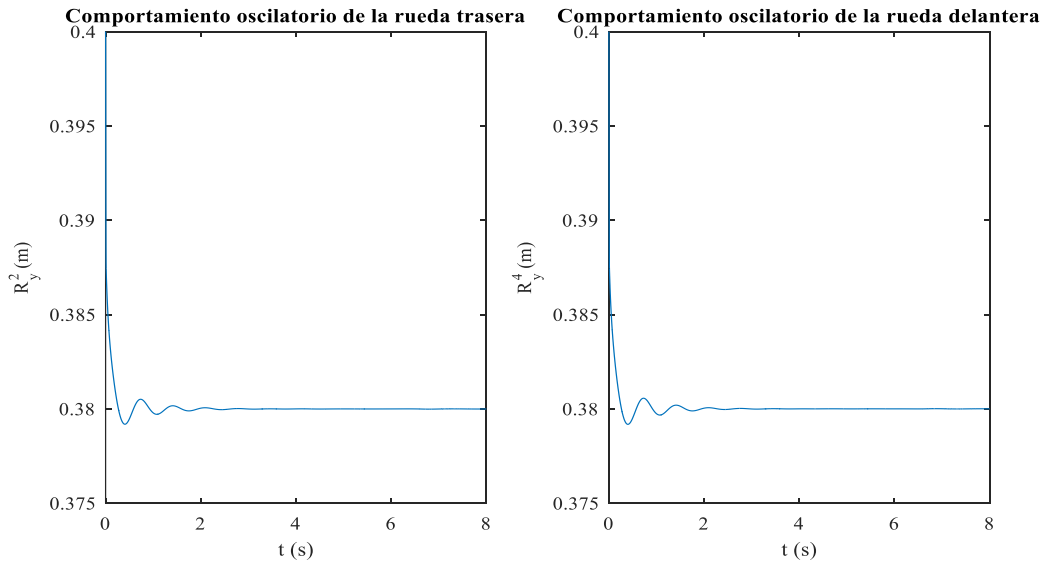


Figura 22. Oscilaciones verticales de las ruedas del vehículo

Los gráficos adjuntados denotan con claridad que tanto la rueda trasera como la rueda delantera del vehículo alcanzan posición estática, los valores de las coordenadas generalizadas R_y^2 y R_y^4 no varían alrededor de los 2 segundos de simulación.

El comportamiento oscilatorio vertical exhibido por ambas ruedas es idéntico, lo cual parece lógico por las siguientes razones:

- Presentan las mismas características geométricas e inerciales.
- Los parámetros que modelan el comportamiento de los muelles a los que se encuentran conectadas son idénticos.
- Los parámetros que modelan el contacto elástico de las mismas con el terreno son idénticos.
- Los puntos de conexión de la suspensión vertical propician que, bajo condiciones estáticas, soporten el mismo porcentaje de peso del chasis.
- Las coordenadas generalizadas iniciales R_{y0}^2 y R_{y0}^4 de ambas ruedas son idénticas.

Se aprecia perfectamente la indentación repentina de ambas ruedas en el terreno, como consecuencia del peso propio y de las fuerzas transmitidas por la suspensión vertical y horizontal del vehículo.

El comportamiento oscilatorio de ambas ruedas estará influenciado por los parámetros de la suspensión vertical y del contacto elástico rueda-terreno. Los resultados obtenidos muestran un comportamiento coherente, las ruedas del vehículo no se encuentran propiamente suspendidas como lo está el chasis. Por ello, parece razonable validar los valores de los parámetros que modelan el contacto elástico rueda-terreno y serán suficientes para poder realizar satisfactoriamente simulaciones dinámicas donde el vehículo y el remolque se desplacen por el terreno.

Finalmente, se analizará el comportamiento oscilatorio vertical del remolque y su rueda:

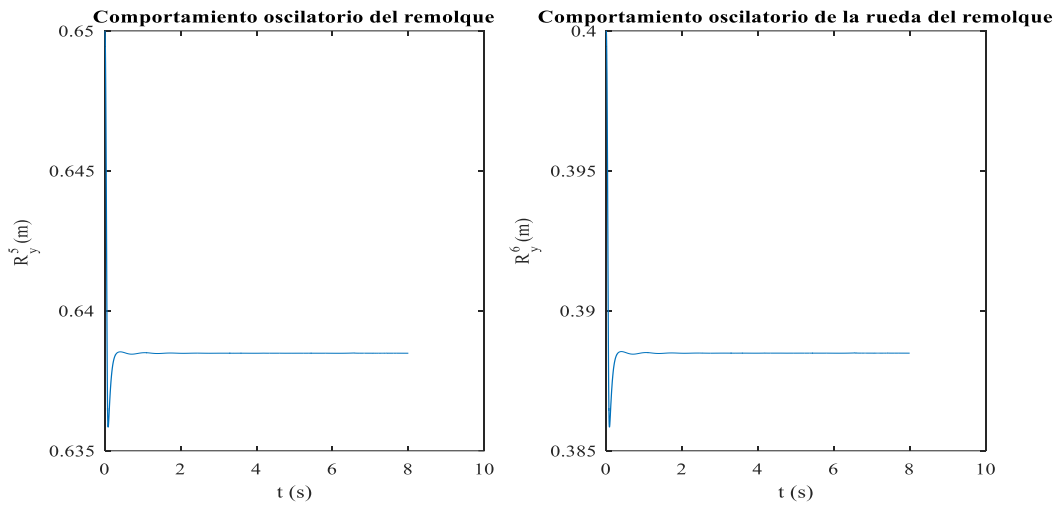


Figura 23. Oscilaciones verticales del remolque y su rueda asociada

Las evoluciones de los valores de las coordenadas generalizadas R_y^5 y R_y^6 indican claramente que tanto el remolque como su rueda asociada han alcanzado una configuración de equilibrio estático. Además, el comportamiento oscilatorio vertical de ambos sólidos es idéntico.

La razón de que oscilen verticalmente de manera idéntica se debe a la unión articulada que conecta ambos sólidos. La unión articulada propicia que el movimiento vertical de un sólido arrastre al otro a moverse de manera idéntica, pero no a girar de manera idéntica. Se aprecia perfectamente como la rueda del remolque al indentar o penetrar en el terreno obliga, mediante la unión articulada, al remolque a descender con ella.

Por otro lado, se aprecia perfectamente el hecho de que el remolque no presente suspensión. Como consecuencia, ambos sólidos han oscilado de un modo más brusco, se aprecia una amortiguación mayor de las oscilaciones, hasta alcanzar una configuración de reposo.

A continuación se estudiarán las longitudes l de los muelle-amortiguadores del modelo multicuerpo a lo largo de la simulación estática:

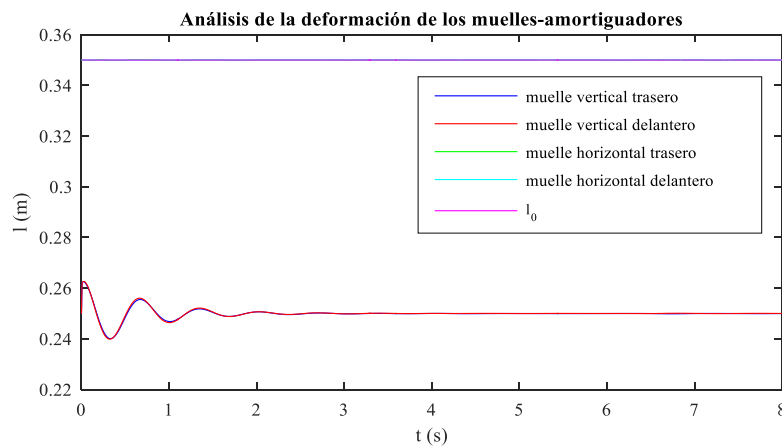


Figura 24. Variaciones de las longitudes de los muelles-amortiguadores del modelo

La figura adjuntada corrobora la hipótesis de que tal definición de la suspensión longitudinal provocaría que el sistema multicuerpo no oscilara horizontalmente, bajo condiciones de reposo, los muelles-amortiguadores longitudinales no han variado su longitud en ningún momento durante la simulación estática.

Por otro lado, las deformaciones de los muelles-amortiguadores verticales han sido idénticas. El hecho de que hayan variado su longitud con el tiempo de manera idéntica se debe a que tienen una rigidez vertical k_v idéntica, un coeficiente de amortiguación c_v idéntico y que soportan el mismo porcentaje de peso del chasis. Como es lógico, los muelles-amortiguadores verticales se encuentran comprimidos en todo momento durante la simulación estática pues el objeto de los mismos es sustentar al chasis.

Finalmente, se analizará la indentación δ en el terreno de cada rueda del modelo multicuerpo en función del tiempo de simulación t :

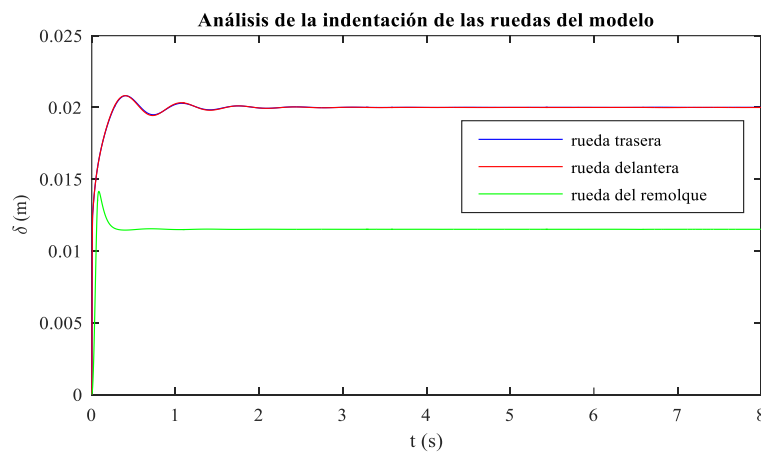


Figura 25. Indentación en el terreno de las ruedas del sistema multicuerpo

Se pueden sacar las siguientes conclusiones a partir de la figura anteriormente adjuntada:

- Se corrobora la suposición realizada acerca de que el vector de coordenadas generalizadas iniciales del sistema multicuerpo $\mathbf{q}_0$, definido a partir de la geometría del modelo, definiría una configuración del modelo multicuerpo donde las ruedas no presentarían indentación en el terreno alguna.
- El hecho de haber modelado los tres contactos elásticos con la misma constante *Hertziana* K_H y que el peso del conjunto remolque-rueda del remolque sea menor que el peso del vehículo derivaría en que la rueda del remolque tenga que indentar en el terreno una cantidad δ menor que las ruedas del vehículo.
- Las indentaciones finales δ de las ruedas del vehículo confirman el modo de calcular y definir las constantes *Hertzianas* del contacto elástico K_H , recuerde que fueron calculadas de modo que bajo condiciones estáticas las ruedas del vehículo tuvieran que indentar una cantidad de 2 centímetros en el terreno.

Los resultados del ensayo estático sugieren que, aunque el sistema multicuerpo haya oscilado verticalmente hasta alcanzar la configuración de equilibrio estático, la configuración inicial definida a partir de la geometría del modelo se encontraba muy cercana a la configuración de equilibrio estático (observe que las oscilaciones verticales son de pequeña magnitud).

El hecho de definir una configuración inicial del sistema multicuerpo muy cercana a la que adquiere el modelo cuando deja de oscilar verticalmente era de esperar pues en la configuración inicial de todas las ruedas se encontraban perfectamente tangentes al terreno. Por ende, sólo se requería la indentación de las ruedas del modelo multicuerpo en el terreno para que el sistema multicuerpo encontrase una configuración de equilibrio. Piense en las diferencias que se hubieran encontrado si se hubiese dejado caer al modelo multicuerpo desde una cierta altura.

Para concluir el ensayo estático, se mostrarán las componentes del vector $\Delta \mathbf{q}_0$ que resulta de restar el vector $\mathbf{q}_0$ definido a partir de la geometría del modelo multicuerpo y el vector $\mathbf{q}$ obtenido tras finalizar la simulación estática:

$$\Delta \mathbf{q}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0,0002 \\ 0,02 \\ 0 \\ 0,0002 \\ 0,02 \\ 0 \\ 0,0002 \\ 0,02 \\ -0,0007 \\ 0,0002 \\ 0,0115 \\ 0,0043 \\ 0,0013 \\ 0,0115 \\ -0,0034 \end{bmatrix} \quad (258)$$

Aunque no difieran mucho los vectores $\mathbf{q}_0$, las posteriores simulaciones dinámicas partirán de unas condiciones iniciales más óptimas.

7.2- Simulación del comportamiento del remolque ante una irregularidad repentina

El presente ensayo tiene por objeto analizar el comportamiento del remolque cuando impacta con una irregularidad repentina o bache. El interés de estudiar el comportamiento del remolque reside en el hecho de que no presenta suspensión, a diferencia del vehículo. Por lo tanto, la rueda del remolque será susceptible de perder el contacto con el terreno.

El terreno, perfectamente plano a excepción de la irregularidad repentina, contiene un bache modelado según la formulación desarrollada en el marco teórico del trabajo.

El ensayo pretende comprobar si el modelo multicuerpo es capaz de simular cómo la rueda del remolque pierde el contacto con el terreno en su paso por el bache, bajo unas condiciones que lo propicien.

Por otro lado, se espera que las ruedas del vehículo no pierdan el contacto con el terreno cuando impacten con la irregularidad repentina, el objeto de la suspensión del vehículo es que las ruedas se encuentren en todo momento en contacto con el terreno. Por ende, la suspensión vertical del vehículo debería de ser capaz de impedir que la irregularidad repentina levante a las ruedas del vehículo.

El vehículo se pondrá en movimiento mediante una restricción de velocidad de giro constante en la rueda trasera del vehículo, simulando de este modo que el vehículo tiene tracción trasera.

Fijando la velocidad de giro de la rueda tractora, la masa del remolque será crucial para que la rueda del remolque pueda perder el contacto con el terreno. Como consecuencia, parece interesante modificar la masa del remolque m^5 para analizar la influencia de la misma en la respuesta del remolque.

Para analizar la respuesta del modelo multicuerpo ante la irregularidad repentina se realizarán un total de tres simulaciones dinámicas:

- Las dos primeras simulaciones analizarán la influencia de la masa del remolque m^5 en la respuesta del mismo al bache.
- La última simulación dinámica pretende analizar la respuesta del remolque ante una irregularidad más abrupta y una velocidad de giro de la rueda trasera $\dot{\theta}^2$ mayor que la definida para las dos primeras simulaciones.

Para las dos primeras simulaciones, la velocidad de giro de la rueda trasera $\dot{\theta}^2$ tendrá un valor constante de 15 rd/s.

En la primera simulación dinámica el remolque tendrá una masa m^5 de 2000 kg mientras que en la segunda simulación el remolque tendrá una masa m^5 de 700 kg. Se espera que al reducir la masa del remolque la rueda del remolque pueda perder el contacto con el terreno cuando contacta con la irregularidad repentina, el peso del conjunto rueda-remolque podría no ser suficiente para equilibrar la fuerza repentina de contacto ejercida por el bache sobre la rueda del remolque.

La irregularidad repentina modelada para las dos primeras simulaciones dinámicas está definida por los valores de los siguientes parámetros, recuerde que se tratan de los parámetros que definen la geometría del bache:

x_1	x_2	x_3	x_4	x_5	m_1	m_2
10 m	10,2 m	10,4 m	10,6 m	10,8 m	10%	-10 %

Tabla 7.1. Parámetros que modelan la primera irregularidad repentina

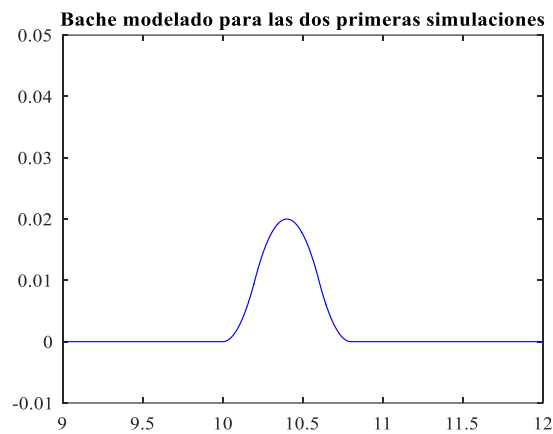


Figura 26. Irregularidad repentina modelada para las dos primeras simulaciones dinámicas

El parámetro del modelo multicuerpo que nos permitirá saber si las ruedas del sistema están en contacto con el terreno en cualquier instante t de la simulación será la indentación δ de cada rueda en el terreno. Posprocesando la indentación δ de cada rueda del modelo multicuerpo, mediante la función *PProceso.m*, se podrá determinar de manera inmediata si el bache ha propiciado que alguna rueda del modelo multicuerpo haya perdido el contacto con el terreno en algún instante.

Se representará la indentación δ de cada rueda frente a su coordenada generalizada R_x^i en vez de frente al instante t de simulación dinámica. De este modo, será sumamente sencillo saber cuándo las ruedas del modelo multicuerpo han impactado con la irregularidad repentina, el valor del parámetro x_1 fija el comienzo del bache en el terreno. Si se hubiese graficado la indentación δ frente al tiempo t no se tendría referencia alguna de cuando las ruedas del sistema multicuerpo han contactado con la irregularidad repentina.

Las siguientes figuras muestran la evolución de la indentación δ de cada rueda frente a su coordenada generalizada R_x^i cuando el remolque tiene una masa m^5 de 2000 kg y cuando tiene una masa m^5 de 700 kg:

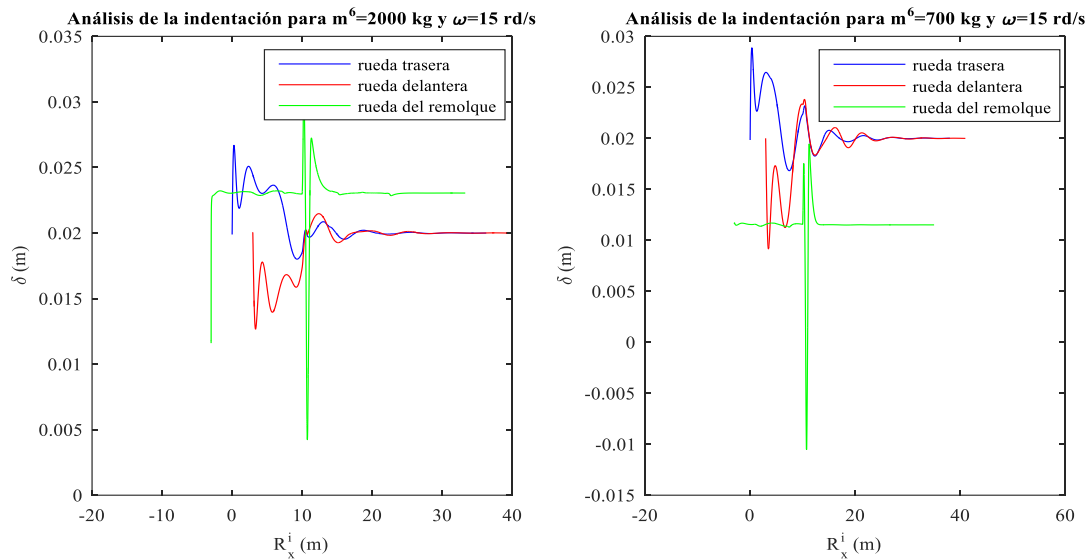


Figura 27. Indentación en el terreno de las ruedas del modelo multicuerpo en las dos primeras simulaciones dinámicas

Los resultados obtenidos reflejan la influencia de la masa m^5 del remolque en su respuesta al bache. Se pueden sacar las siguientes conclusiones de las figuras obtenidas:

- En primer lugar, las ruedas del vehículo no pierden el contacto con el terreno en ningún momento en ambas simulaciones, independientemente de masa m^5 que tenga el remolque. Por ende, la suspensión vertical del vehículo cumple su cometido, que las ruedas del vehículo sigan el perfil del terreno en todo momento.
- Cuando el remolque tiene una masa m^5 de 2000 kg la rueda del remolque no pierde el contacto con el terreno cuando interactúa con el bache. Prueba de ello es que en todo momento la indentación δ de la rueda del remolque es mayor que cero. La fuerza de contacto que ejerce la irregularidad repentina sobre la rueda del remolque no es suficiente para superar el peso del conjunto rueda-remolque.
- Cuando el remolque tiene una masa m^5 de 700 kg, se puede apreciar como la rueda del remolque pierde el contacto con el terreno al contactar con el bache. Prueba de ello es que cuando la coordenada generalizada R_x^6 de la rueda del remolque alcanza un valor de 10 metros la indentación δ toma un valor negativo.

Precisamente, el parámetro x_1 que modela la irregularidad repentina tiene un valor de 10 m para las simulaciones dinámicas analizadas y representa el comienzo del bache en el terreno.

Recuerde, de la formulación del contacto elástico rueda-terreno, que la indentación δ toma valores negativos cuando la rueda y el terreno no han contacto físicamente.

Por todas las conclusiones citadas, se puede concluir que los resultados obtenidos en estas dos primeras simulaciones dinámicas o denotan que el modelo ha sido capaz de reproducir el comportamiento esperado y deseado.

En la tercera simulación dinámica se pretende simular una situación más extrema, obteniendo así una animación más ilustrativa de la interacción rueda-bache. Las anteriores simulaciones trataban de buscar unas condiciones que permitiesen que la rueda del remolque pudiera permanecer en contacto modificando la masa m^5 del remolque.

Para que la rueda del remolque pierda el contacto con el terreno más tiempo se han tomado tres medidas:

- Reducir la masa m^5 del remolque hasta 100 kg.
- Aumentar la velocidad de giro de la rueda trasera del vehículo $\dot{\theta}^2$ hasta 30 rd/s. Con el mismo paso de integración numérica, la rueda del remolque penetrará más en el bache, lo que derivará en una fuerza de contacto mayor.
- El bache ha sido modelado con una geometría más abrupta. Para ello, se han aumentado las pendientes de las funciones matemáticas que modelan su aspecto, definidas por los parámetros m_1 y m_2 .

Por todo ello, se espera que la rueda del remolque se encuentre suspendida en el aire un mayor tiempo.

La siguiente tabla define los valores de los parámetros que definen la geometría de la irregularidad repentina en esta tercera simulación dinámica:

x_1	x_2	x_3	x_4	x_5	m_1	m_2
10 m	10,2 m	10,4 m	10,6 m	10,8 m	22%	-22 %

Tabla 7.2. Parámetros que modelan la segunda irregularidad repentina

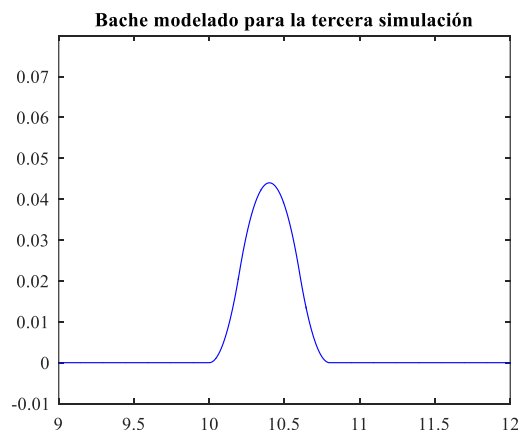


Figura 28. Irregularidad repentina modelada para la tercera simulación dinámica

Se procede seguidamente a analizar el valor de la indentación δ de cada rueda del modelo multicuerpo frente a su coordenada generalizada R_x^i :

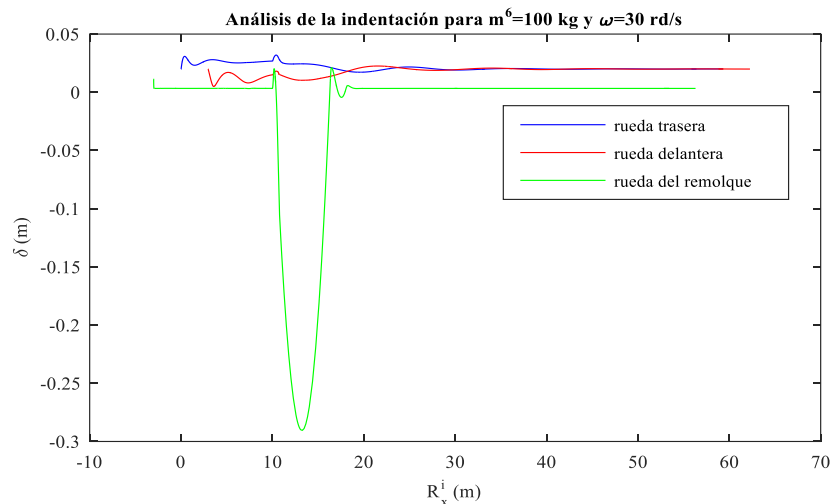


Figura 29. Indentación en el terreno de las ruedas del sistema multicuerpo en la tercera simulación dinámica

La figura obtenida muestra en primera instancia que, de igual modo que en las dos primeras simulaciones dinámicas, las ruedas del vehículo no pierden el contacto con el terreno en ningún momento. Por consiguiente, esta simulación, mucho más exigente que las anteriores, valida una vez más la eficiencia de la suspensión vertical del vehículo.

Por otro lado, se puede apreciar como la indentación de la rueda del remolque δ se mantiene negativa en un rango mayor de valores de la coordenada generalizada R_x^i . Por ende, la rueda del remolque ha recorrido una mayor distancia sobre el terreno sin contactar con él.

Las animaciones de las simulaciones dinámicas del presente ensayo son muy llamativas e ilustrativas del potencial del modelo para simular la realidad.

7.3- Comparativa del comportamiento del modelo multicuerpo en un terreno perfectamente plano y en un terreno con irregularidades

En el presente ensayo se analizará y estudiará el comportamiento del vehículo y del remolque cuando avanzan por un terreno totalmente llano y cuando avanzan por un terreno que tiene irregularidades, modeladas mediante una suma armónicos o funciones sinusoidales. Los restos de parámetros que definen al modelo multicuerpo serán idénticos.

Se activará la dinámica longitudinal del modelo introduciendo un par motor M en la rueda delantera del vehículo, simulando así que el coche presenta tracción delantera. El par motor M tendrá una magnitud de 350 Nm.

Las irregularidades han sido modeladas mediante una función matemática resultado de sumar cinco funciones sinusoidales o armónicos. La siguiente tabla define los parámetros que modelan a cada función sinusoidal o armónico:

	A (Amplitud)	φ (Desfase)	λ (Longitud de onda)
Primer armónico	$0,0075 \text{ m}$	$0,18\pi \text{ rd}$	4 m
Segundo armónico	$0,0037 \text{ m}$	$\pi \text{ rd}$	10 m
Tercer armónico	$0,015 \text{ m}$	$1,46\pi \text{ rd}$	$1,2 \text{ m}$
Cuarto armónico	$0,0113 \text{ m}$	$2\pi \text{ rd}$	8 m
Quinto armónico	$0,0118 \text{ m}$	$2,5\pi \text{ rd}$	12 m

Tabla 7.3. Parámetros que modelan las irregularidades

Las siguientes figuras muestran la función y que define a cada armónico y la función matemática y resultante de sumar las funciones matemáticas de los cinco armónicos:

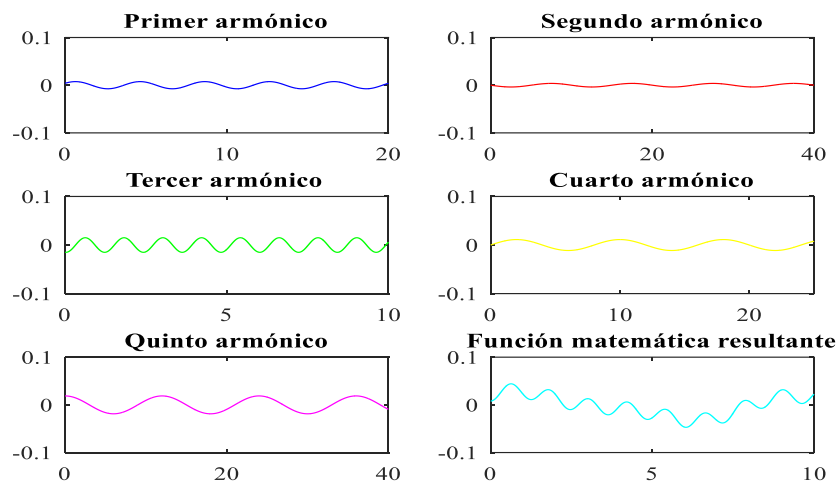


Figura 30. Armónicos y función matemática resultante que modela las irregularidades

Se analizará, en primera instancia, la localización de los puntos de contacto P_j sobre cada rueda del modelo multicuerpo en cada instante t . Unos resultados coherentes validarían la manera en la que el modelo determina los puntos de contacto P_i y P_j en cada instante t .

Cuando el terreno es perfectamente plano, dos puntos P_j de cada rueda del modelo satisfacen las ecuaciones del contacto elástico, el situado en la vertical del centro de gravedad y su homólogo a 180° . Sin embargo, el modelo ha sido programado para que en todo momento tenga claro que sólo el punto P_j en contacto con el terreno ha podido contactar físicamente. Como consecuencia, el modelo debería siempre determinar en cualquier instante t el mismo punto de contacto P_j .

Por otro lado, cuando el terreno es irregular la localización de los puntos de contacto P_j dependerá del valor de la coordenada generalizada R_x^i que tenga cada rueda en cada instante t . El valor de la coordenada generalizada R_x^i de cada rueda del modelo definirá en qué parte del terreno se encuentra la rueda. A la vista de la función matemática y que define la altura

del terreno en cualquier punto, se pueden intuir dónde se localizarán los puntos de contacto P_j .

La visualización y localización de los puntos de contacto P_j en cada rueda del modelo multicuerpo se ha realizado posprocesando las coordenadas curvilíneas s_j , mediante la función *PProceso.m*, que definen a los puntos de contacto P_j en cada instantet.

Las siguientes figuras, obtenidas mediante la función *plotrueda.m*, reflejan la localización de los puntos de contacto P_j en las ruedas del modelo multicuerpo cuando se simula el comportamiento del vehículo y del remolque en un terreno perfectamente llano y en un terreno irregular:

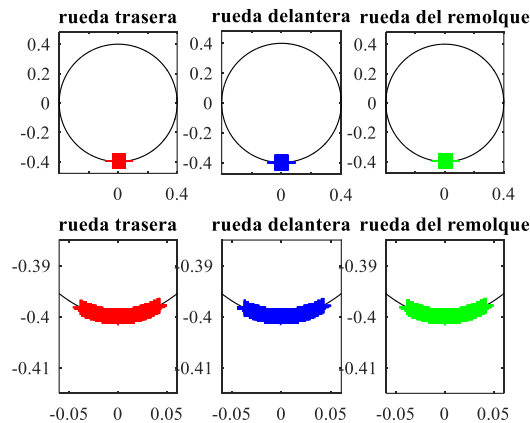


Figura 31. Localización de los puntos de contacto P_j en terreno irregular

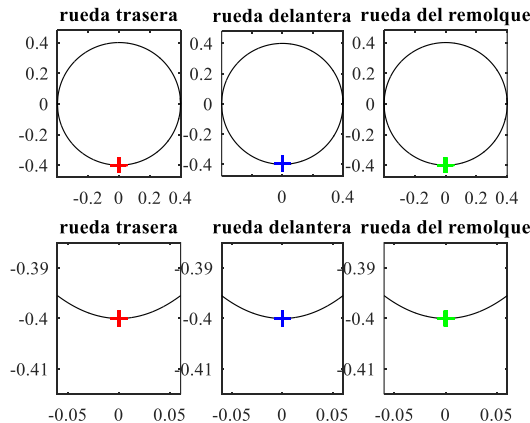


Figura 32. Localización de los puntos de contacto P_j en terreno plano

Los resultados obtenidos validan irrefutablemente la manera en la que el modelo determina las coordenadas curvilíneas s_i y s_j de los puntos P_i y P_j que contactan en cada instante t , resolviendo las ecuaciones no lineales del contacto elástico.

A continuación se analizará el cabeceo del chasis. El cabeceo se estudiará evaluando el valor de la coordenada generalizada θ^3 del chasis en función del tiempo t de simulación.

El vehículo tendrá que girar o cabecear para poder hacer frente a las irregularidades del terreno. Por consiguiente, se espera que los valores de la coordenada generalizada θ^3 del chasis del vehículo difieran de signo cuando el vehículo se encuentra subiendo una irregularidad y cuando se dispone a bajar por ella.

Por otro lado, se espera que el chasis no cabecee cuando avanza por el terreno perfectamente plano. En los primeros instantes, debido a la dificultad de definir unas condiciones iniciales perfectas, se espera que la coordenada generalizada θ^3 oscile de magnitud hasta que el modelo multicuerpo encuentre una configuración estable en su movimiento por el terreno, el modelar el terreno perfectamente plano propiciará que el sistema multicuerpo alcance unas condiciones estacionarias.

Las siguientes figuras representan la coordenada generalizada θ^3 frente al tiempo t de simulación, cuando el vehículo avanza por un terreno perfectamente plano y cuando avanza por un terreno que tiene irregularidades en toda su extensión:

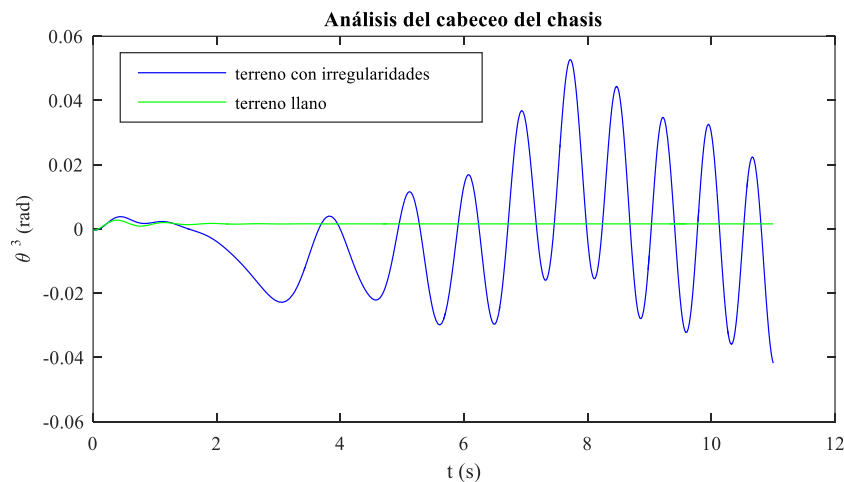


Figura 33. Cabeceo del chasis en terreno irregular y en terreno perfectamente plano

Los resultados obtenidos reflejan con claridad el cabeceo exhibido por el chasis cuando el terreno tiene irregularidades y la estabilidad que presenta la orientación del chasis cuando el terreno es perfectamente plano:

- En primera instancia, cuando el terreno está definido por una función y constante, se observa claramente que en los instantes iniciales, debido que los vectores $\mathbf{q}_0$ y $\dot{\mathbf{q}}_0$ no son los ideales, el vehículo cabecea. Sin embargo, se observa perfectamente como rápidamente la coordenada generalizada θ^3 tiende hacia un valor constante y positivo. Alrededor de los 4 s de simulación el chasis encuentra una orientación θ^3 estable.

La razón de que la coordenada generalizada θ^3 sea positiva cuando el vehículo encuentra unas condiciones estacionarias es consecuencia directa de la inercia del vehículo. Al ser la aceleración generalizada $\ddot{R}_x^3$ del chasis positiva, se producirá una transferencia de peso del vehículo a la rueda trasera del vehículo.

- Cuando la altura del terreno está definida por una función matemática y resultado de sumar las funciones matemáticas y de cinco armónicos se observa claramente que ni la coordenada generalizada θ^3 se estabiliza en un valor constante ni el modelo multicuerpo tiende a unas condiciones estacionarias.

El gráfico adjuntado muestra con total claridad cómo el chasis del vehículo debe que cabecear o girar para superar las irregularidades del terreno.

Se observa claramente como el signo de θ^3 es positivo cuando el vehículo asciende por una irregularidad y negativo cuando desciende por ella.

Además, la dependencia mostrada de la coordenada generalizada θ^3 con el tiempo de simulación t es consecuencia directa de haber modelado el terreno mediante cinco funciones sinusoidales. Si se hubiese definido la altura del terreno mediante un único armónico, los valores máximos y mínimos de la coordenada generalizada θ^3 serían únicos y constantes.

A continuación se estudiará la dinámica longitudinal del modelo analizando la evolución de las fuerza de impulsión en la rueda delantera del vehículo en función del tiempo t de simulación. Unos resultados coherentes con la formulación de la dinámica longitudinal del modelo validarían la manera en la que el modelo determina la fuerza de impulsión o de tracción que desarrolla el vehículo en cualquier instante t .

Evaluando el valor de la fuerza F_{tc} en la rueda delantera del vehículo en función del tiempo de simulación t se han obtenido las siguientes representaciones:

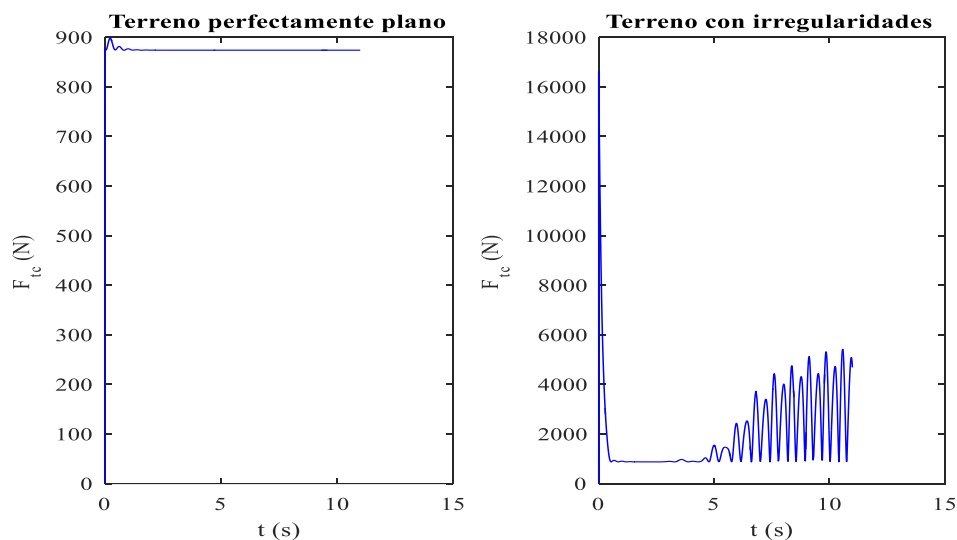


Figura 34. Tracción desarrollada por la rueda tractora en terreno irregular y en terreno plano

De las figuras obtenidas, se pueden sacar las siguientes conclusiones:

- Cuando el terreno es perfectamente plano el modelo multicuerpo encuentra unas condiciones estacionarias y, por ende, la indentación δ de la rueda tractora alcanza un valor constante. Por consiguiente, la máxima tracción que puede desarrollar la rueda delantera del vehículo es constante, tanto el coeficiente de fricción μ como la fuerza

normal de contacto F_{nc} son constantes (si la indentación δ es constante también lo será la fuerza normal de contacto F_{nc}).

Por consiguiente, la rueda delantera solamente deberá de alcanzar un régimen de giro tal que la velocidad absoluta v_{desl} del punto de contacto P_j supere o iguale a la velocidad máxima de deslizamiento v_{lim} .

Se observa claramente como la rueda tractora, por medio del par motor M^i que le provoca girar, alcanza rápidamente un régimen de giro que propicia que desarrolle su máxima capacidad de impulsión. Esto se debe a que la velocidad límite de deslizamiento v_{lim} tiene un valor de 0,1 m/s. Por ende, será sumamente fácil que la rueda tractora desarrolle un régimen de giro tal que la velocidad de deslizamiento v_{desl} iguale o supere a la velocidad límite de deslizamiento v_{lim} .

- Cuando el terreno está modelado mediante armónicos, el valor de la indentación δ de la rueda delantera del vehículo está cambiando continuamente y consecuentemente, la fuerza normal de contacto F_{nc} .

Por ello, recordando que el coeficiente de fricción del terreno μ es constante, la fuerza tangencial de contacto máxima que puede desarrollar la rueda delantera del vehículo está cambiando de valor continuamente.

Como consecuencia, es totalmente lógico obtener una dependencia de la fuerza tangencial de contacto F_{tc} con el tiempo de simulación t mucho menos predecible, como consecuencia de tener que subir y ascender por las irregularidades la fuerza máxima de impulsión que puede desarrollar la rueda delantera del vehículo está cambiando en cada instante t .

Si se representa la indentación δ de cada rueda del modelo multicuerpo en función del tiempo t de simulación se podrán corroborar las hipótesis lanzadas acerca de que en terreno llano la fuerza máxima de impulsión que puede desarrollar la fuerza tractora es constante y que el sistema multicuerpo tiende hacia una configuración estable, tras pasar los instantes iniciales de simulación.

Por otro lado, se espera que la indentación δ de las ruedas del modelo en el terreno esté variando continuamente con el tiempo t de simulación cuando el terreno es irregular. Además, para corroborar que un valor constante de la indentación δ derivará en un valor constante de la fuerza normal de contacto F_{nc} , se representará el valor de F_{nc} en la rueda delantera del vehículo en función del tiempo t .

Las siguientes figuras representan la indentación δ en el terreno de las ruedas del modelo multicuerpo y la fuerza normal de contacto en la rueda delantera del vehículo en función del tiempo de simulación t :

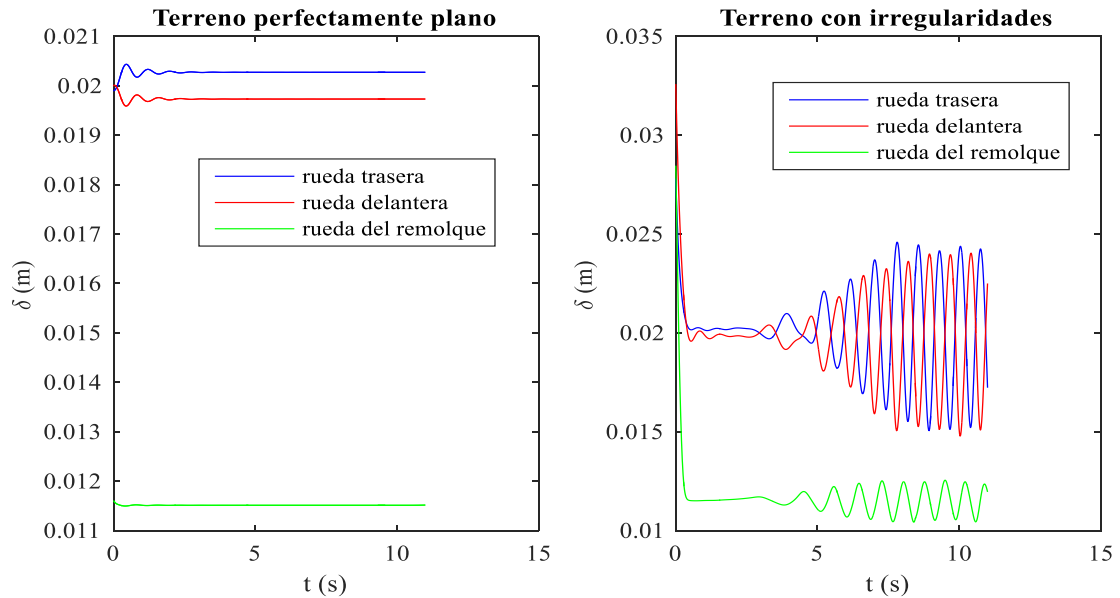


Figura 35. Indentación en el terreno de las ruedas del modelo multicuerpo en terreno irregular y en terreno llano

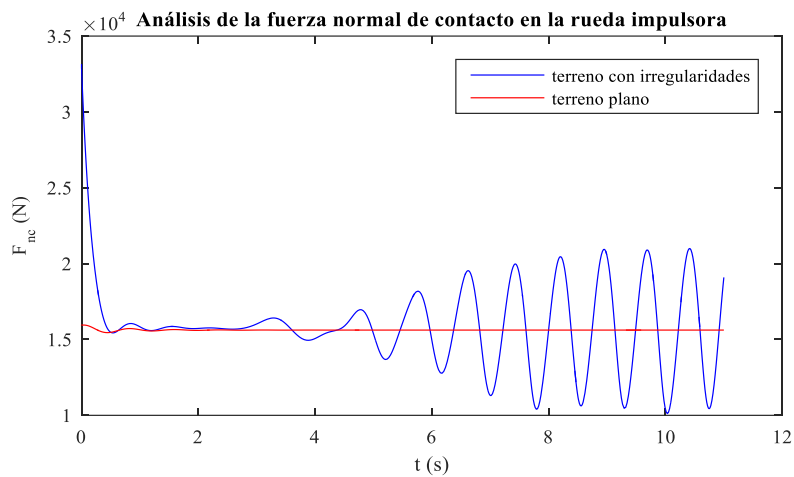


Figura 36. Fuerza normal de contacto actuante en la rueda impulsora en terreno irregular y en terreno llano

Los resultados obtenidos, a la vista de los gráficos obtenidos, confirman totalmente las hipótesis planteadas. En terreno llano la indentación δ de cada rueda se vuelve constante, y por ende la fuerza normal de contacto F_{nc} , y en terreno irregular las ruedas están continuamente indentando cantidades distintas en el terreno.

7.4- Simulación de la influencia del coeficiente de fricción en la tracción del vehículo cuando asciende por una pendiente.

El presente ensayo analizará la influencia del coeficiente de fricción μ del terreno en la capacidad de tracción o impulsión del vehículo. Se pretende simular una situación donde el coeficiente de fricción μ sea determinante para que el vehículo pueda superar un tramo con

una pendiente del 12 %. Para analizar la influencia del coeficiente de fricción μ se simularán un total de tres simulaciones dinámicas:

- En una primera simulación dinámica el vehículo tendrá que ascender por un tramo con una pendiente del 12 % cuando el coeficiente de fricción μ tiene una magnitud de 0,8, el vehículo debería poder superar la pendiente sin problemas. Se podrá en movimiento al vehículo mediante una restricción de usuario de velocidad de giro constante en la rueda trasera del vehículo. La velocidad de giro de la rueda trasera $\dot{\theta}^2$ valdrá 15 rd/s.
- Imagínese ahora que μ reduce su magnitud de manera drástica hasta un valor de 0,2. Suponga, por ejemplo, que ha llovido. El vehículo se seguirá poniendo en movimiento por medio de una restricción de velocidad de giro constante en la rueda trasera del vehículo, la velocidad de giro $\dot{\theta}^2$ seguirá valiendo 15 rd/s.

Una reducción del valor del coeficiente de fricción μ repercutirá en una menor capacidad de tracción o impulsión de cada rueda del vehículo. Recuerde que la máxima capacidad de impulsión en cada rueda está determinada por la fuerza normal, relacionada con la indentación δ de cada rueda en el terreno, y por el coeficiente de fricción μ .

Se espera que dichas condiciones propicien que el vehículo no pueda ascender por la pendiente al no poder entregar una fuerza de impulsión que pueda superar la proyección del peso del vehículo en la dirección de la pendiente. Se espera que, debido a la inercia que ha adquirido el vehículo en el tramo recto que precede al tramo con pendiente, el vehículo sólo pueda superar una parte de la pendiente y que acabe finalmente descendiendo por la misma.

- La tercera simulación dinámica pretende solventar la problemática de que el vehículo no pueda superar la pendiente del terreno con un coeficiente de fricción μ de 0,2. Se espera que el vehículo pueda ascender por la pendiente, bajo las condiciones de baja adherencia de la segunda simulación, introduciendo una restricción de velocidad de giro constante en ambas ruedas del vehículo. De este modo, el vehículo pasará de tener tracción trasera a tener tracción en las dos ruedas. Por consiguiente, las velocidades de giro de ambas ruedas $\dot{\theta}^2$ y $\dot{\theta}^4$ tendrán un valor de 15 rd/s.

En el presente ensayo se hará uso de la formulación necesaria para incluir pendientes en el terreno. El terreno ha sido modelado con dos transiciones definidas mediante funciones polinómicas de grado dos, dos tramos rectos de distintas alturas y un tramo con una pendiente m del 12 %.

Para las tres simulaciones dinámicas del ensayo se han asignado los siguientes valores a los parámetros que modelan la geometría del terreno:

x_i	x_f	m	x_{f1}	x_{f2}
4 m	5 m	12 %	22 m	23 m

Tabla 7.4. Parámetros que modelan el terreno

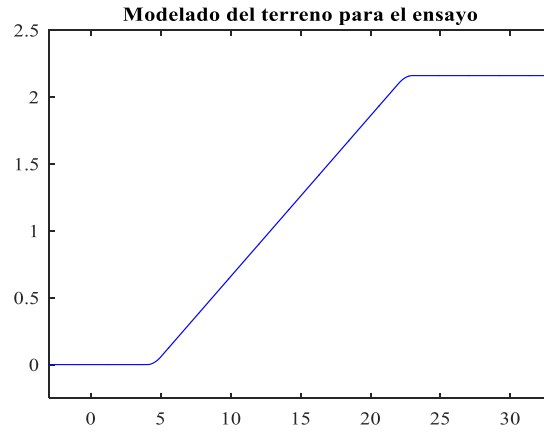


Figura 37. Terreno modelado para el ensayo

La manera más inmediata y clara de poder ver si el vehículo ha podido superar completamente la pendiente, a falta de poder reproducir las animaciones en el presente documento, es representar la coordenada generalizada R_x^3 del chasis en función del tiempo t de simulación.

Una disminución del valor de la coordenada generalizada R_x^3 del chasis indicará que el vehículo y el remolque, tras agotar toda la inercia que presentaban del tramo recto, han comenzado a descender por la pendiente.

Las siguientes figuras muestran la dependencia de R_x^3 con t en las tres simulaciones dinámicas:

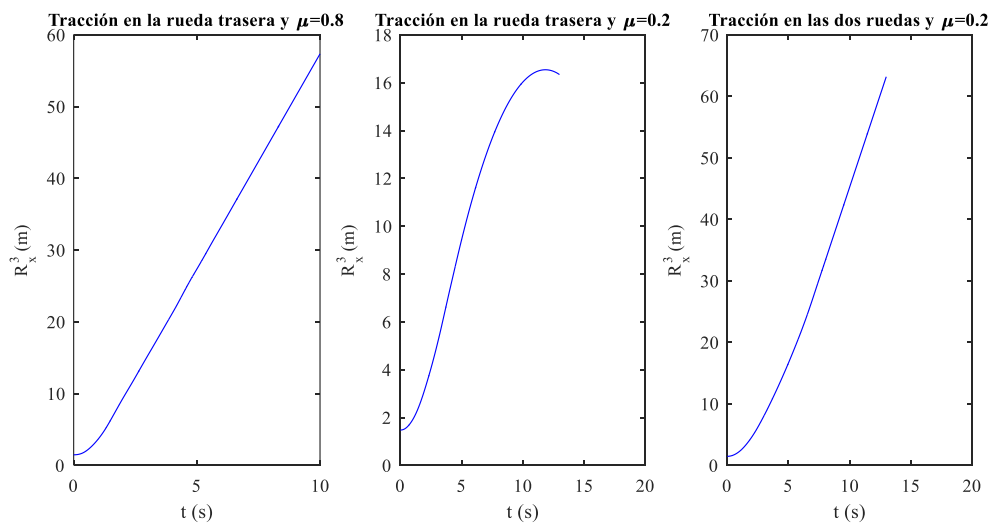


Figura 38. Evolución de la coordenada generalizada R_{x3} del chasis frente al tiempo

Se pueden sacar las siguientes conclusiones acerca de las representaciones mostradas:

- Cuando sólo una rueda del vehículo es tractora y el coeficiente de fricción μ tiene un valor de 0,8, el vehículo asciende por la pendiente sin ninguna dificultad. Prueba de ello es que la coordenada generalizada R_x^3 del chasis crece en todo momento con el paso del tiempo de simulación t .
- Bajo las mismas condiciones de la primera simulación pero con un coeficiente de fricción μ de 0,2, el vehículo no puede superar toda la pendiente del terreno. Observe que R_x^3 comienza a reducir su valor a partir de los 12 segundos de simulación.
- Cuando se añade una restricción de usuario a las dos ruedas del vehículo, dotando al vehículo de doble tracción, el vehículo puede ascender la pendiente aunque las condiciones de adherencia sean pésimas, note que la coordenada generalizada R_x^3 del chasis en ningún momento comienza a reducir su valor.

Por ende, a la vista de las conclusiones citadas, el modelo ha sido capaz de simular a la perfección el cometido del ensayo.

Lo que ha dictado que el vehículo haya podido o no ascender por el tramo con pendiente ha sido la capacidad de impulsión desarrollada por las ruedas del vehículo. La fuerza de tracción generada por las ruedas del vehículo deberá de superar parte del peso del vehículo, el cual está intentado en todo instante intentado que el vehículo descienda. Se trata de un equilibrio elemental de fuerzas.

Si la tracción generada por las ruedas del vehículo es superior a las fuerzas que se oponen a la ascensión del vehículo, el modelo multicuerpo se acelerará en su paso por la pendiente. De lo contrario, los sólidos se desacelerarán en el tramo de pendiente pudiendo avanzar por ella lo que la inercia que presentaban les permita.

Por todo lo descrito anteriormente, se representará la fuerza tangencial de contacto F_{tc} desarrollada por las ruedas tractoras en función del tiempo t de simulación.

Por un lado, se espera la fuerza de impulsión generada por el vehículo en la primera simulación sea mucho superior a la generada en la segunda simulación, el coeficiente de fricción μ reduce su magnitud considerablemente.

Por otro lado, en la tercera simulación dinámica se espera que la fuerza neta de impulsión, suma de la generada por la rueda trasera y la generada por la rueda delantera, sea mayor que la que desarrolla el vehículo en la segunda simulación dinámica como consecuencia de que las dos ruedas sean tractoras.

Las siguientes figuras representan la fuerza tangencial de contacto F_{tc} generada por las ruedas tractores en función del tiempo t de simulación:

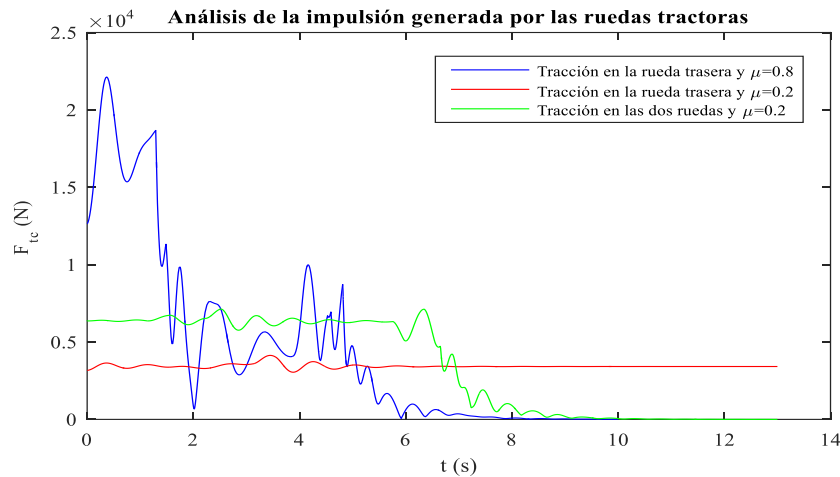


Figura 39. Impulsión generada por el vehículo en cada simulación del ensayo

Se pueden sacar las siguientes conclusiones de la figura adjuntada:

- Las dependencias de F_{tc} con t en las tres simulaciones justifican completamente las figuras anteriormente analizadas que dictaban si el vehículo consiguió o no ascender por la pendiente. Se aprecia cómo la impulsión generada por el vehículo en la primera simulación es mayor que la generada por el vehículo en la segunda simulación. Por dicha razón el vehículo, con una única rueda tractora, no es capaz de superar la pendiente.

Por otro lado, se aprecia que dotando al vehículo de doble tracción se genera la impulsión suficiente para ascender por la pendiente con un coeficiente de fricción μ de valor 0,2.

- En muy pocos instantes la fuerza de impulsión F_{tc} se estabiliza en su valor máximo, la respuesta se halla en la geometría del terreno.

La función y que define la altura del terreno provoca que el vehículo tenga que ascender por una pendiente y pasar por dos transiciones definidas por funciones polinómicas de grado dos. Por consiguiente, la indentación δ de las ruedas del modelo al pasar por las transiciones variará y, por ende, la capacidad máxima de impulsión estará cambiando.

En el primer tramo plano del terreno las ruedas no presentan una capacidad máxima de impulsión constante debido a que las condiciones iniciales, definidas por los vectores $\mathbf{q}_0$ y $\dot{\mathbf{q}}_0$, no son del todo óptimas.

La fuerza de impulsión se estabiliza cuando el vehículo alcanza el segundo tramo perfectamente plano pues el sistema multicuerpo encuentra una configuración estable

Para corroborar las hipótesis planteadas anteriormente, se representará las indentaciones δ de las ruedas tractoras en el terreno en función de sus coordenadas generalizadas R_x^i :

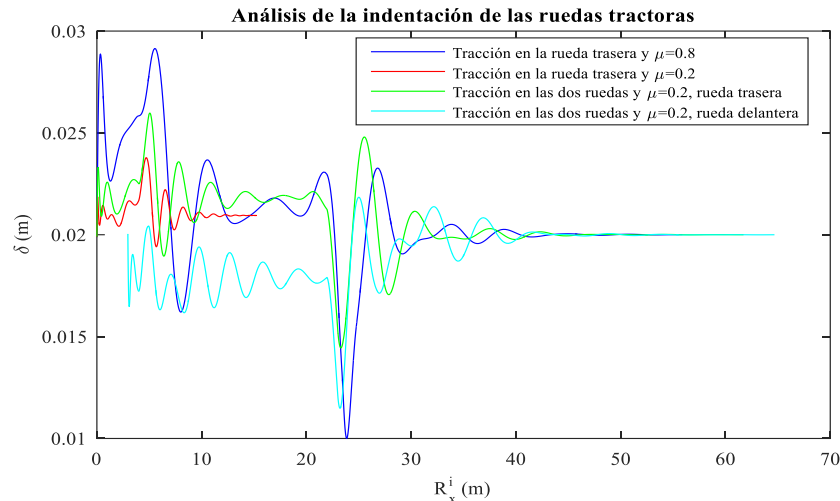


Figura 40. Indentación en el terreno de las ruedas del modelo multicuerpo en cada simulación

Efectivamente, las figuras adjuntadas corroboran la influencia del modelado del terreno en el valor la indentación δ de cada rueda del modelo multicuerpo en el terreno y por consecuencia, en la impulsión que pueda desarrollar el vehículo. Además, las figuras mostradas son un claro indicador, al estar graficando la coordenada generalizada R_x^i en vez del tiempo t de simulación, de hasta dónde son capaces de avanzar en el terreno las ruedas del modelo multicuerpo.

Parece interesante visualizar la localización de los puntos de contacto P_j durante el transcurso de una simulación dinámica. Solamente se representará para una única simulación dinámica del ensayo, en las tres simulaciones la geometría del terreno es idéntica. De igual modo, recuerde que ya fue validada en el tercer ensayo del TFG la manera en la que el modelo determina en cada instante t los puntos de contacto P_i y P_j . Los resultados obtenidos en el ensayo son los adjuntados a continuación:

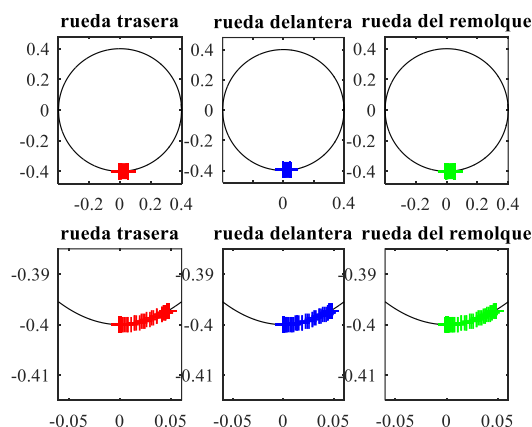


Figura 41. Localización de los puntos de contacto P_j

La figura muestra unos resultados totalmente coherentes y lógicos, confirmando una vez más la validez que tiene el modelo a la hora de buscar los puntos de contacto rueda-terreno en

cada instante t . Se aprecia perfectamente como las transiciones y el tramo con pendiente cambian las localizaciones de los puntos P_j que satisfacen las ecuaciones no lineales del contacto elástico.

Finalmente, para concluir el presente apartado de ensayos realizados con el modelo, se analizará la ley del momento o par de reacción necesario para que la restricción de velocidad de giro constante se cumpla en las ruedas del modelo multicuerpo donde se ha fijado una restricción de usuario.

La restricción de usuario inducirá la aparición de un momento de reacción que provocará que en cualquier instante t de la simulación dinámica el momento o par neto que actúa sobre la rueda sea nulo, de este modo se podrá mantener, por inercia, la velocidad de giro $\dot{\theta}^i$ constante.

Por ende, se espera tener en cada instante t un momento de reacción M de igual magnitud pero sentido contrario al momento neto o resultante que actúe en cada instante t sobre las ruedas que se haya aplicado la restricción de usuario.

Por otro lado, la ley de par estará directamente relacionada con la magnitud de la velocidad absoluta del punto de contacto P_j en cada instante t . Mientras que la velocidad absoluta del punto de contacto P_j o velocidad de deslizamiento v_{desl} sea mayor que cero, un par de reacción M tendrá que actuar sobre la rueda para contrarrestar el momento M ejercido por la fuerza tangencial de contacto F_{tc} . De no actuar el momento de reacción M , la fuerza tangencial de contacto ejercería un momento M tal que la aceleración angular de la rueda α^i será distinta de cero, lo cual se traduciría en un incumplimiento de la restricción de usuario al aumentar o disminuir la velocidad de giro $\dot{\theta}^i$ en la rueda.

Solamente se ha centrado la atención en el momento ejercido por la fuerza F_{tc} pues note que el resto de fuerzas actuantes sobre las ruedas no ejercerán momento M alguno respecto al centro de gravedad de la rueda.

Para no incluir demasiadas representaciones, se analizará únicamente en la tercera simulación dinámica la ley del momento de reacción M en la rueda trasera del vehículo, recuerde que en la tercera simulación dinámica se impuso una restricción de velocidad de giro constante a las dos ruedas del vehículo.

Los resultados obtenidos han sido los siguientes:

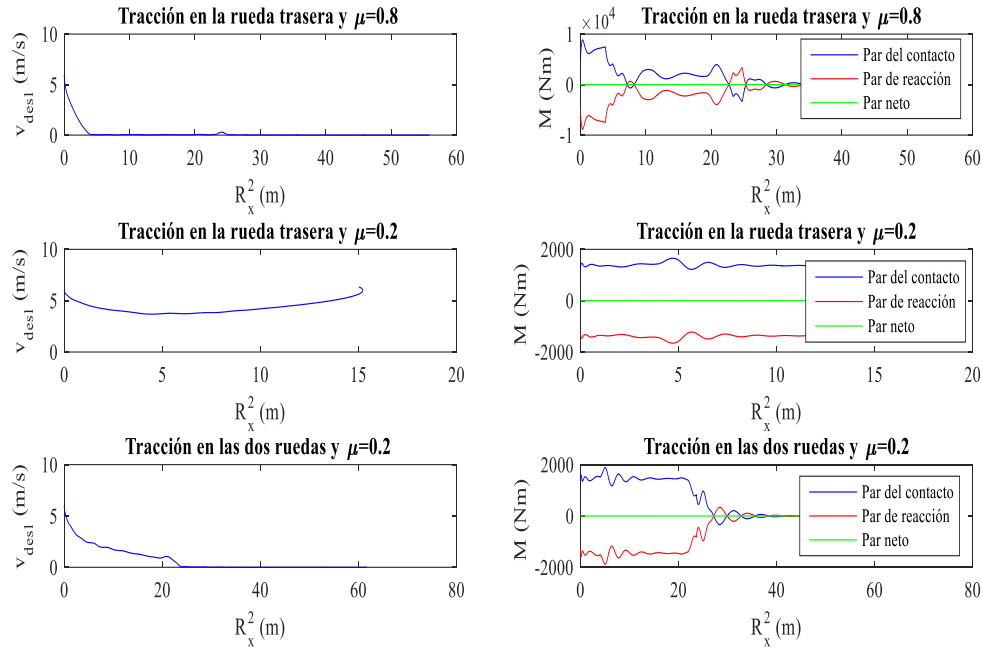


Figura 42. Velocidad de deslizamiento de la rueda tractora y ley del momento de reacción

Se pueden extraer las siguientes conclusiones de las figuras adjuntadas:

- Efectivamente, tal y como se predijo, se observa claramente que el momento M de reacción trata de equilibrar el par ejercido por la fuerza F_{tc} para que el momento neto M se nulo y de este modo mantener constante, por inercia, el valor de la velocidad de giro $\dot{\theta}^i$ en las ruedas sobre las cuales se ha aplicado la restricción de usuario.
- Se observa perfectamente como el par de reacción M es distinto de cero cuando la magnitud de velocidad de deslizamiento v_{desl} es distinta de cero, lo cual promoverá la aparición de una fuerza tangencial de contacto F_{tc} .
- Por otro lado, las representaciones obtenidas muestran la clara dependencia que muestra la velocidad de deslizamiento v_{desl} con la impulsión desarrollada por las ruedas tractoras.

Cuando el vehículo genera la tracción suficiente para superar la pendiente, como ocurre en la primera y tercera simulación, la fuerza de impulsión genera una velocidad lineal, de sentido contrario a la velocidad promovida por la restricción de usuario, en el punto de contacto P_j que promueve que la velocidad de deslizamiento sea nula v_{desl} , y por consecuencia también la fuerza tangencial de contacto F_{tc} .

- Cuando el vehículo no puede ascender la pendiente del terreo, tal y como ocurre en la segunda simulación dinámica, la tracción desarrollada por la rueda trasera del vehículo no puede generar una velocidad lineal que contrarreste la velocidad de la restricción de usuario. Como consecuencia, el par de reacción M tendrá que ser mayor que cero en todo instante t de la simulación.

8-CONCLUSIONES

La realización y desarrollo del *TFG* ha requerido de mucho esfuerzo personal y sacrificio pero a la vez ha supuesto una experiencia enormemente gratificante por la inconmensurable cantidad de conocimientos y habilidades que jamás hubiera adquirido de no haberme decidido a realizar este *TFG*. La elevada especialización de mi director *TFG* en la teoría de sistemas multicuerpo y su excelente labor docente ha sido crucial para que me haya decidido a escoger este trabajo y para que finalmente haya podido afrontar todos los contratiempos que me han surgido a lo largo del desarrollo del *TFG*.

Antes de comenzar a definir o pensar qué sistema multicuerpo modelar para el *TFG*, tuve que introducirme en la teoría de sistemas multicuerpo y estudiar abundantes conceptos teóricos que conformarían mi base de conocimiento para el desarrollo del *TFG*.

Enfrentarme a una ciencia completamente desconocida supuso una dificultad añadida pues tuve que lidiar con una formulación bastante compleja y totalmente desconocida hasta entonces.

La ejecución de un *TFG* de estas características requiere de una elevada habilidad y desparpajo con la programación, en mi caso con el software *Matlab*. Me atrevería a decir que el 90 % del tiempo que he empleado para desarrollar y realizar el *TFG* ha sido trabajando con el software *Matlab*.

Aunque a lo largo del grado, y especialmente en la asignatura de *Informática* cursada en el primer curso del grado, la universidad me ha proporcionado y dotado de una base para programar con *Matlab*, mi elevado interés por la programación y la impecable labor docente de mi tutor de *TFG* ha propiciado que tras finalizar el trabajo haya adquirido unos conocimientos muchos más elevados de programación e utilización del software *Matlab* de los que partía en un principio.

Hasta que se presentó la ocasión de poder desarrollar y realizar este *TFG*, jamás tuve ocasión anteriormente de poder modelar el comportamiento de ningún sistema multicuerpo ni de enfrentarme a simulaciones numéricas. Durante el desarrollo e implementación del modelo multicuerpo, he tenido que lidiar con innumerables errores numéricos debido a la dificultad que presenta el modelo, bajo ciertas condiciones, de determinar las coordenadas curvilíneas s_i y s_j de los puntos P_i y P_j que han contactado.

Por otro lado, todas las figuras que pueda apreciar en el *TFG* han sido elaboradas específicamente para el mismo, no han sido tomadas de ninguna fuente bibliográfica ajena. Mi deseo de no querer adjuntar figuras que no hubiesen sido diseñadas íntegramente por mi supuso tener que aprender a utilizar programas de diseño gráfico, especificados para tales fines.

Puede ocurrir que cuando uno ve las animaciones de las simulaciones dinámicas y observa que el modelo efectivamente, dentro de unos límites coherentes, reproduce el comportamiento que cualquier persona, disponga o no de los conocimientos científicos necesarios, esperaría que tuviera un vehículo en las condiciones de la simulación piense que se trata de una simple animación del movimiento de un vehículo, olvidando toda la formulación y esfuerzo que hay detrás para que sea posible obtener tales animaciones del movimiento del sistema multicuerpo.

Pondré como ejemplo el segundo ensayo del *TFG* para expresar lo complejo que puede llegar ser que el modelo simule el comportamiento deseado y proporcionar los medios necesarios para ello. Este ensayo nació con la simple y llana pregunta de: ¿Qué ocurrirá si se pone un bache en medio del terreno y se intenta que el remolque salte cuando contacte con él?

Desde que uno se plantea dicha pregunta hasta que finalmente se consigue que el modelo simule el comportamiento deseado, se requiere de mucho esfuerzo y dedicación. Simplemente para modelar el bache fue necesario implementar toda la formulación descrita en el marco teórico. Además, recuerde que el modelo está preparado para generar cualquier irregularidad repentina sin más que introducir los valores de los parámetros que la modelan. Por otro lado, no basta con modelar el bache y hacer que el modelo multicuerpo contacte con él. Tuve muchos contratiempos hasta hallar qué velocidades de giro de la rueda trasera del vehículo y qué tipos de irregularidades no derivaban en problemas numéricos. Piense que el modelo puede tener mucha dificultad para encontrar los puntos de contacto P_i y P_j cuando las ruedas del modelo multicuerpo impactan con el bache.

De igual modo, el deseo de simular el comportamiento del vehículo y remolque en un terreno con pendiente condujo a tener que programar toda la formulación descrita en el marco teórico para la inclusión de pendientes en el terreno.

Con lo descrito solamente se desea dejar constancia de la dificultad que puede suponer que el modelo reproduzca el comportamiento esperado y las nuevas necesidades, en cuanto a formulación y programación del modelo, que conlleva querer simular con el modelo multicuerpo nuevos ensayos.

El potencial de mejora y desarrollo al que se presta el modelo multicuerpo es inmenso:

- La formulación de la dinámica longitudinal del vehículo puede ser mejorada con una definición más realista y compleja de la función matemática que define la dependencia de la fuerza tangencial de contacto F_{tc} con la velocidad de deslizamiento v_{desl} .
- Los valores de los parámetros que modelan el comportamiento de los muelles-amortiguadores del sistema multicuerpo podrían ser definidos en base a ensayos experimentales. Si se dispusieran de los medios necesarios se podría diseñar un modelo a escala y realizar ensayos estáticos con él. Registrar las oscilaciones verticales del modelo hasta que encuentre una configuración estática de equilibrio

permitiría ajustar los valores de las rigideces de los muelles verticales y horizontales k_v y k_h , los coeficientes de amortiguación de los muelles-amortiguadores c_v y c_h , las constantes *Hertzianas* K_H de los contactos elásticos rueda-terreno y los coeficientes de amortiguación C de los contactos elásticos

- Modelar el sistema multicuerpo de manera tridimensional. Esto permitirá modelar el movimiento del vehículo y del remolque en una inmensidad de situaciones que no son posibles modelar de manera bidimensional con el actual modelo. Piense en modelar el comportamiento del vehículo y del remolque en una curva. Lógicamente, esto supondría tener que reformular toda la formulación y programación del modelo con la formulación tridimensional de la teoría de sistemas multicuerpo.
- Las propiedades mecánicas e inerciales de los sólidos del sistema multicuerpo se podrían definir de manera más compleja y realista. Recuerde que el chasis y el remolque han sido modelados como rectángulos y las ruedas del vehículo como círculos. De igual modo, se podrá definir de una manera mucho más formal las localizaciones de los centros de gravedad de los sólidos del sistema multicuerpo.
- Definir unas condiciones iniciales mucho más óptimas, definidas por el vector y_0 , para cada simulación dinámica evitando así el cabeceo inicial que presenta el vehículo en los comienzos de las simulaciones. Lógicamente, esto requeriría de estudios y análisis más complejos
- Simular el funcionamiento de sistemas o componentes reales de un vehículo como, por ejemplo, el sistema de transmisión. Se podrían programar y formular leyes del momento M^i o velocidad de giro $\dot{\theta}^i$ que deben de tener en cada momento cada rueda del vehículo en función de la impulsión requerida y de la forma del terreno por la que avanza el vehículo.
- Se podría optimizar y mejorar la manera en la que el modelo determina los puntos de contacto P_i y P_j en cada instante t , que en definitiva consiste en resolver las ecuaciones no lineales del contacto elástico rueda-terreno. Esto derivaría en una mayor eficiencia numérica del modelo y en posibilidades de modelar el terreno con geometrías mucho más complejas.
- Tratar de estudiar e investigar qué tipo de integradores numéricos serían convenientes implementar en el modelo para mejorar la eficiencia computacional y numérica del modelo.

Las posibles mejoras futuras del modelo descritas no son las únicas pero a la vista de los resultados obtenidos en los ensayos presentados se puede concluir que el modelo es capaz de simular una gran cantidad de situaciones de interés y además, de manera óptima.

9-REFERENCIAS BIBLIOGRÁFICAS

- [1] Shabana, A. A. (2009). *Computational dynamics*. John Wiley & Sons.
- [2] Shabana, A. A. (2013). *Dynamics of multibody systems*. Cambridge university press.
- [3] Escalona, J. L. (2019). *Apuntes de dinámica de sistemas multicuerpo*. Universidad de Sevilla

10-ANEXOS

```
function SMCuerpo = Definicion_Modelo
%%Parametrización del terreno%%
SMCuerpo.xi=4;
SMCuerpo.xf=4.1;
SMCuerpo.p=1;
SMCuerpo.m=0.1;
SMCuerpo.yf=0.1;
SMCuerpo.gradopolin=2;
SMCuerpo.bache=0;
SMCuerpo.bachex1=10;
SMCuerpo.bachex2=10.2;
SMCuerpo.bachex3=10.4;
SMCuerpo.bachex4=10.6;
SMCuerpo.bachex5=10.8;
SMCuerpo.bachem1=0.22;
SMCuerpo.bachem2=-0.22;
SMCuerpo.xf1=22;
SMCuerpo.xf2=23;
SMCuerpo.tr=0;
%%PARÁMETROS GENERALES DEL SISTEMA MULTICUERPO%%
SMCuerpo.nsolidos = 6;
SMCuerpo.narticulaciones = 2;
SMCuerpo.nrestrusuario = 1;
SMCuerpo.nma = 4;
SMCuerpo.nce = 3;

%%DETERMINACIÓN DEL NÚMERO DE RESTRICCIONES TOTALES%%
SMCuerpo.nrestricciones = 3+(2*SMCuerpo.narticulaciones)+(SMCuerpo.nrestrusuario);

%%DETERMINACIÓN DEL NÚMERO DE COORDENADAS GENERALIZADAS TOTALES%%
SMCuerpo.ncoortotales=3*SMCuerpo.nsolidos;

%%MAGNITUDES DE INTERÉS%%
SMCuerpo.Par=0;
SMCuerpo.Parrt=0;
SMCuerpo.Parrd=0;
SMCuerpo.Param.g=9.81;
SMCuerpo.Param.L_chasis=4.95;
SMCuerpo.Param.H_chasis=1.84;
SMCuerpo.Param.bata1la=2*0.2985*SMCuerpo.Param.L_chasis;
SMCuerpo.Param.masa_chasis=3230;
SMCuerpo.Param.masa_rueda_chasis=10;
SMCuerpo.Param.radio_rueda_chasis=0.4;
SMCuerpo.Param.masa_remolque=700;
SMCuerpo.Param.L_remolque=2;
SMCuerpo.Param.H_remolque=0.5;
SMCuerpo.Param.masa_rueda_remolque=10;
SMCuerpo.Param.radio_rueda_remolque=0.4;
SMCuerpo.Param.nmuellesamort=SMCuerpo.nma;
SMCuerpo.Param.rigidez_muellesverticales=158431.5;
SMCuerpo.Param.rigidez_muelleshorizontales=158431.5*6;
SMCuerpo.Param.camort_muellesverticales=158431.5*0.03;
SMCuerpo.Param.camort_muelleshorizontales=158431.5*0.3;
```

```

SMcuerpo.Param.li_muelle_vertical=0.35;
SMcuerpo.Param.li_muelle_horizontal=0.35;
SMcuerpo.Param.separacion_remolchas=1;
SMcuerpo.Param.rigidez.Khertz=5636082.988;
SMcuerpo.Param.camort.celas=5636082.988*0.02;
SMcuerpo.Param.friccion_terreno=0.8;
SMcuerpo.Param.vdesliz=0.1;
SMcuerpo.Param.si_rueda_trasera=-pi/2;
SMcuerpo.Param.si_rueda_delantera=-pi/2;
SMcuerpo.Param.si_rueda_remolque=-pi/2;
SMcuerpo.Param.si_suelo_ruedatrasera=0;
SMcuerpo.Param.si_suelo_ruedadelantera=0;
SMcuerpo.Param.si_suelo_ruedaremolque=0;
SMcuerpo.Param.narmonicos=0;
SMcuerpo.Param.aml=0.75*[0.01 0.005 0.02 0.015 0.025]';
SMcuerpo.Param.fi=[0.18*pi 1.0*pi 1.46*pi 2*pi 2.5*pi]';
SMcuerpo.Param.lambda=4*[1 2.5 0.3 2 3]';
SMcuerpo.Param.wrueda_trasera=-10;
SMcuerpo.Param.wrueda_delantera=0;
SMcuerpo.Param.Par=0;
SMcuerpo.Param.alfaBaum=100;
SMcuerpo.Param.betaBaum=100;
SMcuerpo.Param.t0=0;
%%DEFINICIÓN DE LOS SÓLIDOS RÍGIDOS DEL SISTEMA MULTICUERPO%%
%%Eslabón fijo%%
SMcuerpo.Srigidos(1).nombre = 'Eslabón fijo';
SMcuerpo.Srigidos(1).masa = 0;
SMcuerpo.Srigidos(1).Minercia = 0;
SMcuerpo.Srigidos(1).coorcg = [0 0]';
%%Rueda trasera%%
SMcuerpo.Srigidos(2).nombre = 'Rueda trasera';
SMcuerpo.Srigidos(2).masa =SMcuerpo.Param.masa_rueda_chasis;
SMcuerpo.Srigidos(2).Minercia =
0.5*SMcuerpo.Param.masa_rueda_chasis*(SMcuerpo.Param.radio_rueda_chasis^2);
SMcuerpo.Srigidos(2).coorcg = [0 0]';
%%Chasis%%
SMcuerpo.Srigidos(3).nombre='Chasis';
SMcuerpo.Srigidos(3).masa=SMcuerpo.Param.masa_chasis;
SMcuerpo.Srigidos(3).Minercia=(1/12)*SMcuerpo.Param.masa_chasis*((SMcuerpo.Param.L_chasis)^2) + ((SMcuerpo.Param.H_chasis)^2));
SMcuerpo.Srigidos(3).coorcg=[0 0]';
%%Rueda delantera%%
SMcuerpo.Srigidos(4).nombre='Rueda delantera';
SMcuerpo.Srigidos(4).masa=SMcuerpo.Param.masa_rueda_chasis;
SMcuerpo.Srigidos(4).Minercia=0.5*SMcuerpo.Param.masa_rueda_chasis*(SMcuerpo.Param.radio_rueda_chasis^2);
SMcuerpo.Srigidos(4).coorcg=[0 0]';
%%Remolque%%
SMcuerpo.Srigidos(5).nombre='Remolque';
SMcuerpo.Srigidos(5).masa=SMcuerpo.Param.masa_remolque;
SMcuerpo.Srigidos(5).Minercia=(1/12)*SMcuerpo.Param.masa_remolque*((SMcuerpo.Param.L_remolque^2)+(SMcuerpo.Param.H_remolque^2));
SMcuerpo.Srigidos(5).coorcg=[0 0]';
%%Rueda del remolque%%
SMcuerpo.Srigidos(6).nombre='Rueda Remolque';
SMcuerpo.Srigidos(6).masa=SMcuerpo.Param.masa_rueda_remolque;
SMcuerpo.Srigidos(6).Minercia=0.5*SMcuerpo.Param.masa_rueda_remolque*(SMcuerpo.Param.radio_rueda_remolque^2);

```

```

SMcuerpo.Srigidos(6).coorcg=[0 0]';

%%DEFINICIÓN DE LAS UNIONES ARTICULADAS%%
%%Primera articulación (remolque-chasis)%%
SMcuerpo.Pares.revolucion(1).A=5;
SMcuerpo.Pares.revolucion(1).B=3;
SMcuerpo.Pares.revolucion(1).ula=[(0.5*SMcuerpo.Param.L_remolque)+SMcuerpo.Param.separacion
 _remolchas 0]';
SMcuerpo.Pares.revolucion(1).ulb=[-0.5*SMcuerpo.Param.L_chasis -
 0.5*SMcuerpo.Param.H_chasis]';
%%Segunda articulación (remolque-rueda_remolque)%%
SMcuerpo.Pares.revolucion(2).A=5;
SMcuerpo.Pares.revolucion(2).B=6;
SMcuerpo.Pares.revolucion(2).ula=[0 -0.5*SMcuerpo.Param.H_remolque]';
SMcuerpo.Pares.revolucion(2).ulb=[0 0]';

%%DEFINICIÓN DE LOS MUELLES-AMORTIGUADORES%%
%%Muelle-amortiguador_vertical_trasero_chasis%%
SMcuerpo.Ma(1).A=2;
SMcuerpo.Ma(1).B=3;
SMcuerpo.Ma(1).k=SMcuerpo.Param.rigidez_muellesverticales;
SMcuerpo.Ma(1).c=SMcuerpo.Param.camort_muellesverticales;
SMcuerpo.Ma(1).lindf=SMcuerpo.Param.li_muelle_vertical;
SMcuerpo.Ma(1).ua=[0 0]';
SMcuerpo.Ma(1).ub=[-0.2985*SMcuerpo.Param.L_chasis -0.5*SMcuerpo.Param.H_chasis]';
%%Muelle-amortiguador_vertical_delantero_chasis%%
SMcuerpo.Ma(2).A=4;
SMcuerpo.Ma(2).B=3;
SMcuerpo.Ma(2).k=SMcuerpo.Param.rigidez_muellesverticales;
SMcuerpo.Ma(2).c=SMcuerpo.Param.camort_muellesverticales;
SMcuerpo.Ma(2).lindf=SMcuerpo.Param.li_muelle_vertical;
SMcuerpo.Ma(2).ua=[0 0]';
SMcuerpo.Ma(2).ub=[0.2985*SMcuerpo.Param.L_chasis -0.5*SMcuerpo.Param.H_chasis]';
%%Muelle-amortiguador_horizontal_trasero_chasis%%
SMcuerpo.Ma(3).A=2;
SMcuerpo.Ma(3).B=3;
SMcuerpo.Ma(3).k=SMcuerpo.Param.rigidez_muelleshorizontales;
SMcuerpo.Ma(3).c=SMcuerpo.Param.camort_muelleshorizontales;
SMcuerpo.Ma(3).lindf=SMcuerpo.Param.li_muelle_horizontal;
SMcuerpo.Ma(3).ua=[0 0]';
SMcuerpo.Ma(3).ub=[-(0.2985*SMcuerpo.Param.L_chasis +SMcuerpo.Param.li_muelle_horizontal)
 -(SMcuerpo.Param.li_muelle_vertical+(0.5*SMcuerpo.Param.H_chasis)-
 ((9.81*0.5*SMcuerpo.Param.masa_chasis)/SMcuerpo.Param.rigidez_muellesverticales))];
%%Muelle-amortiguador_horizontal_delantero_chasis%%
SMcuerpo.Ma(4).A=4;
SMcuerpo.Ma(4).B=3;
SMcuerpo.Ma(4).k=SMcuerpo.Param.rigidez_muelleshorizontales;
SMcuerpo.Ma(4).c=SMcuerpo.Param.camort_muelleshorizontales;
SMcuerpo.Ma(4).lindf=SMcuerpo.Param.li_muelle_horizontal;
SMcuerpo.Ma(4).ua=[0 0]';
SMcuerpo.Ma(4).ub=[(0.2985*SMcuerpo.Param.L_chasis)+SMcuerpo.Param.li_muelle_horizontal -
 (SMcuerpo.Param.li_muelle_vertical+(0.5*SMcuerpo.Param.H_chasis)-
 ((9.81*0.5*SMcuerpo.Param.masa_chasis)/SMcuerpo.Param.rigidez_muellesverticales))];

%%DEFINICIÓN DE LOS CONTACTOS ELÁSTICOS%%
%%Contacto elástico_rueda_trasera_terreno%%
SMcuerpo.Celas(1).A=1;
SMcuerpo.Celas(1).B=2;

```

```

SMcuerpo.Celas(1).k_khertz=SMcuerpo.Param.rigidez.Khertz;
SMcuerpo.Celas(1).camort=SMcuerpo.Param.camort.celas;
SMcuerpo.Celas(1).coef_friccion=SMcuerpo.Param.friccion_terreno;
SMcuerpo.Celas(1).vlimite=SMcuerpo.Param.vdesliz;
SMcuerpo.Celas(1).sAi=SMcuerpo.Param.si_suelo_ruedatrasera;
SMcuerpo.Celas(1).sBi=SMcuerpo.Param.si_rueda_trasera;
SMcuerpo.Celas(1).sA=SMcuerpo.Celas(1).sAi;
SMcuerpo.Celas(1).sB=SMcuerpo.Celas(1).sBi;
SMcuerpo.Celas(1).Fnormcon=0;
SMcuerpo.Celas(1).inden=0;
SMcuerpo.Celas(1).vinden=0;
%%%Contacto elástico_rueda_delantera_terreno%%%
SMcuerpo.Celas(2).A=1;
SMcuerpo.Celas(2).B=4;
SMcuerpo.Celas(2).k_khertz=SMcuerpo.Param.rigidez.Khertz;
SMcuerpo.Celas(2).camort=SMcuerpo.Param.camort.celas;
SMcuerpo.Celas(2).coef_friccion=SMcuerpo.Param.friccion_terreno;
SMcuerpo.Celas(2).vlimite=SMcuerpo.Param.vdesliz;
SMcuerpo.Celas(2).sAi=SMcuerpo.Param.si_suelo_ruedadelantera;
SMcuerpo.Celas(2).sBi=SMcuerpo.Param.si_rueda_delantera;
SMcuerpo.Celas(2).sA=SMcuerpo.Celas(1).sAi;
SMcuerpo.Celas(2).sB=SMcuerpo.Celas(1).sBi;
SMcuerpo.Celas(2).Fnormcon=0;
SMcuerpo.Celas(2).inden=0;
SMcuerpo.Celas(2).vinden=0;
%%%Contacto elástico_rueda_remolque_terreno%%%
SMcuerpo.Celas(3).A=1;
SMcuerpo.Celas(3).B=6;
SMcuerpo.Celas(3).k_khertz=SMcuerpo.Param.rigidez.Khertz;
SMcuerpo.Celas(3).camort=SMcuerpo.Param.camort.celas;
SMcuerpo.Celas(3).coef_friccion=SMcuerpo.Param.friccion_terreno;
SMcuerpo.Celas(3).vlimite=SMcuerpo.Param.vdesliz;
SMcuerpo.Celas(3).sAi=SMcuerpo.Param.si_suelo_ruedaremolque;
SMcuerpo.Celas(3).sBi=SMcuerpo.Param.si_rueda_remolque;
SMcuerpo.Celas(3).sA=SMcuerpo.Celas(1).sAi;
SMcuerpo.Celas(3).sB=SMcuerpo.Celas(1).sBi;
SMcuerpo.Celas(3).Fnormcon=0;
SMcuerpo.Celas(3).inden=0;
SMcuerpo.Celas(2).vinden=0;

%%%COORDENADAS Y VELOCIDADES INICIALES DEL SISTEMA MULTICUERPO%%%
%%%Eslabón fijo%%%
SMcuerpo.Srigidos(1).q0 = [0 0 0]';
SMcuerpo.Srigidos(1).v0 = [0 0 0]';
%%%Rueda_trasera_chasis%%%
SMcuerpo.Srigidos(2).q0 = [0 SMcuerpo.Param.radio_rueda_chasis 0]';
SMcuerpo.Srigidos(2).v0 = [0 0 0]';
%SMcuerpo.Srigidos(2).q0 = [0.0010 0.3801 0.0000]';
%%%Chasis%%%
SMcuerpo.Srigidos(3).q0 = [ SMcuerpo.Param.L_chasis*0.2985
SMcuerpo.Param.radio_rueda_chasis+SMcuerpo.Param.li_muelle_vertical+(0.5*SMcuerpo.Param.H_c
hasis)-((9.81*0.5*SMcuerpo.Param.masa_chasis)/SMcuerpo.Param.rigidez_muellesverticales)
0]';
SMcuerpo.Srigidos(3).v0 = [0 0 0]';
%SMcuerpo.Srigidos(3).q0 =[ 1.4796 1.5502 -0.0004]';
%%%Rueda_delantera_chasis%%%
SMcuerpo.Srigidos(4).q0 = [SMcuerpo.Param.batalla SMcuerpo.Param.radio_rueda_chasis 0]';
SMcuerpo.Srigidos(4).v0 = [0 0 0]';

```

```
%SMcuerpo.Srigidos(4).q0 =[ 2.9567 0.3800 -0.0037]';
%%%Remolque%%%
SMcuerpo.Srigidos(5).q0 = [-
((0.2015*SMcuerpo.Param.L_chasis)+SMcuerpo.Param.separacion_remolchas+(0.5*SMcuerpo.Param.L
_remolque)) SMcuerpo.Param.radio_rueda_remolque+(0.5*SMcuerpo.Param.H_remolque) 0]';
SMcuerpo.Srigidos(5).v0 = [0 0 0]';
%SMcuerpo.Srigidos(5).q0 = [ -2.9958 0.6384 -0.0036]';
%%%Rueda_remolque%%%
SMcuerpo.Srigidos(6).q0 = [-
((0.2015*SMcuerpo.Param.L_chasis)+SMcuerpo.Param.separacion_remolchas+(0.5*SMcuerpo.Param.L
_remolque)) SMcuerpo.Param.radio_rueda_remolque 0]';
SMcuerpo.Srigidos(6).v0 = [0 0 0]';
%SMcuerpo.Srigidos(6).q0 =[-2.9967 0.3884 -0.0017]';
%%Defino en primera instancia la variable 'SMcuerpo.Srigido(i).q' con las
%%coordenadas generalizadas iniciales''
for i=1:SMcuerpo.nsolidos
    SMcuerpo.Srigidos(i).q=SMcuerpo.Srigidos(i).q0;
    SMcuerpo.Srigidos(i).v=SMcuerpo.Srigidos(i).v0;
end
SMcuerpo.q0=zeros(SMcuerpo.ncoortotales,1);
SMcuerpo.v0=zeros(SMcuerpo.ncoortotales,1);
for i=1:SMcuerpo.nsolidos
    SMcuerpo.q0((3*(i-1))+1:(3*(i-1))+3,1)=SMcuerpo.Srigidos(i).q0;
    SMcuerpo.v0((3*(i-1))+1:(3*(i-1))+3,1)=SMcuerpo.Srigidos(i).v0;
end
SMcuerpo.q = SMcuerpo.q0;
SMcuerpo.v = SMcuerpo.v0;
```

```
function [yp]=EcMovimiento_Baumgarte(t,y)
disp(['t = ' num2str(t)]);
global SMcuerpo

ncoor=SMcuerpo.ncoortotales;
ns=SMcuerpo.nsolidos;
nres=SMcuerpo.nrestricciones;
q=y(1:ncoor);
v=y(ncoor+1:2*ncoor);
yp=zeros(2*ncoor,1);

SMcuerpo.q=q;
SMcuerpo.v=v;

for i=1:ns

    SMcuerpo.Srigidos(i).q=q((3*(i-1))+1:(3*(i-1))+3);
    SMcuerpo.Srigidos(i).v=v((3*(i-1))+1:(3*(i-1))+3);
end
alfa=SMcuerpo.Param.alfaBaum;
beta=SMcuerpo.Param.betaBaum;

M=Mmasa(q);
Qg=Qgravedad(q);
Qma=Fma(q,v);
Qcont=Qcontacto(q,v);
Qus=Qusuario(q,v,t);
```

```

Q=Qg+Qma+Qcont+Qus;
Ceq=Restr(q,t);
Cq=Jacobiana(q,t);
Ct=DtRestricciones(q,t);
Cqt=DtJacobiana(q,v,t);
Ctt=DtdtRestricciones(q,v,t);

A=zeros(ncoor+nres,ncoor+nres);
A(1:ncoor,1:ncoor)=M;
A(1:ncoor,ncoor+1:ncoor+nres)=Cq';
A(ncoor+1:ncoor+nres,1:ncoor)=Cq;
B=zeros(ncoor+nres,1);
B(1:ncoor)=Q;
B(ncoor+1:ncoor+nres)=(-(Ctt+(Cqt*v)))+(-2*alfa*((Cq*v)+Ct))+(-(beta^2)*Ceq));
sol=A\B;
Ml=sol(ncoor+1:ncoor+nres);
FR=-Cq'*Ml;
yp(ncoor+1:2*ncoor)=sol(1:ncoor);
yp(1:ncoor)=v;

```

```

function M=Mmasa(q)
global SMCuerpo
ncoor=SMCuerpo.ncoortotales;
nsr=SMCuerpo.nsolidos;
M=zeros(ncoor,ncoor);

for i=1:nsr
    m=SMCuerpo.Srigidos(i).masa;
    I=SMCuerpo.Srigidos(i).Minercia;
    mm=eye(2)*m;
    M((3*(i-1))+1:(3*(i-1))+2,(3*(i-1))+1:(3*(i-1))+2)=mm;
    M((3*(i-1))+3,(3*(i-1))+3)=I;
end

```

```

function Cq=Jacobiana(q,t)
global SMCuerpo
nq=SMCuerpo.ncoortotales;
nar=SMCuerpo.narticulaciones;
Cqus=Jacobianausuario(q,t);
Cqar=zeros(2*nar,nq);
Cqef=zeros(3,SMCuerpo.ncoortotales);
Cqef(1,1)=1;
Cqef(2,2)=1;
Cqef(3,3)=1;

for i=1:nar
    na=SMCuerpo.Pares.revolucion(i).A;
    ula=SMCuerpo.Pares.revolucion(i).ula;
    qa=q((3*(na-1))+1:(3*(na-1))+3);

```

```

Aa=DerivaMatRot(qa);

nb=SMcuerpo.Pares.revolucion(i).B;
ulb=SMcuerpo.Pares.revolucion(i).ulb;
qb=q((3*(nb-1))+1:(3*(nb-1))+3);
Ab=DerivaMatRot(qb);

Cqar((2*(i-1))+1:(2*(i-1))+2,(3*(na-1))+1:(3*(na-1))+2)=eye(2);
Cqar((2*(i-1))+1:(2*(i-1))+2,(3*(nb-1))+1:(3*(nb-1))+2)=-eye(2);
Cqar((2*(i-1))+1:(2*(i-1))+2,(3*(na-1))+3)=Aa*ula;
Cqar((2*(i-1))+1:(2*(i-1))+2,(3*(nb-1))+3)=-Ab*ulb;

end
Cq=[ Cqef' Cqar' Cqus']';

```

```

function Cqus=Jacobianausuario(q,t)
global SMcuerpo
nrus=SMcuerpo.nrestrusuario;
ncoor=SMcuerpo.ncoortotales;
Cqus=zeros(nrus,ncoor);
if nrus==2
    Cqus(1,6)=1;
    Cqus(2,12)=1;
end
if nrus==1
if SMcuerpo.Param.wrueda_trasera >= 0
    Cqus(1,6)=1;
end
if SMcuerpo.Param.wrueda_delantera >= 0
    Cqus(1,12)=1;
end
end
end

```

```

function C=Restricciones(q,t)
global SMcuerpo

nar=SMcuerpo.narticulaciones;
Car=zeros(2*nar,1);
cef=zeros(1,3)';
Cus=Restrus(q,t);
cef(1)=q(1);
cef(2)=q(2);
cef(3)=q(3);
for i=1:nar
    na=SMcuerpo.Pares.revolucion(i).A;
    qa=q((3*(na-1))+1:(3*(na-1))+3);
    ua=SMcuerpo.Pares.revolucion(i).ula;
    uga=Coor_global(qa,ua);

```

```

        nb=SMcuerpo.Pares.revolucion(i).B;
        qb=q((3*(nb-1))+1:(3*(nb-1))+3);
        ub=SMcuerpo.Pares.revolucion(i).ulb;
        ugb=Coor_global(qb,ub);

        Car((2*(i-1))+1:(2*(i-1))+2)=uga-ugb;
    end

    C=[Cef' Car' Cus']';
end

```

```

function Cu=Restrus(q,t)
global SMcuerpo
nrus=SMcuerpo.nrestrusuario;
Cu=zeros(nrus,1);
wt=SMcuerpo.Param.wrueda_trasera;
wd=SMcuerpo.Param.wrueda_delantera;
if nrus==2
    Cu(1)=q(6)-(wt*t);
    Cu(2)=q(12)-(wd*t);
end
if nrus==1
    if wt>=0
        Cu(1)=q(6)-(wt*t);
    else
        Cu(1)=q(12)-(wd*t);
    end
end
end

```

```

function Ct=DtRestricciones(q,t)
global SMcuerpo
nar=SMcuerpo.narticulaciones;
Ctar=zeros(2*nar,1);
nrus=SMcuerpo.nrestrusuario;
Ctus=zeros(nrus,1);
wt=SMcuerpo.Param.wrueda_trasera;
wd=SMcuerpo.Param.wrueda_delantera;
Ctef=zeros(3,1);
if nrus==2
    Ctus(1)=-wt;
    Ctus(2)=-wd;
end
if nrus==1
    if wt>=0
        Ctus(1)=-wt;
    else
        Ctus(1)=-wd;
    end
end

```

```
end
Ct=[ Ctef' Ctar' Ctus']';
```

```
function Ctt=DtdtRestricciones(q,v,t)
global SMCuerpo
nres=SMCuerpo.nrestricciones;
Ctt=zeros(nres,1);
```

```
function Cqt=DtJacobiana(q,v,t)
global SMCuerpo
ncoor=SMCuerpo.ncoortotales;
nar=SMCuerpo.narticulaciones;
nrus=SMCuerpo.nrestrusuario;
Cqtar=zeros(2*nar,ncoor);
Cqtef=zeros(3,ncoor);
Cqtus=zeros(nrus,ncoor);
for i=1:nar
    na=SMCuerpo.Pares.revolucion(i).A;
    ula=SMCuerpo.Pares.revolucion(i).ula;
    qa=q((3*(na-1))+1:(3*(na-1))+3);
    Aa=Matriz_transformacion(qa);
    wa=v((3*(na-1))+3);

    nb=SMCuerpo.Pares.revolucion(i).B;
    ulb=SMCuerpo.Pares.revolucion(i).ulb;
    qb=q((3*(nb-1))+1:(3*(nb-1))+3);
    Ab=Matriz_transformacion(qb);
    wb=v((3*(nb-1))+3);

    Cqtar((2*(i-1))+1:(2*(i-1))+2,(3*(na-1))+1:(3*(na-1))+2)=zeros(2,2);
    Cqtar((2*(i-1))+1:(2*(i-1))+2,(3*(nb-1))+1:(3*(nb-1))+2)=zeros(2,2);
    Cqtar((2*(i-1))+1:(2*(i-1))+2,(3*(na-1))+3)=-wa*Aa*ula;
    Cqtar((2*(i-1))+1:(2*(i-1))+2,(3*(nb-1))+3)=wb*Ab*ulb;
end
Cqt=[Cqtef' Cqtar' Cqtus']';
```

```
function Qg=Qgravedad(q)
global SMCuerpo
ncoor=SMCuerpo.ncoortotales;
Qg=zeros(ncoor,1);
ns=SMCuerpo.nsolidos;
g=SMCuerpo.Param.g;
for i=1:ns
    m=SMCuerpo.Srigidos(i).masa;
    Qg((3*(i-1))+2)=-m*g;
end
```

```
function Qus=Qusuario(q,v,t)
global SMCuerpo
ncoor=SMCuerpo.ncoortotales;
Qus=zeros(ncoor,1);
if SMCuerpo.Par==1
    if SMCuerpo.Parrrt>=0 && SMCuerpo.Parrrd>=0
        Qus(6)=-SMCuerpo.Parrrt;
        Qus(12)=-SMCuerpo.Parrrd;
    end
    if SMCuerpo.Parrrt>=0 && SMCuerpo.Parrrd==0
        Qus(6)=-SMCuerpo.Parrrt;
    end
    if SMCuerpo.Parrrd>=0 && SMCuerpo.Parrrt==0
        Qus(12)=-SMCuerpo.Parrrd;
    end
end
end
```

```
function Qma=Qma(q,v)
global SMCuerpo
nt=SMCuerpo.ncoortotales;
Qma=zeros(nt,1);
nma=SMCuerpo.Param.nmuelesamort;

for i=1:nma

    na=SMCuerpo.Ma(i).A;
    qa=q((3*(na-1))+1:(3*(na-1))+3);
    va=v((3*(na-1))+1:(3*(na-1))+3);
    ula=SMCuerpo.Ma(i).ua;
    Mata=DerivaMatRot(qa);
    uga=Coor_global(qa,ula);

    nb=SMCuerpo.Ma(i).B;
    qb=q((3*(nb-1))+1:(3*(nb-1))+3);
    vb=v((3*(nb-1))+1:(3*(nb-1))+3);
    ulb=SMCuerpo.Ma(i).ub;
    Matb=DerivaMatRot(qb);
    ugb=Coor_global(qb,ulb);

    Rab=uga-ugb;
    Rabnorm=Rab/norm(Rab);
    l=norm(Rab);
    km=SMCuerpo.Ma(i).k;
    ca=SMCuerpo.Ma(i).c;
    l0=SMCuerpo.Ma(i).l indef;
    delta1=l-l0;
    Drabq=[eye(2) Mata*ula -eye(2) -Matb*ulb];
    qp=[va' vb']';
    lp=Rabnorm'*Drabq*qp;
    fe=km*delta1;
    fa=ca*lp;
    ft=fe+fa;

    Qa=-ft*[eye(2);ula'*Mata']*Rabnorm;
```

```
Qb=ft*[eye(2);ulb'*Matb']*Rabnorm;
Qma((3*(na-1))+1:(3*(na-1))+3)=Qa;
Qmb((3*(nb-1))+1:(3*(nb-1))+3)=Qb;
```

```
end
```

```
function Qc=Qcontacto(q,v)
global SMCuerpo
nt=SMCuerpo.ncoortotales;
ncon=SMCuerpo.nce;
Qc=zeros(nt,1);
global cactual
for i=1:ncon
    cactual=i;
    na=SMCuerpo.Ce1as(i).A;
    qa=q((3*(na-1))+1:(3*(na-1))+3);
    va=v((3*(na-1))+1:(3*(na-1))+3);
    MatRota=Matriz_transformacion(qa);
    SMCuerpo.Srigidos(na).q=qa;
    SMCuerpo.Srigidos(na).v=va;
    sa=SMCuerpo.Ce1as(i).SA;

    nb=SMCuerpo.Ce1as(i).B;
    qb=q((3*(nb-1))+1:(3*(nb-1))+3);
    vb=v((3*(nb-1))+1:(3*(nb-1))+3);
    MatRotb=Matriz_transformacion(qb);
    SMCuerpo.Srigidos(nb).q=qb;
    SMCuerpo.Srigidos(nb).v=vb;
    sb=SMCuerpo.Ce1as(i).SB;

    %%Condiciones iniciales para resolver el sistema no lineal%%
    s0=[sa sb]';
    options = optimset('MaxIter',100,'MaxFunEvals',1000,'TolFun',10^(-10),'TolCon',10^(-
10),'Diagnostics','off','Display','off');
    scontact= fsolve(@Evalua_sisteman1_contacto,s0,options);
    sa=scontact(1);
    sb=scontact(2);
    SMCuerpo.Ce1as(i).SA=sa;
    SMCuerpo.Ce1as(i).SB=sb;
    [y,yp,ypp,vposa,vtanga,vnorma]=Parametriza_terreno(sa);
    [vposb,vtangb,vnormb]=Parametriza_Rueda(sb);
    Ra=Coor_global(qa,vposa);
    Rb=Coor_global(qb,vposb);
    Ta=MatRota*vtanga;
    Tb=MatRotb*vtangb;
    Na=MatRota*vnorma;
    Nb=MatRotb*vnormb;
    Rba=Rb-Ra;
    Nnorma=Na/norm(Na);
    Nnormb=Nb/norm(Nb);
    ind=Nnorma'*Rba;
    indab(1,i)=ind;
    DMatRota=DerivaMatRot(qa);
    DMatRotb=DerivaMatRot(qb);
```

```

vpa=(va(1:2))+(va(3)*DMatRota*vposa);
vpb=(vb(1:2))+(vb(3)*DMatRotb*vposb);
vrelab=vpa-vpb;
vrel=norm(vrelab);
vrelnorm=vrelab/vrel;
vind=-vrelab'*Nnorma;
SMcuerpo.Celas(i).inden=ind;
SMcuerpo.Celas(i).vinden=vind;
khertz=SMcuerpo.Celas(i).k_khertz;
cam=SMcuerpo.Celas(i).camort;
if ind>=0
    fe= khertz*(ind^(1.5));
    fa=cam*vind*sqrt(ind*ind);
    fn=fe+fa;
    if fn<0
        fn=0;
    end
else
    fn=0;
end
Fa=fn*Nnorma;
Fb=fn*Nnormb;
Qc(3*(na-1)+1:3*(na-1)+2,1) = Qc(3*(na-1)+1:3*(na-1)+2,1)+Fa;
Qc(3*(nb-1)+1:3*(nb-1)+2,1) = Qc(3*(nb-1)+1:3*(nb-1)+2,1)+Fb;
Ra=MatRota*vposa;
Rb=MatRotb*vposb;
Fb=[Fb' 0]';
Fa=[Fa' 0]';
Ra=[Ra' 0]';
Rb=[Rb' 0]';
Mna=cross(Ra,Fa);
Mnb=cross(Rb,Fb);
Mna=Mna(3);
Mnb=Mnb(3);
Qc(3*(na-1)+3,1) = Qc(3*(na-1)+3,1)+Mna;
Qc(3*(nb-1)+3,1) = Qc(3*(nb-1)+3,1)+Mnb;
if vrel<SMcuerpo.Celas(i).vlimite
    ft = (vrel/SMcuerpo.Celas(i).vlimite)*SMcuerpo.Celas(i).coef_friccion*fn;
else
    ft = SMcuerpo.Celas(i).coef_friccion*fn;
end
if vrel>0
    Ft=-ft*vrelnorm;
else
    Ft=[0 0]';
end
Qc(3*(na-1)+1:3*(na-1)+2,1) = Qc(3*(na-1)+1:3*(na-1)+2,1)+Ft;
Qc(3*(nb-1)+1:3*(nb-1)+2,1) = Qc(3*(nb-1)+1:3*(nb-1)+2,1)-Ft;
Ft=[Ft' 0]';
Mtana=cross(Ra,Ft);
Mtanb=cross(Rb,-Ft);
Mtana=Mtana(3);
Mtanb=Mtanb(3);
Qc(3*(na-1)+3,1) = Qc(3*(na-1)+3,1)+Mtana;
Qc(3*(nb-1)+3,1) = Qc(3*(nb-1)+3,1)+Mtanb;
end

```

```
function [CC]=Evalua_sisteman1_contacto(s)
global SMCuerpo
global cactual
i=cactual;
sa=s(1);
sb=s(2);
na=SMCuerpo.Celas(i).A;
nb=SMCuerpo.Celas(i).B;
qa=SMCuerpo.q((3*(na-1))+1:(3*(na-1))+3,1);
qb=SMCuerpo.q((3*(nb-1))+1:(3*(nb-1))+3,1);
[y yp ypp vposa vtanga vnorma] = Parametriza_terreno(sa);
[vposb vtangb vnormb] = Parametriza_Rueda(sb);
MatRota=Matriz_transformacion(qa);
MatRotb=Matriz_transformacion(qb);
Ra=Coor_global(qa,vposa);
Rb=Coor_global(qb,vposb);
Ta=MatRota*vtanga;
Nb=MatRotb*vnormb;
Rab=Rb-Ra;
CC(1,1)=Ta'*Rab;
CC(2,1)=Ta'*Nb;
```

```
function ug=Coor_global(q,u1)

R=q(1:2);
Matrot=Matriz_transformacion(q);
ug=R+(Matrot*u1);
```

```
function Matrot=Matriz_transformacion(q)
teta=q(3);
a=cos(teta);
b=sin(teta);
Matrot=[a -b;b a];
```

```
function Ateta=DerivaMatRot (q)

Ateta=zeros(2,2);
teta=q(3);
Ateta=[-sin(teta) -cos(teta);cos(teta) -sin(teta)];
```

```
function [y,yp,ypp,vpos,vtang,vnorm]=Parametriza_terreno(s)
global SMCuerpo
yi=0;
p=SMCuerpo.p;
```

```

xi=SMcuerpo.xi;
xf=SMcuerpo.xf;
m=SMcuerpo.m;
yf=SMcuerpo.yf;
g=SMcuerpo.gradopolin;
batche=SMcuerpo.batche;
x1=SMcuerpo.bachex1;
x2=SMcuerpo.bachex2;
x3=SMcuerpo.bachex3;
x4=SMcuerpo.bachex4;
x5=SMcuerpo.bachex5;
m1=SMcuerpo.bachem1;
m2=SMcuerpo.bachem2;
xf1=SMcuerpo.xf1;
xf2=SMcuerpo.xf2;
tr=SMcuerpo.tr;
nar=SMcuerpo.Param.narmonicos;
ampl=SMcuerpo.Param.ampl;
fi=SMcuerpo.Param.fi;
lambda=SMcuerpo.Param.lambda;
y=0;yp=0;ypp=0;
if batche==1

A1=[2*x1 1 0;x1^2 x1 1;2*x2 1 0];
B1=[0 0 m1]';
sol1=A1\B1;
a1=sol1(1);b1=sol1(2);c1=sol1(3);
A2=[2*x2 1 0;x2^2 x2 1;2*x3 1 0];
B2=[m1 (a1*(x2^2))+(b1*x2)+c1 0]';
sol2=A2\B2;
a2=sol2(1);b2=sol2(2);c2=sol2(3);
A3=[2*x3 1 0;x3^2 x3 1;2*x4 1 0];
B3=[0 (a2*(x3^2))+(b2*x3)+c2 m2]';
sol3=A3\B3;
a3=sol3(1);b3=sol3(2);c3=sol3(3);
A4=[2*x4 1 0;x4^2 x4 1;2*x5 1 0];
B4=[m2 (a3*(x4^2))+(b3*x4)+c3 0]';
sol4=A4\B4;
a4=sol4(1);b4=sol4(2);c4=sol4(3);
    if s<x1
        y=0;
        yp=0;
        ypp=0;
    end
    if s>x1 && s<x2
        y=(a1*(s^2))+(b1*s)+(c1);
        yp=(2*a1*s)+(b1);
        ypp=2*a1;
    end
    if s>x2 && s<x3
        y=(a2*(s^2))+(b2*s)+(c2);
        yp=(2*a2*s)+(b2);
        ypp=2*a2;
    end
    if s>x3 && s<x4
        y=(a3*(s^2))+(b3*s)+(c3);
        yp=(2*a3*s)+(b3);
    end
end

```

```

        ypp=2*a3;

    end
    if s>x4 && s<x5
        y=(a4*(s^2))+(b4*s)+(c4);
        yp=(2*a4*s)+(b4);
        ypp=2*a4;
    end
    if s>x5
        y=(a4*(x5^2))+(b4*x5)+(c4);
        yp=0;
        ypp=0;
    end
end
if p==1
    if g==2
        if tr==0
            A=[2*xi 1 0 ;xi^2 xi 1;2*xf 1 0];
            B=[0;yi;m];
            sol=A\B;
            b=(sol(1)*xf^2)+(sol(2)*xf)+(sol(3))-(m*xf);
            if s<=xi
                y=yi;
                yp=0;
                ypp=0;
            elseif s>=xi && s<=xf
                y=(sol(1)*s^2)+(sol(2)*s)+(sol(3));
                yp=(2*sol(1)*s)+(sol(2));
                ypp=2*sol(1);
            elseif s>=xf
                y=(m*s)+b;
                yp=m;
                ypp=0;
            end
        else
            A=[2*xi 1 0 ;xi^2 xi 1;2*xf 1 0];
            B=[0;yi;m];
            sol=A\B;
            b=(sol(1)*xf^2)+(sol(2)*xf)+(sol(3))-(m*xf);
            A1=[2*xf1 1 0;xf1^2 xf1 1;2*xf2 1 0];
            B1=[m;(m*xf1)+(b);0];
            sol2=A1\B1;
            if s<=xi
                y=yi;
                yp=0;
                ypp=0;
            elseif s>=xi && s<=xf
                y=(sol(1)*s^2)+(sol(2)*s)+(sol(3));
                yp=(2*sol(1)*s)+(sol(2));
                ypp=2*sol(1);
            elseif s>=xf && s<=xf1
                y=(m*s)+b;
                yp=m;
                ypp=0;
            end
        end
    end
end

```

```

elseif s>xf1 && s<=xf2
    y=(sol2(1)*s^2)+(sol2(2)*s)+(sol2(3));
    yp=(2*sol2(1)*s)+(sol2(2));
    ypp=2*sol2(1);

elseif s>xf2
    y=(sol2(1)*xf2^2)+(sol2(2)*xf2)+(sol2(3));
    yp=0;
    ypp=0;

end
    end
    end
    if g==3
        A=[3*xi^2 2*xi 1 0;xi^3 xi^2 xi 1;3*xf^2 2*xf 1 0;xf^3 xf^2 xf 1];
        B=[0;yi;m;yf];
        sol=A\B;
        b=yf-(m*xf);
        if s<=xi
            y=yi;
            yp=0;
            ypp=0;

        elseif s>=xi && s<=xf
            y=(sol(1)*s^3)+(sol(2)*s^2)+(sol(3)*s)+sol(4);
            yp=(3*sol(1)*s^2)+(2*s*sol(2))+sol(3);
            ypp=(6*sol(1)*s)+(2*sol(2));

        elseif s>=xf
            y=(m*s)+b;
            yp=m;
            ypp=0;

        end

    end

end

for i = 1:nar
    y = y + ampl(i)*sin((2*pi/lambda(i))*s+fi(i));
    yp = yp + ampl(i)*cos((2*pi/lambda(i))*s+fi(i))*(2*pi/lambda(i));
    ypp = ypp - ampl(i)*sin((2*pi/lambda(i))*s+fi(i))*((2*pi/lambda(i)))^2;

end
vpos=[s y]';
vtang=[1 yp]';
vnorm=[yp -1]';
    
```

```

function [vpos,vtang,vnorm]=Parametriza_Rueda(alfa)
global SMCuerpo
Radio=SMCuerpo.Param.radio_rueda_chasis;
vpos=[Radio*cos(alfa),Radio*sin(alfa)]';
    
```

```
vtang=[-Radio*sin(alfa),Radio*cos(alfa)]';
vnorm=[-Radio*cos(alfa),-Radio*sin(alfa)]';
```

```
clear all
global SMCuerpo
SMCuerpo=Definicion_Modelo;
dt=1e-3;
tf=0.125;
tspan=[0 tf];
ncoor=SMCuerpo.ncoortotales;
nres=SMCuerpo.nrestricciones;
[t y] = ode15s(@Obten_derivada_Baumgarte,tspan,[SMCuerpo.q0' SMCuerpo.v0']');
y=y';
q=y(1:ncoor,:);
v=y(ncoor+1:2*ncoor,:);
SMCuerpo.Celas(1).sA=0;
SMCuerpo.Celas(1).sB=-pi/2;
SMCuerpo.Celas(2).sA=0;
SMCuerpo.Celas(2).sB=-pi/2;
SMCuerpo.Celas(3).sA=0;
SMCuerpo.Celas(3).sB=-pi/2;

PPQma=zeros(ncoor,length(t));
PPQcont=zeros(ncoor,length(t));
PPQg=zeros(ncoor,length(t));
PPQus=zeros(ncoor,length(t));
PPQ=zeros(ncoor,length(t));

vppma=12;
PPma=zeros(vppma*SMCuerpo.nma,length(t));

vppcont=33;
PPcont=zeros(vppcont*SMCuerpo.nce,length(t));

PPRestr=zeros(nres,length(t));
PPJacob=zeros(nres,ncoor*length(t));
PPCt=zeros(nres,length(t));
PPJacobdt=zeros(nres,ncoor*length(t));
PPCttdt=zeros(nres,length(t));
PPFR=zeros(ncoor,length(t));

PPM=Mmasa(q(:,end));
PPG=Qgravedad(q(:,end));
PPFUS =Qusuario(q(:,end),v(:,end),t);

for i = 1:length(t)
    [PPma(:,i),PPcont(:,i),PPRestr(:,i),PPJacob(:,(ncoor*(i-1))+1:(ncoor*(i-1))+ncoor),PPCt(:,i),PPJacobdt(:,(ncoor*(i-1))+1:(ncoor*(i-1))+ncoor),PPCttdt(:,i),PPFR(:,i),PPQma(:,i),PPQcont(:,i),PPQg(:,i),PPQus(:,i),PPQ(:,i))] =
    Posproceso2(t(i),y(:,i));
end

%%RESAMPLING%%
ttt=t(1):0.001:t(end);
%%Coordenadas generalizadas%%
```

```

q2=interp1(t,q',tst);
v2=interp1(t,v',tst);
%%muelles%%
[m,n]=size(PPma);
for i = 1:m
    PPma2(i,:) = interp1(t,PPma(i,:),tst);
end
%PPma = PPma2; clear PPma2
%%contacto%%
[m,n]=size(PPcont);
for i = 1:m
    PPcont2(i,:) = interp1(t,PPcont(i,:),tst);
end
%PPcont=PPcont2;clear PPcont2
%%restricciones%%
[m,n]=size(PPRestr);
for i=1:m
    PPRestr2(i,:)=interp1(t,PPRestr(i,:),tst);
end
%PPRestr=PPRestr2;clear PPRestr2
%%Derivadas de las restricciones respecto al tiempo%%
[m,n]=size(PPCt);
for i=1:m
    PPCt2(i,:)=interp1(t,PPCt(i,:),tst);
end
% PPCt=PPCt2;clear PPCt2
%%Derivada segunda de las restricciones respecto al tiempo%%
[m,n]=size(PPCtt);
for i=1:m
    PPCtt2(i,:)=interp1(t,PPCtt(i,:),tst);
end
% PPCtt=PPCtt2;clear PPCtt2
%%Fuerzas generalizadas de reacción%%
[m,n]=size(PPFR);
for i=1:m
    PPFR2(i,:)=interp1(t,PPFR(i,:),tst);
end
% PPFR=PPFR2;clear PPFR2
%%Fuerzas generalizadas de los muelles y amortiguadores%%
[m,n]=size(PPQma);
for i=1:m
    PPQma2(i,:)=interp1(t,PPQma(i,:),tst);
end
% PPQma=PPQma2;clear PPQma2
%%Fuerzas generalizadas de contacto%%
[m,n]=size(PPQcont);
for i=1:m
    PPQcont2(i,:)=interp1(t,PPQcont(i,:),tst);
end
% PPQcont=PPQcont2;clear PPQcont2
%%Fuerzas generalizadas de la gravedad%%
[m,n]=size(PPQg);
for i=1:m
    PPQg2(i,:)=interp1(t,PPQg(i,:),tst);
end
%%Fuerzas generalizadas de usuario%%
[m,n]=size(PPQus);
for i=1:m

```

```

        PPQus2(i,:)=interp1(t,PPQus(i,:),tst);
    end
%%Fuerzas generalizadas totales%%
[m,n]=size(PPQ);
for i=1:m
    PPQ2(i,:)=interp1(t,PPQ(i,:),tst);
end

```

```

function [t,y]=Runge_kutta(f,dt,vt,y0)
global SMCuerpo
ncoor=SMCuerpo.ncoortotales;
y=zeros(2*ncoor,length(vt));
t=vt;
y(:,1)=y0;

for i=2:length(vt)
    yev=y(:,i-1);
    tev=vt(i-1);
    f1=feval(f,tev,yev);
    f2=feval(f,tev+(0.5*dt),yev+(0.5*f1));
    f3=feval(f,tev+(0.5*dt),yev+(0.5*f2));
    f4=feval(f,tev+dt,yev+f3);
    y(:,i)=yev+(1/6)*dt*(f1+2*f2+2*f3+f4);
end

```

```

function
[PPma,PPcont,PPrestr,PPJacob,PPCt,PPJacobdt,PPCtt,PPFR,PPQma,PPQcont,PPQg,PPQus,PPQ]=Pproce
so(t,y)
global SMCuerpo
global cactual
%t
disp(['t = ' num2str(t)]);

q=y(1:SMCuerpo.ncoortotales,1);
v=y(SMCuerpo.ncoortotales+1:2*SMCuerpo.ncoortotales,1);

    SMCuerpo.q = q;
    SMCuerpo.v = v;
    for i=1:SMCuerpo.nsolidos
        SMCuerpo.Srigidos(i).q=q((3*(i-1))+1:(3*(i-1))+3);
        SMCuerpo.Srigidos(i).v=v((3*(i-1))+1:(3*(i-1))+3);
    end
    PPQma=Fma(q,v);
    PPQcont=Qcontacto(q,v);
    PPQg=Qgravedad(q);
    PPQus=Qusuario(q,v,t);

%%POST-PROCESO DE LAS VARIABLES DE LOS CONJUNTOS MUELLE-AMORTIGUADOR%%
vppma=12; %longitud de los muelles,velocidad de deformación,fuerza del muelle,fuerza
amortiguadora

```

```

PPma=zeros(vppma*SMcuerpo.nma,1);

for i = 1:SMcuerpo.nma

    na=SMcuerpo.Ma(i).A;
    qa=q((3*(na-1))+1:(3*(na-1))+3);
    va=v((3*(na-1))+1:(3*(na-1))+3);
    ula=SMcuerpo.Ma(i).ua;
    Mata=DerivaMatRot(qa);
    uga=Coor_global(qa,ula);

    nb=SMcuerpo.Ma(i).B;
    qb=q((3*(nb-1))+1:(3*(nb-1))+3);
    vb=v((3*(nb-1))+1:(3*(nb-1))+3);
    ulb=SMcuerpo.Ma(i).ub;
    Matb=DerivaMatRot(qb);
    ugb=Coor_global(qb,ulb);

    Rab=uga-ugb;
    Rabnorm=Rab/norm(Rab);
    l=norm(Rab);
    km=SMcuerpo.Ma(i).k;
    ca=SMcuerpo.Ma(i).c;
    l0=SMcuerpo.Ma(i).l indef;
    delta1=l-l0;
    Drabq=[eye(2) Mata*ula -eye(2) -Matb*ulb];
    qp=[va' vb']';
    lp=Rabnorm'*Drabq*qp;
    fe=km*delta1;
    fa=ca*lp;
    ft=fe+fa;

    Qa=-ft*[eye(2);ula'*Mata']*Rabnorm;
    Fax=Qa(1);
    Fay=Qa(2);
    Ma=Qa(3);
    Qb=ft*[eye(2);ulb'*Matb']*Rabnorm;
    Fbx=Qb(1);
    Fby=Qb(2);
    Mb=Qb(3);
    Qma((3*(na-1))+1:(3*(na-1))+3)=Qa;
    Qmb((3*(nb-1))+1:(3*(nb-1))+3)=Qb;

    PPma((vppma*(i-1))+1:(vppma*(i-1))+vppma)=[l lp fe fa Rab' Fax Fay Ma Fbx Fby Mb]';
end
%%POST-PROCESO DE LAS VARIABLES DE CONTACTO%%
vppcont=33;
PPcont=zeros(vppcont*SMcuerpo.nce,1);
Qc = zeros(SMcuerpo.ncoortotales,1);
for i = 1:SMcuerpo.nce,

    actual=i;
    na=SMcuerpo.Celas(i).A;
    qa=q((3*(na-1))+1:(3*(na-1))+3);
    va=v((3*(na-1))+1:(3*(na-1))+3);
    MatRota=Matriz_transformacion(qa);

```

```

SMcuerpo.Srigidos(na).q=qa;
SMcuerpo.Srigidos(na).v=va;
sa=SMcuerpo.Celas(i).sA;

nb=SMcuerpo.Celas(i).B;
qb=q((3*(nb-1))+1:(3*(nb-1))+3);
vb=v((3*(nb-1))+1:(3*(nb-1))+3);
MatRotb=Matriz_transformacion(qb);
SMcuerpo.Srigidos(nb).q=qb;
SMcuerpo.Srigidos(nb).v=vb;
sb=SMcuerpo.Celas(i).sB;

%%Condiciones iniciales para resolver el sistema no lineal%%
s0=[sa sb]';
options = optimset('MaxIter',100,'MaxFunEvals',1000,'TolFun',10^(-10),'TolCon',10^(-
10),'Diagnostics','off','Display','off');
scontact= fsolve(@Evalua_sistemanl_contacto,s0,options);
sa=scontact(1);
sb=scontact(2);
SMcuerpo.Celas(i).sA=sa;
SMcuerpo.Celas(i).sB=sb;
[y,yp,ypb,vposa,vtanga,vnorma]=Parametriza_terreno(sa);
[vposb,vtangb,vnormb]=Parametriza_Rueda(sb);
Ra=Coor_global(qa,vposa);
Rb=Coor_global(qb,vposb);
Ta=MatRota*vtanga;
Tb=MatRotb*vtangb;
Na=MatRota*vnorma;
Nb=MatRotb*vnormb;
Rba=Rb-Ra;
Nnorma=Na/norm(Na);
Nnormb=Nb/norm(Nb);
ind=Nnorma'*Rba;
indab(1,i)=ind;
DMatRota=DerivaMatRot(qa);
DMatRotb=DerivaMatRot(qb);
vpa=(va(1:2))+(va(3)*DMatRota*vposa);
vpb=(vb(1:2))+(vb(3)*DMatRotb*vposb);

vrelb=vb(3)*DMatRotb*vposb;
vrelbx=vrelb(1);
vrelby=vrelb(2);
vb1=vb(1);
vb2=vb(2);
vrelab=vpa-vpb;
vrel=norm(vrelab);
vrelnorm=vrelab/vrel;
vind=-vrelab'*Nnorma;
SMcuerpo.Celas(i).inden=ind;
SMcuerpo.Celas(i).vinden=vind;
khertz=SMcuerpo.Celas(i).k_khertz;
cam=SMcuerpo.Celas(i).camort;
if ind>=0
    fe= khertz*(ind^(1.5));
    fa=cam*vind*sqrt(ind*ind);
    fn=fe+fa;
    if fn<0
        fn=0;
    end
end

```

```

        end
    else
        fn=0;
    end
    Fa=fn*Nnorma;
    Fb=fn*Nnormb;
    Fnormx=Fb(1);
    Fnormy=Fb(2);
    Qc(3*(na-1)+1:3*(na-1)+2,1) = Qc(3*(na-1)+1:3*(na-1)+2,1)+Fa;
    Qc(3*(nb-1)+1:3*(nb-1)+2,1) = Qc(3*(nb-1)+1:3*(nb-1)+2,1)+Fb;
    Ra=MatRota*vposa;
    Rb=MatRotb*vposb;
    Fb=[Fb' 0]';
    Fa=[Fa' 0]';
    Ra=[Ra' 0]';
    Rb=[Rb' 0]';
    Mna=cross(Ra,Fa);
    Mnb=cross(Rb,Fb);
    Mna=Mna(3);
    Mnb=Mnb(3);
    Qc(3*(na-1)+3,1) = Qc(3*(na-1)+3,1)+Mna;
    Qc(3*(nb-1)+3,1) = Qc(3*(nb-1)+3,1)+Mnb;
    if vrel<SMcuerpo.Celas(i).vlimite
        ft = (vrel/SMcuerpo.Celas(i).vlimite)*SMcuerpo.Celas(i).coef_friccion*fn;
    else
        ft = SMcuerpo.Celas(i).coef_friccion*fn;
    end
    if vrel>0
        Ft=-ft*vrelnorm;
    else
        Ft=[0 0]';
    end
    Ftanx=Ft(1);
    Ftany=Ft(2);
    Qc(3*(na-1)+1:3*(na-1)+2,1) = Qc(3*(na-1)+1:3*(na-1)+2,1)+Ft;
    Qc(3*(nb-1)+1:3*(nb-1)+2,1) = Qc(3*(nb-1)+1:3*(nb-1)+2,1)-Ft;
    Ft=[Ft' 0]';
    Mtana=cross(Ra,Ft);
    Mtanb=cross(Rb,-Ft);
    Ft=Ft(1:2);
    Mtana=Mtana(3);
    Mtanb=Mtanb(3);
    Qc(3*(na-1)+3,1) = Qc(3*(na-1)+3,1)+Mtana;
    Qc(3*(nb-1)+3,1) = Qc(3*(nb-1)+3,1)+Mtanb;

    PPcont((vppcont*(i-1))+1:(vppcont*(i-1))+vppcont)=[sa sb vposa' Ta' Na' vposb' Tb' Nb'
ind vind fe fa fn Ft' vrel vb1 vb2 vrelbx vrelby Fnormx Fnormy Mnb Ftanx Ftany Mtanb ft]';

end

PPRestr= Restr(q,t);
PPJacob = Jacobiana(q,t);
PPCt=DtRestricciones(q,t);
PPJacobdt=DtJacobiana(q,v,t);
PPCtt = DtdtRestricciones(q,v,t);

alfa=SMcuerpo.Param.alfaBaum;

```

```
beta=SMcuerpo.Param.betaBaum;
```

```
M=Mmasa(q);
Qg=Qgravedad(q);
Qma=Fma(q,v);
Qcont=Qcontacto(q,v);
Qus=Qusuario(q,v,t);
Q=Qg+Qma+Qcont+Qus;
Ceq=Restr(q,t);
Cq=Jacobiana(q,t);
Ct=DtRestricciones(q,t);
Cqt=DtJacobiana(q,v,t);
Ctt=DtdtRestricciones(q,v,t);

nres=SMcuerpo.nrestricciones;
ncoor=SMcuerpo.ncoortotales;
A=zeros(ncoor+nres,ncoor+nres);
A(1:ncoor,1:ncoor)=M;
A(1:ncoor,ncoor+1:ncoor+nres)=Cq';
A(ncoor+1:ncoor+nres,1:ncoor)=Cq;
B=zeros(ncoor+nres,1);
B(1:ncoor)=Q;
B(ncoor+1:ncoor+nres)=(-(Ctt+(Cqt*v)))+(-2*alfa*((Cq*v)+Ct))+(-((beta^2)*Ceq));
sol=A\B;
Ml=sol(ncoor+1:ncoor+nres);
PPFR=-Cq'*Ml;
PPQ=PPQma+PPQcont+PPQus+PPFR;
```

```
function plotposiciones(t,q)
Rx2=q((3*(2-1))+1,:);Ry2=q((3*(2-1))+2,:);teta2=q((3*(2-1))+3,:);
Rx3=q((3*(3-1))+1,:);Ry3=q((3*(3-1))+2,:);teta3=q((3*(3-1))+3,:);
Rx4=q((3*(4-1))+1,:);Ry4=q((3*(4-1))+2,:);teta4=q((3*(4-1))+3,:);
Rx5=q((3*(5-1))+1,:);Ry5=q((3*(5-1))+2,:);teta5=q((3*(5-1))+3,:);
Rx6=q((3*(6-1))+1,:);Ry6=q((3*(6-1))+2,:);teta6=q((3*(6-1))+3,:);
figure,plot(t,Rx2,'g',t,Ry2,'r',t,teta2,'b'),grid on,title('Evolución de la posición (cdg) y orientación de la rueda trasera'),legend('posición horizontal','posición vertical','orientación'),xlabel('tiempo (s)'),ylabel('posición (m)')
figure,plot(t,Rx3,'g',t,Ry3,'r',t,teta3,'b'),grid on,title('Evolución de la posición (cdg) y orientación del chasis'),legend('posición horizontal','posición vertical','orientación'),xlabel('tiempo (s)'),ylabel('posición (m)')
figure,plot(t,Rx4,'g',t,Ry4,'r',t,teta4,'b'),grid on,title('Evolución de la posición (cdg) y orientación de la rueda delantera'),legend('posición horizontal','posición vertical','orientación'),xlabel('tiempo (s)'),ylabel('posición (m)')
figure,plot(t,Rx5,'g',t,Ry5,'r',t,teta5,'b'),grid on,title('Evolución de la posición (cdg) y orientación del remolque'),legend('posición horizontal','posición vertical','orientación'),xlabel('tiempo (s)'),ylabel('posición (m)')
figure,plot(t,Rx6,'g',t,Ry6,'r',t,teta6,'b'),grid on,title('Evolución de la posición (cdg) y orientación de la rueda del remolque'),legend('posición horizontal','posición vertical','orientación'),xlabel('tiempo (s)'),ylabel('posición(m)')
```

```
function plotvelocidades(t,v)
vx2=v((3*(2-1))+1,:);vy2=v((3*(2-1))+2,:);omega2=v((3*(2-1))+3,:);
vx3=v((3*(3-1))+1,:);vy3=v((3*(3-1))+2,:);omega3=v((3*(3-1))+3,:);
vx4=v((3*(4-1))+1,:);vy4=v((3*(4-1))+2,:);omega4=v((3*(4-1))+3,:);
```

```
vx5=v((3*(5-1))+1,:);vy5=v((3*(5-1))+2,:);omega5=v((3*(5-1))+3,:);
vx6=v((3*(6-1))+1,:);vy6=v((3*(6-1))+2,:);omega6=v((3*(6-1))+3,:);
figure,plot(t,vx2,'g',t,vy2,'r',t,omega2,'b'),grid on,title('Velocidades de la rueda
trasera'),legend('velocidad horizontal','velocidad vertical','velocidad
rotacional'),xlabel('tiempo (s)'),ylabel('velocidad (m/s)')
figure,plot(t,vx3,'g',t,vy3,'r',t,omega3,'b'),grid on,title('Velocidades del
chasis'),legend('velocidad horizontal','velocidad vertical','velocidad
rotacional'),xlabel('tiempo (s)'),ylabel('velocidad (m/s)')
figure,plot(t,vx4,'g',t,vy4,'r',t,omega4,'b'),grid on,title('Velocidades de la rueda
delantera'),legend('velocidad horizontal','velocidad vertical','velocidad
rotacional'),xlabel('tiempo (s)'),ylabel('velocidad (m/s)')
figure,plot(t,vx5,'g',t,vy5,'r',t,omega5,'b'),grid on,title('Velocidades del
remolque'),legend('velocidad horizontal','velocidad vertical','velocidad
rotacional'),xlabel('tiempo (s)'),ylabel('velocidad (m/s)')
figure,plot(t,vx6,'g',t,vy6,'r',t,omega6,'b'),grid on,title('Velocidades de la rueda del
remolque'),legend('velocidad horizontal','velocidad vertical','velocidad
rotacional'),xlabel('tiempo (s)'),ylabel('velocidad (m/s)')
```

```
function plotmuelles(t,PPma)
vpma=12;
l0=0.35;
l0=ones(1,length(t))*l0;
l1=PPma((vpma*(1-1))+1,:);
l2=PPma((vpma*(2-1))+1,:);
l3=PPma((vpma*(3-1))+1,:);
l4=PPma((vpma*(4-1))+1,:);
figure,plot(t,l0,'g',t,l1,'r',t,l2,'b',t,l3,'m',t,l4,'k'),grid on, title('Deformación de
los muelles a lo largo de la simulación'),legend('l0','muelle vertical trasero','muelle
vertical delantero','muelle horizontal trasero','muelle horizontal
delantero'),xlabel('tiempo (s)'),ylabel('longitud (m)')
```

```
function plotindentaciones(q,t,PPcont)
Rx1=q((3*(2-1))+1,:);
Rx2=q((3*(4-1))+1,:);
Rx3=q((3*(6-1))+1,:);
ind1=PPcont((33*(1-1))+15,:);
ind2=PPcont((33*(2-1))+15,:);
ind3=PPcont((33*(3-1))+15,:);
figure,plot(t,ind1,'b',t,ind2,'r',t,ind3,'g'),title('Evolución de la indentación con el
tiempo de simulación'),legend('rueda trasera','rueda delantera','rueda del remolque'),grid
on,xlabel('tiempo (s)'),ylabel ('indentación (m)')
figure,plot(Rx1,ind1,'b',Rx2,ind2,'r',Rx3,ind3,'g'),title('Evolución de la indentación con
la posición sobre el terreno'),legend('rueda trasera','rueda delantera','rueda del
remolque'),grid on,xlabel('posición (m)'),ylabel ('indentación (m)')
```

```
function plotrueda(PPcont,q)
alf = 0:0.0001:(2*pi+0.1);
s1=PPcont((33*(1-1))+2,:);
```

```

s2=PPcont((33*(2-1))+2,:);
s3=PPcont((33*(3-1))+2,:);
for i = 1:length(alf)
    u1(:,i) = Perfil_Leva_j(alf(i));
end

for j = 1:length(alf)
    ug(:,j) =PosicionPunto([0 0 0]',u1(:,j));
end
X= ug(1,:);
Y = ug(2,:);
for i=1:length(s1)
    u11(:,i)=RotMat_2D(q(6,i))*Perfil_Leva_j(s1(i));

end
for i=1:length(s2)
    u12(:,i)=RotMat_2D(q(12,i))*Perfil_Leva_j(s2(i));

end
for i=1:length(s3)
    u13(:,i)=RotMat_2D(q(18,i))*Perfil_Leva_j(s3(i));

end

for j = 1:length(s1)
    ug1(:,j) =PosicionPunto([0 0 0]',u11(:,j));
end
for j = 1:length(s2)
    ug2(:,j) =PosicionPunto([0 0 0]',u12(:,j));
end
for j = 1:length(s3)
    ug3(:,j) =PosicionPunto([0 0 0]',u13(:,j));
end
X1= ug1(1,:);
Y1 = ug1(2,:);
X2= ug2(1,:);
Y2 = ug2(2,:);
X3= ug3(1,:);
Y3 = ug3(2,:);
ejes = [-0.06 0.06 -0.415 -0.385];

% figure,plot(X,Y,'b'),hold on,title('Registro de los puntos de contacto a lo largo de
la simulación en la rueda trasera'),plot(X1,Y1,'r+'), %axis(ejes)
% figure,plot(X,Y,'r'),hold on,title('Registro de los puntos de contacto a lo largo de
la simulación en la rueda delantera'),plot(X2,Y2,'r+'), %axis(ejes)
% figure,plot(X,Y,'g'),hold on,title('Registro de los puntos de contacto a lo largo de
la simulación en la rueda del remolque'),plot(X3,Y3,'r+'), %axis(ejes)

figure,
subplot(2,3,1), plot(X,Y,'k'),hold on,title('rueda trasera'), plot(X1,Y1,'r+'), axis
equal
subplot(2,3,2), plot(X,Y,'k'),hold on,title('rueda delantera'),plot(X2,Y2,'b+'), axis
equal
subplot(2,3,3), plot(X,Y,'k'),hold on,title('rueda del remolque'),plot(X3,Y3,'g+'),
axis equal

```

```
subplot(2,3,4), plot(X,Y,'k'),hold on,title('rueda trasera'), plot(X1,Y1,'r+'),
axis(ejes)
subplot(2,3,5), plot(X,Y,'k'),hold on,title('rueda delantera'),plot(X2,Y2,'b+'),
axis(ejes)
subplot(2,3,6), plot(X,Y,'k'),hold on,title('rueda del remolque'),plot(X3,Y3,'g+'),
axis(ejes)
```

```
function plotdeslizamiento(t,PPcont,v)
global SMCuerpo
vdesliz=SMCuerpo.Ce1as(1).vlimite;
vlim=ones(1,length(t))*vdesliz;
vcx1=v((3*(2-1))+1,:);
vcy1=v((3*(2-1))+2,:);
vcx2=v((3*(4-1))+1,:);
vcy2=v((3*(4-1))+2,:);
vcx3=v((3*(6-1))+1,:);
vcy3=v((3*(6-1))+2,:);
vpcgx1=PPcont((32*(1-1))+23,:);
vpcgy1=PPcont((32*(1-1))+24,:);
vpcgx2=PPcont((32*(2-1))+23,:);
vpcgy2=PPcont((32*(2-1))+24,:);
vpcgx3=PPcont((32*(3-1))+23,:);
vpcgy3=PPcont((32*(3-1))+24,:);
vrel1=PPcont((32*(1-1))+22,:);
vrel2=PPcont((32*(2-1))+22,:);
vrel3=PPcont((32*(3-1))+22,:);
vprx1=PPcont((32*(1-1))+25,:);
vpry1=PPcont((32*(1-1))+26,:);
vprx2=PPcont((32*(2-1))+25,:);
vpry2=PPcont((32*(2-1))+26,:);
vprx3=PPcont((32*(3-1))+25,:);
vpry3=PPcont((32*(3-1))+26,:);

figure,plot(t,vcx1,'b',t,vcy1,'r',t,vpcgx1,'g',t,vpcgy1,'m',t,vrel1,'c',t,vlim,'k',t,vprx1,
'y',t,vpry1,'r'),legend('componente horizontal del centro','componente vertical del
centro','componente horizontal del punto de contacto','componente vertical global del punto
de contacto','velocidad relativa','velocidad límite','componente horizontal relativa del
punto de contacto','componente vertical relativa del punto de contacto'),title
('Velocidades de interés en el contacto de la rueda trasera'),xlabel('tiempo
(s)'),ylabel('velocidad (m/s)'),grid on
figure,plot(t,vcx2,'b',t,vcy2,'r',t,vpcgx2,'g',t,vpcgy2,'m',t,vrel2,'c',t,vlim,'k',t,vprx2,
'y',t,vpry2,'r'),legend('componente horizontal del centro','componente vertical del
centro','componente horizontal del punto de contacto','componente vertical global del punto
de contacto','velocidad relativa','velocidad límite','componente horizontal relativa del
punto de contacto','componente vertical relativa del punto de contacto'),title
('Velocidades de interés en el contacto de la rueda delantera'),xlabel('tiempo
(s)'),ylabel('velocidad (m/s)'),grid on
figure,plot(t,vcx3,'b',t,vcy3,'r',t,vpcgx3,'g',t,vpcgy3,'m',t,vrel3,'c',t,vlim,'k',t,vprx3,
'y',t,vpry3,'r'),legend('componente horizontal del centro','componente vertical del
centro','componente horizontal del punto de contacto','componente vertical global del punto
de contacto','velocidad relativa','velocidad límite','componente horizontal relativa del
punto de contacto','componente vertical relativa del punto de contacto'),title
```

```
( 'Velocidades de interés en el contacto de la rueda del remolque'),xlabel('tiempo (s)'),ylabel('velocidad (m/s)'),grid on
```

```
function plotfuerzashorizontales(PPma,PPQg,PPFR,PPcont,PPQ,t)
%%Rueda trasera%%
Fmv1x=PPma((12*(1-1))+7,:);
Fmh1x=PPma((12*(3-1))+7,:);
Fgx1=PPQg((3*(2-1))+1,:);
Fnormx1=PPcont((32*(1-1))+27,:);
Ftanx1=PPcont((32*(1-1))+30,:);
Ftx1=PPQ((3*(2-1))+1,:);
%%Chasis%%
Fmv12x=PPma((12*(1-1))+10,:);
Fmv22x=PPma((12*(2-1))+10,:);
Fmh12x=PPma((12*(3-1))+10,:);
Fmh22x=PPma((12*(4-1))+10,:);
Fr2=PPFR((3*(3-1))+1,:);
Fgx2=PPQg((3*(3-1))+1,:);
Ftx2=PPQ((3*(3-1))+1,:);
%%Rueda delantera%%
Fmv3x=PPma((12*(2-1))+7,:);
Fmh3x=PPma((12*(4-1))+7,:);
Fgx3=PPQg((3*(4-1))+1,:);
Fnormx3=PPcont((32*(2-1))+27,:);
Ftanx3=PPcont((32*(2-1))+30,:);
Ftx3=PPQ((3*(4-1))+1,:);
%%Remolque%%
Fgx4=PPQg((3*(5-1))+1,:);
Ftx4=PPQ((3*(5-1))+1,:);
Frchasis4=-Fr2;
Frremolquex4=-PPFR((3*(6-1))+1,:);
%%Rueda del remolque%%
Fgx5=PPQg((3*(5-1))+1,:);
Fnormx5=PPcont((32*(3-1))+27,:);
Ftanx5=PPcont((32*(3-1))+30,:);
Ftx5=PPQ((3*(6-1))+1,:);
Fr5=PPFR((3*(6-1))+1,:);
figure,plot(t,Fmv1x,'b',t,Fmh1x,'g',t,Fgx1,'r',t,Fnormx1,'c',t,Ftanx1,'m',t,Ftx1,'y'),legend('Muelle vertical','Muelle horizontal','Gravedad','Fuerza normal de contacto','Fuerza tangencial de contacto','Fuerza neta horizontal'),title('Análisis de las fuerzas horizontales actuantes en la rueda trasera'),xlabel('tiempo (s)'),ylabel('Fuerza (N)'),grid on
figure,plot(t,Fmv12x,'b',t,Fmv22x,'g',t,Fmh12x,'r',t,Fmh22x,'c',t,Fr2,'y',t,Fgx2,'k',t,Ftx2,'g'),legend('Muelle vertical trasero','Muelle vertical delantero','Muelle horizontal trasero','Muelle horizontal delantero','Rótula del remolque','Gravedad','Fuerza neta'),title('Análisis de las fuerzas horizontales actuantes en el chasis'),xlabel('tiempo(s)'),ylabel('Fuerza (N)'),grid on
figure,plot(t,Fmv3x,'b',t,Fmh3x,'g',t,Fgx3,'r',t,Fnormx3,'c',t,Ftanx3,'y',t,Ftx3,'k'),legend('Muelle vertical','Muelle horizontal','Gravedad','Fuerza normal de contacto','Fuerza tangencial de contacto','Fuerza neta horizontal'),title('Análisis de las fuerzas horizontales actuantes en la rueda delantera'),xlabel('tiempo (s)'),ylabel('Fuerza (N)'),grid on
figure,plot(t,Fgx5,'b',t,Fnormx5,'g',t,Ftanx5,'r',t,Fr5,'c',t,Ftx5,'m'),legend('Gravedad','Fuerza normal de contacto','Fuerza tangencial de contacto','Rótula con el remolque','Fuerza neta horizontal'),title('Análisis de las fuerzas horizontales actuantes
```

```
en la rueda del remolque'),xlabel('tiempo (s)'),ylabel('Fuerza (N)'),grid on
figure,plot(t,Fgx4,'b',t,Frchasisx4,'r',t,Frrremolquex4,'c',t,Ftx4,'g'),legend('Gravedad','R
ótula con el chasis','Rótula con el remolque','Fuerza neta horizontal'),title('Análisis de
la fuerzas horizontales actuantes en el remolque'),xlabel('tiempo (s)'),ylabel('Fuerza
(N)'),grid on
```

```
function plotfuerzasverticales(PPma,PPQg,PPFR,PPcont,PPQ,t)
%%Rueda trasera%%
Fmv1y=PPma((12*(1-1))+8,:);
Fmh1y=PPma((12*(3-1))+8,:);
Fgy1=PPQg((3*(2-1))+2,:);
Fnormy1=PPcont((32*(1-1))+28,:);
Ftany1=PPcont((32*(1-1))+31,:);
Fty1=PPQ((3*(2-1))+2,:);
%%Chasis%%
Fmv12y=PPma((12*(1-1))+11,:);
Fmv22y=PPma((12*(2-1))+11,:);
Fmh12y=PPma((12*(3-1))+11,:);
Fmh22y=PPma((12*(4-1))+11,:);
Fry2=PPFR((3*(3-1))+2,:);
Fgy2=PPQg((3*(3-1))+2,:);
Fty2=PPQ((3*(2-1))+2,:);
%%Rueda delantera%%
Fmv3y=PPma((12*(2-1))+8,:);
Fmh3y=PPma((12*(4-1))+8,:);
Fgy3=PPQg((3*(4-1))+2,:);
Fnormy3=PPcont((32*(2-1))+28,:);
Ftany3=PPcont((32*(2-1))+31,:);
Fty3=PPQ((3*(4-1))+2,:);
%%Remolque%%
Fgy4=PPQg((3*(5-1))+2,:);
Fty4=PPQ((3*(5-1))+2,:);
Fry4=PPFR((3*(5-1))+2,:);
%%Rueda del remolque%%
Fgy5=PPQg((3*(6-1))+2,:);
Fnormy5=PPcont((32*(3-1))+28,:);
Ftany5=PPcont((32*(3-1))+31,:);
Fty5=PPQ((3*(6-1))+2,:);
Fry5=PPFR((3*(6-1))+2,:);
figure,plot(t,Fmv1y,'b',t,Fmh1y,'g',t,Fgy1,'r',t,Fnormy1,'c',t,Ftany1,'m',t,Fty1,'y'),legende('Muelle vertical','Muelle horizontal','Gravedad','Fuerza normal de contacto','Fuerza tangencial de contacto','Fuerza neta vertical'),title('Análisis de las fuerzas verticales actuantes en la rueda trasera'),xlabel('tiempo (s)'),ylabel('Fuerza (N)'),grid on
figure,plot(t,Fmv12y,'b',t,Fmv22y,'g',t,Fmh12y,'r',t,Fmh22y,'c',t,Fry2,'y',t,Fgy2,'k',t,Fty2,'g'),legend('Muelle vertical trasero','Muelle vertical delantero','Muelle horizontal trasero','Muelle horizontal delantero','Rótula del remolque','Gravedad','Fuerza neta vertical'),title('Análisis de las fuerzas verticales actuantes en el chasis'),xlabel('tiempo(s)'),ylabel('Fuerza (N)'),grid on
figure,plot(t,Fmv3y,'b',t,Fmh3y,'g',t,Fgy3,'r',t,Fnormy3,'c',t,Ftany3,'y',t,Fty3,'k'),legende('Muelle vertical','Muelle horizontal','Gravedad','Fuerza normal de contacto','Fuerza tangencial de contacto','Fuerza neta vertical'),title('Análisis de las fuerzas verticales actuantes en la rueda trasera'),xlabel('tiempo (s)'),ylabel('Fuerza (N)'),grid on
figure,plot(t,Fgy5,'b',t,Fnormy5,'g',t,Ftany5,'r',t,Fry5,'c',t,Fty5,'m'),legend('Gravedad','Fuerza normal de contacto','Fuerza tangencial de contacto','Rótula con el remolque','Fuerza neta vertical'),title('Análisis de las fuerzas verticales actuantes en la
```

```
rueda del remolque'),xlabel('tiempo (s)'),ylabel('Fuerza (N) '),grid on
figure,plot(t,Fgy4,'b',t,Fry4,'r',t,Fty4,'r'),legend('Gravedad','Rótulas con el chasis y
rueda','Fuerza neta vertical'),title('Análisis de la fuerzas verticales actuantes en el
remolque'),xlabel('tiempo (s)'),ylabel('Fuerza (N)'),grid on
```

```
function plotmomentos(PPma,PPFR,PPcont,PPQ,t,PPQus)
%%Rueda trasera%%
Mnorm1=PPcont((32*(1-1))+29,:);
Mt1=PPcont((32*(1-1))+32,:);
Mt1=PPQ((3*(2-1))+3,:);
Mfr1=PPFR((3*(2-1))+3,:);
Mfus1=PPQus((3*(2-1))+3,:);
%%Chasis%%
Mmv2=PPma((12*(1-1))+12,:);
Mmv22=PPma((12*(2-1))+12,:);
Mmh12=PPma((12*(3-1))+12,:);
Mmh22=PPma((12*(4-1))+12,:);
Mfr2=PPFR((3*(3-1))+3,:);
Mt2=PPQ((3*(3-1))+3,:);
%%Rueda delantera%%
Mnorm3=PPcont((32*(2-1))+29,:);
Mt3=PPcont((32*(2-1))+32,:);
Mt3=PPQ((3*(4-1))+3,:);
Mfr3=PPFR((3*(4-1))+3,:);
Mfus3=PPQus((3*(4-1))+3,:);
%%Remolque%%
Mt4=PPQ((3*(5-1))+3,:);
Mfr4=PPFR((3*(5-1))+3,:);
%%Rueda del remolque%%
Mnorm5=PPcont((32*(3-1))+29,:);
Mt5=PPcont((32*(3-1))+32,:);
Mt5=PPQ((3*(6-1))+3,:);
Mfr5=PPFR((3*(6-1))+3,:);
figure,plot(t,Mnorm1,'g',t,Mtan1,'r',t,Mt1,'b',t,Mfr1,'m',t,Mfus1,'c'),legend('Fuerza
normal de contacto','Fuerza tangencial de contacto','Par neto','Restricción de velocidad
constante','Par de usuario'),title('Análisis de los momentos actuantes en la rueda
trasera'),xlabel('tiempo (s)'),ylabel('Momento (N*m) '), grid on
figure,plot(t,Mmv2,'b',t,Mmv22,'r',t,Mmh12,'g',t,Mmh22,'c',t,Mfr2,'m',t,Mt2,'y'),legend('Mu
elle vertical trasero','Muelle vertical delantero','Muelle horizontal trasero','Muelle
horizontal delantero','Rótula con el remolque','Par neto'),title('Análisis de los momentos
actuantes en el chasis'),xlabel('tiempo (s)'),ylabel('Momento (N*m)'),grid on
figure,plot(t,Mnorm3,'b',t,Mtan3,'r',t,Mt3,'g',t,Mfr3,'m',t,Mfus3,'c'),legend('Fuerza
normal de contacto','Fuerza tangencial de contacto','Par neto','Restricción de velocidad
rotacional constante','Par de usuario'),title('Análisis de los momentos actuantes en la
rueda delantera'),xlabel('tiempo (s)'),ylabel('Momento (N*m) '),grid on
figure,plot(t,Mnorm5,'b',t,Mtan5,'r',t,Mt5,'g',t,Mfr5,'c'),legend('Fuerza normal de
contacto','Fuerza tangencial de contacto','Par neto','Rótula con el
remolque'),title('Análisis de los momentos actuantes en la rueda del
remolque'),xlabel('tiempo (s)'),ylabel('Momento (N*m)'),grid on
figure,plot(t,Mt4,'b',t,Mfr4,'r'),legend('Par neto','Rótulas con el chasis y con la
rueda'),title('Análisis de los momentos actuantes en el remolque'),xlabel('tiempo
(s)'),ylabel('Momento (N*m)'),grid on
```

```

function Anima(t,qq)
global SMCuerpo
srueda=linspace(0,(2*pi+0.1),200);
sterreno=linspace(-100,100,200);
ulrueda=zeros(2,length(srueda));
ulterreno=zeros(2,length(sterreno));

for i=1:length(srueda)
    ulrueda(:,i)=Parametriza_Rueda_animacion(srueda(i));
end
for i=1:length(sterreno)
    ulterreno(:,i)=Parametriza_terreno_animacion(sterreno(i));
end

for i=1:2:length(t)
    q=qq(:,i);
    q1=q((3*(1-1))+1:(3*(1-1))+3);
    q2=q((3*(2-1))+1:(3*(2-1))+3);
    q3=q((3*(3-1))+1:(3*(3-1))+3);
    q4=q((3*(4-1))+1:(3*(4-1))+3);
    q5=q((3*(5-1))+1:(3*(5-1))+3);
    q6=q((3*(6-1))+1:(3*(6-1))+3);

    for k=1:length(sterreno)
        R1(:,k)=Coor_global(q1,ulterreno(:,k));
    end
    R20(:,1)=Coor_global(q2,[0,0]');

    for k=1:length(srueda)
        R2i(:,k)=Coor_global(q2,ulrueda(:,k));
    end
    R2=[R20 R2i R20];
    R3(:,1) = Coor_global(q3,[0.5*SMCuerpo.Param.L_chasis -0.5*SMCuerpo.Param.H_chasis]');
    R3(:,2) = Coor_global(q3,[0.5*SMCuerpo.Param.L_chasis -0.2]');
    R3(:,3) = Coor_global(q3,[0.2985*SMCuerpo.Param.L_chasis -0.2]');
    R3(:,4) = Coor_global(q3,[0 0.5*SMCuerpo.Param.H_chasis]');
    R3(:,5) = Coor_global(q3,[-0.45*SMCuerpo.Param.L_chasis 0.5*SMCuerpo.Param.H_chasis]');
    R3(:,6) = Coor_global(q3,[-0.45*SMCuerpo.Param.L_chasis -
0.4*SMCuerpo.Param.H_chasis]');
    R3(:,7) = Coor_global(q3,[-0.5*SMCuerpo.Param.L_chasis -0.4*SMCuerpo.Param.H_chasis]');
    R3(:,8) = Coor_global(q3,[-0.5*SMCuerpo.Param.L_chasis -0.5*SMCuerpo.Param.H_chasis]');
    R3(:,9) = Coor_global(q3,[0.5*SMCuerpo.Param.L_chasis -0.5*SMCuerpo.Param.H_chasis]');

    R40(:,1)=Coor_global(q4,[0,0]');
    for k=1:length(srueda)
        R4i(:,k)=Coor_global(q4,ulrueda(:,k));
    end
    R4=[R40 R4i R40];
    R5(:,1) = Coor_global(q5,[0.5*SMCuerpo.Param.L_remo1que
0.5*SMCuerpo.Param.H_remo1que]');
    R5(:,2) = Coor_global(q5,[-0.5*SMCuerpo.Param.L_remo1que
0.5*SMCuerpo.Param.H_remo1que]');
    R5(:,3) = Coor_global(q5,[-0.5*SMCuerpo.Param.L_remo1que -
0.5*SMCuerpo.Param.H_remo1que]');

```

```

R5(:,4) =Coor_global(q5,[0.5*SMcuerpo.Param.L_remolque -
0.5*SMcuerpo.Param.H_remolque]');
R5(:,5) = Coor_global(q5,[0.5*SMcuerpo.Param.L_remolque
0.5*SMcuerpo.Param.H_remolque]');
R60(:,1)=Coor_global(q6,[0,0]');
for k=1:length(srueda)
    R6i(:,k)=Coor_global(q6,u1rueda(:,k));
end
R6=[R60 R6i R60];

R7(:,1)=Coor_global(q3,[-0.5*SMcuerpo.Param.L_chasis -0.5*SMcuerpo.Param.H_chasis]');
R7(:,2)=Coor_global(q5,[0.5*SMcuerpo.Param.L_remolque 0]');

R8(:,1)=Coor_global(q2,[0 0]');
R8(:,2)=Coor_global(q3,[-0.2985*SMcuerpo.Param.L_chasis -
0.5*SMcuerpo.Param.H_chasis]');

R9(:,1)=Coor_global(q4,[0 0]');
R9(:,2)=Coor_global(q3,[0.2985*SMcuerpo.Param.L_chasis -0.5*SMcuerpo.Param.H_chasis]');

R10(:,1)=Coor_global(q2,[0 0]');
R10(:,2)=Coor_global(q3,[-(0.2985*SMcuerpo.Param.L_chasis+SMcuerpo.Ma(3).l indef) -
(SMcuerpo.Ma(1).l indef+(0.5*SMcuerpo.Param.H_chasis)-
((9.81*0.5*SMcuerpo.Srigidos(3).masa)/SMcuerpo.Ma(1).k))]');

R11(:,1)=Coor_global(q4,[0 0]');
R11(:,2)=Coor_global(q3,[(0.2985*SMcuerpo.Param.L_chasis+SMcuerpo.Ma(3).l indef) -
(SMcuerpo.Ma(1).l indef+(0.5*SMcuerpo.Param.H_chasis)-
((9.81*0.5*SMcuerpo.Srigidos(3).masa)/SMcuerpo.Ma(1).k))]');

Rx1=R1(1,:);
Ry1=R1(2,:);
Rx2=R2(1,:);
Ry2=R2(2,:);
Rx3=R3(1,:);
Ry3=R3(2,:);
Rx4=R4(1,:);
Ry4=R4(2,:);
Rx5=R5(1,:);
Ry5=R5(2,:);
Rx6=R6(1,:);
Ry6=R6(2,:);
Rx7=R7(1,:);
Ry7=R7(2,:);
Rx8=R8(1,:);
Ry8=R8(2,:);
Rx9=R9(1,:);
Ry9=R9(2,:);
Rx10=R10(1,:);
Ry10=R10(2,:);
Rx11=R11(1,:);
Ry11=R11(2,:);

```

```

if i==1
    figure(1);
    hold on;
    plot1 = plot(Rx1,Ry1,'g','XDataSource','Rx1','YDataSource','Ry1');
    plot2 = plot(Rx2,Ry2,'r','XDataSource','Rx2','YDataSource','Ry2');
    plot3 = plot(Rx3,Ry3,'b','XDataSource','Rx3','YDataSource','Ry3');
    plot4 = plot(Rx4,Ry4,'r','XDataSource','Rx4','YDataSource','Ry4');
    plot5 = plot(Rx5,Ry5,'b','XDataSource','Rx5','YDataSource','Ry5');
    plot6 = plot(Rx6,Ry6,'r','XDataSource','Rx6','YDataSource','Ry6');
    plot7 = plot(Rx7,Ry7,'k','XDataSource','Rx7','YDataSource','Ry7');
    plot8 = plot(Rx8,Ry8,'k','XDataSource','Rx8','YDataSource','Ry8');
    plot9 = plot(Rx9,Ry9,'k','XDataSource','Rx9','YDataSource','Ry9');
    plot10 = plot(Rx10,Ry10,'k','XDataSource','Rx10','YDataSource','Ry10');
    plot11 = plot(Rx11,Ry11,'k','XDataSource','Rx11','YDataSource','Ry11');
    axis equal;
    axis([q(3*(2-1)+1)-5 q(3*(4-1)+1)+5 -0.2 4-2]);
else
    axis equal;
    axis([qq(3*(2-1)+1,i)-5 qq(3*(4-1)+1,i)+5 -0.2 -2+4+SMcuerpo.Param.L_chasis]);
    refreshdata(plot2,'caller')
    refreshdata(plot1,'caller')
    refreshdata(plot3,'caller')
    refreshdata(plot4,'caller')
    refreshdata(plot5,'caller')
    refreshdata(plot6,'caller')
    refreshdata(plot7,'caller')
    refreshdata(plot8,'caller')
    refreshdata(plot9,'caller')
    refreshdata(plot10,'caller')
    refreshdata(plot11,'caller')
    drawnow;
end
end

```