

---

Universidad Politécnica de Cartagena  
Escuela Técnica Superior de Ingeniería Industrial



## Proyecto fin de carrera

**Cinemática Computacional de Sistemas Multicuerpo.  
Formulación Topológica Eficiente Basada en Ecuaciones  
de Grupo.**

Director:

Mariano Saura Sánchez

Realizado por:

Bernardino Muñoz Pérez

Cartagena, Diciembre de 2013



# Índice general

<b>Índice general</b>	<b>IV</b>
<b>1. Introducción</b>	<b>5</b>
1.1. Antecedentes	7
1.2. Planteamiento del problema	12
1.3. Objetivos	15
1.4. Hipótesis de trabajo y medios empleados	16
<b>2. Fundamentos</b>	<b>19</b>
2.1. Análisis estructural de sistemas multicuerpo	19
2.1.1. Nuevo concepto de Grupo Estructural	20
2.1.2. Grafo estructural de un mecanismo	21
2.1.3. Análisis basado en el grafo. Diagrama estructural	23
2.1.4. Análisis estructural computacional	25
2.2. Formulación cinemática de un Grupo Estructural	25
2.2.1. Método basado en derivada temporal de matriz Jacobiana	25
2.2.2. Método basado en el tensor de tercer orden $\Phi_{qq}$	27
2.3. Cinemática computacional de sistemas multicuerpo	29
2.3.1. Algoritmo para métodos globales	30
2.3.2. Algoritmo propuesto para métodos basados en ecuaciones de grupo	30
<b>3. Materiales y métodos</b>	<b>33</b>
3.1. Modelización de grupo estructural mediante método de derivada temporal	33
3.1.1. Par de rotación 1R	34
3.1.2. Diada 3R	37
3.1.3. Grupo perteneciente a sistema de suspensión 2C3E1R	39
3.1.4. Grupo perteneciente a sistema de suspensión 1C1E1R	44
3.1.5. Grupo perteneciente a sistema de suspensión 1C1E1P	48
3.2. Modelización de sistemas multicuerpo. Casos de estudio	51
3.2.1. Cuadrilátero escalable	51
3.2.2. Sistema de suspensión de un camión	52
3.3. Implementación en Matlab	56
3.3.1. Cuadrilátero escalable	56
3.3.1.1. Fichero raíz MBSBEG_Cuad.m	56
3.3.1.2. Fichero de geometría MBDatosCuad.m	58

3.3.1.3. Archivos de solución de grupos . . . . .	58
3.3.2. Sistema de suspensión del eje de un camión . . . . .	60
3.3.2.1. Archivo raíz MBSuspension5ejes.m . . . . .	60
3.3.2.2. Archivo de geometría MBDatosSuspension5ejesB.m . . . . .	67
3.3.2.3. Archivos de solución de grupos . . . . .	71
<b>4. Resultados</b>	<b>113</b>
4.1. Simulación cinemática de los sistemas analizados . . . . .	113
4.1.1. Cuadrilátero escalable . . . . .	113
4.1.2. Sistema de suspensión de un camión . . . . .	115
4.2. Eficiencia de las diferentes implementaciones . . . . .	119
4.2.1. Formulación global vs la topológica . . . . .	119
4.2.1.1. Cálculo de tiempo para diferentes aproximaciones . . . . .	120
<b>5. Conclusiones y desarrollos futuros</b>	<b>123</b>
5.1. Conclusiones . . . . .	123
5.1.1. Conclusiones generales . . . . .	123
5.1.2. Conclusiones específicas . . . . .	123
5.2. Desarrollos futuros . . . . .	125
<b>6. Anexo de matrices</b>	<b>127</b>
<b>Bibliografía</b>	<b>148</b>

# Índice de figuras

1.1. Cuadrilátero articulado: esquema cinemático y nomenclatura. . . . .	8
1.2. Cuadrilátero articulado: modelización en coordenadas independientes. . . . .	8
1.3. Cuadrilátero articulado: modelización en coordenadas lagrangianas. . . . .	9
1.4. Cuadrilátero articulado: modelización en coordenadas Cartesianas de punto de referencia. . . .	10
1.5. Cuadrilátero articulado: modelización en coordenadas naturales. . . . .	11
2.1. Mecanismos en los que no sería aplicable la teoría de Assur. . . . .	20
2.2. Cuadrilátero articulado: Topología. . . . .	21
2.3. Cuadrilátero articulado y grafo estructural. . . . .	22
2.4. Mecanismos planos en los que no es aplicable la teoría de Assur. . . . .	22
2.5. Grafo estructural de los mecanismos de la Figura 2.4. . . . .	23
2.6. Análisis Estructural del cuadrilátero articulado basado en grafo estructural. . . . .	23
2.7. Diagrama Estructural del cuadrilátero articulado. . . . .	24
2.8. Cuadrilátero escalable dividido en grupos estructurales: SG-I, SG-II y coordenadas . . . . .	26
3.1. Grupo estructural 1R. Esquema cinemático y representación del grafo estructural. . . . .	34
3.2. Elección de la ecuación de restricción de ángulo. . . . .	35
3.3. Grupo estructural 3R. Esquema cinemático y representación del grafo estructural. . . . .	37
3.4. Grupo estructural 2C3E1R. Esquema cinemático y representación del grafo estructural. . . . .	40
3.5. Grupo estructural 1C1E1R. Esquema cinemático y representación del grafo estructural . . . .	44
3.6. Grupo estructural 1C1E1P. Esquema cinemático y representación del grafo estructural . . . .	48
3.7. Esquema cinemático del cuadrilátero escalable . . . . .	52
3.8. Esquema cinemático del sistema de suspensión de un camión . . . . .	53
3.9. Identificación de los eslabones y pares que componen el sistema de suspensión. . . . .	53
3.10. Grafo del sistema de suspensión del camión . . . . .	54
3.11. Análisis del grafo estructural del sistema de suspensión . . . . .	55
3.12. Diagrama estructural del sistema de suspensión . . . . .	55
4.1. Posición del punto $P_3$ respecto del ángulo de entrada $\theta$ . . . . .	114
4.2. Velocidad del punto $P_3$ respecto del ángulo de entrada $\theta$ . . . . .	114
4.3. Aceleración del punto $P_3$ respecto del ángulo de entrada $\theta$ . . . . .	115
4.4. Simulación cinemática del cuadrilátero escalable . . . . .	116
4.5. Ángulo de torsión de la barra 6 – 7 en función de la elevación vertical de la rueda. . . . .	117
4.6. Distancia del par hidráulico en función de la elevación vertical de la rueda. . . . .	117
4.7. Velocidad del par hidráulico en función de la elevación vertical de la rueda. . . . .	118

4.8. Simulación cinemática en detalle del sistema de suspensión. . . . .	118
4.9. Simulación cinemática del sistema de suspensión del camión. . . . .	118
4.10. Evolución del cálculo de tiempo requerido para las soluciones del cuadrilátero escalable . . . .	120
4.11. Cálculo de la distribución de tiempo (%) en el análisis cinemático del cuadrilátero escalable con 17 diadas . . . . .	121
4.12. Distribución del porcentaje de tiempo entre los ítems de las distintas formulaciones . . . . .	121

# Índice de Cuadros

3.1. Fichero raíz del cuadrilátero escalable. . . . .	57
3.2. Fichero que define geometría del cuadrilátero escalable. . . . .	59
3.3. Fichero del grupo estructural <i>1RF</i> ; esquema e identificación de variables . . . . .	61
3.4. Fichero del grupo estructural <i>1RF</i> ; problema de posición. . . . .	62
3.5. Fichero del grupo estructural <i>1RF</i> ; problema de velocidad. . . . .	63
3.6. Fichero del grupo estructural <i>1RF</i> ; problema de aceleración. . . . .	63
3.7. Fichero del grupo estructural <i>1RF</i> ; cálculo de otros POI's. . . . .	64
3.8. Fichero del grupo estructural <i>3RF</i> ; esquema e identificación de variables. . . . .	65
3.9. Fichero del grupo estructural <i>3RF</i> ; problema de posición. . . . .	66
3.10. Fichero del grupo estructural <i>3RF</i> ; problema de velocidad. . . . .	66
3.11. Fichero del grupo estructural <i>3RF</i> ; problema de aceleración. . . . .	67
3.12. Fichero del grupo estructural <i>3RF</i> ; cálculo de otros POI's. . . . .	68
3.13. Fichero raíz del sistema de suspensión del camión (primera parte). . . . .	69
3.14. Fichero raíz del sistema de suspensión del camión (segunda parte). . . . .	70
3.15. Fichero de geometría del sistema de suspensión del camión (primera parte). . . . .	72
3.16. Fichero de geometría del sistema de suspensión del camión (segunda parte). . . . .	73
3.17. Fichero de geometría del sistema de suspensión del camión (tercera parte). . . . .	74
3.18. Fichero de geometría del sistema de suspensión del camión (cuarta parte). . . . .	75
3.19. Fichero de geometría del sistema de suspensión del camión (quinta parte). . . . .	76
3.20. Fichero de geometría del sistema de suspensión del camión (sexta parte). . . . .	77
3.21. Fichero de geometría del sistema de suspensión del camión (séptima parte). . . . .	78
3.22. Fichero de geometría del sistema de suspensión del camión (octava parte). . . . .	79
3.23. Fichero de geometría del sistema de suspensión del camión (novena parte). . . . .	80
3.24. Fichero de geometría del sistema de suspensión del camión (décima parte). . . . .	81
3.25. Fichero de geometría del sistema de suspensión del camión (undécima parte). . . . .	82
3.26. Fichero de geometría del sistema de suspensión del camión (duodécimo parte). . . . .	83
3.27. Fichero de geometría del sistema de suspensión del camión (décimo tercera parte). . . . .	84
3.28. Identificación de constantes y variables del grupo {2,3,4,5}. Primera parte del fichero de grupo. . . . .	85
3.29. Identificación de constantes y variables del grupo {2,3,4,5}. Segunda parte del fichero de grupo. . . . .	86
3.30. Problema de posición del grupo {2,3,4,5}. Tercera parte del fichero de grupo. . . . .	87
3.31. Problema de posición del grupo {2,3,4,5}. Cuarta parte del fichero de grupo. . . . .	88
3.32. Problema de posición del grupo {2,3,4,5}. Quinta parte del fichero de grupo. . . . .	89
3.33. Problema de velocidad del grupo {2,3,4,5}. Sexta parte del fichero de grupo. . . . .	90

3.34. Problema de aceleración del grupo {2,3,4,5}. Séptima parte del fichero de grupo. . . . .	91
3.35. Problema de aceleración del grupo {2,3,4,5}. Octava parte del fichero de grupo. . . . .	92
3.36. Problema de posición de otros POI 's del grupo {2,3,4,5}. Novena parte del fichero de grupo. . . . .	93
3.37. Problema de velocidad de otros POI 's del grupo {2,3,4,5}. Décima parte del fichero de grupo. . . . .	94
3.38. Problema de aceleración de otros POI 's del grupo {2,3,4,5}. Undécima parte del fichero de grupo. . . . .	95
3.39. Identificación de constantes y variables del grupo {7,9}. Primera parte del fichero de grupo. . . . .	96
3.40. Problema de posición del grupo {7,9}. Segunda parte del fichero de grupo. . . . .	97
3.41. Problema de posición del grupo {7,9}. Tercera parte del fichero de grupo. . . . .	98
3.42. Problema de velocidad del grupo {7,9}. Cuarta parte del fichero de grupo. . . . .	99
3.43. Problema de velocidad del grupo {7,9}. Cuarta parte del fichero de grupo. . . . .	100
3.44. Identificación de constantes y variables del grupo {12,13}. Primera parte del fichero de grupo. . . . .	101
3.45. Problema de posición del grupo {12,13}. Segunda parte del fichero de grupo. . . . .	102
3.46. Problema de posición del grupo {12,13}. Tercera parte del fichero de grupo. . . . .	103
3.47. Problema de velocidad del grupo {12,13}. Cuarta parte del fichero de grupo. . . . .	104
3.48. Problema de aceleración del grupo {12,13}. Quinta parte del fichero de grupo. . . . .	105
3.49. Función auxiliar Triedro1P3V. . . . .	106
3.50. Función auxiliar Triedro1P3VC. . . . .	106
3.51. Función auxiliar Triedro2PF2VF. . . . .	107
3.52. Función auxiliar TrdPFVV. . . . .	107
3.53. Función auxiliar CCL1P3V. . . . .	108
3.54. Función auxiliar Cardan. . . . .	108
3.55. Función auxiliar Prismatico. . . . .	109
3.56. Función secundaria dotProduct. . . . .	109
3.57. Función secundaria NormCont. . . . .	109
3.58. Función secundaria RigCont. . . . .	110
3.59. Función auxiliar FiqqpTriedro1P3V. . . . .	110
3.60. Función auxiliar FiqqpTriedro1P3VC. . . . .	111
3.61. Función auxiliar FiqqpTriedro2PF2VF. . . . .	111
3.62. Función auxiliar FiqqpTrdPFVV. . . . .	111
3.63. Función auxiliar FiqqpCCL1P3V. . . . .	111
3.64. Función auxiliar FiqqpCardan. . . . .	112
3.65. Función auxiliar FiqqpPrismatico. . . . .	112



# Índice de Tablas

1.1. Principales características de los métodos de análisis de sistemas multicuerpo. Comparativa . . . . .	12
3.1. Principales características del grupo estructural <i>2C3E1R</i> . Coherencia de coordenadas de definición por eslabón. . . . .	40
3.2. Principales características del grupo estructural <i>1C1E1R</i> . Coherencia de coordenadas de definición por eslabón . . . . .	45
3.3. Principales características del grupo estructural <i>1C1E1P</i> . Coherencia de coordenadas de definición por eslabón. . . . .	49
3.4. Grados de libertad permitidos por los pares cinemáticos . . . . .	52
4.1. Velocidades de las entradas al sistema de suspensión. . . . .	115
5.1. Comparativa de tiempos entre la forma directa y el uso de funciones auxiliares. . . . .	125



# Agradecimientos

Al Director de este Proyecto, Dr. D. Mariano Saura Sánchez, por darme la oportunidad de trabajar bajo su tutela. Por su gran dedicación, interés y constancia en todo momento durante el desarrollo de este proyecto.

A mi familia, por haberme apoyado en mis momentos más difíciles. En concreto a mi madre Ana María, por haberme hecho quien soy, ya que sin ella no estaría hoy aquí. También a mi hermana Ana María, por sacar lo mejor de mí.

A mis amigos, que han aguantado y me han ayudado a seguir adelante.

Finalmente, al Departamento de Ingeniería Mecánica de la Universidad Politécnica de Cartagena.



# Capítulo 1

## Introducción

Un sistema multicuerpo es un modelo matemático de un conjunto de cuerpos, también denominados elementos o eslabones, rígidos o flexibles, interconectados de tal modo que exista movimiento relativo entre ellos. La dinámica de sistemas multicuerpo se puede definir como un área de la mecánica computacional, que reúne varias disciplinas tales como la dinámica estructural, mecánica multifísica, matemática computacional, teoría del control y ciencia de la computación, con la finalidad de proporcionar métodos y herramientas para el prototipado virtual de sistemas mecánicos complejos [30]. La dinámica de sistemas multicuerpo juega hoy en día un papel fundamental en la modelización, análisis, simulación y optimización de sistemas mecánicos en una amplia variedad de campos y para un elevado rango de aplicaciones industriales: máquinas, mecanismos, robótica, vehículos, estructuras espaciales o biomecánica, por citar algunos.

El área de la dinámica estructural se centra en los métodos para el análisis y la síntesis de los sistemas multicuerpo, destacando el análisis cinemático y dinámico. El presente proyecto pretende ofrecer solución sólamante a problemas de índole cinemática.

El análisis cinemático estudia la posición y el movimiento del sistema. Se trata de problemas puramente geométricos, independientes de las fuerzas causantes del movimiento y de las características inerciales de los cuerpos que componen el sistema.

Los problemas más comunes del análisis cinemático son:

- Posición inicial. Se trata de determinar la posición de todos los elementos de un sistema multicuerpo, a partir de la posición de los elementos de entrada. Es un problema de difícil solución que conlleva la resolución de un sistema de ecuaciones algebraicas no lineales, por regla general, con varias soluciones. El número de soluciones es directamente proporcional a la complejidad del sistema.
- Desplazamiento finito. Se trata de encontrar la posición final de los elementos de un sistema multicuerpo, que evoluciona desde una posición inicial mediante un desplazamiento finito, no diferencial, de los elementos de entrada. Pese a que es una variación del problema anterior, en lo que se refiere a una variación del problema de posición inicial, tanto conceptualmente como desde el punto de vista de la herramienta matemática para resolverlo, su resolución es más sencilla dado que la posición inicial del sistema, conocida, puede ser utilizada como punto de partida para el proceso iterativo de resolución del sistema de ecuaciones algebraicas no lineales. La multiplicidad de soluciones no supone un problema en este caso, puesto que sólo resulta de interés la solución más cercana a la inicial del sistema.
- Análisis cinemático. Con este nombre se engloban los análisis de velocidad y de aceleración de los sistemas multicuerpo. El análisis de velocidad trata de determinar la velocidad de todos los elementos del sistema, a partir de su posición y de la velocidad de los elementos de entrada. El análisis de aceleración trata, a su vez, de determinar la aceleración de todos los elementos a partir de su posición, su velocidad y la aceleración de los elementos de entrada. Ambos son problemas sencillos de resolver, puesto que implican la resolución de un sistema de ecuaciones lineales con solución única.
- Simulación cinemática. Se trata de predecir el comportamiento cinemático del sistema y engloba todos los problemas definidos anteriormente. Se utiliza para detectar posibles colisiones entre elementos, estudiar trayectorias de puntos y determinar las secuencias de las posiciones de los elementos.

Los problemas de análisis dinámico son mucho más complejos de resolver que los cinemáticos, que deben resolverse previamente (o simultáneamente). Los problemas dinámicos involucran a las fuerzas que actúan sobre el sistema y además, la masa, el tensor de inercia y la posición del centro de masas de todos los eslabones que lo forman. Los problemas más relevantes de índole dinámica, que no se abordan en este proyecto, son los siguientes:

- Posición de equilibrio estático. Se trata de determinar la posición en que se equilibran las fuerzas exteriores, las fuerzas de enlace y las reacciones que actúan sobre el sistema. Se engloba aquí, a pesar de no tratarse de un problema dinámico, ya que al menos depende de la masa y la posición del centro de masas de los eslabones. La solución al problema exige la resolución de un sistema de ecuaciones mediante procedimientos iterativos.
- Dinámica linealizada. Se linealizan las ecuaciones del movimiento en una posición particular del sistema para, mediante un estudio paso a paso de la historia del sistema o un análisis de valores propios, determinar los modos y las frecuencias de vibración natural que tienen lugar a partir de una posición de equilibrio estático o dinámico del sistema. De esta forma se conoce la rigidez dinámica del sistema.
- Dinámica inversa. Se trata de determinar la forma en que deben evolucionar las fuerzas de entrada, para que el movimiento del sistema tenga unas características determinadas. Conocido el movimiento pueden determinarse las aceleraciones de los centros de masas, por lo que las posiciones y sus primera y segunda derivadas son conocidas. El sistema que se debe resolver es, por tanto, lineal.
- Dinámica directa. Es el problema más complejo dentro de la dinámica. En este caso son conocidas las fuerzas externas que actúan sobre los eslabones y debe determinarse el movimiento que producen sobre ellos, a partir de unas condiciones iniciales de contorno conocidas. Exige formular las ecuaciones diferenciales (ordinarias o algebraicas) del movimiento, como en el caso anterior, pero ahora lo que se puede evaluar son las aceleraciones de los centros de masas. Éstas deberán integrarse dos veces para determinar las velocidades y las posiciones de los eslabones. Supone, por tanto, la utilización de métodos numéricos para integrar sistemas de ecuaciones diferenciales, cuya complejidad dependerá del tipo de coordenadas utilizado para modelizar el problema.
- Simulación dinámica. Se refiere, en la práctica, a la resolución del problema dinámico directo y engloba la simulación cinemática.
- Percusiones e impactos. Se trata de determinar el efecto que, las percusiones e impactos que actúan sobre el sistema, tienen sobre la distribución de velocidades del mismo. Las percusiones, de limitada importancia práctica, producen discontinuidades en las distribuciones de velocidades, que pueden ser calculados introduciendo una ecuación de naturaleza experimental, que incorpore el tipo de impacto y la naturaleza de las superficies en contacto durante el mismo.

El proyecto se estructura en siete capítulos. En el Capítulo 1: Introducción se revisan los antecedentes en modelización de sistemas multicuerpo: tipos de coordenadas y ventajas/desventajas principales. En el Capítulo 2: Fundamentos se presentan los fundamentos teóricos sobre el análisis estructural, se introducen dos formulaciones de análisis cinemático así como las bases de análisis cinemático computacional basado en partición de coordenadas. En el Capítulo 3: Materiales y Métodos se aplican los fundamentos del capítulo anterior a algunos grupos estructurales pertenecientes a mecanismos planos y espaciales, se analizan dos problemas propuestos y se incluyen sus implementaciones en Matlab. En el Capítulo 4: Resultados se muestran los resultados obtenidos de las implementaciones del capítulo anterior. Por un lado se muestran las simulaciones cinemáticas de los problemas propuestos, por otro se compara la eficiencia de esta implementación con otras, para validar el método en cuanto a precisión y coste en términos de tiempo de cálculo. En el Capítulo 5: Conclusiones y desarrollos futuros se exponen las principales conclusiones a las que ha dado lugar el presente trabajo. El Capítulo 6: Anexo de matrices proporciona las matrices relativas al Capítulo 3, y el Capítulo 7: Bibliografía recoge las referencias bibliográficas consultadas.

## 1.1. Antecedentes

### Modelización de sistemas multicuerpo: tipos de coordenadas y métodos de análisis

La selección del tipo de coordenadas a utilizar para describir el comportamiento dinámico de un sistema multicuerpo ha sido desde hace tiempo objeto de gran debate. No en vano, el tipo de coordenadas condiciona de forma determinante el número total de ecuaciones del movimiento a resolver, la complejidad y grado de acoplamiento en que quedan expresadas esas ecuaciones y su tipo: ecuaciones diferenciales ordinarias (ODE) o ecuaciones diferenciales algebraicas (DAE), que a su vez requieren de la selección de diferentes métodos de integración y en su caso de estabilización. Además, el tipo de coordenadas condiciona el método de solución del problema cinemático, el cual puede resultar más o menos adecuado para su implementación mediante programas de propósito general.

Una clasificación general de los tipos de coordenadas las divide en dos grupos: Coordenadas independientes y coordenadas dependientes. Es en este segundo grupo donde existe una mayor diversidad.

Las coordenadas independientes buscan un número mínimo de coordenadas suficientes, en principio, para poder definir la cinemática de todos los sólidos del sistema: posiciones, velocidades y aceleraciones. La dimensión del vector de coordenadas independientes coincide siempre con la movilidad del sistema mecánico a analizar, sea una cadena cinemática abierta o cerrada.

Los métodos de análisis basados en coordenadas dependientes identifican un conjunto de coordenadas suficiente, para definir de forma unívoca la posición de todos los eslabones del mecanismo. El número de coordenadas seleccionadas difiere según su tipo pero es siempre mayor que la movilidad del sistema mecánico, por lo que son necesarias funciones algebraicas que las relacionen; son las llamadas ecuaciones de restricción para las posiciones. En coordenadas dependientes hay dos métodos diferentes de seleccionar el vector de coordenadas: los topológicos, que aprovechan la estructura del sistema mecánico a analizar, y los basados en ecuaciones de restricción impuestas por los pares cinemáticos (llamados métodos globales).

Los topológicos formulan las ecuaciones de restricción y las del movimiento a partir del reconocimiento topológico del sistema mecánico, normalmente representado mediante su correspondiente grafo. Estos métodos buscan el menor número posible de coordenadas y de ecuaciones de restricción, ofreciendo sistemas de ecuaciones ODE reducidos pero fuertemente acopladas. Existe una cierta complejidad en la generación de la matriz de masas y en los términos de fuerza centrípeta y de Coriolis. Entre los métodos topológicos el tipo de coordenadas más utilizadas son lagrangianas.

Los métodos basados en las ecuaciones de restricción de par no consideran la topología del mecanismo, formulan las ecuaciones de restricción en función del número y tipo de pares cinemáticos que aparecen entre sus eslabones. El conjunto de coordenadas dependientes es normalmente mayor al de los métodos topológicos; como ventaja, suelen ser más sistemáticos en la formulación de las ecuaciones de restricción y del movimiento, estas últimas desacopladas y con matrices de masas constantes y términos centrípetos y de Coriolis nulos. Como desventaja, el mayor número de coordenadas obliga a un mayor número de ecuaciones del movimiento y también de ecuaciones de restricción. Dos ejemplos de este tipo de coordenadas son las cartesianas de punto de referencia y las coordenadas naturales.

Los métodos de análisis que utilizan coordenadas dependientes, ya sean topológicos o globales, permiten formular las ecuaciones del movimiento como un sistema de ecuaciones diferenciales algebraicas (DAE) u ordinarias (ODE). En un sistema DAE el número de ecuaciones diferenciales de segundo orden coincide con el de coordenadas utilizadas para describir el sistema; en un sistema ODE coincide con su movilidad.

A continuación se consideran los principales tipos de coordenadas sobre las que más se debate en la actualidad y cómo se aplicarían a la resolución de un sencillo cuadrilátero articulado (Figura 1.1).

#### Coordenadas independientes

Se entiende por coordenadas independientes  $\mathbf{q}$  al conjunto de variables utilizadas para describir el comportamiento de un sistema mecánico, cuando la dimensión del vector de variables coincide con la movilidad del sistema a analizar  $Dim\mathbf{q} = L$ .  $L$  se puede determinar mediante distintos criterios de movilidad. En cadenas cinemáticas abiertas un conjunto de coordenadas independientes, también llamadas generalizadas,

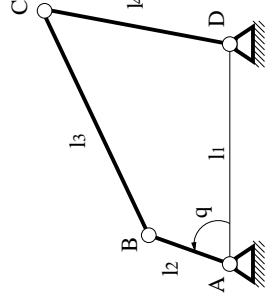


Figura 1.1: Cuadrilátero articulado: esquema cinemático y nomenclatura.

permite definir con precisión las posiciones de todos los eslabones; en una cadena cerrada se eliminan grados de libertad, por tanto se reduce la movilidad del sistema y con ello el número de coordenadas que lo describen. Cuando se selecciona este tipo de coordenadas se deben obtener expresiones que definan de forma explícita a las variables que definen las posiciones del resto de los sólidos en función de las independientes.

En el ejemplo del cuadrilátero articulado se puede determinar su movilidad utilizando el criterio de Grübler:

$$L = 3 \cdot n_m - 2 \cdot P_i - P_s = 3 \cdot 3 - 2 \cdot 4 - 0 = 9 - 8 = 1.$$

Con  $n_m = n^\circ$  eslabones móviles,  $P_i = n^\circ$  pares inferiores y  $P_s = n^\circ$  pares superiores.

De modo que una única coordenada sería suficiente para describir la posición de todo el sistema. Supongamos que se elige como coordenada independiente  $q$  que consiste en el giro absoluto de 2 respecto al bastidor. Para identificar las posiciones del resto de los eslabones (Figura 1.2), se debe realizar un tedioso análisis en base a relaciones trigonométricas complejas, que permitan definir las variables secundarias  $\theta_1 = \theta_1(q)$ ,  $\theta_2 = \theta_2(q)$ ,  $\theta_3 = \theta_3(q)$  como funciones explícitas de la variable independiente (Ec 1.1,1.2,1.3).

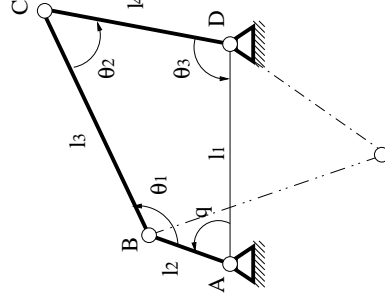


Figura 1.2: Cuadrilátero articulado: modelización en coordenadas independientes.

$$\cos \theta_2 = \frac{l_3^2 + l_4^2 + 2 l_2 l_1 \cdot \cos q}{2 \cdot l_3 \cdot l_4 - l_2^2 - l_1^2} \quad (1.1)$$

$$\sin(\theta_3 + \phi) = \frac{c}{\rho}; \quad \phi = \tan^{-1} \frac{a}{b}; \quad a = l_1 - l_2 \cos q; \quad b = l_2 \sin q; \quad \rho = \sqrt{a^2 + b^2} \quad (1.2)$$

$$\theta_1 = 2 \cdot \pi - q - \theta_2 - \theta_3 \quad (1.3)$$

La ventaja fundamental de este método consiste en obtener, cuando es posible, la dependencia de todas las variables secundarias como función explícita de la independiente, dando lugar a un número mínimo de



ecuaciones de fácil solución [5]. Al estar definidas de forma explícita se pueden determinar, al menos en teoría, mediante derivación directa, la velocidad y la aceleración de éstas variables.

Este método presenta sin embargo grandes inconvenientes. En primer lugar, la forma de obtener las variables secundarias en función de la independiente no siempre es sencillo o incluso posible. Además, las expresiones que se obtienen son sólo útiles para el mecanismo que se está analizando, con lo que existe una pérdida total de generalidad en el método. Otro inconveniente, derivado de las expresiones que se obtienen para las variables secundarias, es que debido al uso de funciones trigonométricas la solución no suele ser única, con lo que finalmente la variable independiente no define de forma unívoca la posición de todo el mecanismo, tal y como se puede apreciar en la Figura 1.2, donde para un mismo valor de  $q$ , el mecanismo puede ensamblarse de dos formas diferentes.

No es fácil encontrar referencias bibliográficas de autores que hayan desarrollado aplicaciones utilizando este tipo de coordenadas en cadenas cerradas. Verlinden presenta un desarrollo [29] y reconoce la complejidad y el elevado conocimiento del análisis cinemático que exige su método.

### Coordenadas lagrangianas

Este tipo de coordenadas se popularizó con el desarrollo de métodos de modelización en robótica, donde se podía aprovechar la condición de cadena abierta de los sistemas a analizar. Cuando la cadena cinemática es abierta (Figura 1.3, izq.), el vector de coordenadas está formado sólo por coordenadas independientes  $[\theta_1 \theta_2 \theta_3]$ , y su dimensión también coincide con la movilidad del sistema. Si la cadena se cierra (Figura 1.3, dcha.) el número de coordenadas independientes disminuye en número igual al grado de cierre del par introducido (si se cierra mediante la creación de un nuevo par inferior disminuye en dos; si es superior en uno). El conjunto de coordenadas lagrangianas sigue siendo el mismo, el necesario para definir de forma unívoca las posiciones de todos los eslabones, pero ahora algunas de las coordenadas son dependientes y se deberán formular ecuaciones algebraicas que las relacionen con las independientes. Estas relaciones se llaman ecuaciones de restricción para las posiciones, o funciones de posición, y se formulan en forma de ecuaciones implícitas que deben resolverse por métodos numéricos.

Este tipo de coordenadas utiliza un método de análisis que resuelve la cinemática de los diferentes eslabones de forma recursiva, mediante transformaciones homogéneas, describiendo la posición de un eslabón en función de su posición relativa a la del anterior. El número de coordenadas es superior a las del tipo independientes (en cadenas cerradas), pero de entre los métodos de variables dependientes es el que menor número utiliza (algunos autores llaman relativas a este tipo de coordenadas [7]).

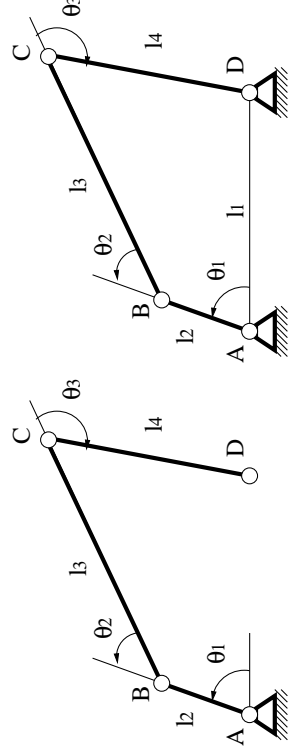


Figura 1.3: Cuadrilátero articulado: modelización en coordenadas lagrangianas. La Figura izquierda responde a una cadena cinemática abierta de movilidad  $L = 3$ , por tanto se utilizan tres coordenadas para describir el sistema. Como el número de coordenadas coincide con la movilidad, estas coordenadas son independientes. En la Figura de la derecha, la misma cadena se ha cerrado uniendo el extremo  $D$  al bastidor. Al introducir un par inferior nuevo la movilidad se reduce a  $L = 1$ . Si se utilizan las tres mismas variables para definir de forma unívoca las posiciones de los sólidos, dos de las variables son ahora dependientes. Harán falta dos ecuaciones que las relacionen con la independiente.

En el caso del cuadrilátero articulado se seleccionarían como coordenadas dependientes el vector de coordenadas:

$$\mathbf{q}^T = [\theta_1 \theta_2 \theta_3]$$

Al tener el sistema un único grado de libertad se hace necesario introducir dos ecuaciones de restricción que garanticen que el comportamiento del nuevo sistema será el mismo que el del original (Ec 1.4).

$$\begin{aligned} L_1 \cdot \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2) + L_3 \cdot \cos(\theta_1 + \theta_2 + \theta_3) - L_4 &= 0 \\ L_1 \cdot \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) + L_3 \cdot \sin(\theta_1 + \theta_2 + \theta_3) &= 0 \end{aligned} \quad (1.4)$$

Con una coordenada independiente se podría plantear una única ecuación del movimiento ODE sin mucha complicación. Un sistema DAE presentaría 3 ecuaciones diferenciales-algebraicas. A ambos sistemas de ecuaciones diferenciales habría que incluir dos ecuaciones de restricción para las posiciones.

### Coordenadas cartesianas de punto de referencia

Se elige un conjunto de coordenadas dependientes de dimensión  $Dim[q] = 3 \cdot n_m$ , con tres coordenadas por eslabón móvil: la posición de un punto de referencia (normalmente el centro de masas) más la orientación absoluta del eslabón (Figura 1.4). De esta forma se garantiza la definición unívoca de la posición de todos los eslabones, aunque a costa de una pérdida considerable en su eficacia.

En el caso del cuadrilátero articulado son necesarias nueve coordenadas:

$$\mathbf{q}^T = [x_2 \ y_2 \ \theta_2 \ x_3 \ y_3 \ \theta_3 \ x_4 \ y_4 \ \theta_4]$$

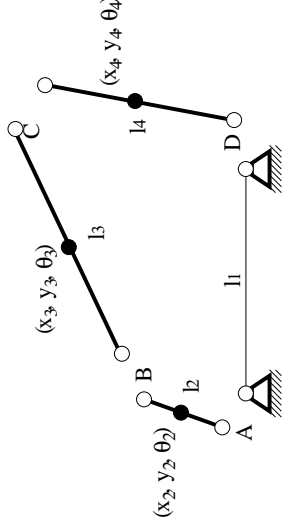


Figura 1.4: Cuadrilátero articulado: modelización en coordenadas Cartesianas de punto de referencia.

Estas coordenadas deben quedar relacionadas mediante ocho ecuaciones de restricción impuestas en función del tipo de par cinemático. En este caso, en el que son todos pares de rotación:

$$\begin{aligned} x_2 - \frac{L_2}{2} \cdot \cos \theta_2 &= 0 \\ x_2 + \frac{L_2}{2} \cdot \cos \theta_2 - x_3 + \frac{L_3}{2} \cdot \cos \theta_3 &= 0 \\ x_3 + \frac{L_3}{2} \cdot \cos \theta_3 - x_4 + \frac{L_4}{2} \cdot \cos \theta_4 &= 0 \\ x_4 + \frac{L_4}{2} \cdot \cos \theta_4 &= L_1 \\ y_2 - \frac{L_2}{2} \cdot \cos \theta_2 &= 0 \\ y_2 + \frac{L_2}{2} \cdot \sin \theta_2 - y_3 + \frac{L_3}{2} \cdot \sin \theta_3 &= 0 \\ y_3 + \frac{L_3}{2} \cdot \sin \theta_3 - y_4 + \frac{L_4}{2} \cdot \sin \theta_4 &= 0 \\ y_4 + \frac{L_4}{2} \cdot \sin \theta_4 &= 0 \end{aligned} \quad (1.5)$$

La dinámica de este sistema quedará definida mediante nueve ecuaciones diferenciales DAE y ocho algebraicas. Este método ofrece una muy sencilla sistematización de las ecuaciones del movimiento muy desacopladas; el problema es el elevado número de ecuaciones resultantes (un total de 17 para este simple mecanismo). Además, la mayoría de las ecuaciones de restricción están definidas en base a funciones trigonométricas, lo que complica la obtención de la solución del sistema (frente a otro tipo de coordenadas). La consideración de eslabones flexibles a partir de este tipo de coordenadas no es más sencillo que en el caso de

coordenadas lagrangianas.

### Coordenadas naturales

Las coordenadas naturales son consideradas por su autor como una evolución «natural» de las coordenadas de punto de referencia [8]. Este método utiliza coordenadas cartesianas definidas en determinados puntos de referencia evitando el uso de ángulos para definir orientaciones, y por tanto eliminando del sistema las clásicas funciones trigonométricas. En el caso del cuadrilátero articulado (Figura 1.5) se seleccionaría un conjunto de coordenadas dependientes formado por las cuatro variables siguientes:

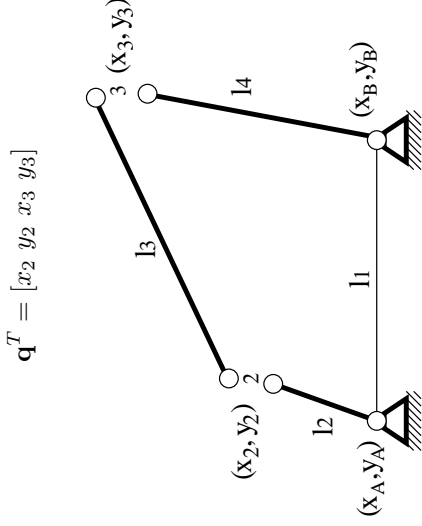


Figura 1.5: Cuadrilátero articulado: modelización en coordenadas naturales.

De las 4 variables necesarias, como el mecanismo tiene movilidad  $L = 1$  serán necesarias 3 ecuaciones de restricción que ligen esas variables. En el método de coordenadas naturales, las restricciones se imponen a partir de dos tipos de condiciones: restricciones de sólido rígido y de par cinemático. En este caso las restricciones de sólido rígido son suficientes para definir esas tres ecuaciones de restricción. No son necesarias establecer condiciones de par cinemático, dado que en este tipo de coordenadas los puntos de referencia se han seleccionado precisamente en la posición de esas articulaciones. Las ecuaciones de restricción son, por tanto:

$$\begin{aligned} (x_2 - x_A)^2 + (y_2 - y_A)^2 - l_1^2 &= 0 \\ (x_3 - x_2)^2 + (y_3 - y_2)^2 - l_2^2 &= 0 \\ (x_4 - x_B)^2 + (y_4 - y_B)^2 - l_3^2 &= 0 \end{aligned} \quad (1.6)$$

Con cuatro coordenadas dependientes y tres ecuaciones de restricción, el comportamiento dinámico de este sistema, en coordenadas naturales, viene definido por un conjunto de cuatro ecuaciones diferenciales ordinarias junto con tres ecuaciones de restricción, es decir un total de siete ecuaciones tipo DAE. Como se puede comprobar, el sistema queda muy reducido respecto al obtenido mediante coordenadas de punto de referencia.

Las ventajas de este tipo de coordenadas son múltiples. El número de coordenadas necesarias está a medio camino entre las lagrangianas y las de punto de referencia. Las ecuaciones se pueden obtener de forma sencilla y sistemática, y ofrecen matrices de masas independientes de la posición del mecanismo. Al no utilizar funciones trigonométricas, su cálculo es más eficiente desde un punto de vista computacional, y los resultados permiten orientar cualquier eslabón con independencia del resto. Además son eficaces también para describir sistemas multicuerpo con eslabones flexibles. Por contra, la única desventaja que se les atribuye, es que exige una gran familiarización con este tipo de coordenadas para modelizar sistemas mecánicos de forma correcta, y que ciertos tipos de pares cinemáticos, como los de mecanismos de leva, son algo más complicados de modelizar que con otros tipos de coordenadas.

## 1.2. Planteamiento del problema

En los apartados anteriores se ha presentado una breve introducción a los sistemas multicuerpo; su utilidad, los tipos de coordenadas más utilizadas para modelizarlos y las formulaciones actuales para resolverlos. Sobre estos últimos existe aún hoy día un gran debate en cuanto a las ventajas e inconvenientes que presentan unos frente a otros. El Cuadro 1.1 recoge algunas de las características fundamentales a las que debe responder una formulación capaz de resolver la dinámica de sistemas multicuerpo, y en qué medida los métodos presentados cubren tales características. Se marcan, para mayor claridad, con fondo de color verde aquellas en las que el método destaca en relación a los otros; en color naranja las que se cubren con cierta dificultad y en rojo funciones no disponibles en el método o verdaderamente difíciles de obtener.

	Coord. indep.	Coordenadas dependientes			Coord. Grupo
		Lagrangianas	Naturales	Punto referencia	
Síntesis de Tipo	No	No	No	No	Si
Síntesis de Numero	No	No	No	No	Si
Síntesis dimensional	Si (*)	Particular	Si	Particular	Si
N Ec restricción	0	m_L <	m_N <	m_P <	m_L
N Ec. diferenciales/Tipo	L ODE	LODE m_L DAE	m_N DAE	m_P DAE	LODE m_L DAE
Generalidad metodo	Particular	S/Librerías	S/Librerías disponibles		S/Librerías
Tratamiento Singularidades	No	Si	Si	Si	Si
Tiempo de calculo	¿ Mínimo ?	Reducido	Moderado	Exagerado	Reducido
Ecuaciones mov. Simbolicas	Si (*)	Si	No	No	Si
Selección Tipo Coord.	No	No	No	No	Si
SeL/Cambio Coord. reducción	Si	Si	No	No	Si
Mecanismos espaciales	C. abiertas (*)	Si	Si	Si	Si
Solidos Flexibles	No	Costosa	Simple	Costosa	S/Coordenadas
Pre/Posprocesado variables	Si (***)	Si	No	No	No

$$m_L = L + 2 C + P_s\_pto + 2 P_s\_S/S + P\_ord + P\_epi + 2 P\_p/c$$

$$m_N = \text{variable}$$

$$m_P = 3 \text{ mm} - L$$

Cuadro 1.1: Principales características de los métodos de análisis de sistemas multicuerpo y grado en que diferentes tipos de coordenadas las satisfacen.

Los tres primeros ítems están relacionados con la capacidad de la formulación de resolver problemas de síntesis: de tipo, número y dimensional. Los dos primeros constituyen la que se conoce como síntesis estructural (tipo y número). Los ítems restantes hacen referencia al método en general.

- Síntesis estructural:** Dado que los métodos de coordenadas naturales y de punto de referencia no utilizan criterios topológicos para formular sus ecuaciones, no están capacitados para afrontar este tipo de problemas. Tampoco lo están los métodos de coordenadas lagrangianas, ni el de independientes. El primero, porque a pesar de considerar la topología del sistema para formular las ecuaciones del movimiento, lo hace en función del número y tipo de lazos cerrados, lo que dificultaría en gran medida la determinación de los parámetros con los que trabaja este tipo de síntesis. Por tanto, hace falta un tipo de coordenadas y una formulación basada en ellas, que permita resolver problemas de síntesis estructural. Para ello un método que evalúe la estructura del sistema, a partir de los conceptos del análisis estructural, estará utilizando variables y parámetros propios de la síntesis estructural y por tanto, estará preparado para este tipo de problemas.
- Síntesis dimensional:** las coordenadas lagrangianas y las de punto de referencia ofrecen soluciones particulares a determinados tipos de mecanismos. Las coordenadas independientes podrían resolver también ciertos tipos de problemas, pero bajo una fuerte carga para el analista y con gran dificultad. Las coordenadas naturales sí parecen adaptarse bien a este tipo de problemas [2]. Dado que este tipo de síntesis es de gran utilidad en el diseño de sistemas mecánicos, será conveniente que el método propuesto pueda aportar soluciones a este problema.

- **Nº Ecuaciones de restricción:** Determina el tamaño del problema de análisis cinemático que se deberá resolver antes o junto con el dinámico, según la formulación. Se observa en la Tabla que el método de coordenadas independientes presenta cero ecuaciones de restricción, ya que el sistema queda determinado con sólo definir los valores de las coordenadas independientes. Esto es parcialmente cierto, ya que existen expresiones - difíciles de obtener - que definen explícitamente algunas variables en función de las independientes (ecuaciones de restricción, en definitiva), que el analista debe obtener (preprocesado) e incluir en las ecuaciones cinemáticas y dinámicas para dejarlas sólo en función de la independiente. Las coordenadas lagrangianas utilizan un número muy reducido de variables en su formulación. Ya se ha dicho que en una cadena abierta su número coincide con las empleadas en las independientes. El número de coordenadas dependientes  $m_L$  es, como se ve, inferior al definido por las coordenadas naturales  $m_N$ , y las de éste inferiores a las de punto de referencia  $m_P$ . Desde luego, interesará que el método que se proponga utilice coordenadas dependientes en el menor número posible  $m_L$ . A pie de Tabla se indican las expresiones que permiten calcular el número de coordenadas dependientes que utiliza cada método. Sin entrar en mucho detalle, conviene observar que en coordenadas naturales se puede utilizar un número variable de coordenadas para definir el modelo. A veces se introducen coordenadas adicionales a las necesarias, simplemente porque es más sencillo modelizar ciertos elementos (por ejemplo resortes o amortiguadores) o porque existen actuadores sobre coordenadas que no precisaba el método.
- **Nº Ecuaciones diferenciales/tipo:** más importante que las ecuaciones de restricción son las del movimiento. Aunque deban resolverse de forma conjunta, las ecuaciones del movimiento son diferenciales de segundo orden, por lo que se requiere su integración por métodos numéricos. A mayor número de variables dependientes, mayor número de ecuaciones de restricción y mayor número de ecuaciones diferenciales y de multiplicadores de Lagrange. Si el tipo de coordenadas lo permite, el sistema de ecuaciones que, en coordenadas dependientes es inicialmente DAE, se puede pasar con mayor o menor coste computacional a sistema ODE, para el que existen métodos de integración rápidos, robustos y precisos. Como se ve en la Tabla, en coordenadas independientes el sistema de ecuaciones del movimiento es ODE, porque se expresa sólo en términos de coordenadas independientes. En los dependientes, por cada coordenada dependiente introducida aparece un multiplicador de Lagrange. Sin embargo, se ha visto anteriormente que mediante la técnica de embebido se puede pasar un DAE a ODE. Esta transformación es sencilla en coordenadas lagrangianas pero complicada (poco eficiente) en coordenadas naturales y de punto de referencia. Será una característica determinante que el nuevo método pueda pasar a ODE con bajo coste computacional. También es importante considerar en el método si los términos de masas son constantes y si aparecen o no términos centripetos y de Coriolis, asociados a las fuerzas de inercia. En coordenadas dependientes naturales y de punto de referencia se dan las mejores características: matriz de masas constante, y términos centrífugos y de Coriolis nulos. En lagrangianas e independientes la matriz de masas depende de la posición y los términos centripetos también.
- **Generalidad del método:** Obviamente el método que se proponga debe poder resolver cualquier tipo de sistema multicuerpo, o estar preparado para que pueda hacerlo. Normalmente la capacidad de un método de abordar diferentes sistemas depende del conjunto de librerías que se hayan implementado: pares cinemáticos de rotación, de traslación, engranajes, levas, posibilidad de incluir resortes lineales o angulares, rozamiento, contacto, etc. Una formulación sistemática y efectiva permitirá implementar nuevas librerías, que amplíen las capacidades de análisis del programa. El método de coordenadas independientes ofrece soluciones particulares, lo que lo convierte en un método poco atractivo. El resto de los métodos analizados son globales y por tanto, totalmente generales. Esta generalidad hace referencia también a la capacidad de analizar sistemas espaciales y con eslabones rígidos o flexibles.
- **Tratamiento de singularidades:** El tipo de coordenadas seleccionado influye en la dimensión de la matriz Jacobiana del sistema y en los valores que se calcularán para sus elementos. Si la matriz Jacobiana se vuelve, en algún instante, deficiente de rango, entonces el sistema alcanzará una posición singular. Es importante que el método propuesto pueda acceder a métodos establecidos o introducir métodos nuevos para el tratamiento de las singularidades. Esta es una de las razones por las que no se utilizan las coordenadas independientes, como se ha visto anteriormente. Los métodos de variables dependientes sí disponen actualmente de algoritmos que les permitan gestionar esas singularidades. El método que se proponga debe poder incluir esos algoritmos o formular nuevos de igual o mayor eficacia.
- **Tiempo de cálculo:** Si se pretende que el método de análisis pueda resolver la dinámica de sistemas multicuerpo complejos, con un elevado número de sólidos, rígidos y flexibles, planos y espaciales en

tiempo real, entonces su formulación debe ser muy eficaz. El método de coordenadas independientes es el más rápido, ya que no tiene que resolver ecuaciones de restricción, el número de ecuaciones del movimiento es mínimo y su tipo ODE. El problema es el trabajo de preprocesado (obtención de las expresiones que permiten hacer desaparecer las variables dependientes en las ecuaciones del movimiento) y de postprocesado: una vez obtenidos los valores de  $\mathbf{q}$ , determinar las posiciones del resto de los eslabones. La de preprocesado puede ser verdaderamente compleja o incluso imposible de acometer. Las coordenadas lagrangianas son las más eficaces en cadenas abiertas. En cadenas cerradas, como el número de variables dependientes es mínimo, también lo serán sus ecuaciones de restricción y del movimiento. Sin embargo, la matriz de masas no es constante; sus elementos dependen de la posición instantánea del sistema y como consecuencia, aparecen términos centripetos y de Coriolis asociados a las fuerzas de inercia. Esto último hace que se tengan que calcular dichas matrices en cada punto de integración y para ello debe resolverse el problema cinemático completo. Los métodos de coordenadas naturales amplían el conjunto de coordenadas dependientes, y por ello el de ecuaciones del movimiento (DAE) pero por contra la matriz de masas es constante y los términos centripetos y de Coriolis nulos. Convertir el sistema DAE a ODE (método de la matriz  $R$ ) tiene un elevado coste computacional que anula su ventaja respecto a otros métodos. Los métodos más eficaces parecen ir en la dirección de resolver el sistema DAE mediante el método de Lagrange aumentado. El elevado número de coordenadas dependientes de punto de referencia hace que los tiempos de cálculo se eleven de forma considerable. Evidentemente, si se busca una nueva formulación y está entre sus objetivos que pueda integrar las ecuaciones del movimiento en tiempo real, deberá emplear tiempos de cálculo muy reducidos. La posibilidad de que el método pueda obtener las ecuaciones en forma simbólica está directamente relacionado con el método de cálculo.

- **Análisis simbólico:** Si el método de análisis es capaz de obtener las ecuaciones del movimiento del sistema en forma simbólica, éstas se podrán integrar fuera del programa utilizado para obtenerlas, con lo que los tiempos de cálculo caerán drásticamente [12] y podrán utilizarse en sistemas en tiempo real. Esto es lo que busca precisamente el método de coordenadas independientes, pero el trabajo tiene que hacerlo el analista con gran dificultad. Los métodos de análisis recursivos en coordenadas lagrangianas están bien orientados al tratamiento simbólico de sus ecuaciones, y se han presentado numerosos trabajos en este sentido [17, 16, 25, 23, 22, 24]. Las coordenadas naturales y de punto de referencia por contra, no están orientadas al tratamiento simbólico y los tiempos de cálculo serán superiores a los de coordenadas lagrangianas con análisis simbólico. El método propuesto debería poder obtener las ecuaciones del movimiento en forma simbólica.
- **Selección del tipo de coordenadas:** Los métodos descritos formulan sus ecuaciones en función de un tipo de coordenadas específico, por tanto ya tienen definido el tipo de coordenadas que van a utilizar y salvo aplicaciones muy particulares, no cambian en el transcurso del análisis. El método de coordenadas naturales es el único que permite introducir, además de las coordenadas cartesianas necesarias para el análisis, un conjunto de coordenadas adicional para facilitar el análisis o incluir en él, el estudio de ciertos parámetros que de otro modo tendrían que realizarse en una fase de postprocesado (un ejemplo característico es la evolución del par motor aplicado sobre un eslabón; en este método no es necesario considerar el giro como coordenada independiente, sin embargo para estudiar esa variable interesa incluirla en el sistema). Un método nuevo que permita seleccionar el tipo de coordenadas que mejor se adapte al estudio a realizar: análisis o síntesis, estructural, cinemática o dinámica y al modelo que se está estudiando, podría presentar ventajas frente al resto de las formulaciones.
- **Selección y cambio de coordenadas de reducción:** Poder elegir las coordenadas a las que se reduce el sistema, es decir para las que se plantean las ecuaciones del movimiento, puede suponer una ventaja en el tiempo de cálculo, bien porque las ecuaciones que se obtienen son más o menos sencillas de integrar [17], bien porque se pueden evitar singularidades durante el análisis, lo que evitaría tener que entrar en los algoritmos que las gestionan, o bien porque un cambio de estado (contacto entre eslabones donde no lo había) cambia la topología del sistema. En coordenadas independientes se pueden elegir las coordenadas de reducción del sistema. El resto de las variables se expresarán en función de éstas. Si se produce un cambio de estado, la formulación no tendrá ninguna validez. Los métodos topológicos, como en el que se basan las coordenadas lagrangianas, están obligados a incluir la posibilidad de cambio de estado ya que, si éste se produce, se modificará la topología del sistema (nuevos lazos cerrados, apertura de algunos ya existentes). En las coordenadas naturales y de punto de referencia no se produce una reducción del sistema a un subconjunto de coordenadas si el sistema es DAE, por tanto en la medida en que el conjunto de coordenadas es capaz de evaluar el contacto entre sólidos,

no serán necesarias transformaciones adicionales que consideren cambios de estado. Los métodos de análisis actuales, basados en criterios topológicos, incluyen en sus formulaciones estas transformaciones y las resuelven con eficacia. Sería aconsejable que, en la medida de lo posible, una nueva formulación incluyera la posibilidad de cambio de estado y selección automática de coordenadas de reducción en forma simbólica, para que no se viera afectado el tiempo de cálculo.

Discutidas las ventajas e inconvenientes que presentan los tipos de coordenadas presentadas, se puede formular el problema a abordar en este proyecto como: el desarrollo e implementación de un método general de análisis cinemático, para sistemas multicuerpo planos y espaciales, con sólidos rígidos y basado en coordenadas de grupo.

Para tratar de resolver el problema planteado, al menos en parte, se formulan los siguientes objetivos.

### 1.3. Objetivos

El objetivo principal de este proyecto es introducir una formulación eficaz que sea capaz de dar respuesta al problema planteado en el apartado anterior, reuniendo las ventajas de los métodos precedentes y evitando sus particulares inconvenientes. Con la intención de que sirva tanto a problemas de análisis como de síntesis, la formulación se desarrollará en base a la teoría del análisis estructural.

También es objetivo de este proyecto el implementar en un software de análisis matemático para la formulación desarrollada, así como validar sus resultados mediante comparación con otros métodos ya contrastados.

Para la consecución de esta meta se plantean una serie de objetivos específicos:

- Revisión bibliográfica de las formulaciones actuales en cinemática de sistemas multicuerpo.
- Revisión de los métodos de análisis estructural analítico y computacional: A partir de la revisión bibliográfica se estudiarán los métodos actuales de análisis estructural y la posibilidad de aplicarlos a mecanismos planos y espaciales.
- Estudio y desarrollo de la formulación necesaria para la obtención de las ecuaciones de restricción, necesarias para resolver el análisis cinemático de un sistema multicuerpo: Se analizarán los métodos de análisis cinemático propuestos en la bibliografía y se seleccionará la formulación más eficaz, para obtener de forma sistemática las ecuaciones de restricción necesarias.
- Extensión de una librería básica para el análisis cinemático de grupos estructurales en sistemas multicuerpo: La formulación cinemática desarrollada en los apartados anteriores se aplicará a un conjunto básico de grupos estructurales. Su implementación permitirá poner a prueba el carácter de generalidad del método. Éste debe dejar la puerta abierta a la incorporación de nuevos grupos, planos y espaciales a la librería. Esta idea permite una gran modularidad en la resolución, lo que resulta útil en problemas de síntesis estructural, de selección del tipo de coordenadas dependientes, y selección y cambio de coordenadas de reducción, lo que a su vez redundará en el tratamiento de singularidades y de situaciones de cambio de estado (transformaciones estructurales, contacto, etc).
- Implementar un módulo de simulación del movimiento: Para visualizar los resultados de forma similar a un programa comercial, se programará la simulación del comportamiento cinemático del sistema a lo largo del tiempo. Esta simulación será opcional, disminuyendo el tiempo de cálculo en caso de no producirse. La representación del movimiento es importante porque puede aclarar en gran medida las soluciones obtenidas de forma numérica, y evaluar y detectar posibles fuentes de error.
- Implementar diferentes formas de modelizar un sistema a estudio.
- Validar el método mediante su aplicación a mecanismos de diferente complejidad y comparar los resultados obtenidos, mediante el uso de diferentes formulaciones: global y topológica.
- Presentar un conjunto de desarrollos futuros: En función de las metas logradas y de las conclusiones obtenidas se presentará un conjunto de desarrollos futuros que complementen el trabajo aquí presentado.

## 1.4. Hipótesis de trabajo y medios empleados

El campo de estudio real de sistemas multicuerpo es complejo. Resultaría pretencioso tratar de introducir e implementar completamente, en ese campo, una nueva metodología que pretenda aprovechar las ventajas de diferentes formulaciones, al tiempo que evitar algunos de sus inconvenientes. El problema planteado debe quedarse en una declaración de intenciones: lo que 'debería ser'. Sin embargo, en el ámbito del desarrollo de un proyecto final de carrera se debe tratar de dar respuesta a unos objetivos generales y específicos, acordes con el problema planteado y al tiempo disponible para cumplirlos. Es por este motivo, por el que se hace necesario limitar el alcance del proyecto a ciertas restricciones recogidas en las hipótesis descritas en esta sección.

### Hipótesis de trabajo

#### Movimiento plano y espacial

Para dotar a la formulación de la mayor generalidad posible, se consideran sistemas mecánicos planos y espaciales.

#### Sólido rígido. Extensible a flexibles

Se considera la hipótesis de sólido rígido, por la cual la distancia entre dos puntos de un mismo sólido permanece invariable. Como consecuencia de ésta, el ángulo formado por dos líneas trazadas entre tres puntos del sólido también permanecerá invariable. De la misma forma que con el tipo de movimiento, la formulación y su implementación deberá poder considerar, en ampliaciones futuras, que uno o varios eslabones puedan ser deformables.

#### Método basado en análisis estructural

El análisis cinemático se formula a partir del resultado del análisis estructural, que puede ser resuelto por métodos computacionales, utilizando una matriz que recoja la topología del sistema, o analíticos.

#### Tipos de grupos estructurales

Los tipos de grupos estructurales que pueden componer un mecanismo son de lo más diversos. En este proyecto se estudiarán mecanismos planos y espaciales, por tanto se tratará un variado tipo de grupos formados por pares inferiores y superiores, entre los que se deben incluir los siguientes:

Grupos estructurales en movimiento plano:

- Grupo  $1RF$ : Grupo formado por un eslabón unido al bastidor mediante un par de rotación, con un movimiento de entrada que suele ser de tipo ángulo.
- Grupo  $1RM$ : Grupo formado por un eslabón unido a otro, en este caso móvil, mediante un par de rotación. Este grupo tendrá un movimiento de entrada que suele ser de tipo ángulo.
- Grupo  $3RF$ : Grupo formado por dos eslabones unidos entre sí por un par interno de rotación, unido por los extremos por pares externos de rotación, por un lado con un eslabón móvil y por otro fijo o bastidor.
- Grupo  $3RM$ : Grupo formado por dos eslabones unidos entre sí por un par interno de rotación, unido por los extremos por pares externos de rotación con otros eslabones móviles.

Grupos estructurales en movimiento espacial:

- Grupo  $2C3E1R$ : Grupo formado por cuatro eslabones. Uno de ellos base al que se unirán los restantes, dos eslabones con dos pares internos cardan y el restante con un par esférico. El grupo se unirá por los extremos mediante tres pares externos, dos esféricos y uno de rotación.



- Grupo *1C1E1R*: Grupo formado por dos eslabones con un par interno cardan y dos pares externos, uno esférico y otro de rotación.
- Grupo *1C1E1P*: Grupo formado por dos eslabones con un par interno prismático y dos pares externos, uno esférico y otro cardan.

#### **Transformación estructural: selección de las coordenadas de reducción**

El método debe permitir la selección de las coordenadas a las que se reduce dinámicamente el sistema multicuerpo. Ello se consigue mediante un análisis estructural sistemático de las posibles transformaciones estructurales que pueden aparecer en el sistema multicuerpo. La transformación estructural está orientada al tratamiento de las posibles singularidades que puede alcanzar el sistema durante su movimiento.

#### **Tratamiento de singularidades**

A pesar de que el análisis incluye el estudio de las posibles transformaciones estructurales del sistema mecánico, y que éstas se orientan al tratamiento de singularidades, el método a implementar no dará solución a la aparición de las mismas. Este estudio se deja para desarrollos futuros.

#### **Simulación**

Como se cita en los objetivos del proyecto, se permite el análisis de mecanismos de geometría compleja. A partir de los resultados del análisis cinemático es posible la simulación de los mecanismos estudiados.

#### **Sistemas multicuerpo con N grados de libertad**

Los sistemas multicuerpo a analizar no tienen limitación en su número de grados de libertad.

#### **Medios empleados**

La programación informática se realiza en lenguaje Matlab. Para ello se emplea el software comercial Matlab versión 7.8.0.347 R2009a.

Para el modelo del mecanismo espacial se ha utilizado el software SolidWorks 2013. Su despiece se ha hecho en Blender y finalmente, para la simulación del movimiento en tres dimensiones se ha utilizado el visor OpenSceneGraph.

El ordenador en el cual se encuentran instalados esos programas posee las siguiente características:

- Procesador: Intel Core i7-3612QM de 2.10 GHz.
- Memoria RAM: 4 GB.



# Capítulo 2

## Fundamentos

En este capítulo se presentan los fundamentos teóricos necesarios para la implementación del análisis cinemático computacional, a partir del análisis estructural computacional del sistema multicuerpo.

Se expone en primer lugar la teoría del análisis estructural más moderna, basada en la obtención de la estructura de un sistema mecánico, plano o espacial, mediante la aplicación sistemática de la ley de formación de grupos estructurales. En esta metodología es de gran ayuda la confección del grafo estructural del sistema, que consiste en la representación gráfica de su topología. Una vez que se adquieren los conceptos básicos del análisis estructural se está en disposición de comprender las ventajas de disponer de un conjunto de cadenas cinemáticas, que agrupadas convenientemente, pueden formar la práctica totalidad de los mecanismos planos y espaciales.

Para cada una de estas cadenas cinemáticas de un mecanismo se puede programar su solución completa de forma que, simplemente atendiendo al número y tipo de cadenas que forman un sistema mecánico concreto (análisis estructural), se pueden escribir e integrar las ecuaciones del movimiento, mediante el acceso a una biblioteca de rutinas que resuelven cada uno de estos grupos estructurales. Las rutinas que resuelven la cinemática de los diferentes grupos estructurales son específicas para cada tipo de grupo, sin embargo sus fundamentos son los mismos. Por tanto, en una segunda parte del capítulo se expondrán dos formulaciones cinemática para un grupo estructural, apartado 2.2.

### 2.1. Análisis estructural de sistemas multicuerpo

Los mecanismos articulados planos están compuestos por un conjunto de miembros o eslabones unidos mediante pares cinemáticos inferiores de rotación, de ejes paralelos o deslizantes, rectilíneos o mediante pares cinemáticos superiores.

El estudio cinemático de estos mecanismos por métodos tradicionales se complica enormemente cuando se acometen mecanismos complejos, esto es, con un elevado número de eslabones y pares cinemáticos.

El profesor L.V. Assur [1876-1920] sentó las bases del análisis estructural de los mecanismos articulados planos, desarrollando un método sistemático que permite dividir un mecanismo, independientemente de su complejidad, en un número determinado de grupos de eslabones y pares cinemáticos básicos, conocidos como *grupos estructurales*. Además comprobó que el número de grupos estructurales que existen es limitado y que por tanto, cualquier mecanismo está formado por una combinación de estos. El análisis estructural de un mecanismo consiste en obtener qué grupos estructurales lo forman y en qué orden.

Assur ha desarrollado también un método sistemático para el análisis cinemático de cada uno de estos grupos individualmente, de forma que cuando se tiene la estructura de un mecanismo, para analizarlo completamente sólo será necesario aplicar los métodos de análisis establecidos a cada grupo, en el orden en el que se han obtenido. La principal ventaja de este método es que las rutinas de análisis de cada tipo de grupo se pueden programar independientemente, de forma que el programa raíz las vaya llamando cada vez que las necesite según la estructura que haya resultado para el mecanismo a analizar.

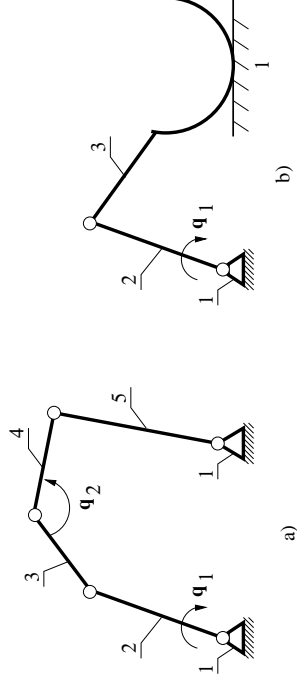


Figura 2.1: Mecanismos en los que no sería aplicable la teoría de Assur.

### 2.1.1.1. Nuevo concepto de Grupo Estructural

Sin embargo, la teoría del análisis estructural propuesta por Assur es aplicable a mecanismos articulados planos, para los que los movimientos de entrada conocidos deben ser absolutos, es decir referidos a un sistema fijo. Por tanto presenta el inconveniente de no ser aplicable a mecanismos como los mostrados en la Figura 2.1., en la que se muestran 2.1.a mecanismo de 5 barras con movimiento de entrada relativo entre los eslabones 3 y 4 y 2.1.b en la que se muestra un mecanismo de tres barras con un par superior entre 1 y 3.

Se hace necesario, por tanto, ampliar el concepto de grupo estructural y establecer nuevos métodos y bases para el análisis estructural de cualquier tipo de mecanismo. Sin embargo, esto no quiere decir que la teoría de Assur sea obsoleta, muy al contrario, resulta fundamental para el análisis por los nuevos métodos en los que, los grupos de Assur pasan a ser un caso particular de grupo estructural.

Un *grupo estructural* es una cadena cinemática en la que el número de movimientos de entrada independientes  $n_c$  sobre sus eslabones coincide con su movilidad  $L_c$ , y no puede subdividirse en otros grupos estructurales más reducidos.

$$L_c = n_c \quad (2.1)$$

La expresión del criterio de Grübler para determinar la movilidad de una cadena cinemática viene dada por:

$$L_c = 3 \cdot N_m - 2 \cdot p_i - p_s$$

Donde si  $N_m$  es el número de eslabones móviles,  $3 \cdot N_m$  representa la movilidad de todos los eslabones móviles sin atender a la forma en que están conectados entre sí, el término  $-2 \cdot p_i - p_s$  son los grados de libertad que eliminan los pares cinemáticos. Analicemos este segundo término para llegar a una formulación que servirá como base para obtener grupos estructurales.

El número de grados de libertad que eliminan los pares se puede expresar como:

$$-2 \cdot p_i - p_s = 3 \cdot P - S_c$$

donde  $P$  = número de pares cinemáticos en la cadena cinemática (sin atender a su tipo).

$S_c$  = Movilidad que dejan los  $p_k$  pares cinemáticos de grado  $k$  dada por (en mecanismos planos):

$$S_c = \sum_{k=1}^2 k \cdot p_k = 1 \cdot p_1 + 2 \cdot p_2$$

De modo que podemos expresar la movilidad de una cadena cinemática como:

$$L_c = 3 \cdot (N_m - P) + S_c \quad (2.2)$$

Como ejemplo, apliquemos la ecuación (Ec. 2.2) al cuadrilátero articulado de la Figura 2.2

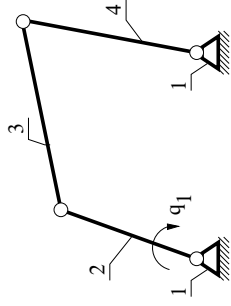


Figura 2.2: Cuadrilátero articulado: Topología.

- $N_m = 3$ ;  $P = 4$ ;  $p_i = 4$ ;  $p_s = 0$
- $S_C = 1 \cdot p_i + 2 \cdot p_s = 4$ .<sup>1</sup>

$$L_c = 3 \cdot (3 - 4) + 4 = -3 + 4 = 1$$

Entonces, sustituyendo en la ecuación (Ec. 2.2) la definición dada de grupo estructural (Ec. 2.1) obtenemos la condición fundamental para que una cadena cinemática sea grupo estructural:

$$L_c = n_c = 3 \cdot (N_m - P) + S_c$$

o bien:

$$S_c - n_c = 3 \cdot (P - N_m) \quad (2.3)$$

A partir de la definición dada se puede observar que el número de combinaciones que se pueden formar entre eslabones y pares cinemáticos para formar grupos estructurales, aunque limitado, es muy superior a las que se obtenían aplicando la teoría de Assur.

Evidentemente, se puede demostrar que los grupos de Assur cumplen la condición fundamental anterior (Ec. 2.3). Ahora, un grupo de Assur cuenta como un caso particular de grupo estructural de movilidad cero, siempre que sobre sus eslabones no actúe ningún movimiento de entrada.

Análogamente al mecanismo plano, la condición fundamental de grupo estructural para un mecanismo espacial queda de la siguiente manera:

$$L_c = n_c = 6 \cdot (N_m - P) + S_c$$

o bien:

$$S_c - n_c = 6 \cdot (P - N_m) \quad (2.4)$$

### 2.1.2. Grafo estructural de un mecanismo

Se puede observar fácilmente que los mecanismos pueden generarse por la unión sucesiva de grupos estructurales simples al bastidor.

Junto al esquema cinemático de un mecanismo, y como base para su análisis estructural, suele dibujarse el *grafo estructural*, otro esquema asociado al mecanismo en el que se muestran los pares existentes entre eslabones, su tipo y entre qué eslabones existe movimiento independiente de entrada. El *grafo estructural* es, además, una herramienta muy útil que permite sistematizar el proceso de obtención de grupos estructurales y por tanto, el análisis estructural del mecanismo.

---

<sup>1</sup>La movilidad que dejan cuatro pares inferiores es 4 ya que de los tres posibles grados de libertad, cada uno restringe dos - permite uno-.

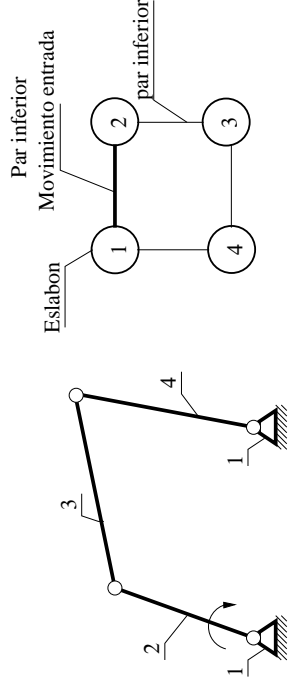


Figura 2.3: Cuadrilátero articulado y grafo estructural.

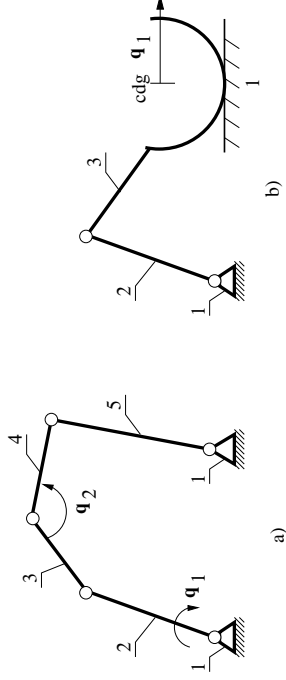


Figura 2.4: Mecanismos planos en los que no es aplicable la teoría de Assur.

Una vez obtenida la estructura de un mecanismo se suele representar por su *diagrama estructural*, el cual muestra el acoplamiento entre los distintos grupos estructurales entre sí, junto con el número de eslabones de cada grupo y su número de grados de libertad.

En este apartado se muestra la construcción de grafos estructurales; la del diagrama estructural se presentará después de introducir el método de análisis basado en el grafo estructural.

### Grafo estructural

Está formado por tantos círculos como eslabones exista en el mecanismo. En el interior de cada círculo se escribe el número del eslabón al que representa. Como ejemplo se aplica al cuadrilátero articulado-Figura 2.3.

Estos círculos se unen si entre ellos existe par cinemático en el mecanismo. En ese caso, se unen con tantas líneas como movilidad tenga el par que formen ( $p_i = 1$ ;  $p_s = 2$ ), y de esas líneas serán de trazo grueso tantas como movimientos de entrada independientes exista entre los dos eslabones del par.

En la Figura 2.3, los eslabones 1 y 2 están unidos mediante un par inferior (una línea), y como entre ellos hay un movimiento de entrada ( $q_1$ ), esa línea deberá ser de trazo grueso. Entre los demás eslabones no hay movimientos de entrada y todos los pares son inferiores, por lo tanto, todos están unidos con una única línea de trazo fino. No existe línea entre 1 y 3 porque entre ellos no hay par cinemático.

La Figura 2.4 muestra otros ejemplos de mecanismos planos, para los que se obtendrá su grafo estructural asociado.

En el mecanismo de la Figura a) obtendremos cinco círculos; conexiones de una sola línea con trazo grueso entre 1 – 2 y 3 – 4, siendo fino entre el resto 2 – 3, 4 – 5 y 1 – 5. El grafo se muestra en la Figura 2.5.a

Para el mecanismo de la Figura b) obtendremos tres círculos; conexión de dos líneas entre 1 – 3 de las cuales una es de trazo grueso y otra fina. Entre 1 – 2 y 2 – 3 líneas simples de trazo fino. El grafo se muestra en la Figura 2.5.b

Un concepto interesante relacionado con el grafo estructural es el de *número ciclomático*, que se define como el número de circuitos (lazos) cerrados independientes en el grafo de un mecanismo, y que puede calcularse según la expresión:

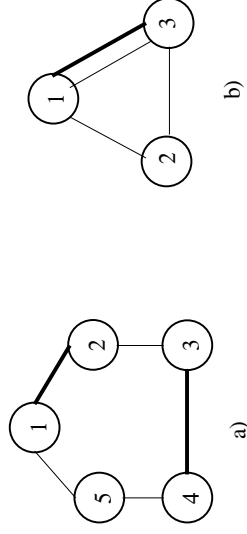


Figura 2.5: Grafo estructural de los mecanismos de la Figura 2.4.

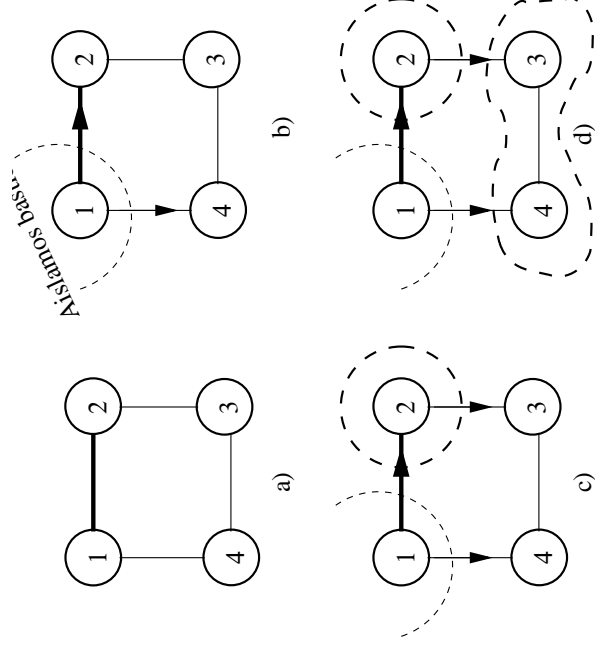


Figura 2.6: Análisis Estructural del cuadrilátero articulado basado en grafo estructural.

$$C = P - N_m$$

Así, para los grafos de la Figura 2.5a y b tendríamos:

$$C_a = P_a - N_{ma} = 5 - 4 = 1$$

$$C_b = P_b - N_{mb} = 3 - 2 = 1$$

### 2.1.3. Análisis basado en el grafo. Diagrama estructural

El análisis estructural de un mecanismo puede realizarse con ayuda del grafo estructural, aplicando la condición fundamental (Ec. 2.3 o 2.4). Para mostrarlo se aplica al análisis del cuadrilátero articulado.

Se representa el grafo estructural del mecanismo a analizar (Figura 2.6.a) y se aísla del resto el círculo correspondiente al bastidor (Figura 2.6.b, se muestra rodeado de una línea de trazo discontinuo). El enlace entre dos eslabones se lo lleva el que se separa. Esto se puede indicar en el grafo mediante una flecha sobre la línea que une dos eslabones. En la Figura 2.6.b, al aislar el bastidor, el par entre 1 - 2 se lo lleva 2 y entre 1 - 4 se lo lleva 4.

Ahora se buscan grupos a partir de uno de los pares externos representados.

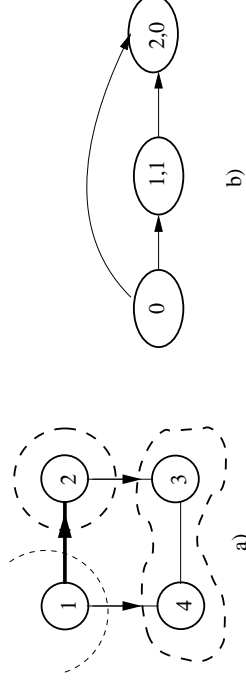


Figura 2.7: Diagrama Estructural del cuadrilátero articulado.

Se puede comenzar por estudiar el eslabón 2 con el enlace que se ha llevado (la flecha sirve también para indicar que ese enlace tiene que ir con ese eslabón). Para ello, rodeamos el eslabón 2 con una línea de trazo discontinuo y señalamos con una flecha que el enlace entre 2 – 3 se lo lleva el eslabón 3 (Figura 2.6.c). Se comprueba si se cumple la condición fundamental (observe que al eslabón 2 le llega un par sólo, el 1 – 2 y que ese par es línea gruesa):  $S_c = 1$ ;  $n_c = 1$ ;  $P = 1$ ;  $N_m = 1$ :

$$1 - 1 = 3 \cdot (1 - 1)$$

Como sí se cumple, el eslabón 2 con ese enlace es un grupo estructural. Ahora tomando el enlace 2 – 3 junto con el eslabón 3, se puede comprobar si se cumple la condición para ese eslabón sólo:

$$1 - 0 = 3 \cdot (1 - 1) \Rightarrow 1 \neq 0$$

por lo que no lo es. Para que un eslabón sólo sea grupo estructural, debe estar unido al bastidor o a otro grupo estructural mediante sólo líneas gruesas, ya que como  $P = 1$ ;  $N_m = 1$  el segundo término de la condición se anula, y para ello debe ser cero también el primero:  $S_c = n_c$ .  $S_c$  se puede determinar como la suma de líneas gruesas « $l_g$ » y finas « $l_f$ » que llegan a los eslabones del grupo, y  $n_c$  se puede determinar como número de líneas gruesas que lleguen, entonces tenemos:

$$l_f + l_g - l_g = 0$$

Por lo tanto, para que un solo eslabón sea grupo, « $l_f$ » debe ser cero.

En el mecanismo cuadrilátero articulado, como el eslabón 3 sólo no cumple esa condición, el próximo grupo debe ser el formado por los eslabones 3 y 4, para el cual sí se cumple la condición fundamental:

$$3 - 0 = 3 \cdot (3 - 2)$$

Ahora el número de pares es tres, entre los que se cuentan los dos externos: 1 – 4 y 2 – 3 y el interno 3 – 4 entre los dos eslabones. Para terminar marcamos con trazo discontinuo el grupo obtenido (Figura 2.6.d).

### Diagrama estructural

Está formado por tantos círculos como grupos estructurales simples formen el mecanismo más uno (este último corresponde al bastidor).

En el círculo que corresponde al bastidor se escribe el número 0 porque no representa ningún grupo.

Para el resto de grupos estructurales se escribe, en un círculo por grupo, el número de eslabones que lo forman y el número de grados de libertad del mismo.

Dos círculos del diagrama se unen con flechas si contienen eslabones que en el mecanismo original están también unidos. Las líneas que unen dos círculos en el diagrama estructural llevan el mismo sentido que la asignación de los pares cinemáticos en el grafo.

Aplicando lo descrito al cuadrilátero articulado se obtiene el diagrama de la Figura 2.7.b.



Se puede observar cómo en el diagrama estructural (Figura 2.7.b) las líneas van del círculo que representa al bastidor (0) al grupo estructural (1,1) - formado por el eslabón 2, tal y como se obtuvo en su grafo (el par se lo lleva 2). De la misma forma el grupo (3-4) se lleva dos pares inferiores, y por eso hacia él van dirigidas las flechas correspondientes en el diagrama, una desde el bastidor y otra desde el grupo estructural (1,1).

### Consideración de la rodadura pura

La condición de rodadura pura es una excepción a la aplicación de los criterios de movilidad. Por cada condición de rodadura pura se debe eliminar un grado de libertad a los obtenidos aplicando el criterio de Grübler. La forma en que la rodadura pura debe considerarse en el grafo estructural, depende de si existe o no cierre de cadena entre los eslabones en los que aparece esa condición. Si existe cierre de cadena (Ej. Tren ordinario o epicicloidal) se representa como un par superior, es decir, con dos líneas de trazo fino. Si no existe cierre de cadena se representará con dos líneas (par superior) pero una de trazo fino y otra de trazo grueso, indicando que existe una ecuación que liga las variables cinemáticas que definen la posición de los eslabones.

## 2.1.4. Análisis estructural computacional

La aplicación de la separación de grupos estructurales por métodos gráficos-analíticos y computacionales, basados en la teoría de Assur [40, 52, 34], y más recientemente, basados en procesos combinatorios [39], han sido tratada ampliamente en literatura para mecanismos planos con limitación a pares cinemáticos inferiores, grupos de Assur y un grado de libertad.

El método que se describe en este proyecto aplica un algoritmo, desarrollado previamente para el análisis estructural computacional, que usa un concepto extendido de grupos estructurales, haciéndolo útil no solamente a mecanismos planos con grupos de Assur y un grado de libertad, sino también a sistemas multicuerpo planos.

La solución empieza, como se muestra en el algoritmo 2 que después se presenta, leyendo información detallada del sistema multicuerpo plano, a través de un archivo separado (*MBDatos*). Esta información incluye la matriz adjunta, que contiene la topología del sistema y es suficiente para desarrollar el análisis estructural computacional (línea 2.3), y obtiene la estructura cinemática *KinStruc* del sistema multicuerpo, accediendo a algoritmos separados que han sido desarrollados y explicados en trabajos previos.

## 2.2. Formulación cinemática de un Grupo Estructural

La estructura cinemática de un sistema multicuerpo define los grupos estructurales que lo componen y el orden específico en que tienen que ser resueltos. La solución del sistema multicuerpo se lleva a cabo mediante la resolución, en cada etapa de cálculo, de la cinemática de todos sus grupos estructurales. Estos pueden ser programados en subrutinas específicas, separadas, y cuya creación puede ser estandarizada siguiendo los pasos que se describen a continuación. Se consideran dos formulaciones cinemáticas diferentes: en la primera, la matriz Jacobiana del sistema se deriva con respecto al tiempo; en la segunda se deriva con respecto a las coordenadas independientes de todo el sistema, dando lugar a un tensor de tercer orden.

### 2.2.1. Método basado en derivada temporal de matriz Jacobiana

La cinemática de cualquier grupo estructural puede resolverse si se selecciona el conjunto apropiado de coordenadas de grupo,  $\mathbf{q}_G$ , y quedan definidas sus correspondientes ecuaciones de restricción  $\Phi$ . La subrutina específica que resuelve el grupo se puede programar en consecuencia con los siguientes pasos.

**Identificación de las coordenadas de grupo y parámetros:** Se define un sistema de coordenadas local, solidario a cada eslabón, y se selecciona un conjunto de coordenadas (de cualquier tipo) que definan la

cadena cinemática. Presentamos dos subconjuntos de coordenadas de grupo: dependientes  $\phi$  e independiente  $\mathbf{h}$ . Las segundas son diferentes de las coordenadas independientes de todo el sistema (en adelante  $\mathbf{z}$ , como en muchas formulaciones recursivas). Otros parámetros, que dependerán del grupo estructural específico a resolver, serán determinados a partir de la geometría del problema, así como los resultados del análisis estructural computacional; los puntos de referencia necesarios para identificar las coordenadas  $\mathbf{h}$ , dimensiones de los cuerpos, etc. A modo de ejemplo, un grupo estructural 3R de Assur se muestra en la Figura 2.8.c. Los sistemas locales  $(\{\eta_i \xi_i\}, \{\eta_j \xi_j\})$  son solidarios a los eslabones  $(i, j)$ , las coordenadas dependientes quedan  $\varphi = [x_4, y_4]$  y las coordenadas independientes  $\mathbf{h} = [x_3, y_3, x_2, y_2]$ .

**Solución del problema de posición por grupos estructurales:** Para resolver el problema de posición de cualquier grupo estructural, se han de definir las ecuaciones de restricción correspondientes, de conformidad con el tipo seleccionado de coordenadas (2.5). Para el sistema dado de ecuaciones de restricción, los términos de la matriz Jacobiana  $\Phi_\varphi$  se obtienen analíticamente, y por el método iterativo de Newton-Raphson se obtendrán los valores de las coordenadas dependientes de grupo (2.5). Dependiendo de la geometría del grupo estructural, puede existir la solución explícita del problema de posición y debería de ser tenida en cuenta para reducir el tiempo de cálculo.

$$\Phi = 0 \quad \rightarrow \quad \varphi_k = \varphi_{k-1} - (\Phi_\varphi)_{k-1}^{-1} \cdot \Phi_{k-1} \quad (2.5)$$

**Solución del problema de velocidad:** Como los valores de las velocidades independientes de grupo  $\dot{\mathbf{h}}$  son conocidos, el problema de velocidad puede formularse derivando las ecuaciones de restricción respecto del tiempo, y resolviendo las coordenadas dependientes (2.6). No solamente la matriz Jacobiana  $\Phi_{\mathbf{h}}$ , sino la expresión completa  $-(\Phi_\varphi)^{-1} \Phi_{\mathbf{h}}$  se puede obtener analíticamente, en la mayoría de casos, para grupos estructurales planos y espaciales.

$$\dot{\Phi}(\mathbf{q}, t) = 0 \quad \rightarrow \quad \dot{\varphi} = -(\Phi_\varphi)^{-1} [\Phi_{\mathbf{h}} \dot{\mathbf{h}}] \quad (2.6)$$

**Solución del problema de aceleración:** La solución del problema de aceleración, para coordenadas dependientes de grupo, puede obtenerse derivando la velocidad de las ecuaciones de restricción respecto del tiempo (2.7). De nuevo, debido a las reducidas dimensiones de las matrices involucradas, la mayoría de los cálculos se desarrollan analíticamente y se incluyen en cada subrutina de los grupos estructurales. En los casos más complicados, como los grupos estructurales espaciales, se necesitará la formulación general aquí presentada.

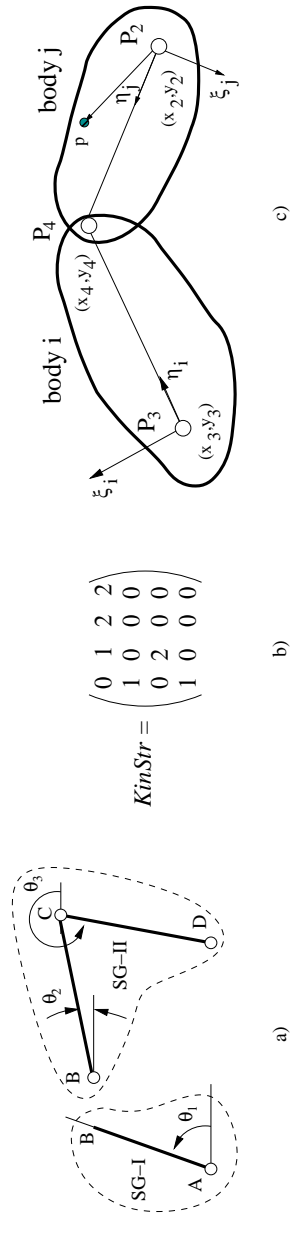


Figura 2.8: a) Cuadrilátero escalable dividido en grupos estructurales: SG-I, SG-II y coordenadas. b) La estructura cinemática computacional está representado por la matriz *KinStr*. c) Sistemas de coordenadas locales y coordenadas de grupo en el grupo estructural 3R.

$$\Phi_{\varphi}\dot{\varphi} + \dot{\Phi}_{\mathbf{q}_G}\mathbf{q}_G = 0 \quad \rightarrow \quad \dot{\varphi} = -(\Phi_{\varphi})^{-1} \left[ \Phi_{\mathbf{h}}\dot{\mathbf{h}} + \dot{\Phi}_{\mathbf{q}_G}\mathbf{q}_G \right] \quad (2.7)$$

**Solución cinemática de otros POI's:** Aparte de las coordenadas dependientes, se pueden necesitar los resultados de otros puntos de interés (POI's), como centro de masas o puntos de referencia para otros grupos estructurales. La posición, velocidad y aceleración de un POI que pertenece a un eslabón ( $p \in j$ , Figura 2.8.c) se obtiene fácilmente siempre que sus coordenadas locales sean conocidas  $\bar{\mathbf{r}}_p = [\bar{x}_p \bar{y}_p]$  (2.8). Para ello se requiere del cálculo de la matriz de rotación y sus derivadas temporales (2.9).

$$\bar{\mathbf{r}}_p = \mathbf{r}_{ref} + \mathbf{A}\bar{\mathbf{r}}_p; \quad \dot{\bar{\mathbf{r}}}_p = \dot{\mathbf{r}}_{ref} + \dot{\mathbf{A}}\bar{\mathbf{r}}_p; \quad \ddot{\bar{\mathbf{r}}}_p = \ddot{\mathbf{r}}_{ref} + \ddot{\mathbf{A}}\bar{\mathbf{r}}_p \quad (2.8)$$

$$\mathbf{A} = \begin{bmatrix} x_j - x_i & y_i - y_j \\ y_j - y_i & x_j - x_i \end{bmatrix}; \quad \dot{\mathbf{A}} = \begin{bmatrix} \dot{x}_j - \dot{x}_i & \dot{y}_i - \dot{y}_j \\ \dot{y}_j - \dot{y}_i & \dot{x}_j - \dot{x}_i \end{bmatrix}; \quad \ddot{\mathbf{A}} = \begin{bmatrix} \ddot{x}_j - \ddot{x}_i & \ddot{y}_i - \ddot{y}_j \\ \ddot{y}_j - \ddot{y}_i & \ddot{x}_j - \ddot{x}_i \end{bmatrix} \quad (2.9)$$

### 2.2.2. Método basado en el tensor de tercer orden $\Phi_{qq}$

En algunas formulaciones cinemáticas y dinámicas es preferible expresar las velocidades y aceleraciones de las coordenadas dependientes de grupo, respecto de las coordenadas independientes de todo el sistema (conocidas como  $\mathbf{z}$ ). Estas formulaciones se pueden usar también con ayuda de la descomposición en grupos estructurales. Los dos primeros pasos son similares a la primera formulación. Los problemas de velocidad y aceleración se resuelven introduciendo las matrices  $\mathbf{R}$  y  $\mathbf{S}$  (2.10), que son extensamente utilizadas en recursivas, como un método alternativo a la estabilización de las ecuaciones de restricción, propuesta por de Baumgarte. [8].

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}^d \\ \mathbf{S}^i \end{bmatrix} = \begin{bmatrix} (\Phi_{\varphi})^{-1} \\ 0_{f \times m} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{R}^d \\ \mathbf{R}^i \end{bmatrix} = \begin{bmatrix} -(\Phi_{\varphi})^{-1}\Phi_{\mathbf{q}_G}^i \\ I_f \end{bmatrix}, \quad \mathbf{K}_{\varphi\mathbf{h}} = \mathbf{R}^d \quad (2.10)$$

**Solución al problema de velocidad:** Los elementos de la matriz  $\mathbf{K}_{\varphi\mathbf{h}}$  corresponden a los coeficientes de velocidad, y muestran que las velocidades dependientes de grupo  $\dot{\varphi}$  pueden ser expresadas como una combinación lineal de las independientes  $\dot{\mathbf{h}}$  (2.11).

$$\mathbf{K}_{\varphi\mathbf{h}} = -(\Phi_{\varphi})^{-1}\Phi_{\mathbf{h}} \rightarrow \dot{\varphi} = \mathbf{K}_{\varphi\mathbf{h}}\dot{\mathbf{h}} \quad (2.11)$$

Los coeficientes de velocidad  $\mathbf{K}_{\mathbf{h}\mathbf{z}}$ , que definen la velocidad de las coordenadas independientes de grupo, como una combinación lineal de las velocidades independientes del sistema  $\dot{\mathbf{z}}$ , se han obtenido para las coordenadas de los grupos estructurales previos. Por tanto, es sencillo expresar la velocidad de las coordenadas dependientes de grupo en términos de las independientes del sistema, por medio de sus respectivos coeficientes de velocidad  $\mathbf{K}_{\varphi\mathbf{z}}$ , muy útiles en formulaciones dinámicas, y con ello calcular las velocidades dependientes (2.12).

$$\mathbf{K}_{\varphi\mathbf{z}} = \mathbf{K}_{\varphi\mathbf{h}}\mathbf{K}_{\mathbf{h}\mathbf{z}} \quad \rightarrow \quad \dot{\varphi} = \mathbf{K}_{\varphi\mathbf{z}}\dot{\mathbf{z}} \quad (2.12)$$

**Solución del problema de aceleración:** Para resolver el problema de aceleración, en términos de coordenadas independientes del sistema, se ha de calcular la derivada temporal de (2.12). Considerando la regla de la cadena, se obtiene el sistema mostrado en (2.13). En esta aproximación, en vez de derivar los coeficientes de velocidad respecto al tiempo, como en la anterior formulación, serán derivados respecto de las coordenadas  $\mathbf{z}$ .

$$\dot{\varphi} = \mathbf{K}_{\varphi\mathbf{z}} \cdot \dot{\mathbf{z}} + \sum_{i=1}^f \dot{z}_i \cdot (\mathbf{L}_{\varphi\mathbf{z}})_{z_i} \cdot \dot{\mathbf{z}} \quad (2.13)$$

Los términos  $(\mathbf{L}_{\varphi\mathbf{z}})_{z_i}$  en el sumatorio se definen como derivadas de los coeficientes de velocidad: derivada de la matriz de coeficientes de velocidad  $\mathbf{K}_{\varphi\mathbf{z}}$  respecto a una coordenada independiente dada  $z_i$ .

$$(\mathbf{L}_{\varphi\mathbf{z}})_{z_i} = \frac{d \mathbf{K}_{\varphi\mathbf{z}}}{d z_i}$$

El cálculo de las derivadas de los coeficientes de velocidad es lo más complicado y laborioso de esta aproximación. Sin embargo, por comparación de la aceleración de las coordenadas dependientes de grupo, expresadas en términos de las coordenadas independientes del sistema  $\mathbf{z}$  (2.13) y de las independientes de grupo  $\mathbf{h}$ , se determina la expresión analítica (2.14).

$$\ddot{\varphi} = \mathbf{K}_{\varphi\mathbf{h}} \cdot \ddot{\mathbf{h}} + \sum_{j=1}^r \dot{h}_j \cdot (\mathbf{L}_{\varphi\mathbf{h}})_{h_j} \cdot \dot{\mathbf{h}} \quad (2.14)$$

Las aceleraciones  $\ddot{\mathbf{h}}$  se expresan en términos de las coordenadas independientes del sistema  $z_i$  (2.15).

$$\ddot{\mathbf{h}} = \mathbf{K}_{\mathbf{h}\mathbf{z}} \cdot \ddot{\mathbf{z}} + \sum_{i=1}^f \dot{z}_i \cdot (\mathbf{L}_{\mathbf{h}\mathbf{z}})_{z_i} \cdot \dot{\mathbf{z}} \quad (2.15)$$

Y las velocidades independientes de grupo, en términos de las velocidades independientes del sistema (2.16):

$$\dot{\mathbf{h}} = \mathbf{K}_{\mathbf{h}\mathbf{z}} \cdot \dot{\mathbf{z}} \quad (2.16)$$

Sustituyendo (2.15) y (2.16) en (2.14) se obtiene:

$$\ddot{\varphi} = \mathbf{K}_{\varphi\mathbf{h}} \mathbf{K}_{\mathbf{h}\mathbf{z}} \cdot \ddot{\mathbf{z}} + \sum_{i=1}^f \dot{z}_i \cdot \left( \mathbf{K}_{\varphi\mathbf{h}} \cdot (\mathbf{L}_{\mathbf{h}\mathbf{z}})_{z_i} + \sum_{j=1}^r \mathbf{K}_{h_j z_i} \cdot (\mathbf{L}_{\varphi\mathbf{h}})_{h_j} \cdot \mathbf{K}_{\mathbf{h}\mathbf{z}} \right) \cdot \dot{\mathbf{z}} \quad (2.17)$$

Comparando (2.17) y (2.13) parece que los coeficientes de las aceleraciones independientes  $\ddot{\mathbf{z}}$  corresponden a aquellos ya introducidos en (2.12), y que los coeficientes de los términos de aceleración centrípeta y de Coriolis  $\mathbf{L}_{\varphi\mathbf{z}}$  que estamos buscando, corresponden a las expresión entre paréntesis en (2.17). Por tanto se puede escribir (2.18) para calcular sus valores.

$$(\mathbf{L}_{\varphi\mathbf{z}})_{z_i} = \left( \mathbf{K}_{\varphi\mathbf{h}} \cdot (\mathbf{L}_{\mathbf{hz}})_{z_i} + \sum_{j=1}^r \mathbf{K}_{h_j z_i} \cdot (\mathbf{L}_{\varphi\mathbf{h}})_{h_j} \cdot \mathbf{K}_{\mathbf{hz}} \right) \quad (2.18)$$

En (2.18) las derivadas de los coeficientes de velocidad  $(\mathbf{L}_{\mathbf{hz}})_{z_i}$  son conocidos a través de POI's que pertenecen a grupos estructurales previos. Las derivadas de los coeficientes de velocidad  $(\mathbf{L}_{\varphi\mathbf{h}})_{h_j}$  de las coordenadas dependientes de grupo  $\varphi$ , respecto a las coordenadas  $\mathbf{h}$ , pueden ser obtenidos analíticamente y numéricamente usando (2.19).

$$(\mathbf{L}_{\varphi\mathbf{h}})_{h_j} = \mathbf{S} \cdot \left( \frac{d\Phi_{\mathbf{h}}}{dh_j} - \frac{d\Phi_{\varphi}}{dh_j} \cdot \mathbf{K}_{\varphi} \right) \quad (2.19)$$

Ambas formulaciones topológicas presentadas en esta sección han sido implementadas en el entorno de programación de Matlab, junto con una formulación global (apartado 2.3) y su eficiencia aparecerá comparada en capítulos posteriores.

### 2.3. Cinemática computacional de sistemas multicuerpo

En este apartado se presentan dos algoritmos para el análisis cinemático de sistemas multicuerpo planos, extrapolable a espaciales. El primero corresponde a una formulación global y el segundo a un desarrollo específico, desarrollado en este proyecto, que también considera el análisis estructural computacional y la modelación automática, así como la solución de sistemas planos multicuerpo.

**Algoritmo 1:** Solución cinemática global

```

1.1: MBDatos /* Lee datos MBS */
1.2: for t = t0 : pasoTemporal : tf do
1.3: | z = z + Δz /* establece valores
1.4: | indep. */
1.5: | % Problema de posición %
1.6: | evalúa Φ → mFi
1.7: | error = norm(Φ)
1.8: | while error > tolerance do
1.9: | | evalúa Φq → Jacob
1.10: | | extrae Φqd
1.11: | | resuelve
1.12: | |  $\mathbf{q}_k^d = \mathbf{q}_{k-1}^d - \left(\Phi_{\mathbf{q}}^d\right)_{k-1}^{-1} \cdot \Phi_{k-1}$ 
1.13: | | evalúa Φ → mFi
1.14: | | error = norm(Φ)
1.15: | | end
1.16: | | % Problema de velocidad % % %
1.17: | | evalúa Φq → Jacob
1.18: | | extrae Φqd
1.19: | | extrae Φqi
1.20: | | resuelve  $\dot{\mathbf{q}}^d = - \left(\Phi_{\mathbf{q}}^d\right)^{-1} \Phi_{\mathbf{q}}^i \dot{\mathbf{q}}^i$ 
1.21: | | % Problema de
1.22: | | aceleración % % % %
1.23: | | evalúa Φqđ → Fiqqq
1.24: | | evalúa - [ Φqiđi + Φqđ ]
1.25: | | resuelve
1.26: | |  $\ddot{\mathbf{q}}^d = - \left(\Phi_{\mathbf{q}}^d\right)^{-1} \left[ \Phi_{\mathbf{q}}^i \ddot{\mathbf{q}}^i + \dot{\Phi}_{\mathbf{q}} \dot{\mathbf{q}} \right]$ 
1.27: | | z = z + Δz
1.28: | end

```

**Algoritmo 2:** Solución cinemática topológica basada en grupos estructurales

```

2.1: MBDatos; /* Lee datos MBS */
2.2: % Análisis estructural
2.3: KInstr → CompSG /* Estructura
2.4: | cinemática */
2.5: | % Modelización automática MBS
2.6: | MGrupos; POI → MBSModelling
2.7: | % Análisis cinemático MBS
2.8: | for t = t0 : pasoTemporal : tf do
2.9: | | z = z + Δz /* establece valores
2.10: | | indep. */
2.11: | | for ng = 2 : length(MGrupos) do
2.12: | | | /* resuelve cada GE */
2.13: | | | switch MGrupos(ng).tipo do
2.14: | | | | case MGrupos(ng).tipo == 1R
2.15: | | | | | POI → Resuelve_1R
2.16: | | | | case MGrupos(ng).tipo == 3R
2.17: | | | | | POI → Resuelve_3R
2.18: | | | endsw
2.19: | | end
2.20: | end

```

**Algoritmo 3:** Solución cinemática global sparse

```

/* Llama la función Sparse */
3.1: ...
3.2: evalúa Φq → Jacob
3.3: extrae Φqd
3.4: Φqd → Sparse(Φqd) ...
3.5: resuelve  $\mathbf{q}_k^d = \mathbf{q}_{k-1}^d - \left(\Phi_{\mathbf{q}}^d\right)_{k-1}^{-1} \cdot \Phi_{k-1}$ 
3.6: ...

```

**2.3.1. Algoritmo para métodos globales**

En el algoritmo 1 se presenta una formulación global para el análisis cinemático de sistemas planos multicuerpo. Se ha utilizado una porción de coordenadas para evaluar las velocidades y aceleraciones de las coordenadas dependientes  $\mathbf{q}_d$  del sistema, una vez que las independientes  $\mathbf{q}_i$  se conocen. Para este fin, el problema de posición se resuelve usando el método de Newton-Raphson (líneas 1.4 - 1.13). Las ecuaciones de restricción  $\Phi$  y la matriz Jacobiana  $\Phi^d$  son evaluadas en subrutinas por separado, *mFi* y *Jacob* respectivamente. El problema de velocidad se resuelve en las líneas (1.14 - 1.18) y el problema de aceleración en (1.19 - 1.22). Solamente  $\Phi_{\mathbf{q}} \dot{\mathbf{q}}$  tiene que ser evaluado en una subrutina por separado *Fiqqq*. Esta solución permanece en un bucle hasta que se alcanza un valor final (o se define otro criterio de parada).

**2.3.2. Algoritmo propuesto para métodos basados en ecuaciones de grupo**

En el algoritmo 2 se muestra una representación esquemática del método computacional desarrollado para la cinemática de los sistemas multicuerpo, basado en su estructura cinemática. Está dividido principalmente en tres pasos, que corresponden al análisis estructural computacional, la modelización automática de

sistemas multicuerpo planos y finalmente el análisis cinemático computacional del sistema.

### Modelización automática de sistemas multicuerpo

Este procedimiento se divide en tres pasos. El primero almacena el identificador de los eslabones que pertenecen a cada grupo estructural en la estructura de datos *MGrupos.eslabones*. Además, analizando las filas de *MatrizAdjunta*, las coordenadas  $\mathbf{q}_G$  para cada grupo estructural son almacenadas en *MGrupos.coordenadas*, distinguiendo entre dependientes e independientes.

En el segundo paso, se accede de nuevo a *MatrizAdjunta* para determinar el tipo de cada grupo estructural, que será guardado en *MGrupos.eslabones*. Este resultado es usado en el bucle de cierre (2.9 - 2.17) para desarrollar el análisis cinemática completo, accediendo a subrutinas específicas (switch-case, en 2.10 - 2.16) de acuerdo a cada uno de los grupos estructurales que han sido obtenidos.

En el tercer y último paso, el algoritmo permite al usuario introducir información adicional, con respecto a cada de los grupos estructurales obtenidos por dos métodos: usando un interfaz gráfico de usuario o introduciendo el nombre del archivo que contiene la correspondiente información.

### Análisis cinemático computacional

Aunque existe mucha literatura referente a métodos globales y topológicos para el análisis cinemático y dinámico de sistemas multicuerpo, pocos trabajos han utilizado soluciones basadas en descomposición de grupos estructurales. En [54] se presenta un algoritmo para el análisis cinemático computacional de mecanismos, formados por cualquier número de grupos de Assur 3R (dos eslabones y tres pares cinemáticos de rotación). En [51] se introduce una solución cinemática eficiente para sistemas multicuerpo planos, un grado de libertad, reordenando los elementos de su matriz Jacobiana. Varbanov [36] presenta la integración de síntesis cinemática y análisis de mecanismos planos, formados por un eslabón, con un grado de libertad, combinado con diferentes clases de diadas (grupos de Assur de clase I). Hay muy pocos trabajos que desarrollen automáticamente el análisis estructural computacional de mecanismos con cualquier número de grados de libertad, planos o espaciales, y que usen sus resultados para la modelización automática y análisis cinemático de un sistema multicuerpo dado.

Una vez que se ha modelizado el sistema multicuerpo plano, un análisis cinemático basado en las formulaciones presentadas en los apartados (2.2.1) o (2.2.2), puede ser desarrollado automáticamente, llamando subrutinas (Algoritmo 2, líneas 2.9 - 2.17). Esas subrutinas podrían seguir un esquema similar al de las líneas (1.3 - 1.21, Algoritmo 1), pero adaptado a cada grupo estructural específico.





## Capítulo 3

# Materiales y métodos

Una vez presentados en el Capítulo 2 los fundamentos de análisis estructural y las distintas formulaciones para el análisis cinemático, sólo queda la puesta en práctica a casos de estudio.

En una primera parte de este capítulo se presentan una serie de grupos estructurales de lo más variado, tanto planos como espaciales, para los que se da solución a los problemas de posición, velocidad y aceleración. Se establece el procedimiento pautado y extrapolable de análisis para que sea de utilidad práctica tanto para esos grupos estructurales como para otros distintos a ellos.

En la segunda parte se analizan dos mecanismos escalables, uno plano y otro espacial. Por ser escalables, las incógnitas para resolver los problemas varían en saltos finitos, lo que permite posteriormente estudiar los tiempos de resolución cinemática en función del número de coordenadas, de ahí la elección de estos mecanismos y no otros. Además, se presenta un problema plano y otro espacial para demostrar que la solución cinemática se alcanza tanto para dos como para tres dimensiones.

Para finalizar el capítulo se expone la implementación del análisis cinemático computacional en Matlab. Para los casos de estudio se seleccionarán unas variables características, las cuales serán analizadas en el capítulo posterior.

### 3.1. Modelización de grupo estructural mediante método de derivada temporal

Las rutinas que resuelven la cinemática de los diferentes grupos estructurales son específicas para cada tipo de grupo. Sin embargo, los fundamentos del análisis cinemático son los mismos; se basan en una formulación expuesta en el apartado 2.2 del capítulo de fundamentos.

Utilizando el método basado en la derivada temporal de la matriz Jacobiana, presentado en el Capítulo 2: Fundamentos, sub-apartado 2.2.1, se seguirán los pasos para resolver la cinemática de cualquier tipo de grupo estructural. Este procedimiento se aplica para la obtención de las expresiones que resuelven la cinemática completa de los grupos planos *1RF*, *1RM*, *3RF*, *3RM*, y los grupos espaciales *2C3E1R*, *1C1E1R* y *1C1E1P*.

Entre las coordenadas más utilizadas para la modelización de sistemas se encuentran tres. En primer lugar las de punto de referencia, que definen la posición de cada sólido mediante las coordenadas cartesianas de un punto y tres o más parámetros que describen su orientación angular. En segundo lugar las coordenadas relativas, que describen la posición de cada sólido respecto al sólido anterior en la cadena cinemática, utilizando las coordenadas de movimiento relativo permitidas por dicho par. Por último las coordenadas naturales, que definen la posición de todos los sólidos mediante las coordenadas cartesianas de punto y vectores unitarios, compartiendo cuando sea posible dichos puntos y vectores para definir pares sin ecuaciones adicionales. Para cada tipo de coordenadas se asocia un tipo de ecuaciones de restricción, resultando más sencillas las asociadas a las coordenadas naturales. Por ello en este capítulo se utilizan estas coordenadas. Por otra parte, las coordenadas relativas son las que necesitan un número menor de variables y las coordenadas de punto de referencia las que permiten plantear las ecuaciones de dinámica de manera más sencilla.

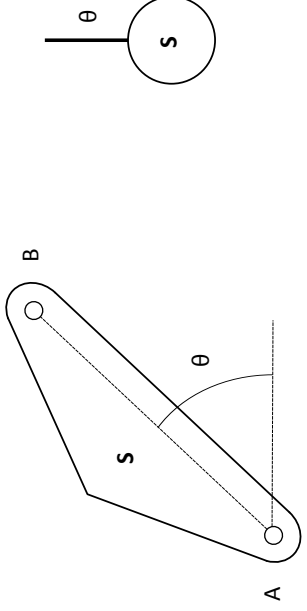


Figura 3.1: Grupo estructural 1R. Esquema cinemático y representación del grafo estructural. Este grupo consta de un único eslabón sujeto a un giro absoluto definido por la coordenada independiente  $\theta$ . El eslabón está articulado al menos en  $A$ .

En lo que sigue se aplica el método de análisis cinemático a diferentes tipos de grupos estructural.

### 3.1.1. Par de rotación 1R

La Figura 3.1 muestra un grupo estructural tipo 1R. Está formado por un único eslabón rígido  $S$ , articulado a otro sólido que puede ser fijo (grupo 1RF) o móvil (grupo 1RM); por mayor generalidad se va a resolver en este apartado el grupo 1RM.

Este grupo tiene movilidad uno, movimiento de entrada representado por la línea de trazo grueso en el grafo. La coordenada asociada a ese grado de libertad es  $\theta$ , con valores iniciales  $(\theta)_0$ ,  $(\dot{\theta})_0$  y  $(\ddot{\theta})_0$  conocidos. Como se supone que el eslabón está articulado en el punto  $A$ , este punto tiene coordenadas y derivadas de coordenadas asociadas conocidas. El punto  $B$  es un punto de interés, en adelante POI (point of interest).

#### 1.- Identificación de las coordenadas de grupo y parámetros

Se modeliza el grupo estructural en coordenadas naturales. Para que la posición del eslabón quede perfectamente definida, se toman dos puntos básicos  $A$  y  $B$ . Entre las coordenadas de grupo se hace la distinción de: dependientes  $\varphi$  e independientes  $\mathbf{h}$ .

Al estar articulado el grupo en el punto  $A$ , se consideran las coordenadas asociadas a este punto como independientes de grupo. La coordenada de entrada  $\theta$  también se considera conocida y por tanto independiente del grupo. El punto  $B$  tiene asociadas dos coordenadas que se consideran dependientes de grupo.

En definitiva, se tiene  $\varphi = [x_B, y_B, ]$  coordenadas dependientes de grupo estructural y  $\mathbf{h} = [x_A, y_A, \theta]$  coordenadas independientes.

#### 2.- Solución del problema de posición por grupos estructurales

El número de ecuaciones de restricción  $r_i$ , de sólido rígido, es la diferencia entre el número de coordenadas dependientes utilizado  $n_i$  y los grados de libertad del mecanismo. Para el caso plano de un grupo estructural se tienen tres grados de libertad, por tanto la expresión queda:

$$r_i = n_i - 3 \quad (3.1)$$

Las ecuaciones de restricción asociadas a este grupo estructural son:

$$(x_B - x_A)^2 + (y_B - y_A)^2 - L_1^2 = 0 \quad (3.2)$$

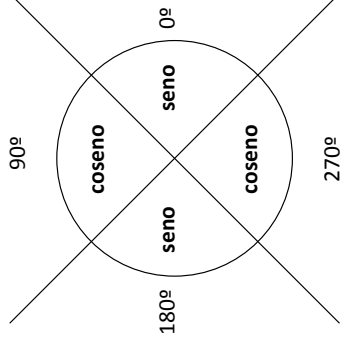


Figura 3.2: Elección de la ecuación de restricción mixta de ángulo en función de  $\theta$ .

$$y_B - y_A - L_1 \cdot \sin \theta = 0 \quad (3.3)$$

$$x_B - x_A - L_1 \cdot \cos \theta = 0 \quad (3.4)$$

Con ellas se forman los vectores  $\Phi$ :

$$\Phi = \begin{bmatrix} (x_B - x_A)^2 + (y_B - y_A)^2 - L_1^2 \\ y_B - y_A - L_1 \cdot \sin \theta \end{bmatrix} \quad (3.5)$$

$$\Phi = \begin{bmatrix} (x_B - x_A)^2 + (y_B - y_A)^2 - L_1^2 \\ x_B - x_A - L_1 \cdot \cos \theta \end{bmatrix} \quad (3.6)$$

La ecuación (3.2) define la restricción de sólido rígido del eslabón, estableciendo las distancias entre puntos constante. No se requieren de ecuaciones de par cinemático ya que el punto  $A$  es perteneciente a la articulación con otro grupo; la articulación es de clase I, elimina un grado de libertad, y por tanto requiere que existan dos ecuaciones de restricción que obliguen a que haya un punto en común entre ambos elementos. Al elegir como punto básico el punto de la articulación quedan implícitas las dos ecuaciones de restricción. Las restantes son ecuaciones mixtas que introducen el ángulo  $\theta$ . Cuando el valor del ángulo se aproxima a cero, la ecuación del coseno (3.4) se hace no válida, lo mismo ocurre con la del seno (3.3) cuando el ángulo se aproxima a 90 grados. Por tanto, se utilizará una u otra dependiendo del valor de  $\theta$ , como muestra la Figura 3.2. Lo mismo sucede con  $\Phi$ .

Para resolver el problema de posición se requiere, además de  $\Phi$  y las posiciones iniciales de las coordenadas dependientes, la matriz Jacobiana  $\Phi_\varphi$ . Esta puede ser analíticamente obtenida y por el método iterativo de Newton-Raphson (3.7) se obtienen los valores de las coordenadas dependientes de grupo.

$$\Phi = 0 \quad \rightarrow \quad \varphi_k = \varphi_{k-1} - (\Phi_\varphi)_{k-1}^{-1} \cdot \Phi_{k-1} \quad (3.7)$$

La matriz Jacobiana  $\Phi_\varphi$ , derivada de  $\Phi$  respecto de las coordenadas dependientes de grupo  $\varphi$ , es dependiendo del caso una u otra de las siguientes:

$$\Phi_\varphi = \begin{bmatrix} 2 \cdot (x_B - x_A) & 2 \cdot (y_B - y_A) \\ 0 & 1 \end{bmatrix} \quad (3.8)$$

$$\Phi_\varphi = \begin{bmatrix} 2 \cdot (x_B - x_A) & 2 \cdot (y_B - y_A) \\ 1 & 0 \end{bmatrix} \quad (3.9)$$

Al igual que sucedía con las ecuaciones de restricción, se obtienen, dependiendo del ángulo  $\theta$  dos matrices  $\Phi_\varphi$ ; cuando el ángulo se aproxime a cero se utilizará (3.8) y cuando el ángulo se aproxime a 90 grados se usará (3.9).

En el caso de tener el grupo  $1RF$ , por de rotación unido al bastidor, el problema de posición coincide con el de  $1RM$ .

### 3.- Solución del problema de velocidad

Para resolver este problema se necesita la matriz  $\Phi_{\mathbf{h}}$ , ya que  $\Phi_\varphi$  ha sido obtenida en el paso anterior. También se requiere del vector de velocidades independientes de grupo  $\dot{\mathbf{h}}$  que es conocido.

$$\dot{\Phi}(\mathbf{q}, t) = 0 \quad \rightarrow \quad \dot{\varphi} = -(\Phi_\varphi)^{-1} [\Phi_{\mathbf{h}} \dot{\mathbf{h}}] \quad (3.10)$$

Derivando  $\Phi$  respecto de las coordenadas independientes de grupo  $\mathbf{h}$ :

$$\Phi_{\mathbf{h}} = \begin{bmatrix} -2 \cdot (x_B - x_A) & -2 \cdot (y_B - y_A) & 0 \\ 0 & -1 & -L1 \cdot \cos \theta \end{bmatrix} \quad (3.11)$$

$$\Phi_{\mathbf{h}} = \begin{bmatrix} -2 \cdot (x_B - x_A) & -2 \cdot (y_B - y_A) & 0 \\ -1 & 0 & L1 \cdot \sin \theta \end{bmatrix} \quad (3.12)$$

Al igual que sucede con las matrices Jacobianas, se tienen, dependiendo del ángulo  $\theta$ , dos matrices  $\Phi_{\mathbf{h}}$ ; cuando el ángulo se aproxime a cero se utilizará (3.11) y cuando el ángulo se aproxime a 90 grados se usará (3.12).

En el caso del grupo  $1RF$  el problema de velocidad varía del planteado.

Como el punto  $A$  pertenece al bastidor resulta que  $\mathbf{h} = [\theta]$  es ahora la coordenada independiente del grupo estructural. Por tanto cambia la matriz  $\Phi_{\mathbf{h}}$ ; en este caso se toman de la matriz de grupo  $1RM$  las columnas correspondientes a  $\mathbf{h}$ , eliminando el resto. También se cambia el vector  $\dot{\mathbf{h}} = [\dot{\theta}]$ .

### 4.- Solución del problema de aceleración

$$\Phi_\varphi \ddot{\varphi} + \dot{\Phi}_{\mathbf{q}_c} \dot{\mathbf{q}}_G = 0 \quad \rightarrow \quad \ddot{\varphi} = -(\Phi_\varphi)^{-1} [\Phi_{\mathbf{h}} \ddot{\mathbf{h}} + \dot{\Phi}_{\mathbf{q}_c} \dot{\mathbf{q}}_G] \quad (3.13)$$

Por último, para resolver dicho problema se necesita  $\dot{\Phi}_{\mathbf{q}_c}$ , derivada temporal de la matriz que resulta de derivar  $\Phi$  respecto a todas las coordenadas del grupo estructural  $\mathbf{q}_G$ , tanto dependientes como independientes. El orden de derivación es:

$$\mathbf{q}_G = [x_B, y_B, x_A, y_A, \theta]$$

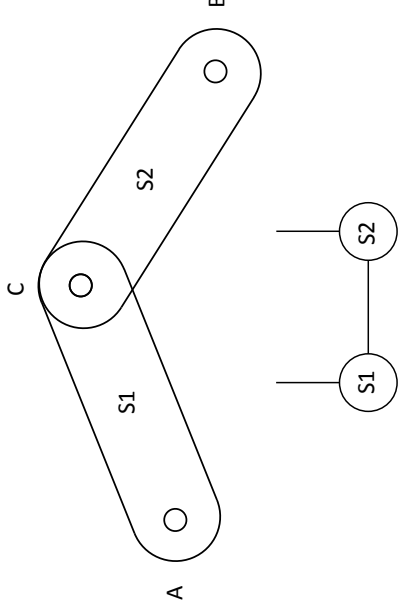


Figura 3.3: Grupo estructural 3R. Esquema cinemático y representación del grafo estructural. El grupo consta de dos eslabones y tres pares de rotación. Los pares externos  $A$  y  $B$  están determinados por enlaces cedidos por eslabones de otros grupos.

Los resultados son también dos matrices, dependiendo la elección de una de ellas del ángulo  $\theta$ . El vector  $\mathbf{q}_G$  es conocido al igual que  $\dot{\mathbf{h}}$ . El resto de matrices requeridas para resolver el algoritmo vienen determinados en apartados anteriores.

$$\dot{\Phi}_{\mathbf{q}_G} = \begin{bmatrix} 2 \cdot (\dot{x}_B - \dot{x}_A) & 2 \cdot (\dot{y}_B - \dot{y}_A) & -2 \cdot (\dot{x}_B - \dot{x}_A) & -2 \cdot (\dot{y}_B - \dot{y}_A) & 0 & 0 \\ 0 & 0 & 0 & 0 & L1 \cdot \sin \theta \cdot \dot{\theta} & 0 \end{bmatrix} \quad (3.14)$$

$$\dot{\Phi}_{\mathbf{q}_G} = \begin{bmatrix} 2 \cdot (\dot{x}_B - \dot{x}_A) & 2 \cdot (\dot{y}_B - \dot{y}_A) & -2 \cdot (\dot{x}_B - \dot{x}_A) & -2 \cdot (\dot{y}_B - \dot{y}_A) & 0 & 0 \\ 0 & 0 & 0 & 0 & L1 \cdot \cos \theta \cdot \dot{\theta} & 0 \end{bmatrix} \quad (3.15)$$

En el caso 1RF el problema de aceleración varía del planteado.

Como el punto  $A$  pertenece al bastidor,  $\mathbf{q}_G = [x_B, y_B, \theta]$  son ahora las coordenadas del grupo estructural. Por tanto cambia la matriz  $\dot{\Phi}_{\mathbf{q}_G}$ ; en este caso se toman de la matriz de grupo 1RM las columnas correspondientes a  $\mathbf{q}_G$ , eliminando el resto. También cambia el vector  $\dot{\mathbf{h}} = [\dot{\theta}]$  y se eliminan las derivadas temporales asociadas al punto  $A$ , ya que es bastidor.

### 3.1.2. Diada 3R

La Figura 3.3 muestra este grupo estructural. Está formado por dos eslabones rígidos  $S1$  y  $S2$ , articulados entre sí por un par interno de rotación. A su vez, ambos eslabones se pueden articular por los extremos con otros sólidos, formando pares externos de rotación. Uno de sus extremos puede estar unido a un sólido fijo (grupo 3RF) o móvil (grupo 3RM); por mayor generalidad se va a resolver en este apartado el grupo 3RM.

#### 1.- Identificación de las coordenadas de grupo y parámetros

Se modeliza el grupo estructural en coordenadas naturales. Para que la posición del grupo quede perfectamente definida, se toman tres puntos básicos  $A$ ,  $B$  y  $C$  en las tres articulaciones. Así cada eslabón se define por dos puntos básicos y quedan definidos también los pares externos e interno de rotación. Entre las coordenadas de grupo se hace la distinción: dependientes  $\varphi$  e independientes  $\mathbf{h}$ .

Al pertenecer los puntos  $A$  y  $B$  a otros sólidos se consideran las coordenadas asociadas a ellos como independientes de grupo, por tanto se conoce también su velocidad y aceleración inicial. El punto  $C$  en cambio tiene coordenadas dependientes de grupo estructural.

En definitiva, se tienen  $\varphi = [x_C, y_C, ]$  coordenadas dependientes de grupo estructural y  $\mathbf{h} = [x_A, y_A, x_B, y_B]$  coordenadas independientes.

## 2.- Solución del problema de posición por grupos estructurales

Se deben introducir ecuaciones de restricción de sólido rígido. No hay ecuaciones de par cinemático ya que la articulaciones se han definido como puntos básicos; la razón es la misma que la mencionada en el grupo 1R. Las ecuaciones de restricción asociadas son:

- Eslabón 1 
$$(x_C - x_A)^2 + (y_C - y_A)^2 - L_1^2 = 0 \quad (3.16)$$

- Eslabón 2 
$$(x_C - x_B)^2 + (y_C - y_B)^2 - L_2^2 = 0 \quad (3.17)$$

Con ellas se forma el vector  $\Phi$ :

$$\Phi = \begin{bmatrix} (x_C - x_A)^2 + (y_C - y_A)^2 - L_1^2 \\ (x_C - x_B)^2 + (y_C - y_B)^2 - L_2^2 \end{bmatrix} \quad (3.18)$$

Para resolver el problema de posición, además se requiere de la matriz Jacobiana  $\Phi_\varphi$  (3.19). Mediante el método iterativo de Newton-Raphson, ecuación (3.7), se obtienen los valores de las coordenadas dependientes de grupo.

$$\Phi_\varphi = \begin{bmatrix} 2 \cdot (x_C - x_A) & 2 \cdot (y_C - y_A) \\ 2 \cdot (x_C - x_B) & 2 \cdot (y_C - y_B) \end{bmatrix} \quad (3.19)$$

En el caso del grupo 3RF, diada unida por un extremo al bastidor, el problema de posición coincide con el de 3RM.

## 3.- Solución del problema de velocidad

Para resolver este problema se utiliza la ecuación (3.10). Necesita la matriz  $\Phi_{\mathbf{h}}$  ya que el resto de matrices y vectores son conocidos.

Derivando  $\Phi$  respecto de las coordenadas independientes  $\mathbf{h}$ :

$$\Phi_{\mathbf{h}} = \begin{bmatrix} -2 \cdot (x_C - x_A) & -2 \cdot (y_C - y_A) & 0 & 0 \\ 0 & 0 & -2 \cdot (x_C - x_B) & -2 \cdot (y_C - y_B) \end{bmatrix} \quad (3.20)$$

En el caso del grupo 3RF el problema de velocidad varía del planteado.

Como el punto B pertenece al bastidor resulta que  $\mathbf{h} = [x_A, y_A]$  es ahora la coordenada independiente del grupo estructural. Por tanto cambia la matriz  $\Phi_{\mathbf{h}}$ ; en este caso se toman de la matriz de grupo 3RM las columnas correspondientes a  $\mathbf{h}$ , eliminando el resto. También cambia el vector  $\mathbf{h} = [\dot{x}_A, \dot{y}_A]$ .

#### 4.- Solución del problema de aceleración

Para resolver dicho problema se utiliza la ecuación (3.13), que requiere del cálculo de  $\dot{\Phi}_{\mathbf{q}_G}$ , derivada temporal de la matriz que resulta de derivar  $\Phi$  respecto a todas las coordenadas del grupo estructural  $\mathbf{q}_G$ , tanto dependientes como independientes, donde el orden de integración es:

$$\mathbf{q}_G = [x_C, y_C, x_A, y_A, x_B, y_B].$$

Las matrices y vectores restantes necesarios para resolver el algoritmo son conocidos de apartados anteriores.

$$\dot{\Phi}_{\mathbf{q}_G} = \begin{bmatrix} 2 \cdot (\dot{x}_C - \dot{x}_A) & 2 \cdot (\dot{y}_C - \dot{y}_A) & -2 \cdot (\dot{x}_C - \dot{x}_A) & -2 \cdot (\dot{y}_C - \dot{y}_A) & 0 & 0 \\ 2 \cdot (\dot{x}_C - \dot{x}_B) & 2 \cdot (\dot{y}_C - \dot{y}_B) & 0 & 0 & -2 \cdot (\dot{x}_C - \dot{x}_B) & -2 \cdot (\dot{y}_C - \dot{y}_B) \end{bmatrix} \quad (3.21)$$

En el caso del grupo *3RF* el problema de aceleración varía del planteado.

Como el punto *B* pertenece al bastidor resulta que  $\mathbf{q}_G = [x_C, y_C, x_A, y_A]$  son ahora las coordenadas del grupo estructural. Por tanto cambia la matriz  $\dot{\Phi}_{\mathbf{q}_G}$ ; en este caso de la matriz de grupo *3RM* se toman las columnas correspondientes a  $\mathbf{q}_G$ , eliminando el resto. También cambia el vector  $\ddot{\mathbf{h}} = [\ddot{x}_A, \ddot{y}_A]$  y se eliminan las derivadas temporales asociadas al punto *A*, ya que es bastidor.

### 3.1.3. Grupo perteneciente a sistema de suspensión 2C3E1R

La Figura 3.4 muestra el grupo estructural *2C3E1R*. Es un grupo complejo, formado por cuatro eslabones rígidos, *S2*, *S3*, *S4* y *S5*, conectados entre sí por pares internos. Al eslabón principal *S2* se unen los demás; *S4* y *S5* se conectan a él mediante pares cardan y *S3* a través de un par esférico. A su vez, el grupo estructural está conectado a otro sólido *S1*, utilizando dos pares externos esféricos y uno de revolución.

Cabe la posibilidad de añadir otros sólidos a este grupo estructural, como se verá en el caso de estudio del sistema de suspensión de un camión, mediante un par de revolución y dos pares cardan, dada la geometría del eslabón *S2*. En este caso se calcularán los otros POI's del eslabón *S2* por medio de la formulación expuesta en el Capítulo 2, apartado 2.2.1.

El grupo estructural tiene movilidad 2, por tanto tendrá dos coordenadas de entrada consideradas conocidas,  $z_8$  y  $z_9$ , ya que corresponden al desplazamiento vertical de cada extremo ( $P_8$  y  $P_9$ ) del eje. También se suponen conocidas las derivadas temporales de estas coordenadas,  $\dot{z}_8$ ,  $\dot{z}_9$ ,  $\ddot{z}_8$  y  $\ddot{z}_9$ .

#### 1.- Identificación de las coordenadas de grupo y parámetros

Se modeliza el grupo estructural en coordenadas naturales. Para que la posición del grupo quede perfectamente determinada se define cada eslabón del grupo adecuadamente. Se usan como mínimo 12 coordenadas por eslabón; tres vectores y un punto, dos vectores y dos puntos, etc. En la Tabla 3.1 viene detallada esta definición.

En las coordenadas de grupo se hace la distinción entre: dependientes  $\varphi$  e independientes  $\mathbf{h}$ . Como los puntos  $P_{11}$ ,  $P_{12}$ ,  $P_{10}$ , y el vector  $V_1$  pertenecen a otros sólidos, en concreto a *S1*, se consideran las coordenadas asociadas a estos puntos y vector como independientes de grupo. Por tanto, los puntos  $P_1$ ,  $P_2$ ,  $P_3$ ,  $P_8$ ,  $P_9$  y los vectores  $V_2$ ,  $V_3$ ,  $V_4$ ,  $V_5$ ,  $V_6$ ,  $V_7$ ,  $V_8$  y  $V_9$  definen el conjunto de coordenadas dependientes de grupo estructural. Los puntos  $P_4$ ,  $P_5$ ,  $P_6$  y  $P_7$  podrían ser otros POI's necesarios para los casos de adición de otros grupos.

En definitiva, se tiene:

- Coordenadas dependientes de grupo estructural:  $\varphi = [x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_8, y_8, x_9, y_9, v_{2x}, v_{2y}, v_{2z}, v_{3x}, v_{3y}, v_{3z}, v_{4x}, v_{4y}, v_{4z}, v_{5x}, v_{5y}, v_{5z}, v_{6x}, v_{6y}, v_{6z}, v_{7x}, v_{7y}, v_{7z}, v_{8x}, v_{8y}, v_{8z}, v_{9x}, v_{9y}, v_{9z}]$

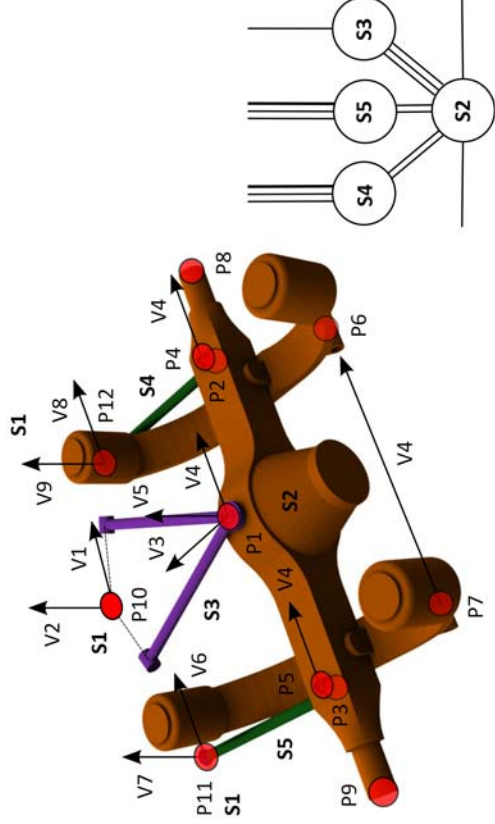


Figura 3.4: Grupo estructural  $2C3E1R$ . Esquema cinemático y representación del grafo estructural. Este grupo consta de cuatro eslabones rígidos  $S2$ ,  $S3$ ,  $S4$  y  $S5$ , conectados entre sí por pares esféricos. A su vez, se une por tres extremos a otro sólido  $S1$ , con dos pares esféricos y un par de revolución. Tiene dos movimientos de entrada y por su geometría la capacidad de unirse con otros sólidos.

SÓLIDO RÍGIDO						
ID	SÓLIDO	PUNTOS	VECTORES	COORDS.	ECS. S.R.	COORDS. LIBRES
1	2	$P1, P2, P3, P8, P9$	$V3, V4, V5$	$3 \times 5 = 15$	$6(\text{triédro}) + 6(\text{c.l.})$	$15 - 12 = 3$
3	3	$P1, P10$	$V1, V2$	$3 \times 2 = 6$	$5(\text{triédro})$	$6 - 5 = 1$
4	4	$P2, P12$	$V8, V9$	$3 \times 3 = 9$	$6(\text{triédro}) + 3(\text{c.l.})$	$9 - 9 = 0$
5	5	$P3, P11$	$V6, V7$	$3 \times 3 = 9$	$6(\text{triédro}) + 3(\text{c.l.})$	$9 - 9 = 0$
6						
PAR CINEMÁTICO						
8	PAR	TIPO	ECS. de par	ECS. nuevas	OBSERVACIONES	
9	1-3	R(Revolución)	5	0	Por tener puntos y vector en eje de rotación.	
10	1-4	S(Esférico)	3	0	Porque comparten P12.	
11	1-5	S(Esférico)	3	0	Porque comparten P11.	
12	2-3	S(Esférico)	3	0	Porque comparten P1.	
13	2-4	CD(Cardan)	4	1		
14	2-5	CD(Cardan)	4	1		
15			Los pares cinemáticos añaden dos ecuaciones adicionales de par.			

Cuadro 3.1: Principales características del grupo estructural  $2C3E1R$ . Coordenadas por eslabón, ecuaciones de restricción de sólido rígido y de par. Se marcan en cursiva las coordenadas compartidas.

- Coordenadas independientes de grupo estructural:

$$\mathbf{h} = [z_8, z_9, x_{10}, y_{10}, z_{10}, x_{11}, y_{11}, z_{11}, x_{12}, y_{12}, z_{12}, v_{1x}, v_{1y}, v_{1z}]$$

## 2.- Solución del problema de posición por grupos estructurales

El número de ecuaciones de restricción  $r_i$ , asociadas al caso espacial para un grupo estructural, se obtiene a partir del número de coordenadas para definirlo  $n_i$  por de la siguiente expresión:



$$r_i = n_i - 6 \quad (3.22)$$

En la Tabla 3.1 se determina, para cada eslabón, el número de ecuaciones de restricción asociadas, tanto de restricción de sólido rígido como de par cinemático. A continuación se muestran las ecuaciones de restricción.

Ecuaciones de restricción de par cinemático:

- Cardan
 
$$\begin{aligned} v_{4x} \cdot v_{7x} + v_{4y} \cdot v_{7y} + v_{4z} \cdot v_{7z} &= 0 \\ v_{4x} \cdot v_{9x} + v_{4y} \cdot v_{9y} + v_{4z} \cdot v_{9z} &= 0 \end{aligned} \quad (3.23)$$

Ecuaciones de restricción de sólido rígido:

- Eslabón 2
 
$$\begin{aligned} v_{3x}^2 + v_{3y}^2 + v_{3z}^2 - 1 &= 0 \\ v_{4x}^2 + v_{4y}^2 + v_{4z}^2 - 1 &= 0 \\ v_{5x}^2 + v_{5y}^2 + v_{5z}^2 - 1 &= 0 \\ v_{3x} \cdot v_{4x} + v_{3y} \cdot v_{4y} + v_{3z} \cdot v_{4z} &= 0 \\ v_{3x} \cdot v_{5x} + v_{3y} \cdot v_{5y} + v_{3z} \cdot v_{5z} &= 0 \\ v_{4x} \cdot v_{5x} + v_{4y} \cdot v_{5y} + v_{4z} \cdot v_{5z} &= 0 \end{aligned} \quad (3.24)$$

- Eslabón 3

$$\begin{aligned} (x_1 - x_{10})^2 + (y_1 - y_{10})^2 + (z_1 - z_{10})^2 - L_{11-10}^2 &= 0 \\ v_{2x}^2 + v_{2y}^2 + v_{2z}^2 - 1 &= 0 \\ v_{1x} \cdot v_{2x} + v_{1y} \cdot v_{2y} + v_{1z} \cdot v_{2z} &= 0 \\ (x_1 - x_{10}) \cdot v_{1x} + (y_1 - y_{10}) \cdot v_{1y} + (z_1 - z_{10}) \cdot v_{1z} &= 0 \\ (x_1 - x_{10}) \cdot v_{2x} + (y_1 - y_{10}) \cdot v_{2y} + (z_1 - z_{10}) \cdot v_{2z} &= 0 \end{aligned} \quad (3.25)$$

- Eslabón 4

$$\begin{aligned} (x_2 - x_{12})^2 + (y_2 - y_{12})^2 + (z_2 - z_{12})^2 - L_{2-12}^2 &= 0 \\ v_{8x}^2 + v_{8y}^2 + v_{8z}^2 - 1 &= 0 \\ v_{9x}^2 + v_{9y}^2 + v_{9z}^2 - 1 &= 0 \\ (x_2 - x_{12}) \cdot v_{8x} + (y_2 - y_{12}) \cdot v_{8y} + (z_2 - z_{12}) \cdot v_{8z} &= 0 \\ (x_2 - x_{12}) \cdot v_{9x} + (y_2 - y_{12}) \cdot v_{9y} + (z_2 - z_{12}) \cdot v_{9z} &= 0 \\ v_{8x} \cdot v_{9x} + v_{8y} \cdot v_{9y} + v_{8z} \cdot v_{9z} &= 0 \end{aligned} \quad (3.26)$$

- Eslabón 5

$$\begin{aligned} (x_3 - x_{11})^2 + (y_3 - y_{11})^2 + (z_3 - z_{11})^2 - L_{3-11}^2 &= 0 \\ v_{6x}^2 + v_{6y}^2 + v_{6z}^2 - 1 &= 0 \\ v_{7x}^2 + v_{7y}^2 + v_{7z}^2 - 1 &= 0 \\ v_{6x} \cdot v_{7x} + v_{6y} \cdot v_{7y} + v_{6z} \cdot v_{7z} &= 0 \\ (x_3 - x_{11}) \cdot v_{6x} + (y_3 - y_{11}) \cdot v_{6y} + (z_3 - z_{11}) \cdot v_{6z} &= 0 \\ (x_3 - x_{11}) \cdot v_{7x} + (y_3 - y_{11}) \cdot v_{7y} + (z_3 - z_{11}) \cdot v_{7z} &= 0 \end{aligned} \quad (3.27)$$

- Combinación lineal en el eslabón 2

$$\begin{aligned}
 (x_2 - x_1) - 0,758 \cdot v_{4x} - 0,131 \cdot v_{3x} + 0,265 \cdot v_{5x} &= 0 \\
 (y_2 - y_1) - 0,758 \cdot v_{4y} - 0,131 \cdot v_{3y} + 0,265 \cdot v_{5y} &= 0 \\
 (z_2 - z_1) - 0,758 \cdot v_{4z} - 0,131 \cdot v_{3z} + 0,265 \cdot v_{5z} &= 0 \\
 (x_3 - x_1) + 0,758 \cdot v_{4x} - 0,131 \cdot v_{3x} + 0,265 \cdot v_{5x} &= 0 \\
 (y_3 - y_1) + 0,758 \cdot v_{4y} - 0,131 \cdot v_{3y} + 0,265 \cdot v_{5y} &= 0 \\
 (z_3 - z_1) + 0,758 \cdot v_{4z} - 0,131 \cdot v_{3z} + 0,265 \cdot v_{5z} &= 0 \\
 (x_8 - x_1) - 1,202 \cdot v_{4x} + 0,265 \cdot v_{5x} &= 0 \\
 (y_8 - y_1) - 1,202 \cdot v_{4y} + 0,265 \cdot v_{5y} &= 0 \\
 (z_8 - z_1) - 1,202 \cdot v_{4z} + 0,265 \cdot v_{5z} &= 0 \\
 (x_9 - x_1) + 1,202 \cdot v_{4x} + 0,265 \cdot v_{5x} &= 0 \\
 (y_9 - y_1) + 1,202 \cdot v_{4y} + 0,265 \cdot v_{5y} &= 0 \\
 (z_9 - z_1) + 1,202 \cdot v_{4z} + 0,265 \cdot v_{5z} &= 0
 \end{aligned} \tag{3.28}$$

A diferencia de lo visto hasta ahora, en este grupo estructural los eslabones aparecen definidos por un triedro y un punto o por dos puntos y dos vectores. En el primer caso, eslabón 2 (3.24), en las ecuaciones de restricción se determinan las direcciones del triedro, multiplicando componentes dos a dos, y se determina el módulo de los vectores. En el segundo caso, eslabón 3 (3.25), 4 (3.26) y 5 (3.27), uno de los vectores que define un vector del triedro se obtiene por la resta de los puntos. Como excepción el eslabón 3 tiene una ecuación de restricción menos, debido a que uno de los vectores que definen su triedro pertenece a otro sólido y por tanto su módulo está ya resuelto.

En el caso de que se quiera definir un eslabón con más puntos básicos de los mínimos, como es el caso del eslabón 2, habrá que utilizar las ecuaciones de combinación lineal (3.28), que determinan cada punto adicional a partir de combinación lineal del triedro local de dicho eslabón. Las constantes que aparecen en las ecuaciones se determinan por la geometría del eslabón, resolviendo el sistema de ecuaciones donde el centro del triedro se establece origen local.

Respecto a las ecuaciones de par sólo se dispone de las de cardan (3.23); al ser de clase II restringe cuatro grados de libertad, al elegirse como punto básico la unión de ambos eslabones se restringen tres movimientos relativos y por tanto sólo necesita de una ecuación de restricción por par cardan. Para el par esférico, clase III, se restringen tres grados de libertad; como se ha elegido como punto básico el punto de unión se cumplen automáticamente las tres restricciones. Para el par de revolución, clase I, se restringen cinco grados de libertad; como se ha elegido un punto y vector pertenecientes a la dirección de la rotación se cumplen automáticamente las cinco restricciones.

Con las ecuaciones de restricción se forma el vector  $\Phi$ :

$$\begin{aligned}
& \left[ \begin{array}{l}
v_{4x} \cdot v_{7x} + v_{4y} \cdot v_{7y} + v_{4z} \cdot v_{7z} \\
v_{4x} \cdot v_{9x} + v_{4y} \cdot v_{9y} + v_{4z} \cdot v_{9z} \\
v_{3x}^2 + v_{3y}^2 + v_{3z}^2 - 1 \\
v_{4x}^2 + v_{4y}^2 + v_{4z}^2 - 1 \\
v_{5x}^2 + v_{5y}^2 + v_{5z}^2 - 1 \\
v_{3x} \cdot v_{4x} + v_{3y} \cdot v_{4y} + v_{3z} \cdot v_{4z} \\
v_{3x} \cdot v_{5x} + v_{3y} \cdot v_{5y} + v_{3z} \cdot v_{5z} \\
v_{4x} \cdot v_{5x} + v_{4y} \cdot v_{5y} + v_{4z} \cdot v_{5z} \\
(x_1 - x_{10})^2 + (y_1 - y_{10})^2 + (z_1 - z_{10})^2 - L_{1-10}^2 \\
v_{2x}^2 + v_{2y}^2 + v_{2z}^2 - 1 \\
v_{1x} \cdot v_{2x} + v_{1y} \cdot v_{2y} + v_{1z} \cdot v_{2z} \\
(x_1 - x_{10}) \cdot v_{1x} + (y_1 - y_{10}) \cdot v_{1y} + (z_1 - z_{10}) \cdot v_{1z} \\
(x_1 - x_{10}) \cdot v_{2x} + (y_1 - y_{10}) \cdot v_{2y} + (z_1 - z_{10}) \cdot v_{2z} \\
(x_2 - x_{12})^2 + (y_2 - y_{12})^2 + (z_2 - z_{12})^2 - L_{2-12}^2 \\
v_{8x}^2 + v_{8y}^2 + v_{8z}^2 - 1 \\
v_{9x}^2 + v_{9y}^2 + v_{9z}^2 - 1 \\
(x_2 - x_{12}) \cdot v_{8x} + (y_2 - y_{12}) \cdot v_{8y} + (z_2 - z_{12}) \cdot v_{8z} \\
(x_2 - x_{12}) \cdot v_{9x} + (y_2 - y_{12}) \cdot v_{9y} + (z_2 - z_{12}) \cdot v_{9z} \\
v_{8x} \cdot v_{9x} + v_{8y} \cdot v_{9y} + v_{8z} \cdot v_{9z} \\
(x_3 - x_{11})^2 + (y_3 - y_{11})^2 + (z_3 - z_{11})^2 - L_{3-11}^2 \\
v_{6x}^2 + v_{6y}^2 + v_{6z}^2 - 1 \\
v_{7x}^2 + v_{7y}^2 + v_{7z}^2 - 1 \\
v_{6x} \cdot v_{7x} + v_{6y} \cdot v_{7y} + v_{6z} \cdot v_{7z} \\
(x_3 - x_{11}) \cdot v_{6x} + (y_3 - y_{11}) \cdot v_{6y} + (z_3 - z_{11}) \cdot v_{6z} \\
(x_3 - x_{11}) \cdot v_{7x} + (y_3 - y_{11}) \cdot v_{7y} + (z_3 - z_{11}) \cdot v_{7z} \\
(x_2 - x_1) - 0,758 \cdot v_{4x} - 0,131 \cdot v_{3x} + 0,265 \cdot v_{5x} \\
(y_2 - y_1) - 0,758 \cdot v_{4y} - 0,131 \cdot v_{3y} + 0,265 \cdot v_{5y} \\
(z_2 - z_1) - 0,758 \cdot v_{4z} - 0,131 \cdot v_{3z} + 0,265 \cdot v_{5z} \\
(x_3 - x_1) + 0,758 \cdot v_{4x} - 0,131 \cdot v_{3x} + 0,265 \cdot v_{5x} \\
(y_3 - y_1) + 0,758 \cdot v_{4y} - 0,131 \cdot v_{3y} + 0,265 \cdot v_{5y} \\
(z_3 - z_1) + 0,758 \cdot v_{4z} - 0,131 \cdot v_{3z} + 0,265 \cdot v_{5z} \\
(x_8 - x_1) - 1,202 \cdot v_{4x} + 0,265 \cdot v_{5x} \\
(y_8 - y_1) - 1,202 \cdot v_{4y} + 0,265 \cdot v_{5y} \\
(z_8 - z_1) - 1,202 \cdot v_{4z} + 0,265 \cdot v_{5z} \\
(x_9 - x_1) + 1,202 \cdot v_{4x} + 0,265 \cdot v_{5x} \\
(y_9 - y_1) + 1,202 \cdot v_{4y} + 0,265 \cdot v_{5y} \\
(z_9 - z_1) + 1,202 \cdot v_{4z} + 0,265 \cdot v_{5z}
\end{array} \right]
\end{aligned}$$

$$\Phi =$$

(3.29)

Para resolver el problema de posición se precisa de una posición inicial de todas las coordenadas dependientes, del vector  $\Phi$  y de la matriz Jacobiana  $\Phi_\varphi$ . Esta última puede ser analíticamente obtenida, por el método iterativo de Newton-Raphson se obtienen los valores de las coordenadas dependientes de grupo (3.7).

La Jacobiana de  $\Phi$ , derivando respecto a las variables dependientes de grupo  $\varphi$ , aparece en el Anexo de matrices de este capítulo. Será la matriz  $\Phi_\varphi$  dividida por columnas en tres bloques de referencias(6.1, 6.2 y 6.3).

### 3.- Solución del problema de velocidad

Para resolver este problema se utiliza la ecuación (3.10). Se necesita la matriz  $\Phi_{\mathbf{h}}$ , ya que  $\Phi_\varphi$  se ha obtenido en el paso anterior. También se necesita  $\dot{\mathbf{h}}$ , conocido. Derivando  $\Phi$  respecto de las coordenadas independientes  $\mathbf{h}$  se obtiene  $\Phi_{\mathbf{h}}$ , matriz (6.4) del Anexo al capítulo.

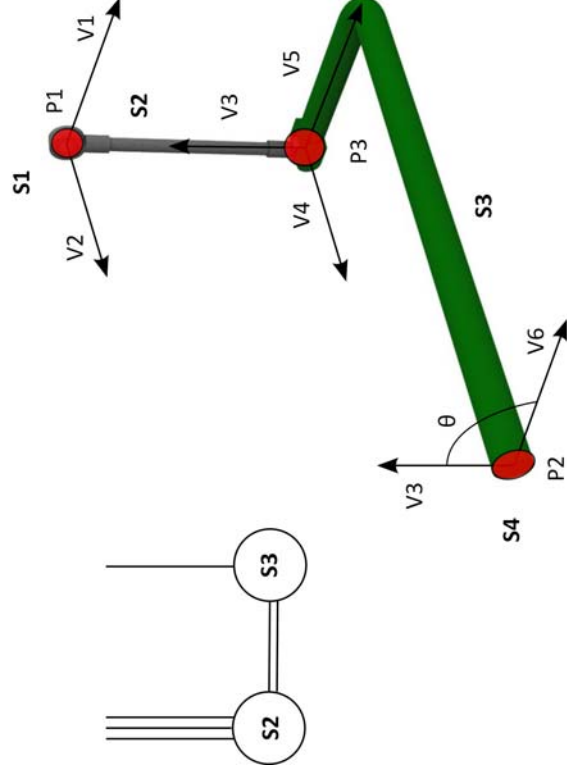


Figura 3.5: Grupo estructural 1C1E1R. Esquema cinemático y representación del grafo estructural. Este grupo consta de dos eslabones rígidos  $S2$  y  $S3$ , articulados entre sí por un cardán. A su vez, conectados con dos sólidos por dos pares extremos; esférico y de revolución.

#### 4.- Solución del problema de aceleración

Para resolver dicho problema se utiliza la ecuación (3.13), que requiere del cálculo de  $\dot{\Phi}_{\mathbf{q}_G}$ , que será la derivada temporal de la matriz que resulta de derivar  $\Phi$  respecto a todas las coordenadas del grupo estructural  $\mathbf{q}_G$ , tanto dependientes como independientes. El orden de derivación para las coordenadas de grupo es:

$$\mathbf{q}_G = [x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_8, y_8, z_8, x_9, y_9, z_9, x_{10}, y_{10}, z_{10}, x_{11}, y_{11}, z_{11}, x_{12}, y_{12}, z_{12}, v_{1x}, v_{1y}, v_{1z}, v_{2x}, v_{2y}, v_{2z}, v_{3x}, v_{3y}, v_{3z}, v_{4x}, v_{4y}, v_{4z}, v_{5x}, v_{5y}, v_{5z}, v_{6x}, v_{6y}, v_{6z}, v_{7x}, v_{7y}, v_{7z}, v_{8x}, v_{8y}, v_{8z}, v_{9x}, v_{9y}, v_{9z}].$$

Las matrices y vectores  $\Phi_{\mathbf{h}}$ ,  $\Phi_{\varphi}$ ,  $\ddot{\mathbf{h}}$  y  $\mathbf{q}_G$  son conocidos de apartados anteriores y de las correspondientes condiciones de los sólidos externos de unión.

La matriz  $\dot{\Phi}_{\mathbf{q}_G}$  se muestra en el Anexo. Será la matriz dividida por columnas en cuatro bloques con referencias (6.5, 6.6, 6.7 y 6.8).

### 3.1.4. Grupo perteneciente a sistema de suspensión 1C1E1R

La Figura 3.5 muestra el grupo estructural 1C1E1R. Es un grupo menos complejo que el anterior, formado por dos eslabones rígidos  $S2$  y  $S3$ , articulados entre ellos formando un par interno cardán. A su vez, este grupo estructural está conectado a dos sólidos distintos,  $S1$  y  $S4$ , por dos pares externos; un par esférico y de revolución respectivamente.

#### 1.- Identificación de las coordenadas de grupo y parámetros

Se modeliza el grupo estructural en coordenadas naturales. Para que la posición del grupo quede perfectamente determinada se define cada eslabón del grupo adecuadamente. Se usa como mínimo 12 coordenadas por eslabón; tres vectores y un punto, dos vectores y dos puntos, etc. En la Tabla 3.2 viene detallada esta definición.

Entre las coordenadas de grupo se hace la distinción de: dependientes  $\varphi$  e independientes  $\mathbf{h}$ . Como los puntos  $P_1$ ,  $P_2$  y los vectores  $V_4$  y  $V_6$  pertenecen a otros sólidos, se consideran las coordenadas asociadas a ellos como independientes de grupo. Por tanto, el punto  $P_3$  y los vectores  $V_1$ ,  $V_2$ ,  $V_3$  y  $V_5$  definen el conjunto de coordenadas dependientes de grupo estructural. Además, es importante introducir en este conjunto la variable  $\theta$ , que define el movimiento relativo de  $S_3$  con el sólido al que está unido,  $S_4$ .

SÓLIDO RÍGIDO					
ID	SÓLIDO	PUNTOS	VECTORES	COORDS.	COORDS. LIBRES
1	2	$P_1, P_3$	$V_1, V_2$	$3 \times 3 = 9$	6(triedro)
2	3	$P_2, P_3$	$V_3, V_4, V_5$	$3 \times 2 = 6$	$5(\text{triedro}) + 3(\text{c.l})$
3	4	Queda una coordenada libre. Falta ecuación de par.			
PAR CINEMÁTICO					
PAR	TIPO	ECS. de par	ECS. nuevas	OBSERVACIONES	
1-2	S(Esférico)	3	0	Porque comparten P1.	
2-3	CD(Cardan)	4	1		
3-4	R(Revolución)	5	0	Por tener punto y vector en eje de rotación.	
10	Los pares cinemáticos añaden una ecuación adicional de par.				

Cuadro 3.2: Principales características del grupo estructural  $1C1E1R$ . Coordenadas por eslabón, ecuaciones de restricción de sólido rígido y de par. Se marcan en cursiva las coordenadas compartidas.

En definitiva, se tienen:

- Coordenadas dependientes de grupo estructural:

$$\varphi = [x_3, y_3, z_3, v_{1y}, v_{1z}, v_{2x}, v_{2y}, v_{2z}, v_{3x}, v_{3y}, v_{3z}, v_{5x}, v_{5y}, v_{5z}, \theta]$$

- Coordenadas independientes de grupo estructural:

$$\mathbf{h} = [x_1, y_1, z_1, x_2, y_2, z_2, v_{4x}, v_{4y}, v_{4z}]$$

## 2.- Solución del problema de posición por grupos estructurales

En la Tabla 3.2 se determinan para cada eslabón el número de ecuaciones de restricción asociadas, tanto de restricción de sólido rígido como de par cinemático. A continuación se muestran las ecuaciones de restricción.

Ecuaciones de restricción de sólido rígido:

▪ Eslabón 2	$(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 - L_{13}^2 = 0$ $v_{1x}^2 + v_{1y}^2 + v_{1z}^2 - 1 = 0$ $v_{2x}^2 + v_{2y}^2 + v_{2z}^2 - 1 = 0$ $v_{1x} \cdot (x_3 - x_1) + v_{1y} \cdot (y_3 - y_1) + v_{1z} \cdot (z_3 - z_1) = 0$ $v_{2x} \cdot (x_3 - x_1) + v_{2y} \cdot (y_3 - y_1) + v_{2z} \cdot (z_3 - z_1) = 0$ $v_{1x} \cdot v_{2x} + v_{1y} \cdot v_{2y} + v_{1z} \cdot v_{2z} = 0$	(3.30)
▪ Eslabón 3	$v_{4x} \cdot v_{3x} + v_{4y} \cdot v_{3y} + v_{4z} \cdot v_{3z} = 0$ $v_{4x} \cdot v_{5x} + v_{4y} \cdot v_{5y} + v_{4z} \cdot v_{5z} = 0$ $v_{5x} \cdot v_{3x} + v_{5y} \cdot v_{3y} + v_{5z} \cdot v_{3z} = 0$ $v_{5x}^2 + v_{5y}^2 + v_{5z}^2 - 1 = 0$ $v_{3x}^2 + v_{3y}^2 + v_{3z}^2 - 1 = 0$	(3.31)

- Combinación lineal en el eslabón 3

$$\begin{aligned} (x_2 - x_3) - 0,199 \cdot v_{4x} - 0,3606 \cdot v_{5x} - 0,00003 \cdot v_{3x} &= 0 \\ (y_2 - y_3) - 0,199 \cdot v_{4y} - 0,3606 \cdot v_{5y} - 0,00003 \cdot v_{3y} &= 0 \\ (z_2 - z_3) - 0,199 \cdot v_{4z} - 0,3606 \cdot v_{5z} - 0,00003 \cdot v_{3z} &= 0 \end{aligned} \quad (3.32)$$

Ecuaciones de restricción de par cinemático:

- Cardan

$$v_{1x} \cdot v_{4x} + v_{1y} \cdot v_{4y} + v_{1z} \cdot v_{4z} = 0 \quad (3.33)$$

Ecuación de restricción mixta:

- Ángulo relativo 3-4

$$v_{3x} \cdot v_{6x} + v_{3y} \cdot v_{6y} + v_{3z} \cdot v_{6z} - \cos \theta = 0 \quad (3.34)$$

$$v_{3y} \cdot v_{6x} - v_{3z} \cdot v_{6y} - v_{4x} \cdot \sin \theta = 0 \quad (3.35)$$

Con ellas se formarán los vectores ecuaciones de restricción  $\Phi$ :

$$\Phi = \begin{bmatrix} (x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 - L_{13}^2 \\ v_{1x}^2 + v_{1y}^2 + v_{1z}^2 - 1 \\ v_{2x}^2 + v_{2y}^2 + v_{2z}^2 - 1 \\ v_{1x} \cdot (x_3 - x_1) + v_{1y} \cdot (y_3 - y_1) + v_{1z} \cdot (z_3 - z_1) \\ v_{2x} \cdot (x_3 - x_1) + v_{2y} \cdot (y_3 - y_1) + v_{2z} \cdot (z_3 - z_1) \\ v_{1x} \cdot v_{2x} + v_{1y} \cdot v_{2y} + v_{1z} \cdot v_{2z} \\ v_{4x} \cdot v_{3x} + v_{4y} \cdot v_{3y} + v_{4z} \cdot v_{3z} \\ v_{4x} \cdot v_{5x} + v_{4y} \cdot v_{5y} + v_{4z} \cdot v_{5z} \\ v_{5x} \cdot v_{3x} + v_{5y} \cdot v_{3y} + v_{5z} \cdot v_{3z} \\ v_{5x}^2 + v_{5y}^2 + v_{5z}^2 - 1 \\ v_{3x}^2 + v_{3y}^2 + v_{3z}^2 - 1 \\ (x_2 - x_3) - 0,199 \cdot v_{4x} - 0,3606 \cdot v_{5x} - 0,00003 \cdot v_{3x} \\ (y_2 - y_3) - 0,199 \cdot v_{4y} - 0,3606 \cdot v_{5y} - 0,00003 \cdot v_{3y} \\ (z_2 - z_3) - 0,199 \cdot v_{4z} - 0,3606 \cdot v_{5z} - 0,00003 \cdot v_{3z} \\ v_{1x} \cdot v_{4x} + v_{1y} \cdot v_{4y} + v_{1z} \cdot v_{4z} \\ v_{3x} \cdot v_{6x} + v_{3y} \cdot v_{6y} + v_{3z} \cdot v_{6z} - \cos \theta \end{bmatrix} \quad (3.36)$$

$$\Phi = \begin{bmatrix} (x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 - L_{13}^2 \\ v_{1x}^2 + v_{1y}^2 + v_{1z}^2 - 1 \\ v_{2x}^2 + v_{2y}^2 + v_{2z}^2 - 1 \\ v_{1x} \cdot (x_3 - x_1) + v_{1y} \cdot (y_3 - y_1) + v_{1z} \cdot (z_3 - z_1) \\ v_{2x} \cdot (x_3 - x_1) + v_{2y} \cdot (y_3 - y_1) + v_{2z} \cdot (z_3 - z_1) \\ v_{1x} \cdot v_{2x} + v_{1y} \cdot v_{2y} + v_{1z} \cdot v_{2z} \\ v_{4x} \cdot v_{3x} + v_{4y} \cdot v_{3y} + v_{4z} \cdot v_{3z} \\ v_{4x} \cdot v_{5x} + v_{4y} \cdot v_{5y} + v_{4z} \cdot v_{5z} \\ v_{5x} \cdot v_{3x} + v_{5y} \cdot v_{3y} + v_{5z} \cdot v_{3z} \\ v_{5x}^2 + v_{5y}^2 + v_{5z}^2 - 1 \\ v_{3x}^2 + v_{3y}^2 + v_{3z}^2 - 1 \\ (x_2 - x_3) - 0,199 \cdot v_{4x} - 0,3606 \cdot v_{5x} - 0,00003 \cdot v_{3x} = 0 \\ (y_2 - y_3) - 0,199 \cdot v_{4y} - 0,3606 \cdot v_{5y} - 0,00003 \cdot v_{3y} = 0 \\ (z_2 - z_3) - 0,199 \cdot v_{4z} - 0,3606 \cdot v_{5z} - 0,00003 \cdot v_{3z} = 0 \\ v_{1x} \cdot v_{4x} + v_{1y} \cdot v_{4y} + v_{1z} \cdot v_{4z} \\ v_{3y} \cdot v_{6x} - v_{3z} \cdot v_{6y} - v_{4x} \cdot \sin \theta \end{bmatrix} \quad (3.37)$$

Aparecen ecuaciones de sólido rígido (3.30 o 3.30), de cardan (3.33) y de combinación lineal (3.32), todas ya comentadas en el grupo anterior 3.1.3. No aparecen más ecuaciones de par por la elección de los puntos básicos, razones presentadas también en el grupo estructural anterior.

En este caso por interés se ha introducido  $\theta$  en una ecuación de restricción mixta, ángulo relativo entre el vector  $V_3$ , perteneciente a  $S_3$ , y  $V_6$ , fijo y solidario a  $S_2$ . Cuando el valor del ángulo relativo a 3-4 se aproxima a cero, la ecuación (3.34) se hace no válida, lo mismo ocurre con (3.35) cuando el ángulo se aproxima a 90 grados. Se utilizará una u otra dependiendo del valor de  $\theta$ . Lo mismo sucede con  $\Phi$ .

Para resolver el problema de posición, además se requiere de la matriz Jacobiana  $\Phi_\varphi$  que se obtiene a continuación. Por el método iterativo de Newton-Raphson, ecuación (3.7), se obtienen los valores de las coordenadas dependientes de grupo.

Las Jacobianas de  $\Phi$ , derivando respecto a las variables dependientes de grupo  $\varphi$ , aparecen en el Anexo como (6.9) y (6.10).

### 3.- Solución del problema de velocidad

Para resolver este problema se utiliza la ecuación (3.10). Necesita la matriz  $\Phi_h$ , ya que el resto de matrices y vectores son conocidos de apartados anteriores.

Derivando  $\Phi$  respecto de las coordenadas independientes  $h$  se obtienen las matrices (6.11) y (6.12).

### 4.- Solución del problema de aceleración

Para resolver dicho problema se utiliza la ecuación (3.13), que requiere del cálculo de  $\Phi_{q_G}$ , que será la derivada temporal de la matriz que resulta de derivar  $\Phi$  respecto a todas las coordenadas del grupo estructural  $q_G$ , tanto dependientes como independientes. El orden de derivación es:

$$q_G = [x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, v_{1x}, v_{1y}, v_{1z}, v_{2x}, v_{2y}, v_{2z}, v_{3x}, v_{3y}, v_{3z}, v_{4x}, v_{4y}, v_{4z}, v_{5x}, v_{5y}, v_{5z}, v_{6x}, v_{6y}, v_{6z}, \theta]$$

Se obtienen dos matrices, las cuales aparecen en el Anexo divididas por columnas en dos bloques; la asociada al coseno viene con las referencias (6.13 y refec:Figp 79 2,cos) y la del seno (6.15 y 6.16). Las matrices y vectores necesarias para terminar el problema son conocidos de apartados anteriores y de las correspondientes condiciones de los sólidos externos de unión.

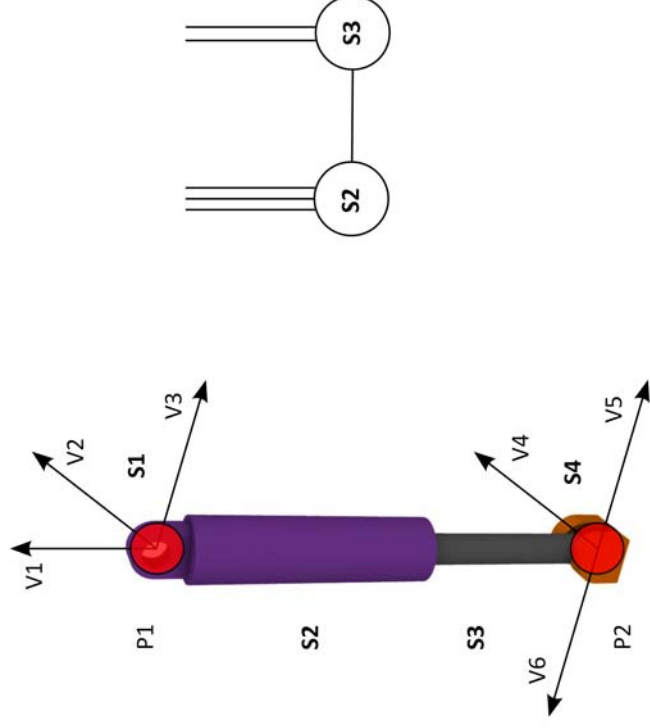


Figura 3.6: Grupo estructural 1C1E1P. Esquema cinemático y representación del grafo estructural. Este grupo consta de dos eslabones rígidos  $S2$  y  $S3$ , unidos mediante un par prismático. A su vez, está conectado con dos sólidos distintos por dos pares extremos; esférico y cardan.

### 3.1.5. Grupo perteneciente a sistema de suspensión 1C1E1P

La Figura 3.6 muestra el grupo estructural 1C1E1P. Es un grupo sencillo, formado por dos eslabones rígidos  $S2$  y  $S3$ , conectados entre sí formando un par interno prismático. A su vez, este grupo estructural está conectado con dos sólidos distintos  $S1$  y  $S4$ , por dos pares extremos; por un lado utiliza un par esférico y por otro un par cardan.

#### 1.- Identificación de las coordenadas de grupo y parámetros

Se modeliza el grupo estructural en coordenadas naturales. Para que la posición del grupo quede perfectamente determinada, se define cada eslabón del grupo adecuadamente. Se usa como mínimo 12 coordenadas por eslabón; tres vectores y un punto, dos vectores y dos puntos, etc. En la Tabla 3.3 viene detallada esta definición.

Entre las coordenadas de grupo se hace la distinción de: dependientes  $\varphi$  e independientes  $\mathbf{h}$ . Como los puntos  $P_1$ ,  $P_2$  y el vector  $V_6$  pertenecen a otros sólidos, se consideran sus coordenadas asociadas como independientes de grupo. Por tanto, los vectores  $V_1$ ,  $V_2$ ,  $V_3$ ,  $V_4$  y  $V_5$  definen el conjunto de coordenadas dependientes de grupo estructural. Además, es importante introducir en este conjunto la variable  $\mathbf{s}$ , que define la distancia asociada al par prismático.

En definitiva, se tiene:

- Coordenadas dependientes de grupo estructural:

$$\varphi = [v_{1x}, v_{1y}, v_{1z}, v_{2y}, v_{2z}, v_{3x}, v_{3y}, v_{3z}, v_{4x}, v_{4y}, v_{4z}, v_{5x}, v_{5y}, v_{5z}, \mathbf{s}]$$

- Coordenadas independientes de grupo estructural:

$$\mathbf{h} = [x_1, y_1, z_1, x_2, y_2, z_2, v_{6x}, v_{6y}, v_{6z}]$$



SÓLIDO RÍGIDO							
ID	SOLIDO	PUNTOS	VECTORES	COORDS.	ECS. S.R.	COORDS. LIBRES	
1	2	P1	V1, V2, V3	3 x 3 = 9	6(triedro)	9 - 6 = 3	
3	3	P2	V1, V4, V5	3 x 2 = 6	5(triedro)	6 - 5 = 1	
4	Quedan cuatro coordenadas libres. Faltan ecuaciones de par.						
PAR CINEMÁTICO							
PAR	TIPO	ECS. de par	ECS. nuevas	OBSERVACIONES			
1-2	S(Esférico)	3	0	Porque comparten P1.			
2-3	P(Prismático)	5	3	Las otras dos se imponen al compartir V1.			
3-4	CD(Cardan)	4	1				
10	Los pares cinemáticos añaden cuatro ecuaciones adicionales de par.						

Cuadro 3.3: Principales características del grupo estructural 1C1E1P. Coordenadas por eslabón, ecuaciones de restricción de sólido rígido y de par. Se marcan en cursiva las coordenadas compartidas.

## 2.- Solución del problema de posición por grupos estructurales

En la Tabla 3.3 se determinan para cada eslabón el número de ecuaciones de restricción, asociadas tanto de restricción de sólido rígido como de par cinemático. A continuación se muestran las anteriores.

Ecuaciones de restricción de sólido rígido:

### ■ Eslabón 2

$$\begin{aligned}
 v_{1x}^2 + v_{1y}^2 + v_{1z}^2 - 1 &= 0 \\
 v_{2x}^2 + v_{2y}^2 + v_{2z}^2 - 1 &= 0 \\
 v_{3x}^2 + v_{3y}^2 + v_{3z}^2 - 1 &= 0 \\
 v_{1x} \cdot v_{2x} + v_{1y} \cdot v_{2y} + v_{1z} \cdot v_{2z} &= 0 \\
 v_{1x} \cdot v_{3x} + v_{1y} \cdot v_{3y} + v_{1z} \cdot v_{3z} &= 0 \\
 v_{2x} \cdot v_{3x} + v_{2y} \cdot v_{3y} + v_{2z} \cdot v_{3z} &= 0
 \end{aligned} \tag{3.38}$$

### ■ Eslabón 3

$$\begin{aligned}
 v_{4x}^2 + v_{4y}^2 + v_{4z}^2 - 1 &= 0 \\
 v_{5x}^2 + v_{5y}^2 + v_{5z}^2 - 1 &= 0 \\
 v_{4x} \cdot v_{5x} + v_{4y} \cdot v_{5y} + v_{4z} \cdot v_{5z} &= 0 \\
 v_{1x} \cdot v_{4x} + v_{1y} \cdot v_{4y} + v_{1z} \cdot v_{4z} &= 0 \\
 v_{1x} \cdot v_{5x} + v_{1y} \cdot v_{5y} + v_{1z} \cdot v_{5z} &= 0
 \end{aligned} \tag{3.39}$$

Ecuación de restricción mixta:

### ■ Distancia asociada al par prismático

$$(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 - s^2 = 0 \tag{3.40}$$

Ecuaciones de restricción de par cinemático:

### ■ Par prismático

$$\begin{aligned}
 (x_2 - x_1) \cdot v_{1y} - (y_2 - y_1) \cdot v_{1x} &= 0 \\
 (z_2 - z_1) \cdot v_{1x} - (x_2 - x_1) \cdot v_{1z} &= 0 \\
 (y_2 - y_1) \cdot v_{1z} - (z_2 - z_1) \cdot v_{1y} &= 0 \\
 v_{5x} \cdot v_{2x} + v_{5y} \cdot v_{2y} + v_{5z} \cdot v_{2z} &= 0
 \end{aligned} \tag{3.41}$$

- Cardan

$$v_{4x} \cdot v_{6x} + v_{4y} \cdot v_{6y} + v_{4z} \cdot v_{6z} = 0 \quad (3.42)$$

Se tiene las ya comentadas ecuaciones de restricción de sólido rígido, (3.38 o 3.39), y de cardan (3.42). Con diferencia a grupos anteriores, aquí aparecen una ecuación de restricción mixta que introduce la variable  $s$ , usada cuando el signo de la coordenada no es relevante (3.40), y las ecuaciones de restricción de par prismático (3.41).

El par prismático es de clase I, que restringe cinco grados de libertad. Al definir un vector básico en la dirección del par se cumplen automáticamente dos restricciones, y por tanto se necesitarán tres ecuaciones de restricción. De estas últimas sólo tres de ellas son independientes. Esto nos lleva a distintas opciones; resolver el sistema redundante o considerar dos de ellas y descartar una por la propia configuración del grupo. Aquí se ha elegido resolver el sistema redundante.

A través de las ecuaciones de restricción se forma  $\Phi$ :

$$\Phi = \begin{bmatrix} v_{1x}^2 + v_{1y}^2 + v_{1z}^2 - 1 \\ v_{2x}^2 + v_{2y}^2 + v_{2z}^2 - 1 \\ v_{3x}^2 + v_{3y}^2 + v_{3z}^2 - 1 \\ v_{1x} \cdot v_{2x} + v_{1y} \cdot v_{2y} + v_{1z} \cdot v_{2z} \\ v_{1x} \cdot v_{3x} + v_{1y} \cdot v_{3y} + v_{1z} \cdot v_{3z} \\ v_{2x} \cdot v_{3x} + v_{2y} \cdot v_{3y} + v_{2z} \cdot v_{3z} \\ v_{4x}^2 + v_{4y}^2 + v_{4z}^2 - 1 \\ v_{5x}^2 + v_{5y}^2 + v_{5z}^2 - 1 \\ v_{4x} \cdot v_{5x} + v_{4y} \cdot v_{5y} + v_{4z} \cdot v_{5z} \\ v_{1x} \cdot v_{4x} + v_{1y} \cdot v_{4y} + v_{1z} \cdot v_{4z} \\ v_{1x} \cdot v_{5x} + v_{1y} \cdot v_{5y} + v_{1z} \cdot v_{5z} \\ (x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 - s^2 \\ (x_2 - x_1) \cdot v_{1y} - (y_2 - y_1) \cdot v_{1x} \\ (z_2 - z_1) \cdot v_{1x} - (x_2 - x_1) \cdot v_{1z} \\ (y_2 - y_1) \cdot v_{1z} - (z_2 - z_1) \cdot v_{1y} \\ v_{5x} \cdot v_{2x} + v_{5y} \cdot v_{2y} + v_{5z} \cdot v_{2z} \\ v_{4x} \cdot v_{6x} + v_{4y} \cdot v_{6y} + v_{4z} \cdot v_{6z} \end{bmatrix} \quad (3.43)$$

Para resolver el problema de posición además se requiere de la matriz Jacobiana  $\Phi_\varphi$ , que se muestra en el Anexo (6.17).

Por el método iterativo de Newton-Raphson, teniendo un sistema redundante, se obtienen los valores de las coordenadas dependientes de grupo a partir de (3.44). Esto sucede ya que al no ser una Jacobiana cuadrada no se podrá calcular su inversa, por tanto para obtener la Jacobiana invertible se calcula  $\Phi_\varphi^T \cdot \Phi_\varphi$ .

$$\varphi_k = \varphi_{k-1} - \left( \Phi_\varphi^T \cdot \Phi_\varphi \right)_{k-1}^{-1} \cdot \Phi_\varphi^T \cdot \Phi_{k-1} \quad (3.44)$$

### 3.- Solución del problema de velocidad

Para resolver este problema se utiliza la ecuación (3.45) por ser un sistema redundante. Necesita la matriz  $\Phi_h$ , ya que el resto de vectores y matrices son conocidos por apartados anteriores o de entrada.

$$\dot{\varphi} = - \left( \Phi_\varphi^T \cdot \Phi_\varphi \right)^{-1} \left[ \Phi_\varphi^T \cdot \Phi_h \dot{h} \right] \quad (3.45)$$

Derivando  $\Phi$  respecto de las coordenadas independientes  $\mathbf{h}$  se obtiene la matriz (6.18) del Anexo.

#### 4.- Solución del problema de aceleración

Para resolver dicho problema se utiliza la ecuación (3.46) para un sistema redundante. Además se requiere del cálculo de  $\dot{\Phi}_{\mathbf{q}_G}$ , que es la matriz dividida por columnas en dos bloques del Anexo con referencias (6.19 y 6.20), derivada temporal de la matriz que resulta de derivar  $\Phi$  respecto a todas las coordenadas del grupo estructural  $\mathbf{q}_G$ , donde el orden de derivación es:

$$\mathbf{q}_G = [x_1, y_1, z_1, x_2, y_2, z_2, v_{1x}, v_{1y}, v_{1z}, v_{2x}, v_{2y}, v_{2z}, v_{3x}, v_{3y}, v_{3z}, v_{4x}, v_{4y}, v_{4z}, v_{5x}, v_{5y}, v_{5z}, v_{6x}, v_{6y}, v_{6z}, \mathbf{s}].$$

$$\dot{\varphi} = - \left( \Phi_{\varphi}^T \cdot \Phi_{\varphi} \right)^{-1} \cdot \Phi_{\varphi}^T \cdot \left[ \Phi_{\mathbf{h}} \ddot{\mathbf{h}} + \dot{\Phi}_{\mathbf{q}_G} \dot{\mathbf{q}}_G \right] \quad (3.46)$$

## 3.2. Modelización de sistemas multicuerpo. Casos de estudio

Se plantean dos casos de estudio, el cuadrilátero articulado y el sistema de suspensión de un camión. Analizando el movimiento de todos los elementos del mecanismo cuando se conoce el movimiento de sus grados de libertad o entradas.

Establecidos todos los fundamentos del método basado en la derivada temporal de la matriz Jacobiana, y visto cómo se aplica a distintos tipos de grupos estructurales, a continuación se aplican a los casos propuestos. En principio se lleva a cabo el análisis estructural, identificando el número de eslabones, movimientos de entrada, pares de los que está compuesto y posteriormente se identifican los grupos estructurales a partir del análisis del grafo. Además se da orden de resolución al mecanismo a través del diagrama estructural. Por último, para cada grupo estructural se resuelve el problema de posición, velocidad y aceleración.

### 3.2.1. Cuadrilátero escalable

El primer problema de estudio viene representado en la Figura 3.7. El cuadrilátero escalable es un caso particular del cuadrilátero articulado. Al igual que éste, es un mecanismo plano de movilidad uno, con movimiento de entrada  $\theta$ , formado por tres eslabones base unidos por dos extremos al bastidor, definidos mediante pares cinemáticos de rotación entre eslabones o bastidor. Se diferencia al cuadrilátero articulado en que a esos tres eslabones base pueden añadirse  $\mathbf{k}$  diadas  $3RF$ , una a continuación de la otra, de manera que para el caso  $\mathbf{k} = 1$  se contaría con la definición de cuadrilátero articulado, y por ejemplo, para  $\mathbf{k} = 5$ , con el cuadrilátero articulado más cuatro diadas  $3RF$  unidas una tras otra.

El cuadrilátero articulado ha sido analizado a lo largo de todo este proyecto como ejemplo. Rememorando, en el Capítulo 2: Fundamentos, apartado 2.1.3, el mecanismo fue separado en grupos estructurales a través de su grafo. A partir de este se obtuvo que los grupos estructurales que componen el cuadrilátero escalable son el par de rotación  $1RF$  y la diada  $3RF$ . Con la información que se obtuvo del diagrama estructural, 2.1.3, se dotó de orden a la resolución del mecanismo, primero resolviendo el grupo  $1RF$  y después los grupos  $3RF$ .

En este capítulo, en el apartado 3.1.1 y 3.1.2, se resuelve la cinemática de los grupos estructurales que componen este caso de estudio, por tanto podemos dar como resuelto este caso de estudio.

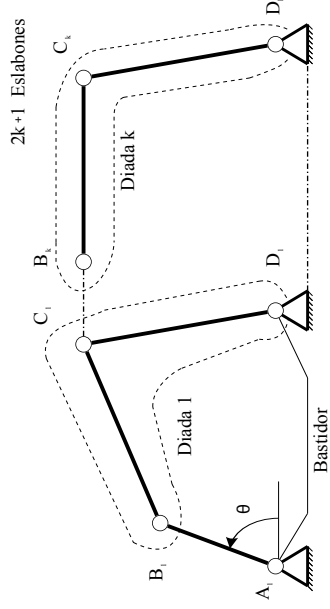


Figura 3.7: Esquema cinemático del cuadrilátero escalable. Al eslabón formado por los puntos  $A$  y  $B$ , se añade el número de diadas  $k$ .

### 3.2.2. Sistema de suspensión de un camión

El segundo problema de estudio se representa en la Figura 3.8. El sistema de suspensión de un camión es un mecanismo espacial complejo. Está definido por trece eslabones y una gran diversidad de pares cinemáticos, entre los que se encuentran pares prismáticos, de revolución, esféricos y tipo cardan.

El mecanismo tiene dos movimientos de entrada. Al tratarse de un sistema de suspensión se definen dos coordenadas verticales, en extremos del eje como entradas, para simular el empuje real del suelo sobre las ruedas.

La metodología a seguir es la misma que en el caso de estudio anterior. Una vez obtenido el grafo se analiza cada uno de los grupos estructurales sencillos en los que se puede descomponer el mecanismo. Ya identificados los grupos estructurales se analiza su cinemática por la formulación basada en la derivada temporal de la matriz Jacobiana.

#### ■ Obtención del grafo estructural

En el mecanismo se pueden identificar los trece eslabones que lo componen (Figura 3.9, a), así como los distintos pares cinemáticos que los conectan, (Figura 3.9, b). Hay que destacar que se consideran por separado los eslabones 7 y 6 para tener en cuenta la torsión en la barra rotatoria. A continuación se expone una Tabla a modo de recordatorio, donde se muestran los grados de libertad permitidos por cada par cinemático.

Par cinemático 2	Grados de libertad
Esférico (S)	3
Rotación (R)	1
Prismático (P)	1
Cardan (CD)	2

Cuadro 3.4: Grados de libertad permitidos por los pares cinemáticos espaciales.

Cada grado de libertad tiene asociada una línea de trazo fino, de trazo grueso vendrán determinados los movimientos de entrada. Identificados los pares y eslabones se puede encontrar con facilidad el grafo estructural, que se representa en la Figura 3.10.

#### ■ Análisis del grafo estructural

Siguiendo las pautas definidas en el Capítulo 2: Fundamentos, apartado 2.1.3, para analizar el grafo estructural del mecanismo. El principal objetivo es encontrar un conjunto de eslabones que sea grupo estructural, y para ello se ha de cumplir, en tres dimensiones, la condición fundamental de grupo que se muestra a continuación.

$$S_c - n_c = 6 \cdot (P - N_m) \quad (3.47)$$

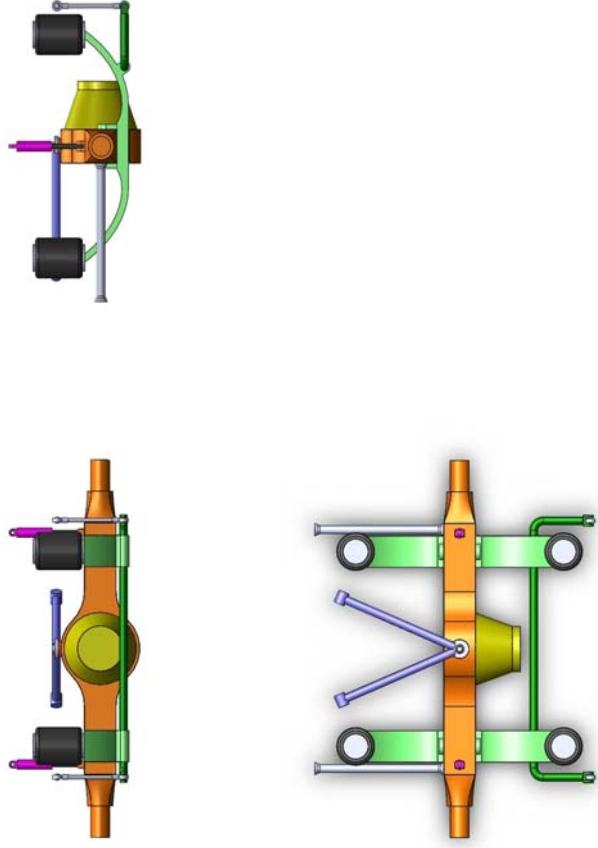


Figura 3.8: Esquema cinemático del sistema de suspensión de un camión.

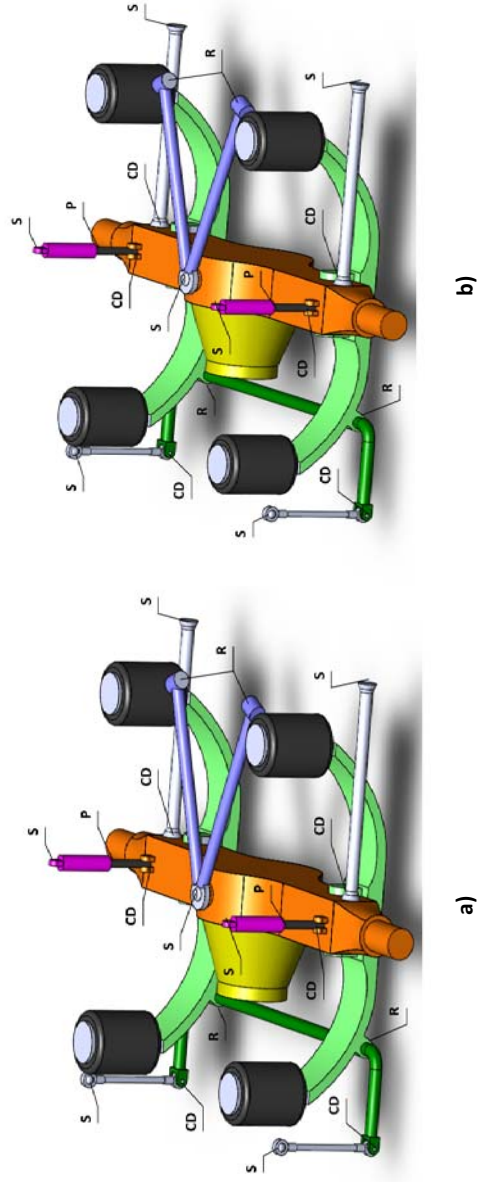


Figura 3.9: a) Identificación de eslabones. b) Identificación de pares que componen el sistema de suspensión del camión.

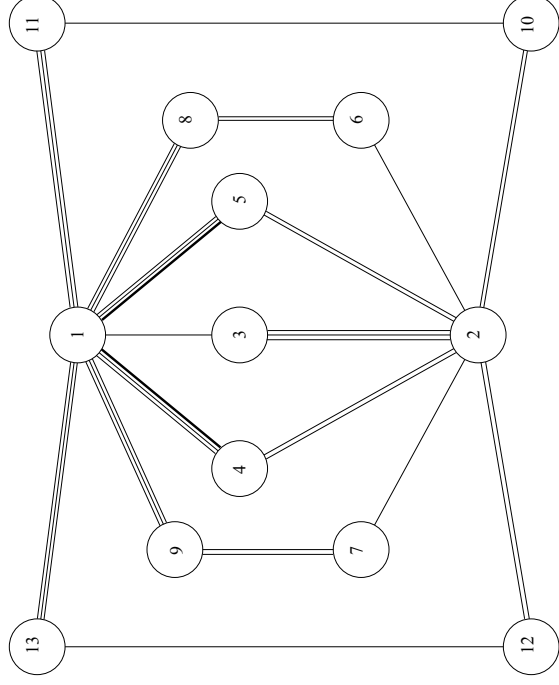


Figura 3.10: Grafo del sistema de suspensión del camión.

Donde  $S_c$  es la movilidad que dejan los pares cinemáticos que intervienen,  $n_c$  son los movimientos de entrada,  $P$  los pares cinemáticos que recibe y  $N_m$  los eslabones del grupo. En la Figura 3.11 se describe esquemáticamente el proceso que numéricamente se detalla bajo estas líneas.

*Grupo 2, 3, 4, 5*

$$N_m = 4; P = 6; S_c = 14; n_c = 2.$$

$$14 - 2 = 6 \cdot (6 - 4)$$

*Grupo 7, 9*

$$N_m = 2; P = 3; S_c = 6; n_c = 0.$$

$$6 - 0 = 6 \cdot (3 - 2)$$

*Grupo 6, 8*

$$N_m = 2; P = 3; S_c = 6; n_c = 0.$$

$$6 - 0 = 6 \cdot (3 - 2)$$

*Grupo 12, 13*

$$N_m = 2; P = 3; S_c = 6; n_c = 0.$$

$$6 - 0 = 6 \cdot (3 - 2)$$

*Grupo 10, 11*

$$N_m = 2; P = 3; S_c = 6; n_c = 0.$$

$$6 - 0 = 6 \cdot (3 - 2)$$

■ Diagrama estructural

El diagrama estructural se obtiene a partir del grafo generado en el punto anterior. Éste nos proporciona la misma información del grafo pero de forma más intuitiva. Se dota de orden de resolución al mecanismo, así como el número de eslabones del que está compuesto y los movimientos de entrada que recibe. En la Figura 3.12 aparece el resultado.

El primer grupo que se resuelve será el *Grupo 2, 3, 4, 5*, después se puede resolver indistintamente *Grupo 6, 8*, *Grupo 7, 9*, *Grupo 10, 11* y *Grupo 12, 13*.

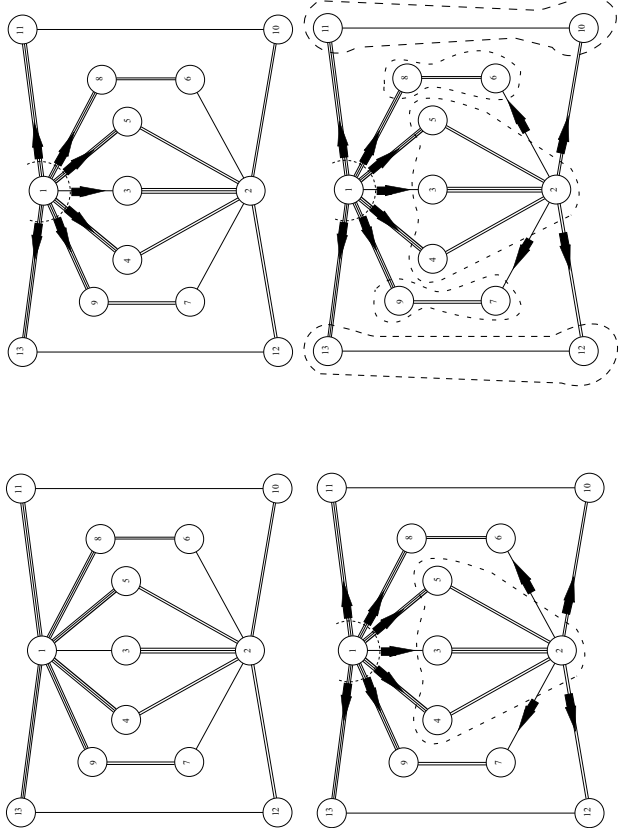


Figura 3.11: Análisis del grafo estructural del sistema de suspensión del camión.

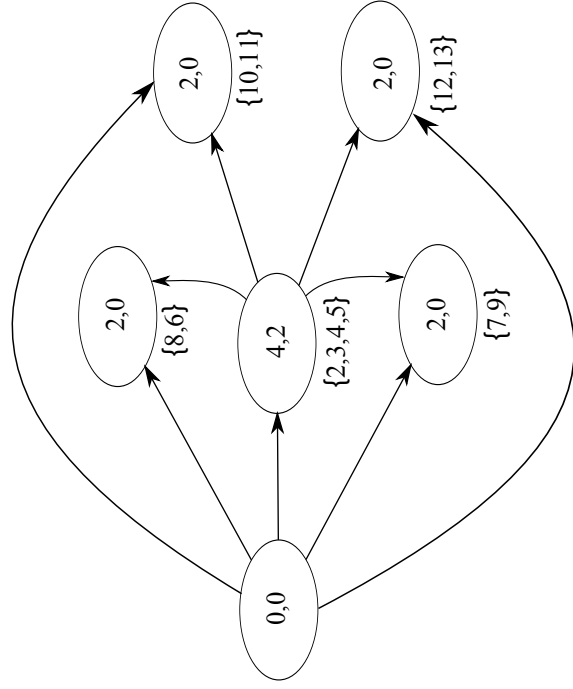


Figura 3.12: Diagrama estructural del sistema de suspensión del camión.

- Análisis por grupo estructural

El sistema de suspensión de una cabeza tractora, como se ha podido apreciar a través del análisis del grafo, está compuesto por tres tipos distintos de grupo estructural. El primero de ellos, el *Grupo 2, 3, 4, 5* corresponde al tipo de grupo estructural *2C3E1R*, apartado 3.1.3; el resto de grupos, *Grupo 6, 8* o *Grupo 7, 9* y *Grupo 10, 11* o *Grupo 12, 13*, corresponden a los tipos *1C1E1R* y *1C1E1P* respectivamente, apartados 3.1.4 y 3.1.5.

Los grupos estructurales que componen el sistema de suspensión aparecen resueltos en el apartado Modelización de grupo estructural mediante método de derivada temporal, 3.3.1.3. Ha sido resuelta la cinemática de todos ellos de forma generalizada, es decir, no se ha tratado en ningún momento de uniones con bastidor. Por ello, cuando se utilicen las matrices que resuelven cada grupo estructural, habrá que identificar qué coordenadas pertenecen al bastidor, para no tenerlas en cuenta como coordenadas independientes, además la derivada temporal de cualquier coordenada asociada al bastidor será nula. También habrá que considerar que para el *Grupo 6, 8* cambia el signo de uno de los coeficientes de combinación lineal (el relativo a coordenada x). Con estas consideraciones se puede dar por resuelto el segundo caso de estudio.

### 3.3. Implementación en Matlab

En este último apartado se explica cómo se ha implementado el código para el análisis cinemático de los problemas propuestos, a partir de los resultados previos obtenidos en el análisis estructural. Tal implementación se lleva a cabo utilizando Matlab como soporte de cálculo, un potente entorno de programación para análisis matemático donde la mayoría de funciones utilizadas son de elaboración propia.

Cada implementación cuenta con un fichero raíz, otro de geometría, ficheros de resolución de grupo específico e incluso funciones auxiliares o secundarias, todos necesarios para la solución. En el Capítulo 4 se presentarán los resultados de aplicar esta metodología.

#### 3.3.1. Cuadrilátero escalable

Se emplean cuatro ficheros para resolver la cinemática del cuadrilátero escalable. Un fichero raíz principal, un fichero de datos y dos ficheros auxiliares que resuelven los grupos estructurales de los que está compuesto el mecanismo. A continuación se muestra y explica su contenido.

##### 3.3.1.1. Fichero raíz MBSBEG\_Cuad.m

El fichero raíz del cuadrilátero escalable, Cuadro 3.1, es el encargado de llamar al fichero que define la geometría del mecanismo, *MBDatosCuad.m*, y a los que ejecutan la resolución de la cinemática de cada grupo estructural específico, *SolucionGrupo1RFMatR2.m* y *SolucionGrupo3RFMatR.m*. Es aquí donde se dota de orden de resolución y por ello es el fichero principal.

Al principio del fichero se indica donde se encuentran los ficheros que resuelven los grupos estructurales, carpeta *SolucionGrupos*. Después se define el número *m* de diadas (*3RF*) de adición a la manivela *1RF*. Seguidamente se establecen los valores de la coordenada independiente de todo el sistema, su velocidad y aceleración. También se da un valor de incremento que determinará momentos en los que ha de ser resuelta la cinemática del mecanismo.

En el bucle final *for* exterior permite resolver el sistema  $2*n$  veces, incrementando de la coordenada de entrada en *delta\_alfa(1,1)*. En el bucle *for* interior, dependiendo del valor de *ng*, se llama a un fichero de resolución de grupo u otro. Estos bucles están diseñados para resolver el mecanismo cuando la manivela da dos vueltas completas.



```
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
% % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %

addpath SolucionGrupos
clear all
clc

m=5; % Número de diadas 3R.

% Valores de gdl e incremento.
alfa(1,1)=pi/2; n=72; delta_alfa(1,1)=2*pi/n;
alfap(1,1)=1; alfas(1,1)=0;

MBDatosCudad; % Define la geometría del sistema.

zp=1;
zs=0;

for i=1:2*n
    i=i+1;
    for ng = 2:length(MGrupos)
        z=alfa;
        if ng==2
            [POI] = SolucionGrupoIRFMatR2(z,zp,ng,zs,POI,MGrupos);
            % Resuelve el grupo IRF.
        else
            [POI] = SolucionGrupo3RFMatR(z,zp,ng,zs,POI,MGrupos);
            % Resuelve el grupo 3R.
        end
    end
end

alfa = alfa+delta_alfa;

end
```

Cuadro 3.1: Fichero raíz del cuadrilátero escalable.

### 3.3.1.2. Fichero de geometría MBDatosCuad.m

Este fichero, Cuadro 3.2, es al que accede el fichero raíz, `MBSBEG_Cuad.m`, para definir la geometría del sistema. Ésta se logra gracias a la creación de unas estructuras de datos auxiliares, como `MGrupos.poi`, `MGrupos.eslabones` y `MGrupos_tipo`, y otras definitivas como `POI.PVA`. A continuación se describe cada uno de estos tipos de estructuras:

- `MGrupos(i).poi` Hace referencia a los puntos de interés correspondientes al grupo estructural `i`. Esta estructura de datos fila tiene por columnas el identificador del POI en la estructura de datos global, el punto de referencia para la toma de coordenadas locales, el eslabón al que pertenece el POI y coordenadas `x` e `y` locales; `numPoi`, `Pref`, `Eslabptnce` y `cord x` y `local`, respectivamente.
- `MGrupos(i).eslabones` Hace referencia a los eslabones correspondientes al grupo estructural `i`. Esta estructura fila tiene por columnas los eslabones de los que está compuesto dicho grupo.
- `POI(numPoi).PVA` Hace referencia a la posición, velocidad y aceleración del `numPoi`. Esta estructura de datos fila tiene por columnas las posiciones en `x` e `y`, velocidades y aceleraciones del punto `numPoi`.

Al principio del fichero se define el origen global y se almacenan en `MGrupos(1).poi` los datos relativos a los POI's del bastidor. Con una estructura `for`, dependiendo del número de diadas `m`, se definen los más o menos puntos de bastidor. Después se determinan los eslabones correspondientes a cada grupo, utilizando la estructura `for` con `MGrupos(i).eslabones`, también en función de `m`, y el tipo de grupo con `MGrupos_tipo`. Acto seguido se almacena la información correspondiente a los demás POI's en sus correspondientes `MGrupos.poi`. Para finalizar se guardan todos los datos contenidos en las estructuras auxiliares en `POI.PVA`, con ayuda de más estructuras `for`.

Las estructuras de datos definitivas almacenan datos absolutos del sistema. En este fichero lo que se hace es almacenar en `POI.PVA` datos relativos a puntos del bastidor y a otros POI's. Los datos relativos a otros POI's han sido recibidos de estructuras de datos auxiliares, que proporcionan información, posiciones iniciales de coordenadas, a los ficheros de resolución de grupo estructural. Una vez que los ficheros de grupo obtengan los resultados del análisis cinemático, estos son almacenados de nuevo en `POI.PVA`.

### 3.3.1.3. Ficheros de solución de grupos

A continuación se explican los ficheros que resuelven los grupos `1RF` y `3RF`. Se unifica en este apartado las explicaciones de `SoluciónGrupo1RFMatR` y `SoluciónGrupo3RFMatR`, debido a que el procedimiento seguido para resolver los grupos es el mismo, y por tanto su implementación en Matlab.

Debido a la extensión de `SoluciónGrupo1RFMatR` se divide el fichero en cinco partes: Cuadros 3.3-3.7. Al igual sucede con `SoluciónGrupo3RFMatR`: Cuadros 3.8-3.12.

Los ficheros de resolución de grupo reciben como entradas `z`, `zp`, `zs`, `ng`, `POI` y `MGrupos`; posición, velocidad y aceleración de la coordenada de entrada, número de grupo, estructura de datos global y auxiliar, respectivamente.

A continuación se describe brevemente lo más característico relativo a estos ficheros:

- Cuadro Identificación: En él se identifican las coordenadas de grupo, dependientes e independientes, y se crean con `POI.PVA`, guardando las posiciones iniciales de todas ellas. Además se calculan las constantes necesarias, como por ejemplo longitudes de barras necesarias para las ecuaciones de restricción.
- Cuadro Problema de posición: En él se definen las ecuaciones de restricción mencionadas para cada grupo en con sus correspondientes condiciones, como por ejemplo el que se escoja una restricción u otra en función del ángulo de entrada. Se calcula la Jacobiana respecto de las coordenadas dependientes y se itera por Newton-Raphson. Una vez que converge el algoritmo, las coordenadas dependientes de almacenan en la estructura de datos `POI.PVA`.
- Cuadro Problema de velocidad: En él se calculan las Jacobianas asociadas a las variables independientes y se resuelve el problema de velocidad. Resuelto, se almacenan las velocidades dependientes en `POI.PVA`.

```

%% %% %% %% %% %% %% %%
%% %% MBDATOS %% %%
%% %% %% %% %% %% %% %%

% Puntos sobre el bastidor.
x1=0; y1=0;

% Valores iniciales para las variables.

% Coordenadas de puntos sobre bastidor.

k=2;
MGrupos(1).poi(1,:) = [2 0 1 0 0];

for i=1:m % Resto de puntos sobre el bastidor.
    MGrupos(1).poi(i+1,:) = [2+k 0 1 6*i 0];
    k=k+2;
end

% Eslabones.

MGrupos(1).eslabones = 1;
MGrupos(2).eslabones = 2;
k=0;

for i=1:m % Resto eslabones en grupos.
    MGrupos(i+2).eslabones = [k+3 k+4];
    MGrupos_tipo(1,i+2) = 4;
    k=k+2;
end

MGrupos_tipo(1,1) = 0;
MGrupos_tipo(1,2) = 1;

% numPoi Pref Eslabptnce cord x y local
MGrupos(2).poi = [1 2 2 2 0];

for i=1:m
    % numPoi Pref Eslabptnce cord x y local
    MGrupos(i+2).poi = [2*i+1 2*i-1 2*i+1 6 0
                       2*i+1 2*i+2 2*i+2 6 0 ];
    POI(2*i+1).PVA(1:2) = [6*i 5.8];
    POI(2*i+1).PVA(3:6) = 0;
end

POI(1).PVA(1:2) = [0 2];
POI(1).PVA(3:6) = 0;

for i = 1:size(MGrupos(1).poi,1)
    POI(MGrupos(1).poi(i,1)).PVA(1:2) = MGrupos(1).poi(i,4:5);
    POI(MGrupos(1).poi(i,1)).PVA(3:6) = 0;
end

```

Cuadro 3.2: Fichero que define geometría del cuadrilátero escalable.

- Cuadro Problema de aceleración: En él se resuelve el problema de aceleración básicamente. Se calcula la matriz derivada temporal de la Jacobiana de las funciones de posición respecto a todas las coordenadas de grupo y se multiplica por las velocidades de grupo. Resuelto, se almacenan las aceleraciones dependientes en `POI.PVA`.
- Cuadro Otros POI: Se utiliza en el caso de que tener más POI en el grupo estructural. Se crean las matrices de transformación de posiciones, `Rotacion`, `Rotacion`, `Rotacion`, `Rotacion`, y aceleraciones, `Rotacions`, y a partir de ellas se obtienen las posiciones, `Rotacion`, `Rotacion` y aceleraciones de los POI's adicionales, junto a los datos de los POI's ya resueltos en el mismo fichero de grupo. En el mecanismo no hay otros POI's y por tanto no llega a ejecutarse nunca esta parte del fichero.

### 3.3.2. Sistema de suspensión del eje de un camión

La resolución cinemática del sistema de suspensión de un camión se divide en cinco ficheros. Estos son: un fichero raíz, otro de datos y tres ficheros que resuelven los grupos estructurales de los que está compuesto el mecanismo. Además, se definen una serie de funciones auxiliares y secundarias que facilitan distintos cálculos, que son llamadas en los ficheros que resuelven los grupos.

Para la ejecución del programa la estructura queda: El fichero raíz y el de datos en el exterior de la carpeta `SolucionGrupos`, en la que se encuentran ficheros que resuelven los grupos, las funciones auxiliares y las secundarias.

La implementación en Matlab de este caso de estudio, se desarrolla resolviendo, en primer lugar, la cinemática del mecanismo para un solo eje. Posteriormente se extiende la solución a un número variable de ejes  $m$ , que fluctúa del 1 al 5. Con ello se consigue otro sistema mecánico escalable, en este caso para tres dimensiones, con otros tipos de grupos estructurales más complejos. En este apartado se presenta dicha implementación.

#### 3.3.2.1. Fichero raíz `MBSuspension5ejes.m`

El fichero raíz (Cuadros 3.13 y 3.14) se encarga básicamente de llamar a los ficheros `MBDatosSuspension5ejesB.m`, `SolucionGrupo2345C`, `SolucionGrupo79` y `SolucionGrupo1213_b`; el primero define la geometría del mecanismo y el resto resuelve los grupos estructurales.

Al principio del fichero se establece el número de ejes  $m$  y se definen con matrices fila los grados de libertad del sistema: posiciones iniciales de partida (`Z150` y `Z160`), posiciones (`Z15` y `Z16`), velocidades (`Z15p` y `Z16p`) y aceleraciones (`Z15s` y `Z16s`). Para la entrada a la rueda derecha del camión se utiliza la numeración 16, para la izquierda la 15. A continuación, una estructura condicional determina el número de grupo estructural final `ngf` en función de  $m$ . Después se llama a `MBDatosSuspension5ejesB.m`.

Posteriormente se crean las variables `t` y `incr`, para que se analice la cinemática del sistema en un intervalo tiempo cada ciertos intervalos. Se logra a partir de la estructura `for` más externa.

Acto seguido se van resolviendo todos los grupos estructurales. Con una estructura `for` y otra condicional, en función de `ng`, se llama a un fichero u otro que resuelven la cinemática. Estos la resuelven en un sistema local y los resultados se graban en las estructuras de datos del sistema global `POI`, `VEC` y `MGrupos`, estructuras de retorno al fichero raíz.

Para que cada eje posea grados de libertad distintos, se incluye en este raíz una estructura condicional incluida en el bucle de tiempo. Con ello se tiene libertad e independencia para simular cada eje del sistema de suspensión del camión.

Por último, destacar que el fichero raíz puede contar con una serie de comandos que guardan y borran resultados, `save` y `delete` respectivamente. Estos se incluyen ya que se utilizan para la simulación espacial del movimiento, que se presenta en el capítulo posterior.

```

funcion[POI] = SolucionGrupoIRF(z,zp,ng,zs,POI,MGrupos);

%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %%
%% %% Este script se encarga de resolver el GE IRF.
%% %%
%% %%
% Se reciben:
%% -z y sus derivadas, coordenadas independientes globales.
%% -ng, número de grupo estructural.
%% -POI, matriz puntos de interés.
%% -MGrupos, matriz de matrices.

%% Esquema del grupo.

%% Variables del GE: q=[X2 Y2 alfa]. Las separamos en:
%% Dependientes: qG=[X2 Y2], coordenadas de P2.
%% Independientes: h=[Alfa], giro de la barra.

%% Identificación de variables, puntos y longitudes.

numpoi = MGrupos(ng).poi(1,1); % Id del POI P2.
puntoreferencia = MGrupos(ng).poi(1,2); % N° del punto de ref.

% La longitud L entre puntos de modelo es:
L1 = sqrt((MGrupos(ng).poi(1,4))^ 2 + MGrupos(ng).poi(1,5) ^ 2);

X1 = POI(puntoreferencia(1)).PVA(1);
Y1 = POI(puntoreferencia(1)).PVA(2);

Alfa1=z;
hp=zp;
hs=zs;

X2 = POI(numpoi).PVA(1);
Y2 = POI(numpoi).PVA(2);

qgrupo0(1:2,1) = [X2 Y2]';

```

Cuadro 3.3: Fichero del grupo estructural *IRF*; esquema e identificación de variables

```

%% Problema de posición: Newton-Raphson.
% mFiG, ecuaciones de restricción.
% FiQG, Jacobiana de miFG respecto de coordenadas dependientes de GE.
i = 0; tol = 1e-8; Res = 20; % Asignamos valores distintos de '0' para comenzar.

while Res > tol % Criterio de parada.
    if abs(sin(Alfa1)) < sqrt(2)/2
        mFiG(1:2,1) = [(X2-X1)^ 2 + (Y2-Y1)^ 2 - L1^ 2
                      Y2 - Y1 - L1*sin(Alfa1)];
    else
        mFiG(1:2,1) = [(X2-X1)^ 2 + (Y2-Y1)^ 2 - L1^ 2
                      X2 - X1 - L1*cos(Alfa1)];
    end

    FiQG(1,1:2) = [2*(X2-X1) 2*(Y2-Y1)];

    if abs(sin(Alfa1)) < sqrt(2)/2
        FiQG(2,1:2) = [0 1];
    else
        FiQG(2,1:2) = [1 0];
    end

    qgrupo = qgrupo0 - mFiG/FiQG;

    X2 = qgrupo(1); Y2 = qgrupo(2);
    qgrupo0 = [X2 Y2]';

    Res = norm(mFiG);

end

POI(numpoi).PVA(1) = X2;
POI(numpoi).PVA(2) = Y2;

```

Cuadro 3.4: Fichero del grupo estructural 1RF; problema de posición.

```

%% Problema de velocidad.
% h es [x3 y3 x2 y2].
% Fih, Jacobiana de mFiG respecto h.
% Fihhp, vector producto Fih y velocidades de h.

if abs(sin(Alfa1)) < sqrt(2)/2
    FihGhp(1:2,1) = [ 0
                    -L1*cos(Alfa1)*hp];
else
    FihGhp(1:2,1) = [ 0
                    L1*sin(Alfa1)*hp];
end

POI(numpoi).PVA(3:4) = -FihGhp/FiqG; % ya tendríamos los nuevos X2p Y2p.

X2p = POI(numpoi).PVA(3); Y2p = POI(numpoi).PVA(4);

```

Cuadro 3.5: Fichero del grupo estructural *IRF*; problema de velocidad.

```

%% Problema de aceleración.
% Fihhs, vector producto Fih y aceleraciones de h.
% Fiqp, derivada temporal de Jacobiana de mFiG respecto de todas las variables de GE.
% Fiqppq, vector producto Fiqp y velocidades de todas las variables de GE.
% Aquí se calculan Fiqppq (Fiq * qp) y Fihhs (Fih * hs). Ambas se suman en el mismo vector,
que llamamos Fiqppq.

if abs(sin(Alfa1)) < sqrt(2)/2
    Fiqppq = [2*X2p^ 2+2*Y2p^ 2
             L1*sin(Alfa1)*hp^ 2-L1*cos(Alfa1)*hs];
else
    Fiqppq = [2*X2p^ 2+2*Y2p^ 2
             L1*cos(Alfa1)*hp^ 2+L1*sin(Alfa1)*hs];
end

POI(numpoi).PVA(5:6) = - Fiqppq/FiqG;

```

Cuadro 3.6: Fichero del grupo estructural *IRF*; problema de aceleración.

```

%% Otros POI.

if size(MGrupos(ng).poi,1) >1

for npto = 2: size(MGrupos(ng).poi,1)
Rotacion = [X2-X1 Y1-Y2
            Y2-Y1 X2-X1];

Rotacion = (1/L1)*Rotacion;

X2s = POI(numpoi).PVA(1,5);
Y2s = POI(numpoi).PVA(1,6);

Rotacionp = [X2p -Y2p
            Y2p X2p];

Rotacionp = (1/L1)*Rotacionp;

Rotacions = [X2s -Y2s
            Y2s X2s];

Rotacions = (1/L1)*Rotacions;

POI(MGrupos(ng).poi(npto,1)).PVA(1:2)=POI(MGrupos(ng).poi(npto,2)).PVA(1:2)'
+ Rotacion * MGrupos(ng).poi(npto,4:5)';
POI(MGrupos(ng).poi(npto,1)).PVA(3:4)=POI(MGrupos(ng).poi(npto,2)).PVA(3:4)'
+ Rotacionp * MGrupos(ng).poi(npto,4:5)';
POI(MGrupos(ng).poi(npto,1)).PVA(5:6)=POI(MGrupos(ng).poi(npto,2)).PVA(5:6)'
+ Rotacions * MGrupos(ng).poi(npto,4:5)';

end

end

```

Cuadro 3.7: Fichero del grupo estructural *1RF*; cálculo de otros POI's.



```

funcion[POI] = SolucionGrupo3RFMatR(z,zp,ng,zs,POI,MGrupos);

%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %%
%% %% Este script se encarga de resolver el GE 3RM.
%% %%
%% %%
% Se reciben:
%% -z y sus derivadas, coordenadas independientes globales.
%% -ng, número de grupo estructural.
%% -POI, matriz puntos de interés.
%% -MGrupos, matriz de matrices.

%% Esquema del grupo.

%% Variables del GE: q=[X2 Y2 X3 Y3 X4 Y4]. Las separamos en:
%% Dependientes: qG=[X4 Y4], coordenadas de P4.
%% Independientes: h=[X2 Y2 X3 Y3], coordenadas de P2 (de bastidor) y P3.

%% Identificación de variables, puntos y longitudes.

numpoi = MGrupos(ng).poi(1,1); % Id del POI P4.
Solido(1:2,1) = MGrupos(ng).eslabones(1,1:2); % Identifica eslabones 3 y 4 de GE.

pref3 = MGrupos(ng).poi(1,2); % Punto de referencia P3.
pref2 = MGrupos(ng).poi(2,2); % Punto de referencia P2.

%% Las longitudes entre puntos de modelo son:
L1 = sqrt((MGrupos(ng).poi(1,4))^2 + MGrupos(ng).poi(1,5)^2);
L2 = sqrt((MGrupos(ng).poi(2,4))^2 + MGrupos(ng).poi(2,5)^2);

X3 = POI(pref3).PVA(1);
Y3 = POI(pref3).PVA(2);
X4 = POI(numpoi).PVA(1);
Y4 = POI(numpoi).PVA(2);
X2 = POI(pref2).PVA(1);
Y2 = POI(pref2).PVA(2);

qgrupo(1:2,1) = [X4 Y4];

```

Cuadro 3.8: Fichero del grupo estructural *3RF*; esquema e identificación de variables.

```

%% Problema de posición: Newton-Raphson.
% mFiG, ecuaciones de restricción.
% FiqG, Jacobiana de mFiG respecto de coordenadas dependientes de GE.
i=0; tol = 1e-8; Res = 20; % Asignamos valores distintos de '0' para comenzar.

while Res >tol % Criterio de parada.

mFiG(1:2,1) = [(X4-X3)^ 2 + (Y4-Y3)^ 2 - L1^ 2
              (X4-X2)^ 2 + (Y4-Y2)^ 2 - L2^ 2];

FiqG = [2*(X4-X3) 2*(Y4-Y3)
        2*(X4-X2) 2*(Y4-Y2)];

qgrupo1 = qgrupo - mFiG/FiqG;
qgrupo = qgrupo1;
X4 = qgrupo(1); Y4 = qgrupo(2);

Res = norm(mFiG);
end

end

POI(numpoi).PVA(1) = X4;
POI(numpoi).PVA(2) = Y4;

```

% Observación. Si hay otros POI's se actualizan ahí. La posición % del punto adicional se da relativa al punto fijo si está en el eslabón % con punto fijo o respecto al externo si está en el móvil.

Cuadro 3.9: Fichero del grupo estructural *3RF*; problema de posición.

```

%% Problema de velocidad.
% h es [x3 y3 x2 y2].
% Fih, Jacobiana de mFiG respecto h.
% Fihhp, vector producto Fih y velocidades de h.

X3p = POI(pref3).PVA(1,3);
Y3p = POI(pref3).PVA(1,4);

Fihhp = [-2*(X4-X3)*X3p-2*(Y4-Y3)*Y3p
         0];
POI(numpoi).PVA(3:4) = -Fihhp/FiqG;

X4p = POI(numpoi).PVA(1,3);
Y4p = POI(numpoi).PVA(1,4);

```

Cuadro 3.10: Fichero del grupo estructural *3RF*; problema de velocidad.

```

% % Problema de aceleración.
% Fihhs, vector producto Fih y aceleraciones de h.
% Fiqp, derivada temporal de Jacobiana de mFiG respecto de todas las variables de GE.
% Fiqpqp, vector producto Fiqp y velocidades de todas las variables de GE.

Fiqpqp = 2*[(X4p-X3p) ^ 2+(Y4p-Y3p) ^ 2
            X4p ^ 2+Y4p ^ 2];

X3s = POI(pref3).PVA(1,5);
Y3s = POI(pref3).PVA(1,6);

Fihhs = -2*[(X4-X3)*X3s+(Y4-Y3)*Y3s
            0];

POI(numpoi).PVA(5:6) = -(Fiqpqp+Fihhs)/FiqG;

```

Cuadro 3.11: Fichero del grupo estructural *3RF*; problema de aceleración.

### 3.3.2.2. Fichero de geometría *MBDatosSuspension5ejesB.m*

Éste es el fichero al que accede el raíz *MBSuspension5ejes.m*. Su finalidad es definir la geometría de todo el sistema de suspensión en función de  $m$  ejes. Por su gran extensión se divide en trece partes, Cuadros 3.15-3.27.

Al principio este fichero fue creado para definir la geometría relativa a un solo eje. Después se ha automatizado para  $m$  ejes. Esto se logra con una estructura *for*, junto con una condicional y una serie de parámetros aquí destacadas: *distEje*, *np*, *nv*, *ngEje*, *nSol* y *cont*.

- *distEje*: Se usa para establecer la distancia variable entre ejes.
- *np*: Se usa para establecer el identificador de los POI's. Para un primer eje hay 19 POI's, por ello en cada eje se suma esta cantidad al identificador.
- *nv*: Se usa para establecer el identificador de los VEC's. Para un primer eje hay 31 VEC's, donde la posición 28 y 29 está libre. Por ello en cada eje se suma esta cantidad al identificador.
- *ngEje*: Se usa para definir en función de  $m$  el adecuado grupo estructural.
- *nSol*: Se usa para definir en función de  $m$  el adecuado sólido o eslabón.
- *cont*: Es el contador que determina las pasadas por el fichero en función de  $m$  ejes.

En el fichero se define al principio el sistema de referencia global, origen y direcciones de su triedro (*numPoi* 19 y *numVec* 1+*nv*, 30 y 31). Se establecen los eslabones de grupo y luego la especificación de la geometría de cada grupo. En dicha especificación, para cada punto, vector, ángulo o distancia del sistema global se le da una numeración de sistema local, que utiliza después el fichero de resolución de grupo. Además se determinan los triedros locales de cada eslabón y sus inversas, necesarios para la simulación espacial y el cálculo de otros POI's, respectivamente. También se definen otros parámetros como *coef*, *coef79* y *coef68*, necesarios para la resoluciones de grupo.

Para llevar a cabo todo ello se utilizan las estructuras de datos similares a las comentadas en el programa del cuadrilátero articulado. Son las siguientes:

- *MGrupos*(*ng*).*poi*(:,: ,*n°cuerpo*): Determina puntos de interés relativos al modelo de un grupo estructural *ng*. Para un grupo, tiene por columnas un identificador global, el punto de referencia del POI, el eslabón al que pertenece, y las coordenadas locales (respecto al punto de referencia) y globales; *numPoi*, *Pref*, *Estabptnce*, *xyzlocal* y *xyzglobal*, respectivamente. Además un mismo POI puede venir referido a dos cuerpos distintos, por ello se incluye un tercer parámetro en esta estructura de datos.

```

%% Otros POI:
if size(MGrupos(ng).poi,1) >2
    for npto = 3: size(MGrupos(ng).poi,1)
        if MGrupos(ng).poi(npto,3) == Solido(1)
            Rotacion = [X4-X3 Y3-Y4
                        Y4-Y3 X4-X3];
            Rotacion = (1/L1)*Rotacion;
        else
            Rotacion = [X4-X2 Y2-Y4
                        Y4-Y2 X4-X2];
            Rotacion = (1/L2)*Rotacion;
        end
    P0I(MGrupos(ng).poi(npto,1)).PVA(1:2) = P0I(MGrupos(ng).poi(npto,2)).PVA(1:2) +
    + Rotacion * MGrupos(ng).poi(npto,4:5)';
end
X4s = P0I(numpoi).PVA(1,5);
Y4s = P0I(numpoi).PVA(1,6);
for npto = 3: size(MGrupos(ng).poi,1)
    if MGrupos(ng).poi(npto,3) == Solido(1)
        Rotacionp = [X4p-X3p Y3p-Y4p
                     Y4p-Y3p X4p-X3p];
        Rotacionp = (1/L1)*Rotacionp;
        Rotacions = [X4s-X3s Y3s-Y4s
                     Y4s-Y3s X4s-X3s];
        Rotacions = (1/L1)*Rotacions;
    else
        Rotacionp = [X4p -Y4p
                     Y4p X4p];
        Rotacionp = (1/L2)*Rotacionp;
        Rotacions = [X4s -Y4s
                     Y4s X4s];
        Rotacions = (1/L2)*Rotacions;
    end
    P0I(MGrupos(ng).poi(npto,1)).PVA(3:4)=P0I(MGrupos(ng).poi(npto,2)).PVA(3:4) +
    + Rotacionp * MGrupos(ng).poi(npto,4:5)';
    P0I(MGrupos(ng).poi(npto,1)).PVA(5:6)=P0I(MGrupos(ng).poi(npto,2)).PVA(5:6) +
    + Rotacions * MGrupos(ng).poi(npto,4:5)';
end
end
end

```

Cuadro 3.12: Fichero del grupo estructural 3RF; cálculo de otros POI's.

```

%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %%
%% SISTEMA DE SUSPENSIÓN DEL CAMIÓN %%
%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %%

addpath SolucionGrupos
delete Suspension.out
clear all

clc
format long

% Valores de los gdl del sistema. Es sistema de suspensión tiene como
% entrada el desplazamiento vertical en el eje de la rueda. Serán las
% variables Z15 y Z16, para m=1, entrada en el eslabón 2.

m=5; % Número de ejes.
Z150 = zeros(1,m);
Z160 = zeros(1,m);

Z15p = [0.0 0 0.03 0.013 0.013]; %mm/s
Z16p = [0.01 0.01 0 0.01 0];
Z15s = zeros(1,m); %mm/s^2
Z16s = zeros(1,m);

if m==1; ngf=6;
elseif m==2; ngf=11;
elseif m==3; ngf=16;
elseif m==4; ngf=21;
else ngf=26;
end

```

MBDatosSuspension5EjesB

Cuadro 3.13: Fichero raíz del sistema de suspensión del camión (primera parte).

```

t=0;
incr=0.005;
cont=0;
cont2=1;

for i=0:incr:20
    cont=cont+1;
    for k=1:m
        Z15(k) = Z150(k)+i*Z15p(k);
        Z16(k) = Z160(k)+i*Z16p(k);
    end

    save('C:Suspension.out','t','-ascii','-double','-append')

for ng = 2:ngf;
    if (ng==2 || ng==7 || ng==12 || ng==17 || ng==22)

        if ng==2; k=1;
            elseif ng==7; k=2;
                elseif ng==12; k=3;
                    elseif ng==17; k=4;
                        elseif ng==22; k=5;
                            end

        Gd1(1,:)=[Z15(k) Z15p(k) Z15s(k)];
        Gd1(2,:)=[Z16(k) Z16p(k) Z16s(k)];
        [POI,VEC,MGrupos]=SolucionGrupo2345C(Gd1,ng,POI,VEC,MGrupos,coef);

    elseif (ng==3 || ng==8 || ng==13 || ng==18 || ng==23)

        [POI,VEC,ANG,MGrupos]=SolucionGrupo79(Gd1,ng,POI,VEC,ANG,MGrupos,coef79);

    elseif (ng==4 || ng==9 || ng==14 || ng==19 || ng==24)

        [POI,VEC,ANG,MGrupos]=SolucionGrupo79(Gd1,ng,POI,VEC,ANG,MGrupos,coef68);

        % Cálculo del ángulo relativo de la barra a torsión.
        AngTor67(cont2,cont)=abs(ANG(MGrupos(ng)).ang(1,1)).PVA
            - ANG(MGrupos(ng-1)).ang(1,1)).PVA(1);

        cont2=cont2+1;

    elseif (ng==5 || ng==6 || ng==10 || ng==11 || ng==15 || ng==16 || ng==20 ||
        ng==21 || ng==25 || ng==26)

        [POI,VEC,DIST,MGrupos]=SolucionGrupo1213_b(Gd1,ng,POI,VEC,DIST,MGrupos);

    end

end
t=t+incr;
cont2=1;

end

```

Cuadro 3.14: Fichero raíz del sistema de suspensión del camión (segunda parte).

- `MGrupos(ng).vec`: Determina los vectores relativos al modelo de un grupo estructural `ng`. Tiene las mismas columnas que la estructura de datos anterior, identificador global, el punto de referencia del `VEC`, el eslabón al que pertenece y las coordenadas locales y globales. Un `VEC` no viene referido a dos cuerpos distintos, por ello no se incluye el parámetro al que se dotó `MGrupos(ng).poi`.
- `MGrupos(n°+ngEje).eslabones`: Determina los eslabones que pertenecen al grupo estructural `n°+ngEje`. Se utiliza ese parámetro y no `ng` por simplicidad.
- `MGrupos(ng).ang`: Determina ángulos relativos al modelo de un grupo estructural `ng`. Para un grupo, tiene por columnas un identificador global, el vector de inicio y fin de la medida, el eje de rotación y el valor de partida; `numAng`, `vec.inic`, `vec.fin`, `eje.rot` y `valor`, respectivamente.
- `MGrupos(ng).dist`: Determina distancias de interés relativos al modelo de un grupo estructural `ng`. Para un grupo, tiene por columnas un identificador global, el punto de inicio y fin de la medida, y el valor inicial; `numDis`, `pto.inic`, `pto.fin` y `valor.inicio`, respectivamente.
- `MGrupos(ng).triidroLoc(:, : , n°cuerpo)`: Determina el triidro local relativo a cada eslabón o cuerpo, necesario para la simulación espacial del mecanismo. El eslabón al que hace referencia el triidro viene definido por el tercer parámetro.
- `MGrupos(ng).invTriidroLoc(:, : , n°cuerpo)`: Determina la inversa del triidro local `MGrupos(ng).triidroLoc(:, : , n°cuerpo)`. Se utiliza para determinar otros puntos de interés en los ficheros que resuelven los grupos estructurales.
- `POI(numPoi).PVA`: Es la estructura de datos `POI` global, que en principio recibe todas las posiciones iniciales y después, cuando se llama el fichero que resuelve el grupo, recibe las posiciones, velocidades y aceleraciones de todos los `POI`'s.
- `VEC(numVec).PVA`: Es la estructura de datos `VEC` global, que en principio recibe todas las posiciones iniciales y después, cuando se llama el fichero que resuelve el grupo, recibe las posiciones, velocidades y aceleraciones de todos los `VEC`'s.
- `ANG(numAng).PVA`: Es la estructura de datos `ANG` global, que en principio recibe todas las posiciones iniciales y después, cuando se llama el fichero que resuelve el grupo, recibe las posiciones, velocidades y aceleraciones de todos los `ANG`'s.
- `DIST(numDis).PVA`: Es la estructura de datos `DIST` global, que en principio recibe todas las posiciones iniciales y después, cuando se llama el fichero que resuelve el grupo, recibe las posiciones, velocidades y aceleraciones de todos los `DIST`'s.

Una vez precisado todo lo referente a cada grupo, se guardan los valores iniciales definidos en las estructuras de datos globales, citadas arriba. Las velocidades y aceleraciones de estas estructuras son puestas a cero. Todo esto se logra con las estructuras `for` que se encuentran al final del fichero.

### 3.3.2.3. Ficheros de solución de grupos

Los ficheros que resuelven los grupos estructurales son llamados por el fichero raíz en su debido tiempo. Estos reciben la matriz `GdL`, el grupo `ng`, las estructuras de datos `POI`, `VEC`, `DIST`, `ANG` y `MGrupos`, además de otras matrices adicionales constantes. Devuelven `POI`, `VEC`, `DIST` y `ANG` con los resultados cinemáticos.

Las matrices de adicionales constantes son necesarias para la resolución de los grupos estructurales. Son los coeficientes que aparecen cuando en un modelo de grupo un `POI` se expresa como combinación lineal del triidro local de dicho grupo. Son calculados aparte y por eso aparecen como constantes en `MBDatosSuspension5ejesB.m`

Debido a que los ficheros `SolucionGrupo2345C`, `SolucionGrupo79` y `SolucionGrupo1213_b` poseen estructuras más o menos semejantes, se explica en este apartado de manera genérica, incidiendo en situaciones específicas para cada grupo.

- El fichero `SolucionGrupo2345C` está formado por los Cuadros 3.28-3.38.

```

%% %% %% %% %% %% %% %%
%% %% MBDATOS %% %%
%% %% %% %% %% %% %% %%
%% Origen de coordenadas globales.

% Definimos el origen global en el punto P19.
MGrupos(1).poi(8,:) = [19 0 1 0 0 0 0];

distEje=0;
np=0;
nv=0;
nda=0;
ngEje=0;
nSol=0;
mult=1;
cte1=0;
cte2=0;
cte3=0;
cte4=0;
cte5=0;
cte6=0;

```

Cuadro 3.15: Fichero de geometría del sistema de suspensión del camión (primera parte).

- El fichero `SolucionGrupo79` está formado por los Cuadros 3.39-3.43.
- El fichero `SolucionGrupo1213_b` está formado por los Cuadros 3.44-3.48.

### Identificación de constantes y variables de este grupo

Todos los ficheros comienzan con este apartado, específico para cada grupo. En él se identifican las coordenadas dependientes e independientes de grupo. Para cada una de ellas se asignan los valores iniciales procedentes de las estructuras globales `POI` y `VEC`, que fueron llenadas previamente en el fichero `MBDatosSuspensi on5ejesB.m`. En el caso de tener como coordenadas independientes grados de libertad, como en el *Grupo* 2, 3, 4, 5, se asignan los `GDL` a `POI`, y de ellos a las coordenadas de grupo.

Después se define `qgrupo`, vector de coordenadas dependientes de grupo, `Fig`, Jacobiana respecto de todas las coordenadas de grupo, y los índices correspondientes a cada coordenada. Esto es, en un vector de coordenadas de grupo, cada punto o vector tendrá un índice, que determine la primera coordenada asociada a un punto (`idpt`) o vector (`idvc`). En el apartado siguiente quedará aclarada su utilidad.

### Problema de posición

Todos los ficheros resuelven este problema. Inicialmente se define la variable `id`, que hace referencia a las filas de las ecuaciones de restricción, y también se establece el criterio de parada para el algoritmo que resuelve el problema de posición. Cuando converge se extraen las coordenadas dependientes, que pasan a las variables de grupo y de ahí a `POI` o `VEC`. Hasta ahí todo coincide más o menos con los ficheros que resolvían los grupos estructurales del cuadrilátero escalable. La novedad se encuentra en la obtención de las ecuaciones de restricción y la Jacobiana respecto de las variables dependientes.

Básicamente lo que se hace es, a través de funciones auxiliares de par o de sólido rígido, con orden en cuanto a `id`, obtener evaluadas sus ecuaciones de restricción `mFi` y Jacobiana `Jacob`. Después se mete `mFi` en `Fig` y `Jacob` en `Fig`. Los índices se utilizan para identificar en las columnas correspondientes la Jacobiana que proporciona la función auxiliar en la Jacobiana del grupo.



```
cont=1; % Contador que define las pasadas por MBGrupos. Se incrementa al final.
for k=1:m; % Recorrido para los m ejes.
if cont==2
    distEje=4.640;
    np=19; nv=31;
    nda=2;
    nSol=12;
    ngEje=5;
elseif cont==3
    distEje=6.660;
    np=2*19; nv=2*31;
    nda=2*2;
    nSol=2*12;
    ngEje=2*5;
    mult=-1;
    cte1=2*0.131;
    cte2=2*0.0004;
    cte3=2*0.4727;
    cte4=2*0.889;
    cte5=2*0.917;
    cte6=2*0.833;
elseif cont==4
    cte1=0;
    cte2=0;
    cte3=0;
    cte4=0;
    cte5=0;
    cte6=0;
    distEje=11.73465;
    np=3*19; nv=3*31;
    nda=3*2;
    nSol=3*12;
    ngEje=3*5;
    mult=1;
elseif cont==5
    distEje=13.73465;
    np=4*19; nv=4*31;
    nda=4*2;
    nSol=4*12;
    ngEje=4*5;
    mult=-1;
    cte1=2*0.131;
    cte2=2*0.0004;
    cte3=2*0.4727;
    cte4=2*0.889;
    cte5=2*0.917;
    cte6=2*0.833;
end
```

% Para un eje hay 19 pois y 31 vec. Por eso se añaden  
% esos valores a la numeración de un eje. Con los esla  
% bones y grupos pasa más menos lo mismo. Siguiente eje  
% al primero es grupo 7, por eso al grupo 2 se suma 5.  
% El siguiente eslabón es 14, por eso al 2 se suma 12.

Cuadro 3.16: Fichero de geometría del sistema de suspensión del camión (segunda parte).

```

%% Coordenadas de vectores sobre bastidor.
% numVec Pref Eslabptnce x y z loc/glob
MGrupos(1).vec(1,:) = [1+nv 19 1 mult*1 0 0]; % V1 bastidor, compare con eslabón 3.
MGrupos(1).vec(2,:) = [30 19 1 0 1 0];
MGrupos(1).vec(3,:) = [31 19 1 0 0 1];

%% Eslabones.
MGrupos(1).eslabones = 1;
MGrupos(2+ngEje).eslabones = [2+nSol 3+nSol 4+nSol 5+nSol];
MGrupos(3+ngEje).eslabones = [7+nSol 9+nSol];
MGrupos(4+ngEje).eslabones = [6+nSol 8+nSol];
MGrupos(5+ngEje).eslabones = [10+nSol 11+nSol];
MGrupos(6+ngEje).eslabones = [12+nSol 13+nSol];

```

Cuadro 3.17: Fichero de geometría del sistema de suspensión del camión (tercera parte).

Las ecuaciones de restricción del eslabón 2 incluido en `SolucionGrupo2345C` son las seis del triedro, correspondientes a las seis primeras filas. Se usa la función `Triedro1P3V(P1, V4, V3, V5)`, definida para que a partir del origen y direcciones del triedro, proporcione las restricciones evaluadas y la Jacobiana de éstas respecto de las coordenadas variables asociadas. Luego se mete `mFi` en `FiG`, seis primeras filas, y `Jacob` en `Fiq`, en las seis primeras filas y columnas correspondientes a las coordenadas del vector 4, 3 y 5 en el sistema local.

Obtenidas la Jacobiana, respecto de todas las coordenadas de grupo, se crea la Jacobiana respecto de las coordenadas dependientes, extrayendo columnas adecuadas, también utilizando índices.

A continuación describen las funciones auxiliares utilizadas para resolver el problema de posición.

- `Triedro1P3V(P1, V1, V2, V3)`: Esta función auxiliar, Cuadro 3.49, devuelve las ecuaciones de restricción y la contribución a la Jacobiana, derivada respecto de todas las variables de grupo estructural, de sólido rígido (ecuaciones del triedro) formado por un punto P1 y tres vectores unitarios V1, V2 y V3.
- `Triedro1P3VC(P1, V1, V2, V3)`: Esta función auxiliar, Cuadro 3.50, devuelve las ecuaciones de restricción y la contribución a la Jacobiana, derivada respecto de todas las variables de grupo estructural, de sólido rígido (ecuaciones del triedro) formado por un punto P1 y tres vectores unitarios V1, V2 y V3. Es un caso especial al anterior ya que V1 es compartido por otro grupo.
- `Triedro2PF2VF(P1, P2, V1, V2, dist)`: Esta función auxiliar, Cuadro 3.51, devuelve las ecuaciones de restricción y la contribución a la Jacobiana, derivada respecto de todas las variables de grupo estructural, de sólido rígido (ecuaciones del triedro) formado por dos puntos, P1 y P2, y dos vectores unitarios, V1 y V2. El punto P1 y vector V1 son fijos por pertenecer al bastidor, mientras que P2 y V2 son móviles.
- `TrdPFVW(P1, P2, V1, V2, dist)`: Esta función auxiliar, Cuadro 3.52, devuelve las ecuaciones de restricción y la contribución a la Jacobiana, derivada respecto de todas las variables de grupo estructural, de sólido rígido (ecuaciones del triedro) formado por dos puntos, P1 y P2, y dos vectores unitarios, V1 y V2. El punto P1 es fijo por pertenecer al bastidor, mientras que P2, V1 y V2 son móviles.
- `CCL1P3V(P2, P1, V1, V2, V3, coef)`: Esta función auxiliar, Cuadro 3.53, devuelve las ecuaciones de restricción de combinación lineal de un punto P2 respecto a un sólido rígido, definido por P1 y el triedro V1, V2 y V3, además de la contribución a la Jacobiana, derivada respecto de todas las variables de grupo estructural.
- `Cardan(V1, V2)`: Esta función auxiliar, Cuadro 3.54, devuelve las ecuaciones de restricción de cardan, relativas a los vectores V1 y V2, y las contribuciones a la Jacobiana respecto de todas las variables de grupo estructural.

```

%% Puntos y vectores del grupo 2 {2,3,4,5}.
ng=2+ngEje;

% Siguen la estructura: Poi:(1,2,3,4,5,6,7,8,9,10,11,12) / Vec:(1,2,3,4,5,6,7,8,9).
% Para este grupo son: Poi: (14,8,9,10,11,12,13,15,16,1,3,2) / Vec(1,2,8,9,7,5,6,3,4).

% Se consideran:
% Cuerpos 1, 2, 3 y 4 = sólidos 2, 3, 4 y 5.

MGrupos(ng).poi=zeros(12,9);
MGrupos(ng).vec=zeros(9,9);

% Los puntos y vectores del GE.
%fil/col/sol% numPoi/Vec Pref Eslabptnce x y z local x y z global
MGrupos(ng).poi(1, :, 1) = [14+np 14+np 2+nSol 0 -0.0012 0 mult*0 0+distEje
0.265];
MGrupos(ng).poi(1, :, 2) = [14+np 10+np 3+nSol 0 -0.889 0.003727 mult*0 0+distEje
0.265];

MGrupos(ng).vec(4, :) = [9+nv 14+np 2+nSol 1 0 0 mult*1 0 0 ];
MGrupos(ng).vec(3, :) = [8+nv 14+np 2+nSol 0 1 0 0 mult*1 0 0 ];
MGrupos(ng).vec(5, :) = [7+nv 14+np 2+nSol 0 0 1 0 0 0 1 ];

MGrupos(ng).poi(2, :, 1) = [8+np 14+np 2+nSol 0.758 0.131 -0.265 mult*0.76
0.12+distEje-cte1 -0.265+0.265 ];
MGrupos(ng).poi(2, :, 3) = [8+np 2+np 4+nSol 0 -0.8 0 mult*0.76 0.12+distEje-cte1
-0.265+0.265 ];
MGrupos(ng).poi(3, :, 1) = [9+np 14+np 2+nSol -0.758 0.131 -0.265 mult*-0.76
0.12+distEje-cte1 -0.265+0.265];
MGrupos(ng).poi(3, :, 4) = [9+np 3+np 5+nSol 0 -0.8 0 mult*-0.76 0.12+distEje-cte1
-0.265+0.265 ];

MGrupos(ng).poi(4, :, 1) = [10+np 14+np 2+nSol 0.7373 -0.0004 -0.134 mult*0.7392
-0.0022+distEje+cte2 0.114];
MGrupos(ng).poi(5, :, 1) = [11+np 14+np 2+nSol -0.7373 -0.0004 -0.134 mult*-0.7392
-0.0022+distEje+cte2 0.114];
MGrupos(ng).poi(6, :, 1) = [12+np 14+np 2+nSol 0.6246 -0.4727 -0.4133 mult*0.6246
-0.4727+distEje+cte3 -0.425+0.265];
MGrupos(ng).poi(7, :, 1) = [13+np 14+np 2+nSol -0.6246 -0.4727 -0.4133
mult*-0.6246 -0.4727+distEje+cte3 -0.425+0.265];
MGrupos(ng).poi(8, :, 1) = [15+np 14+np 2+nSol 1.202 0 -0.265 mult*1.202 0+distEje
-0.265+0.265];
MGrupos(ng).poi(9, :, 1) = [16+np 14+np 2+nSol -1.202 0 -0.265 mult*-1.202
0+distEje -0.265+0.265];

MGrupos(ng).poi(10, :, 2) = [1+np 19 1 0 0 0.003727 mult*0 0.889+distEje-cte4
0.265];
MGrupos(ng).poi(12, :, 3) = [2+np 19 1 0 0 0 mult*0.76 0.917+distEje-cte5 0];
MGrupos(ng).poi(11, :, 4) = [3+np 19 1 0 0 0 mult*-0.76 0.917+distEje-cte5 0];

MGrupos(ng).poi(10, :, 1) = [1+np 19 1 0 0 0.003727 mult*0 0.889+distEje-cte4
0.265];
MGrupos(ng).poi(12, :, 1) = [2+np 19 1 0 0 0 mult*0.76 0.917+distEje-cte5 0];
MGrupos(ng).poi(11, :, 1) = [3+np 19 1 0 0 0 mult*-0.76 0.917+distEje-cte5 0 ];

```

Cuadro 3.18: Fichero de geometría del sistema de suspensión del camión (cuarta parte).

```

MGrupos(ng).vec(1,:) = [1 19 1 1 0 0 mult*1 0 0];
MGrupos(ng).vec(2,:) = [2+nv 1+np 3+nSol 0 0 1 0 0 1];
MGrupos(ng).vec(6,:) = [5+nv 3+np 5+nSol 1 0 0 mult*1 0 0];
MGrupos(ng).vec(7,:) = [6+nv 3+np 5+nSol 0 0 1 0 0 1];
MGrupos(ng).vec(8,:) = [3+nv 2+np 4+nSol 1 0 0 mult*1 0 0];
MGrupos(ng).vec(9,:) = [4+nv 2+np 4+nSol 0 0 1 0 0 1];

% Escribimos matrices triedro local y sus inversas.

MGrupos(ng).triedroLoc(1:3,1:3,1)=zeros; % Matrices de 3x3 para cuatro sólidos del grupo.
MGrupos(ng).triedroLoc(1:3,1:3,2)=zeros;
MGrupos(ng).triedroLoc(1:3,1:3,3)=zeros;
MGrupos(ng).triedroLoc(1:3,1:3,4)=zeros;

MGrupos(ng).triedroLoc(:,1)= [MGrupos(ng).vec(4,4:6)
MGrupos(ng).vec(3,4:6)
MGrupos(ng).vec(5,4:6)]; % La del sólido 2.

% Vnorm=norm(MGrupos(ng).poi(10,4:6,2)-MGrupos(ng).poi(1,4:6,2));

MGrupos(ng).triedroLoc(:,2)= [MGrupos(ng).vec(1,4:6)
(MGrupos(ng).poi(1,4:6,2)-MGrupos(ng).poi(10,4:6,2)) % Vnorm
MGrupos(ng).vec(2,4:6)]; % La del sólido 3.

% Vnorm=norm(MGrupos(ng).poi(12,4:6,3)-MGrupos(ng).poi(2,4:6,3));

MGrupos(ng).triedroLoc(:,3)= [MGrupos(ng).vec(8,4:6)
(MGrupos(ng).poi(2,4:6,3)-MGrupos(ng).poi(12,4:6,3)) % Vnorm
MGrupos(ng).vec(9,4:6)]; % La del sólido 4.

% Vnorm=norm(MGrupos(ng).poi(11,4:6,4)-MGrupos(ng).poi(3,4:6,4));

MGrupos(ng).triedroLoc(:,4)= [MGrupos(ng).vec(6,4:6)
(MGrupos(ng).poi(3,4:6,4)-MGrupos(ng).poi(11,4:6,4)) % Vnorm
MGrupos(ng).vec(7,4:6)]; % La del sólido 5.

MGrupos(ng).invTriedroLoc(:,1) = inv(MGrupos(ng).triedroLoc(:,1));
MGrupos(ng).invTriedroLoc(:,2) = inv(MGrupos(ng).triedroLoc(:,2));
MGrupos(ng).invTriedroLoc(:,3) = inv(MGrupos(ng).triedroLoc(:,3));
MGrupos(ng).invTriedroLoc(:,4) = inv(MGrupos(ng).triedroLoc(:,4));

```

Cuadro 3.19: Fichero de geometría del sistema de suspensión del camión (quinta parte).

```
% Matriz de coeficientes para el grupo 2, necesaria para resolver su
% cinemática del grupo, ec. restricción de CL. Se han despejado
% multiplicando las constantes en el orden V9, V8 y V7.
```

```
coef=[ 0.758 %Parámetros asociados a P8.
       0.131
      -0.265000000000000000
      -0.758 %Parámetros asociados a P9.
       0.131
      -0.265000000000000000
       1.2020000000000000 %Parámetros asociados a P15.
       0
      -0.265000000000000000
      -1.2020000000000000 %Parámetros asociados a P16.
       0
      -0.265000000000000000];
```

Cuadro 3.20: Fichero de geometría del sistema de suspensión del camión (sexta parte).

- **Prismatico(P1,P2,V1,V2,V3):** Esta función auxiliar, Cuadro 3.55, devuelve las ecuaciones de restricción y contribuciones a la Jacobiana, respecto de todas las coordenadas de grupo estructural, del par prismático. Éste ha de estar definido entre P1 y P2, donde V1 define esa dirección, y los vectores V2 y V3, pertenecientes a P1 y P2 respectivamente, son perpendiculares. El punto P1 es un fijo de bastidor. Con esta aportación se crea un sistema redundante, en el caso de querer un sistema linealmente independiente se deberá eliminar por configuración una de las tres primeras ecuaciones.

Además de las anteriores, se definen funciones secundarias, para agilizar la resolución y facilitar el cálculo. Son las siguientes:

- **dotProduct(V1,V3):** Función que devuelve el producto escalar de dos vectores, V1 y V3 (Cuadro 3.56).
- **NormCont(V1):** Función que devuelve el resultado de evaluar la ecuación de restricción de normalidad de un vector (Cuadro 3.57).
- **RigCont(V1,dist):** Función que devuelve el resultado de evaluar la ecuación de restricción de sólido rígido entre dos puntos, formados por el vector V1 (Cuadro 3.58).

### Problema de velocidad

Todos los ficheros resuelven este problema. Por un lado se crea el vector velocidades independientes, por otro se extrae la Jacobiana respecto de las coordenadas independientes Fih, a partir de la matriz Jacobiana respecto de todas las coordenadas de grupo Fiq, definida en el problema anterior. Los valores de las velocidades dependientes se almacenan en qgrupo, y de ahí pasan a las columnas de velocidad de las estructuras de datos globales POI.PVA(4:6) y VEC.PVA(4:6).

### Problema de aceleración

Todos los ficheros resuelven este problema. Por un lado se extraen de POI.PVA(4:6) y VEC.PVA(4:6) los valores de las velocidades y se almacenan en las variables correspondientes, además de definir el vector aceleraciones independientes. Por otro se define Fiqpp a partir de funciones auxiliares. Con todo esto se tendrá lo necesario para ejecutar el algoritmo que proporciona las aceleraciones dependientes.

Es sencillo el procedimiento de creación de Fiqpp, será similar a como se obtuvo mFi y Jacob. Con unas funciones auxiliares se obtiene evaluado el dicho vector, que es organizado en las filas id correspondientes. Las funciones auxiliares utilizadas son las siguientes:

```

%% Puntos y vectores del grupo 3 {7,9}.
ng=3+ngEje;

% Siguen la estructura: Poi(1,2,3); Vec:(1,2,3,4,5).
% Para este grupo son: Poi: (7,13,18) / Vec(23,24,25,9,8).

% Se considera:
% Cuerpo 1 = sólido 7.
% Cuerpo 2 = sólido 9.

MGrupos(ng).poi=zeros(3,9);
MGrupos(ng).vec=zeros(5,9);
MGrupos(ng).ang=zeros(1,5);

%numAng vec.inic vec.fin eje rot. valor
MGrupos(ng).ang(1,:)=[1+nda 8+nv 25+nv 9+nv pi/4]; % Orientación relativa sólido 2 y 7.

% Los nuevos puntos y vectores del GE.

% numPoi/Vec Pref Eslabptnce x y z local x y z Global
MGrupos(ng).poi(3, :, 1) = [18+np 13+np 7+nSol 0 0 0 mult*-0.819
-0.833+distEje+cte6 -0.160];
MGrupos(ng).poi(3, :, 2) = [18+np 7+np 9+nSol 0 0 0.419 mult*-0.819
-0.833+distEje+cte6 -0.160];

MGrupos(ng).vec(1, :) = [23+nv 7+np 9+nSol 0 1 0 0 mult*1 0 0 ];
MGrupos(ng).vec(2, :) = [24+nv 7+np 9+nSol 1 0 0 mult*1 0 0 0 ];
MGrupos(ng).vec(3, :) = [25+nv 18+np 7+nSol 0 0 1 0 0 1 0 ];
MGrupos(ng).vec(5, :) = [26+nv 18+np 7+nSol 0 1 0 0 mult*1 0 0 ];

% Los ya definidos anteriormente.

MGrupos(ng).poi(1, :, 2) = [7+np 7+np 9+nSol 0 0 0 mult*-0.819 -0.833+distEje+cte6
0.261];
MGrupos(ng).poi(1, :, 1) = [7+np 19 1 -0.819 -0.833 0.261 mult*-0.819
-0.833+distEje+cte6 0.261];

MGrupos(ng).poi(2, :, 1) = [13+np 18+np 7+nSol 0.199 0.3606 0.0000323 mult*-0.622
-0.473+distEje+cte3 -0.425+0.265];

MGrupos(ng).vec(4, :) = [ 9+nv 14+np 2+nSol 1 0 0 mult*1 0 0 ];
MGrupos(ng).vec(6, :) = [ 8+nv 14+np 2+nSol 0 1 0 0 mult*1 0 ];

% Matriz de coeficientes para el grupo 3, necesaria para resolver su
% cinemática del grupo; ec. restricción de CL. Se han despejado
% multiplicando las constantes en el orden V4, V5 y V3.

coef79 = [ 0.199000 %Parámetros asociados al P13.
0.3606
0.00003];

```

Cuadro 3.21: Fichero de geometría del sistema de suspensión del camión (séptima parte).

```
% Escribimos matrices triedro local y sus inversas.
MGrupos(ng).triedroLoc(1:3,1:3,1)=zeros;% matrices de 3x3 para cuatro sólidos del grupo.
MGrupos(ng).triedroLoc(1:3,1:3,2)=zeros;

MGrupos(ng).triedroLoc(:,1,1)= [MGrupos(ng).vec(4,4:6)
    MGrupos(ng).vec(5,4:6)
    MGrupos(ng).vec(3,4:6)]; % La del sólido 7.

% Vnorm=norm(MGrupos(ng).poi(3,4:6,2)-MGrupos(ng).poi(1,4:6,2));

MGrupos(ng).triedroLoc(:,1,2)= [MGrupos(ng).vec(2,4:6)
    MGrupos(ng).vec(1,4:6)
    (MGrupos(ng).poi(3,4:6,2)-MGrupos(ng).poi(1,4:6,2))]; % La del sólido 9.

MGrupos(ng).invTriedroLoc(:,1,1) = inv(MGrupos(ng).triedroLoc(:,1,1));
MGrupos(ng).invTriedroLoc(:,1,2) = inv(MGrupos(ng).triedroLoc(:,1,2));
```

Cuadro 3.22: Fichero de geometría del sistema de suspensión del camión (octava parte).

- **FiqpqTrietro1P3V(V1p,V2p,V3p)**: Es la relacionada con las ecuaciones de restricción proporcionadas por la función **Trietro1P3V.m**. Esta función auxiliar, Cuadro 3.59, devuelve la contribución a **Fiqpq** de un triedro formado por un punto y tres vectores, donde **V1p**, **V2p** y **V3p** son las velocidades de dichos vectores.
- **FiqpqTrietro1P3VC(V1p,V2p,V3p)**: Es la relacionada con las ecuaciones de restricción proporcionadas por la función **Trietro1P3VC.m**. Esta función auxiliar, Cuadro 3.60, devuelve la contribución a **Fiqpq** de un triedro formado por un punto y tres vectores, donde uno de los vectores es compartido con otro grupo. Son **V1p**, **V2p** y **V3p** las velocidades del los vectores móviles.
- **FiqpqTrietro2PF2VF(P2p,V2p)**: Es la relacionada con las ecuaciones de restricción proporcionadas por la función **Trietro2PF2VF.m**. Esta función auxiliar, Cuadro 3.61, devuelve la contribución a **Fiqpq** de un triedro formado por dos punto y dos vectores, donde uno de los puntos y vectores son hijos. Son **P2p** y **V2p** las velocidades del punto y vector móvil.
- **FiqpqTrdPFVW(P2p,V1p,V2p)**: Es la relacionada con las ecuaciones de restricción proporcionadas por la función **TrdPFVW.m**. Esta función auxiliar, Cuadro 3.62, devuelve la contribución a **Fiqpq** de un triedro formado por dos punto y dos vectores, donde uno de los puntos es fijo. Son **P2p**, **V1p** y **V2p** las velocidades del punto y vectores móviles.
- **FiqpqCCL1P3V**: Es la relacionada con las ecuaciones de restricción proporcionadas por la función **CCL1P3V.m**. Esta función auxiliar, Cuadro 3.63, devuelve la contribución a **Fiqpq** de la combinación lineal.
- **FiqpqCardan(V1p,V2p)**: Es la relacionada con las ecuaciones de restricción proporcionadas por la función **Cardan.m**. Esta función auxiliar, Cuadro 3.64, devuelve la contribución a **Fiqpq** del cardan, donde **V1p** y **V2p** son las velocidades de los vectores móviles.
- **FiqpqPrismatico(P2p,V1p,V2p,V3p)**: Es la relacionada con las ecuaciones de restricción proporcionadas por la función **Prismatico.m**. Esta función auxiliar, Cuadro 3.65, devuelve la contribución a **Fiqpq** de un par prismático, donde **P2p**, **V1p**, **V2p** y **V3p** son las velocidades de los vectores y punto móviles.

Para finalizar, los valores de las aceleraciones dependientes se almacenan en **qsgrupo**, y de ahí pasan a las columnas de aceleración de las estructuras de datos globales, **POI.PVA(7:9)** y **VEC.PVA(7:9)**.

```

%% Puntos y vectores del grupo 4 {6,8}.
ng=4+ngEje;

% Siguen la estructura: Poi:(1,2,3); Vec:(1,2,3,4,5).
% Para este grupo son: Poi: (6,12,17) / Vec(20,21,22,9,8).

% Se considera:
% Cuerpo 1 = sólido 6.
% Cuerpo 2 = sólido 8.
% Cuerpo 3 = bastidor.

MGrupos(ng).poi=zeros(3,9);
MGrupos(ng).vec=zeros(5,9);
MGrupos(ng).ang=zeros(1,5);

% numAng vec.inic vec.fin eje rot. valor
MGrupos(ng).ang(1,:)=[2+nda 8+nv 22+nv 9+nv pi/4]; % Orientación relativa sólido 2 y 6.

% Los nuevos puntos y vectores del GE.

% numPoi/Vec Pref Eslabptnce x y z local x y z global
MGrupos(ng).poi(3,,:)= [17+np 12+np 6+nSol 0 0 0 mult*0.819
-0.833+distEje+cte6 -0.160];
MGrupos(ng).poi(3,,:)= [17+np 6+np 8+nSol 0 0 0.419 mult*0.819
-0.833+distEje+cte6 -0.160];

MGrupos(ng).vec(1,:) = [20+nv 6+np 8+nSol 0 1 0 0 mult*1 0 0 ];
MGrupos(ng).vec(2,:) = [21+nv 6+np 8+nSol 1 0 0 mult*1 0 0 ];
MGrupos(ng).vec(3,:) = [22+nv 12+np 6+nSol 0 0 1 0 0 1 ];
MGrupos(ng).vec(5,:) = [27+nv 17+np 6+nSol 0 1 0 0 mult*1 0 0 ];

% Los ya definidos anteriormente.

MGrupos(ng).poi(1,,:)= [6+np 6+np 8+nSol 0 0 0 mult*0.819 -0.833+distEje+cte6
0.261];
MGrupos(ng).poi(1,,:)= [6+np 19 1 0.819 -0.833 0.261 mult*0.819
-0.833+distEje+cte6 0.261];

MGrupos(ng).poi(2,,:)= [12+np 14+np 2+nSol -0.199 0.3606 0.000323 mult*0.622
-0.473+distEje+cte6 -0.160];

MGrupos(ng).vec(4,:) = [9+nv 14+np 2+nSol 1 0 0 mult*1 0 0];
MGrupos(ng).vec(6,:) = [8+nv 14+np 2+nSol 0 1 0 0 mult*1 0];

% Matriz de coeficientes para el grupo 4, necesaria para resolver su cinemática del grupo; ec.
restriccion de CL. Se han despejado multiplicando las constantes en el orden V4, V5 y V3.

coef68 = [ -0.199000 %Parámetros asociados a P12.
           0.3606
           0.00003];

```

Cuadro 3.23: Fichero de geometría del sistema de suspensión del camión (novena parte).



```
% Escribimos matrices triedro local y sus inversas.
MGrupos(ng).triedroLoc(1:3,1:3,1)=zeros;% matrices de 3x3 para cuatro sólidos del grupo.
MGrupos(ng).triedroLoc(1:3,1:3,2)=zeros;

MGrupos(ng).triedroLoc(:, :, 1) = [MGrupos(ng).vec(4,4:6)
    MGrupos(ng).vec(5,4:6)
    MGrupos(ng).vec(3,4:6)]; % La del sólido 6.

% Vnorm=norm(MGrupos(ng).poi(3,4:6,2)-MGrupos(ng).poi(1,4:6,2));

MGrupos(ng).triedroLoc(:, :, 2) = [MGrupos(ng).vec(2,4:6)
    MGrupos(ng).vec(1,4:6)
    (MGrupos(ng).poi(3,4:6,2)-MGrupos(ng).poi(1,4:6,2))]; % La del sólido 8.

MGrupos(ng).invTriedroLoc(:, :, 1) = inv(MGrupos(ng).triedroLoc(:, :, 1));
MGrupos(ng).invTriedroLoc(:, :, 2) = inv(MGrupos(ng).triedroLoc(:, :, 2));
```

Cuadro 3.24: Fichero de geometría del sistema de suspensión del camión (décima parte).

### Otros POI's

Sólamante el fichero `SolucionGrupo2345C` resuelve el cálculo de otros puntos de interés. Esto es debido a que al *Grupo* 2, 3, 4, 5 se le añaden grupos, siendo los POI's calculados aquí coordenadas independientes de grupo posteriores.

Para resolver la posición de los cuatro POI's necesarios se calcula primero `MGrupos(ng).triedroGlo(1:3,1:3,1)`, matriz del triedro global del eslabón 2 (para un eje) o `2+ngEje` (para *m* ejes), que multiplicada por la matriz `MGrupos(ng).invTriedroLoc(:, :, 1)`, procedente del fichero de geometría, proporcionará la matriz de transformación `MGrupos(ng).matTrans(:, :, 1)`. Al vector de posición del origen del triedro local se suma el producto de la matriz de transformación por `MGrupos(ng).poi(n°POI,4:6)` y se obtienen las posiciones de los puntos correspondientes.

Para resolver las velocidades y aceleraciones de los cuatro POI's necesarios se opera de manera similar a la comentada. Se calculan las matrices de transformación asociadas a velocidades y aceleraciones, donde `MGrupos(ng).invTriedroLoc(:, :, 1)` no cambia, se multiplican por las velocidades y aceleraciones locales de los POI's, y finalmente se le suman las velocidades o aceleraciones del origen del triedro local.

```

%% Puntos y vectores del grupo 5 {10,11}.
ng=5+ngEje;

%% Siguen la estructura: Poi:(1,2); Vec:(1,2,3,4,5).
%% Para este grupo son: Poi: (4,10) / Vec(10,11,12,13,14)

% Se considera:
% Cuerpo 1 = sólido 10.
% Cuerpo 2 = sólido 11.

MGrupos(ng).poi=zeros(2,9);
MGrupos(ng).vec=zeros(5,9);
MGrupos(ng).dist=zeros(1,4);

% numDis pto.inic pto.fin valor inicio
MGrupos(ng).dist(1,:)= [2+nda 4+np 10+np 0.427 ]; % Distancia par prismático 10 y 11.

% Los nuevos vectores del GE.

% numVec Pref EsLabptnce x y z local x y z global
MGrupos(ng).vec(2,:) = [10+nv 4+np 11+nSol 0 1 0 0 mult*1 0 ];
MGrupos(ng).vec(1,:) = [11+nv 4+np 11+nSol 0 0 1 0 0 1 ];
MGrupos(ng).vec(3,:) = [12+nv 4+np 11+nSol 1 0 0 mult*1 0 0 ];
MGrupos(ng).vec(5,:) = [13+nv 10+np 10+nSol 1 0 0 mult*1 0 0 ];
MGrupos(ng).vec(4,:) = [14+nv 10+np 10+nSol 0 1 0 0 mult*1 0 ];

% Los ya definidos anteriormente.

MGrupos(ng).poi(1,:) = [4+np 19 1 0.7373 0.10 0.5539 mult*0.7343
0.003+distEje+cte2 0.5539];
MGrupos(ng).poi(2,:) = [10+np 10+np 2+nSol 0 0 0 mult*0.7343 0.0004+distEje+cte2
0.114];

MGrupos(ng).vec(6,:) = [8+nv 14+np 2+nSol 0 1 0 0 mult*1 0 ];

% Escribimos matrices triedro local y sus inversas.

MGrupos(ng).triedroLoc(1:3,1:3,1)=zeros; % Matrices de 3x3 para cuatro sólidos del grupo.
MGrupos(ng).triedroLoc(1:3,1:3,2)=zeros;

MGrupos(ng).triedroLoc(:,1)= [MGrupos(ng).vec(5,4:6)
MGrupos(ng).vec(4,4:6)
MGrupos(ng).vec(1,4:6)]'; % La del sólido 10.

MGrupos(ng).triedroLoc(:,2)= [MGrupos(ng).vec(3,4:6)
MGrupos(ng).vec(2,4:6)
MGrupos(ng).vec(1,4:6)]'; % La del sólido 11.

MGrupos(ng).invTriedroLoc(:,1) = inv(MGrupos(ng).triedroLoc(:,1));
MGrupos(ng).invTriedroLoc(:,2) = inv(MGrupos(ng).triedroLoc(:,2));

```

Cuadro 3.25: Fichero de geometría del sistema de suspensión del camión (undécima parte).

```

%% Puntos y vectores del grupo 6 {12,13}.
ng=6+ngEje;

% Siguen la estructura: Poi:(1,2); Vec:(1,2,3,4,5).
% Para este grupo son: Poi: (5,11) / Vec(16,15,17,19,18)

% Se considera.
% Cuerpo 1 = sólido 12.
% Cuerpo 2 = sólido 13.

MGrupos(ng).poi=zeros(2,9);
MGrupos(ng).vec=zeros(5,9);
MGrupos(ng).dist=zeros(1,4);

% numDis pto.inic pto.fin valor inicio
MGrupos(ng).dist(1,:)= [1+nda 5+np 11+np 0.427 ]; % Distancia par prismático 12 y 13.

% Los vectores dependientes de GE.

% numVec Pref Eslabptnce x y z local x y z global
MGrupos(ng).vec(2,:) = [15+nv 5+np 13+nSol 0 1 0 0 mult*1 0 ];
MGrupos(ng).vec(1,:) = [16+nv 5+np 13+nSol 0 0 1 0 0 1 ];
MGrupos(ng).vec(3,:) = [17+nv 5+np 13+nSol 1 0 0 mult*1 0 0 ];
MGrupos(ng).vec(5,:) = [18+nv 11+np 12+nSol 1 0 0 mult*1 0 0 ];
MGrupos(ng).vec(4,:) = [19+nv 11+np 12+nSol 0 1 0 0 mult*1 0 0 ];

% Los puntos ya definidos anteriormente.

MGrupos(ng).poi(1,:) = [5+np 19 1 -0.7373 0.05 0.5539 mult*-0.7373
0.0003+distEje+cte2 0.5539];
MGrupos(ng).poi(2,:) = [11+np 14+np 2+nSol 0 0 0 mult*-0.739 0+distEje+cte2
0.114];

MGrupos(ng).vec(6,:) = [8+nv 14+np 2+nSol 0 1 0 0 mult*1 0 ];

% Escribimos matrices triedro local y sus inversas.

MGrupos(ng).triedroLoc(1:3,1:3,1)=zeros; % Matrices de 3x3 para cuatro sólidos del grupo.
MGrupos(ng).triedroLoc(1:3,1:3,2)=zeros;

MGrupos(ng).triedroLoc(:,1)= [MGrupos(ng).vec(5,4:6)
MGrupos(ng).vec(4,4:6)
MGrupos(ng).vec(1,4:6)]; % La del sólido 12.

MGrupos(ng).triedroLoc(:,2)= [MGrupos(ng).vec(3,4:6)
MGrupos(ng).vec(2,4:6)
MGrupos(ng).vec(1,4:6)]; % La del sólido 13.

MGrupos(ng).invTriedroLoc(:,1) = inv(MGrupos(ng).triedroLoc(:,1));
MGrupos(ng).invTriedroLoc(:,2) = inv(MGrupos(ng).triedroLoc(:,2));

```

Cuadro 3.26: Fichero de geometría del sistema de suspensión del camión (duodécimo parte).

```

%% Completamos valores Globales, a partir de iniciales del triedro + coord locales.
i=0;
for i = 1:size(MGrupos(1).vec,1)
VEC(MGrupos(1).vec(i,1)).PVA(1:3) = MGrupos(1).vec(i,4:6);
VEC(MGrupos(1).vec(i,1)).PVA(4:9) = 0;
end

% Escribimos valores de PVA globales iniciales en matrices, tanto
% poi,vec, ang y dist.
i=0;
for ng = 2:length(MGrupos)
for i = 1:size(MGrupos(ng).poi,1)
POI(MGrupos(ng).poi(i,1,1)).PVA(1:3) = MGrupos(ng).poi(i,7:9);
POI(MGrupos(ng).poi(i,1,1)).PVA(4:9) = 0;
end
for i = 1:size(MGrupos(ng).vec,1)
VEC(MGrupos(ng).vec(i,1)).PVA(1:3) = MGrupos(ng).vec(i,7:9);
VEC(MGrupos(ng).vec(i,1)).PVA(4:9) = 0;
end
for i = 1:size(MGrupos(ng).ang,1)
ANG(MGrupos(ng).ang(i,1)).PVA(1) = MGrupos(ng).ang(i,5);
end
for i = 1:size(MGrupos(ng).dist,1)
DIST(MGrupos(ng).dist(i,1)).PVA(1) = MGrupos(ng).dist(i,4);
end
end
cont=cont+1;
end

```

Cuadro 3.27: Fichero de geometría del sistema de suspensión del camión (décimo tercera parte).

```

function [POI,VEC,MGrupos] = SolucionGrupo2345C(GDL,ng,POI,VEC,MGrupos,coef)

%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %%
%% %% Este script se encarga de resolver el GE{2,3,4,5}.
%% %%
%% %%
%% Se reciben:
%% -z,y sus derivadas, coordenadas independientes globales.
%% -ng, número de grupo estructural.
%% -POI, matriz puntos de interés.
%% -MGrupos, matriz de matrices.

%% Identificación de constantes y de variables de este grupo.
%% Son variables dependientes las coordenadas X Y Z de las entidades siguientes:
%% P1, P2, P3, P8(1:2), P9(1:2), V2, V3, V4, V5, V6, V7, V8, V9 = 3*11+4 = 37.

%% Son variables independientes las coordenadas:
%% P8(3), P9(3) = 2*1 = 2;
%% Hay que considerar que aquí, en el propio grupo, los P1, P2, etc tienen la nomenclatura propia
del grupo, según su esquema, independientemente de cuántos grupos como éste haya. Eso ya de
define en MBDatos.

%% Asignación de variables independientes.
POI(MGrupos(ng).poi(8,1)).PVA(1,3) = GDL(1,1); POI(MGrupos(ng).poi(8,1)).PVA(1,6)
= GDL(1,2); POI(MGrupos(ng).poi(8,1)).PVA(1,9) = GDL(1,3);
POI(MGrupos(ng).poi(9,1)).PVA(1,3) = GDL(2,1); POI(MGrupos(ng).poi(9,1)).PVA(1,6)
= GDL(2,2); POI(MGrupos(ng).poi(9,1)).PVA(1,9) = GDL(2,3);

%% Posiciones de puntos y vectores de GE.
P1 = POI(MGrupos(ng).poi(1,1,1)).PVA(1:3);
P2 = POI(MGrupos(ng).poi(2,1,1)).PVA(1:3);
P3 = POI(MGrupos(ng).poi(3,1,1)).PVA(1:3);
P8 = POI(MGrupos(ng).poi(8,1,1)).PVA(1:3);
P9 = POI(MGrupos(ng).poi(9,1,1)).PVA(1:3);
P10 = POI(MGrupos(ng).poi(10,1,2)).PVA(1:3);
P12 = POI(MGrupos(ng).poi(12,1,3)).PVA(1:3);
P11 = POI(MGrupos(ng).poi(11,1,4)).PVA(1:3);
V1 = VEC(MGrupos(ng).vec(1,1)).PVA(1:3);
V2 = VEC(MGrupos(ng).vec(2,1)).PVA(1:3);
V3 = VEC(MGrupos(ng).vec(3,1)).PVA(1:3);
V4 = VEC(MGrupos(ng).vec(4,1)).PVA(1:3);
V5 = VEC(MGrupos(ng).vec(5,1)).PVA(1:3);
V6 = VEC(MGrupos(ng).vec(6,1)).PVA(1:3);
V7 = VEC(MGrupos(ng).vec(7,1)).PVA(1:3);
V8 = VEC(MGrupos(ng).vec(8,1)).PVA(1:3);
V9 = VEC(MGrupos(ng).vec(9,1)).PVA(1:3);

```

Cuadro 3.28: Identificación de constantes y variables del grupo {2,3,4,5}. Primera parte del fichero de grupo.

```
% Obtención de valores iniciales de las variables (desde POI).  
% qgrupo, coordenadas dependientes del GE.  
qgrupo(1:37,1) = [P1, P2, P3, P8(1), P8(2), P9(1), P9(2), V2, V4, V3, V5, V6,  
V7, V8, V9];  
Fiq = zeros(37,39);  
  
idpt1 = 1; idpt2 = 4; idpt3 = 7; idpt8 = 10; idpt9 = 13; %Cuenta los 39 huecos.  
idvc2 = 16; idvc4 = 19; idvc3 = 22; idvc5 = 25; idvc6 = 28;  
idvc7 = 31; idvc8 = 34; idvc9 = 37;  
  
% Considero P8z y P9z coordenadas independientes, de grupo y globales del sistema, ya que la  
movilidad global es dos y éste es el grupo que primero que se resuelve.  
  
Cuadro 3.29: Identificación de constantes y variables del grupo {2,3,4,5}. Segunda parte del fichero  
de grupo.
```

```

%% Problema de posición.
% mFi, vector de funciones de restricción.
% FigG, matriz jacobiana de mFi respecto a las coordenadas dependientes de grupo (qG).

id=1; tol = 1e-8; n = 0; Res = 20; % Criterio de parada de solución aceptable.
while Res >tol
    n = n+1;

    %% Triedro sólido 2.
    [mFi, Jacob] = Triedro1P3V(P1, V4, V3, V5); % Sólido 2 (1:6);
    Fig(id:id+5,1) = mFi;
    % formamos la jacobiana:
    Fig(id:id+5, idvc4: idvc4+2) = Jacob(:, 1:3);
    Fig(id:id+5, idvc3: idvc3+2) = Jacob(:, 4:6);
    Fig(id:id+5, idvc5: idvc5+2) = Jacob(:, 7:9);
    id = 7;

    %% Triedro sólido 3.
    dist = norm(MGrupos(ng).poi(1,4:6,2) - MGrupos(ng).poi(10,4:6,2));
    [mFi, Jacob] = Triedro2PF2VF(P10, P1, V1, V2, dist); % Sólido 3 (7:11);
    Fig(id:id+4,1) = mFi;
    % formamos la jacobiana:
    Fig(id:id+4, idpt1: idpt1+2) = Jacob(:, 1:3);
    Fig(id:id+4, idvc2: idvc2+2) = Jacob(:, 4:6);
    id = 12;

    %% Triedro sólido 4.
    dist = norm(MGrupos(ng).poi(2,4:6,3) - MGrupos(ng).poi(12,4:6,3));
    [mFi, Jacob] = TrdPFPVV(P12, P2, V8, V9, dist); % Sólido 4 (12:17);
    Fig(id:id+5,1) = mFi;
    % formamos la jacobiana:
    Fig(id:id+5, idpt2: idpt2+2) = Jacob(:, 1:3);
    Fig(id:id+5, idvc8: idvc8+2) = Jacob(:, 4:6);
    Fig(id:id+5, idvc9: idvc9+2) = Jacob(:, 7:9);
    id = 18;

    %% Triedro sólido 5.
    dist = norm(MGrupos(ng).poi(3,4:6,4) - MGrupos(ng).poi(11,4:6,4));
    [mFi, Jacob] = TrdPFPVV(P11, P3, V6, V7, dist); % Sólido 5 (18:23);
    Fig(id:id+5,1) = mFi;
    % formamos la jacobiana:
    Fig(id:id+5, idpt3: idpt3+2) = Jacob(:, 1:3);
    Fig(id:id+5, idvc6: idvc6+2) = Jacob(:, 4:6);
    Fig(id:id+5, idvc7: idvc7+2) = Jacob(:, 7:9);
    id = 24;

    %% CL punto P2.
    [mFi, Jacob] = CCL1P3V(P2, P1, V4, V3, V5, coef(1:3,1)); % CL (24:26);
    Fig(id:id+2,1) = mFi;
    % formamos la jacobiana:
    Fig(id:id+2, idpt2: idpt2+2) = Jacob(:, 1:3);
    Fig(id:id+2, idpt1: idpt1+2) = Jacob(:, 4:6);
    Fig(id:id+2, idvc4: idvc4+2) = Jacob(:, 7:9);
    Fig(id:id+2, idvc3: idvc3+2) = Jacob(:, 10:12);
    Fig(id:id+2, idvc5: idvc5+2) = Jacob(:, 13:15);
    id=27;

```

Cuadro 3.30: Problema de posición del grupo {2,3,4,5}. Tercera parte del fichero de grupo.

```

%% CL punto P3.
[mFi, Jacob] = CCL1P3V(P3, P1, V4, V3, V5, coef(4:6, 1)); % CL (27:29);
FiG(id:id+2, 1) = mFi;
% formamos la jacobiana:
Fiq(id:id+2, idpt3: idpt3+2) = Jacob(:, 1:3);
Fiq(id:id+2, idpt1: idpt1+2) = Jacob(:, 4:6);
Fiq(id:id+2, idvc4: idvc4+2) = Jacob(:, 7:9);
Fiq(id:id+2, idvc3: idvc3+2) = Jacob(:, 10:12);
Fiq(id:id+2, idvc5: idvc5+2) = Jacob(:, 13:15);
id = 30;
%% CL punto P8.
[mFi, Jacob] = CCL1P3V(P8, P1, V4, V3, V5, coef(7:9, 1)); % CL (30:32);
FiG(id:id+2, 1) = mFi;
% formamos la jacobiana:
Fiq(id:id+2, idpt8: idpt8+2) = Jacob(:, 1:3);
Fiq(id:id+2, idpt1: idpt1+2) = Jacob(:, 4:6);
Fiq(id:id+2, idvc4: idvc4+2) = Jacob(:, 7:9);
Fiq(id:id+2, idvc3: idvc3+2) = Jacob(:, 10:12);
Fiq(id:id+2, idvc5: idvc5+2) = Jacob(:, 13:15);
id = 33;
%% CL punto P9.
[mFi, Jacob] = CCL1P3V(P9, P1, V4, V3, V5, coef(10:12, 1)); % CL (33:35);
FiG(id:id+2, 1) = mFi;
% formamos la jacobiana:
Fiq(id:id+2, idpt9: idpt9+2) = Jacob(:, 1:3);
Fiq(id:id+2, idpt1: idpt1+2) = Jacob(:, 4:6);
Fiq(id:id+2, idvc4: idvc4+2) = Jacob(:, 7:9);
Fiq(id:id+2, idvc3: idvc3+2) = Jacob(:, 10:12);
Fiq(id:id+2, idvc5: idvc5+2) = Jacob(:, 13:15);
id = 36;
%% Cardan 2-4.
[mFi, Jacob] = Cardan(V4, V9); % (36);
FiG(id, 1) = mFi;
% formamos la jacobiana:
Fiq(id, idvc4: idvc4+2) = Jacob(:, 1:3);
Fiq(id, idvc9: idvc9+2) = Jacob(:, 4:6);
id = 37;
%% Cardan 2-5.
[mFi, Jacob] = Cardan(V4, V7); % (37);
FiG(id, 1) = mFi;
% formamos la jacobiana:
Fiq(id, idvc4: idvc4+2) = Jacob(:, 1:3);
Fiq(id, idvc7: idvc7+2) = Jacob(:, 4:6);

```

% Obtenemos la Jacobiana respecto a las variables dependientes (jacobDep):

```

jacobDep(:, 1:11) = Fiq(:, idpt1: idpt8+1);
jacobDep(:, 12:13) = Fiq(:, idpt9: idpt9+1);
jacobDep(:, 14:37) = Fiq(:, idvc2: idvc9+2);

```

Cuadro 3.31: Problema de posición del grupo {2,3,4,5}. Cuarta parte del fichero de grupo.



```

% Solución mediante Newton-Raphson.

qgrupo1 = qgrupo - jacobDep \ FiG;
qgrupo = qgrupo1;

P1(1,:) = qgrupo(1:3);
P2(1,:) = qgrupo(4:6);
P3(1,:) = qgrupo(7:9);
P8(1,1:2) = qgrupo(10:11);
P9(1,1:2) = qgrupo(12:13);
V2(1,:) = qgrupo(14:16);
V4(1,:) = qgrupo(17:19);
V3(1,:) = qgrupo(20:22);
V5(1,:) = qgrupo(23:25);
V6(1,:) = qgrupo(26:28);
V7(1,:) = qgrupo(29:31);
V8(1,:) = qgrupo(32:34);
V9(1,:) = qgrupo(35:37);

Res = norm(FiG);
id = 1;
end

% Reescribimos posiciones calculadas en POI.PVA real, el valor obtenido para su id de GE:
% Ej. POI(MGrupos(ng).poi(1,1)).PVA(1:3) ->punto 14 del modelo = P1 del GE.

POI(MGrupos(ng).poi(1,1,1)).PVA(1:3) = P1 ; % Escribe en el POI(14) valores de P1.
POI(MGrupos(ng).poi(2,1,1)).PVA(1:3) = P2 ; % Escribe en el POI(8) valores de P2.
POI(MGrupos(ng).poi(3,1,1)).PVA(1:3) = P3 ; % Escribe en el POI(9) valores de P3.
POI(MGrupos(ng).poi(8,1,1)).PVA(1:3) = P8 ; % Escribe en el POI(15) valores de P8.
POI(MGrupos(ng).poi(9,1,1)).PVA(1:3) = P9 ; % Escribe en el POI(16) valores de P9.
POI(MGrupos(ng).poi(8,1,1)).PVA(1,3) = GDL(1,1) ;
POI(MGrupos(ng).poi(9,1,1)).PVA(1,3) = GDL(2,1) ;
VEC(MGrupos(ng).vec(2,1)).PVA(1:3) = V2; % Escribe en el VEC(2) valores de V2.
VEC(MGrupos(ng).vec(3,1)).PVA(1:3) = V3; % Escribe en el VEC(8) valores de V3.
VEC(MGrupos(ng).vec(4,1)).PVA(1:3) = V4; % Escribe en el VEC(9) valores de V4.
VEC(MGrupos(ng).vec(5,1)).PVA(1:3) = V5; % Escribe en el VEC(7) valores de V5.
VEC(MGrupos(ng).vec(6,1)).PVA(1:3) = V6; % Escribe en el VEC(5) valores de V6.
VEC(MGrupos(ng).vec(7,1)).PVA(1:3) = V7; % Escribe en el VEC(6) valores de V7.
VEC(MGrupos(ng).vec(8,1)).PVA(1:3) = V8; % Escribe en el VEC(3) valores de V8.
VEC(MGrupos(ng).vec(9,1)).PVA(1:3) = V9; % Escribe en el VEC(4) valores de V9.

```

Cuadro 3.32: Problema de posición del grupo {2,3,4,5}. Quinta parte del fichero de grupo.

```

%% Problema de velocidad.
% Fih, matriz jacobiana de las mFi respecto a las coordenadas independientes de grupo (h).
% Fihhp, vector producto de Fih por el vector velocidades independientes de grupo (hp).

Fih(:,1) = Fiq(:,idpt8+2);
Fih(:,2) = Fiq(:,idpt9+2);
hp = [GDL(1,2) GDL(2,2)]';
Fihhp = Fih * hp;
qpgrupo = -jacobDep \ Fihhp;

% Reescribimos velocidades calculadas en POI.PVA real, el valor obtenido para su id de GE:
% Ej. POI(MGrupos(ng).poi(1,1,1)).PVA(4:6) -> punto 14 del modelo = P1 del GE.

POI(MGrupos(ng).poi(1,1,1)).PVA(4:6) = qpgrupo(1:3);
POI(MGrupos(ng).poi(2,1,1)).PVA(4:6) = qpgrupo(4:6);
POI(MGrupos(ng).poi(3,1,1)).PVA(4:6) = qpgrupo(7:9);
POI(MGrupos(ng).poi(8,1,1)).PVA(4:5) = qpgrupo(10:11);
POI(MGrupos(ng).poi(8,1,1)).PVA(6) = GDL(1,2);
POI(MGrupos(ng).poi(9,1,1)).PVA(4:5) = qpgrupo(12:13);
POI(MGrupos(ng).poi(9,1,1)).PVA(6) = GDL(2,2);
VEC(MGrupos(ng).vec(2,1)).PVA(4:6) = qpgrupo(14:16);
VEC(MGrupos(ng).vec(3,1)).PVA(4:6) = qpgrupo(20:22);
VEC(MGrupos(ng).vec(4,1)).PVA(4:6) = qpgrupo(17:19);
VEC(MGrupos(ng).vec(5,1)).PVA(4:6) = qpgrupo(23:25);
VEC(MGrupos(ng).vec(6,1)).PVA(4:6) = qpgrupo(26:28);
VEC(MGrupos(ng).vec(7,1)).PVA(4:6) = qpgrupo(29:31);
VEC(MGrupos(ng).vec(8,1)).PVA(4:6) = qpgrupo(32:34);
VEC(MGrupos(ng).vec(9,1)).PVA(4:6) = qpgrupo(35:37);

```

Cuadro 3.33: Problema de velocidad del grupo {2,3,4,5}. Sexta parte del fichero de grupo.

```

%% Problema de aceleración.
% Filhs, vector producto de Fih por el vector aceleraciones independientes de grupo (hs).
% Fiqp, matriz derivada temporal de la jacobiana de las mFi respecto a todas las coordenadas
de grupo.
% Fiqqp, vector producto de Fiqp por el vector derivada temporal de todas las coordenadas de
grupo.
P1p = POI(MGrupos(ng) ,poi(1,1,1)) .PVA(4:6);
P2p = POI(MGrupos(ng) ,poi(2,1,1)) .PVA(4:6);
P3p = POI(MGrupos(ng) ,poi(3,1,1)) .PVA(4:6);
P8p = POI(MGrupos(ng) ,poi(8,1,1)) .PVA(4:6);
P9p = POI(MGrupos(ng) ,poi(9,1,1)) .PVA(4:6);
P10p = POI(MGrupos(ng) ,poi(10,1)) .PVA(4:6);
P12p = POI(MGrupos(ng) ,poi(12,1)) .PVA(4:6);
P11p = POI(MGrupos(ng) ,poi(11,1)) .PVA(4:6);
V1p = VEC(MGrupos(ng) ,vec(1,1,1)) .PVA(4:6);
V2p = VEC(MGrupos(ng) ,vec(2,1)) .PVA(4:6);
V3p = VEC(MGrupos(ng) ,vec(3,1)) .PVA(4:6);
V4p = VEC(MGrupos(ng) ,vec(4,1)) .PVA(4:6);
V5p = VEC(MGrupos(ng) ,vec(5,1)) .PVA(4:6);
V6p = VEC(MGrupos(ng) ,vec(6,1)) .PVA(4:6);
V7p = VEC(MGrupos(ng) ,vec(7,1)) .PVA(4:6);
V8p = VEC(MGrupos(ng) ,vec(8,1)) .PVA(4:6);
V9p = VEC(MGrupos(ng) ,vec(9,1)) .PVA(4:6);

%% Triedro sólido 2.
% formamos Fiqqp:
Fiqqp(id:id+5,1) = FiqqpTriedro1P3V(V4p, V3p, V5p);
id = 7;
%% Triedro sólido 3.
% formamos Fiqqp:
Fiqqp(id:id+4,1) = FiqqpTriedro2PF2VF(P2p, V2p);
id = 12;
%% Triedro sólido 4.
% formamos Fiqqp:
Fiqqp(id:id+5,1) = FiqqpTriedro2PF2VF(P2p, V8p, V9p);
id = 18;
%% Triedro sólido 5.
% formamos Fiqqp:
Fiqqp(id:id+5,1) = FiqqpTrdPFVW(P2p, V8p, V9p);
id = 24;
%% CL punto P2.
% formamos Fiqqp:
Fiqqp(id:id+2,1:1) = FiqqpCCL1P3V;
id = 27;
%% CL punto P3.
% formamos Fiqqp:
Fiqqp(id:id+2,1) = FiqqpCCL1P3V;
id = 30;
%% CL punto P8.
% formamos Fiqqp:
Fiqqp(id:id+2,1) = FiqqpCCL1P3V;
id = 33;

```

Cuadro 3.34: Problema de aceleración del grupo {2,3,4,5}. Séptima parte del fichero de grupo.

```

%% CL punto P9.
% formamos Fiqpqp:
Fiqpqp(id:id+2,1) = FiqpqpCCL1P3V;
id = 36;
%% Cardan 2-4.
% formamos Fiqpqp:
Fiqpqp(id,1) = FiqpqpCardan(V5p,V8p);
id = 37;
%% Cardan 2-5.
% formamos Fiqpqp:
Fiqpqp(id,1) = FiqpqpCardan(V5p,V6p);

hs = [GDL(1,3) GDL(2,3)]';
Fihhs = Fih * hs;
qsgrupo = -jacobDep \ (Fiqpqp + Fihhs);

% Reescribimos aceleraciones calculadas en POI.PVA real, el valor obtenido para su id de GE:
% Ej. POI(MGrupos(ng).poi(1,1)).PVA(7:9) -> punto 14 del modelo = P1 del GE.

POI(MGrupos(ng).poi(1,1)).PVA(7:9) = qsgrupo(1:3);
POI(MGrupos(ng).poi(2,1)).PVA(7:9) = qsgrupo(4:6);
POI(MGrupos(ng).poi(3,1)).PVA(7:9) = qsgrupo(7:9);
POI(MGrupos(ng).poi(8,1)).PVA(7:8) = qsgrupo(10:11);
POI(MGrupos(ng).poi(8,1)).PVA(9) = GDL(1,3);
POI(MGrupos(ng).poi(9,1)).PVA(7:8) = qsgrupo(12:13);
POI(MGrupos(ng).poi(9,1)).PVA(9) = GDL(2,3);
VEC(MGrupos(ng).vec(2,1)).PVA(7:9) = qsgrupo(14:16);
VEC(MGrupos(ng).vec(3,1)).PVA(7:9) = qsgrupo(20:22);
VEC(MGrupos(ng).vec(4,1)).PVA(7:9) = qsgrupo(17:19);
VEC(MGrupos(ng).vec(5,1)).PVA(7:9) = qsgrupo(23:25);
VEC(MGrupos(ng).vec(6,1)).PVA(7:9) = qsgrupo(26:28);
VEC(MGrupos(ng).vec(7,1)).PVA(7:9) = qsgrupo(29:31);
VEC(MGrupos(ng).vec(8,1)).PVA(7:9) = qsgrupo(32:34);
VEC(MGrupos(ng).vec(9,1)).PVA(7:9) = qsgrupo(35:37);

```

Cuadro 3.35: Problema de aceleración del grupo {2,3,4,5}. Octava parte del fichero de grupo.

```

% Posición de Otros POI´s.
% Matriz de transformación del triedro matTrans = X \ Xloc
MGrupos(ng).matTrans(1:3,1:3,1) = zeros(3,3);
MGrupos(ng).matTrans(1:3,1:3,2) = zeros(3,3);
MGrupos(ng).matTrans(1:3,1:3,3) = zeros(3,3);
MGrupos(ng).matTrans(1:3,1:3,4) = zeros(3,3);

MGrupos(ng).triedroGlo(1:3,1:3,1) = [VEC(MGrupos(ng).vec(4,1)).PVA(1:3)
    VEC(MGrupos(ng).vec(3,1)).PVA(1:3)
    VEC(MGrupos(ng).vec(5,1)).PVA(1:3)]'; % La del sólido 2.

% Vnorm=norm(POI(MGrupos(ng).poi(10,1,2)).PVA(1:3) -
    POI(MGrupos(ng).poi(1,1,2)).PVA(1:3));

MGrupos(ng).triedroGlo(1:3,1:3,2) = [VEC(MGrupos(ng).vec(1,1)).PVA(1:3)
    (POI(MGrupos(ng).poi(1,1,2)).PVA(1:3)-POI(MGrupos(ng).poi(10,1,2)).PVA(1:3))
    VEC(MGrupos(ng).vec(2,1)).PVA(1:3)]'; % La del sólido 3.

% Vnorm=norm(POI(MGrupos(ng).poi(12,1,3)).PVA(1:3) -
    POI(MGrupos(ng).poi(2,1,3)).PVA(1:3));

MGrupos(ng).triedroGlo(1:3,1:3,3) = [VEC(MGrupos(ng).vec(8,1)).PVA(1:3)
    (POI(MGrupos(ng).poi(2,1,3)).PVA(1:3)-POI(MGrupos(ng).poi(12,1,3)).PVA(1:3))
    VEC(MGrupos(ng).vec(9,1)).PVA(1:3)]'; % La del sólido 4.

% Vnorm=norm(POI(MGrupos(ng).poi(11,1,4)).PVA(1:3) -
    POI(MGrupos(ng).poi(3,1,4)).PVA(1:3));

MGrupos(ng).triedroGlo(1:3,1:3,4) = [VEC(MGrupos(ng).vec(6,1)).PVA(1:3)
    (POI(MGrupos(ng).poi(3,1,4)).PVA(1:3)-POI(MGrupos(ng).poi(11,1,4)).PVA(1:3))
    VEC(MGrupos(ng).vec(7,1)).PVA(1:3)]'; % La del sólido 5.

for j=1:4
MGrupos(ng).matTrans(:,j) = MGrupos(ng).triedroGlo(:,j) *
    MGrupos(ng).invTriedroLoc(:,j); % A= X * XLoc^-1
end

rpt = POI(MGrupos(ng).poi(1,1,1)).PVA(1:3)';
POI(MGrupos(ng).poi(4,1,1)).PVA(1:3) = rpt +
    MGrupos(ng).matTrans(:,1)*MGrupos(ng).poi(4,4:6:6)';
POI(MGrupos(ng).poi(5,1,1)).PVA(1:3) = rpt +
    MGrupos(ng).matTrans(:,1)*MGrupos(ng).poi(5,4:6:6)';
POI(MGrupos(ng).poi(6,1,1)).PVA(1:3) = rpt +
    MGrupos(ng).matTrans(:,1)*MGrupos(ng).poi(6,4:6:6)';
POI(MGrupos(ng).poi(7,1,1)).PVA(1:3) = rpt +
    MGrupos(ng).matTrans(:,1)*MGrupos(ng).poi(7,4:6:6)';

```

Cuadro 3.36: Problema de posición de otros POI´s del grupo {2,3,4,5}. Novena parte del fichero de grupo.

```

% Velocidades de Otros POI's.
MGrupos(ng).triédroGlop = [VEC(MGrupos(ng).vec(4,1)).PVA(4:6)
    VEC(MGrupos(ng).vec(3,1)).PVA(4:6)
    VEC(MGrupos(ng).vec(5,1)).PVA(4:6)]';

MGrupos(ng).matTransp = zeros(3,3);

MGrupos(ng).matTransp(1:3,1:3) = MGrupos(ng).triédroGlop *
    MGrupos(ng).invTriédroLoc(:,:,1);

rptp = POI(MGrupos(ng).poi(1,1,1)).PVA(4:6)';

POI(MGrupos(ng).poi(4,1)).PVA(4:6) = rptp +
    MGrupos(ng).matTransp*MGrupos(ng).poi(4,4:6)';
POI(MGrupos(ng).poi(5,1)).PVA(4:6) = rptp +
    MGrupos(ng).matTransp*MGrupos(ng).poi(5,4:6)';
POI(MGrupos(ng).poi(6,1)).PVA(4:6) = rptp +
    MGrupos(ng).matTransp*MGrupos(ng).poi(6,4:6)';
POI(MGrupos(ng).poi(7,1)).PVA(4:6) = rptp +
    MGrupos(ng).matTransp*MGrupos(ng).poi(7,4:6)';

```

Cuadro 3.37: Problema de velocidad de otros POI's del grupo {2,3,4,5}. Décima parte del fichero de grupo.

```

% Aceleraciones de Otros POI's.
MGrupos(ng).triedroGlos = [VEC(MGrupos(ng).vec(4,1)).PVA(7:9)
    VEC(MGrupos(ng).vec(3,1)).PVA(7:9)
    VEC(MGrupos(ng).vec(5,1)).PVA(7:9)]';

MGrupos(ng).matTransp = zeros(3,3);

MGrupos(ng).matTrans(1:3,1:3) = MGrupos(ng).triedroGlos *
    MGrupos(ng).invTriedroLoc(:,:,1);

rpts = POI(MGrupos(ng).poi(1,1,1)).PVA(7:9)';

POI(MGrupos(ng).poi(4,1)).PVA(7:9) = rpts +
    MGrupos(ng).matTrans * MGrupos(ng).poi(4,4:6)';
POI(MGrupos(ng).poi(5,1)).PVA(7:9) = rpts +
    MGrupos(ng).matTrans * MGrupos(ng).poi(5,4:6)';
POI(MGrupos(ng).poi(6,1)).PVA(7:9) = rpts +
    MGrupos(ng).matTrans * MGrupos(ng).poi(6,4:6)';
POI(MGrupos(ng).poi(7,1)).PVA(7:9) = rpts +
    MGrupos(ng).matTrans * MGrupos(ng).poi(7,4:6)';

%% Grabamos resultados:

sólido2=zeros(4,4);
sólido3=zeros(4,4);
sólido4=zeros(4,4);
sólido5=zeros(4,4);

sólido2(1:4,4) = [POI(MGrupos(ng).poi(1,1,1)).PVA(1:3) 1];
sólido2(1:3,1:3) = MGrupos(ng).matTrans(:,:,1);
sólido3(1:4,4) = [POI(MGrupos(ng).poi(10,1,2)).PVA(1:3) 1];
sólido3(1:3,1:3) = MGrupos(ng).matTrans(:,:,2);
sólido4(1:4,4) = [POI(MGrupos(ng).poi(12,1,3)).PVA(1:3) 1];
sólido4(1:3,1:3) = MGrupos(ng).matTrans(:,:,3);
sólido5(1:4,4) = [POI(MGrupos(ng).poi(11,1,4)).PVA(1:3) 1];
sólido5(1:3,1:3) = MGrupos(ng).matTrans(:,:,4);

save('C:\ Users\ Bernardino\ Desktop\ PFC\ Simulacion\ src_cpp\ Debug\
Suspension.out','sólido2','-ascii','-double','-append')
save('C:\ Users\ Bernardino\ Desktop\ PFC\ Simulacion\ src_cpp\ Debug\
Suspension.out','sólido3','-ascii','-double','-append')
save('C:\ Users\ Bernardino\ Desktop\ PFC\ Simulacion\ src_cpp\ Debug\
Suspension.out','sólido4','-ascii','-double','-append')
save('C:\ Users\ Bernardino\ Desktop\ PFC\ Simulacion\ src_cpp\ Debug\
Suspension.out','sólido5','-ascii','-double','-append')

```

Cuadro 3.38: Problema de aceleración de otros POI's del grupo {2,3,4,5}. Undécima parte del fichero de grupo.

```

function [POI,VEC,ANG,MGrupos]=SolucionGrupo79(Gd1,ng,POI,VEC,ANG,MGrupos,coef79)
%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %%
%% %% Este script se encarga de resolver el GE{7,9}.
%% %%
%% Se reciben:
%% -z y sus derivadas, coordenadas independientes globales.
%% -ng, número de grupo estructural.
%% -POI, matriz puntos de interés.
%% -VEC, matriz de vectores.
%% -MGrupos, matriz de matrices.
%%
%% Identificación de constantes y de variables de este grupo:
%%
%% Son variables dependientes las coordenadas X Y Z de las entidades:
%% P2, V1, V2, V3, V5 = 3*5 = 15.
%% En total 15 variables dependientes.
%%
%% Son variables independientes las coordenadas X Y Z de las entidades:
%% P1, P3, V4 = 3*3 = 9 variables independientes.
%% Hay que considerar que aquí, en el propio grupo, los V1, V2, etc tienen la nomenclatura propia
del grupo, según su esquema, independientemente de cuántos grupos como éste haya. Eso ya se
define en MIBDatos.
%%
%% Las coordenadas de GE quedan ordenadas: P2, P3, V1, V2, V3, V4, V5.
%%
%% Posiciones.
P1 = POI(MGrupos(ng).poi(1,1,1)).PVA(1:3); % PVA del punto 7 del modelo. No contabiliza
a coordenadas independientes porque es punto de bastidor.
P2 = POI(MGrupos(ng).poi(2,1,1)).PVA(1:3); % PVA del punto 13 del modelo. Es variable
independiente.
%%
%% Puntos dependientes.
P3 = POI(MGrupos(ng).poi(3,1,1)).PVA(1:3); % PVA de punto 18 del modelo.
V4 = VEC(MGrupos(ng).vec(4,1)).PVA(1:3); % PVA del vector 9 del modelo. Es coordenada
independiente de GE, perteneciente al GE 2345.
V6 = VEC(MGrupos(ng).vec(6,1)).PVA(1:3); % PVA del vector 8 del modelo. Se usa para
calcular Theta, ángulo relativo entre 7 y 2.
%%
%% Vectores dependientes.
V1 = VEC(MGrupos(ng).vec(1,1)).PVA(1:3); % PVA del vector 23 del modelo.
V2 = VEC(MGrupos(ng).vec(2,1)).PVA(1:3); % PVA del vector 24 del modelo.
V3 = VEC(MGrupos(ng).vec(3,1)).PVA(1:3); % PVA del vector 25 del modelo.
V5 = VEC(MGrupos(ng).vec(5,1)).PVA(1:3); % PVA del vector 26 del modelo.
%%
%% grupo, coordenadas dependientes del GE.
qgrupo(1:15,1) = [P3, V1, V2, V3, V5];
Fig = zeros(15,15);
%%
%% Índices de coordenadas de GE:
idpt2 = 1; idpt3 = 4; % Cuenta los 21 huecos.
idvc1 = 7; idvc2 = 10; idvc3 = 13; idvc4 = 16; idvc5 = 19;

```

Cuadro 3-39: Identificación de constantes y variables del grupo {7,9}. Primera parte del fichero de grupo.



```

%% Problema de posición.
% mFi, vector de funciones de restricción.
% FigG, matriz jacobiana de mFi respecto a las coordenadas dependientes de grupo (qG).
id = 1; tol = 1e-8; n = 0; Res = 20; % Criterio de parada de solución aceptable.
while Res >tol
    n = n+1;
    %% Triedro sólido 9.
    dist = norm(MGrupos(ng).poi(1,4:6,2) - MGrupos(ng).poi(3,4:6,2));
    [mFi, Jacob] = TrdPPVV(P1,P3,V1,V2,dist); % Sólido 9 (1:6);
    FIG(id:id+5,1) = mFi;
    % formamos la jacobiana:
    Fig(id:id+5,idpt3:idpt3+2) = Jacob(:,1:3);
    Fig(id:id+5,idvc1:idvc1+2) = Jacob(:,4:6);
    Fig(id:id+5,idvc2:idvc2+2) = Jacob(:,7:9);
    id = 7;
    %% Triedro sólido 7.
    [mFi, Jacob] = Triedro1P3VC(P3,V4,V5,V3); % Sólido 9 (7:11);
    FIG(id:id+4,1) = mFi;
    % formamos la jacobiana:
    Fig(id:id+4,idvc4:idvc4+2) = Jacob(:,1:3);
    Fig(id:id+4,idvc5:idvc5+2) = Jacob(:,4:6);
    Fig(id:id+4,idvc3:idvc3+2) = Jacob(:,7:9);
    id = 12;
    %% Cardan 7-9.
    [mFi, Jacob] = Cardan(V1,V4); % (12);
    FIG(id,1) = mFi;
    % formamos la jacobiana:
    Fig(id,idvc1:idvc1+2) = Jacob(:,1:3);
    Fig(id,idvc4:idvc4+2) = Jacob(:,4:6);
    id = 13;
    %% CL punto P2.
    [mFi, Jacob] = CCL1P3V(P2,P3,V4,V5,V3,coef79(1:3,1)); % CL (13:15);
    FIG(id:id+2,1) = mFi;
    % formamos la jacobiana:
    Fig(id:id+2,idpt2:idpt2+2) = Jacob(:,1:3);
    Fig(id:id+2,idpt3:idpt3+2) = Jacob(:,4:6);
    Fig(id:id+2,idvc4:idvc4+2) = Jacob(:,7:9);
    Fig(id:id+2,idvc5:idvc5+2) = Jacob(:,10:12);
    Fig(id:id+2,idvc3:idvc3+2) = Jacob(:,13:15);

% Obtenemos la Jacobiana respecto a las variables dependientes (jacobDep).
% Las coordenadas de GE quedan ordenadas: P2, P3, V1, V2, V3, V4, V5.

jacobDep(:,1:12) = Fig(:,idpt3:idvc3+2);
jacobDep(:,13:15) = Fig(:,idvc5:idvc5+2);

```

Cuadro 3.40: Problema de posición del grupo {7,9}. Segunda parte del fichero de grupo.

```

% Solución mediante Newton-Raphson.
qgrupo1 = qgrupo - jacobDep\ Fig;
qgrupo = qgrupo1;

P3(1,:) = qgrupo(1:3);
V1(1,:) = qgrupo(4:6);
V2(1,:) = qgrupo(7:9);
V3(1,:) = qgrupo(10:12);
V5(1,:) = qgrupo(13:15);

Res = norm(Fig);
id = 1;
end

% Ahora se calcula Theta: cos(Theta) = dotProduct(V3,V6).
Theta=acos(dotProduct(V3,V6)); % Ángulo relativo entre eslabón 2 y 7.

% Reescribimos posiciones calculadas en POI.PVA real, el valor obtenido para su id de GE.
% Ejemplo: POI(MGrupos(ng).poi(3,1,1)).PVA(1:3) ->18 del modelo = P3 del GE.

POI(MGrupos(ng).poi(3,1,1)).PVA(1:3) = P3;
VEC(MGrupos(ng).vec(1,1)).PVA(1:3) = V1;
VEC(MGrupos(ng).vec(2,1)).PVA(1:3) = V2; % Escribe en el VEC(2) valores de V2.
VEC(MGrupos(ng).vec(3,1)).PVA(1:3) = V3; % Escribe en el VEC(8) valores de V3.
VEC(MGrupos(ng).vec(5,1)).PVA(1:3) = V5;

ANG(MGrupos(ng).ang(1,1)).PVA(1) = Theta;

% Matriz de transformación del triedro matTrans = X\ Xloc.

MGrupos(ng).matTrans(1:3,1:3,1) = zeros(3,3);
MGrupos(ng).matTrans(1:3,1:3,2) = zeros(3,3);

MGrupos(ng).triedroGlo(1:3,1:3,1) = [VEC(MGrupos(ng).vec(4,1)).PVA(1:3)
    VEC(MGrupos(ng).vec(5,1)).PVA(1:3)
    VEC(MGrupos(ng).vec(3,1)).PVA(1:3)]; % La del sólido 7.

% Vnorm = norm(POI(MGrupos(ng).poi(3,1,2)).PVA(1:3) -
    % POI(MGrupos(ng).poi(1,1,2)).PVA(1:3));

MGrupos(ng).triedroGlo(1:3,1:3,2) = [VEC(MGrupos(ng).vec(2,1)).PVA(1:3)
    VEC(MGrupos(ng).vec(1,1)).PVA(1:3)
    (POI(MGrupos(ng).poi(3,1,2)).PVA(1:3) - POI(MGrupos(ng).poi(1,1,2)).PVA(1:3))];

for j = 1:2
    MGrupos(ng).matTrans(:,j) = MGrupos(ng).triedroGlo(:,j) *
        MGrupos(ng).invTrietroLoc(:,j); %A= X * XLoc^-1
end

```

Cuadro 3.41: Problema de posición del grupo {7,9}. Tercera parte del fichero de grupo.

```

%% Problema de velocidad.

% Fih, matriz Jacobiana de las mFi respecto a las coordenadas independientes de GE (h).
% Fihhp, vector producto de Fih por el vector velocidades independientes de GE (hp).

P2p = POI(MGrupos(ng).poi(2,1)).PVA(4:6);
V4p = VEC(MGrupos(ng).vec(4,1)).PVA(4:6);

Fih(:,1:3) = Fiq(:,idpt2:idpt2+2); % De la Jacobiana sacamos las respecto h.
Fih(:,4:6) = Fiq(:,idvc4:idvc4+2);
hp = [P2p V4p]';

Fihhp = Fih * hp;
qpgrupo = -jacobDep\ Fihhp;

% Escribimos las velocidades calculadas en POI.PVA real, el valor obtenido en su id de GE.
% Ej: POI(MGrupos(ng).poi(3,1)).PVA(1:3) -> punto 18 del modelo = P3 del GE.

POI(MGrupos(ng).poi(3,1)).PVA(4:6) = qpgrupo(1:3);
VEC(MGrupos(ng).vec(1,1)).PVA(4:6) = qpgrupo(4:6);
VEC(MGrupos(ng).vec(2,1)).PVA(4:6) = qpgrupo(7:9);
VEC(MGrupos(ng).vec(3,1)).PVA(4:6) = qpgrupo(10:12);
VEC(MGrupos(ng).vec(5,1)).PVA(4:6) = qpgrupo(13:15);

P3p = POI(MGrupos(ng).poi(3,1)).PVA(4:6);
V1p = VEC(MGrupos(ng).vec(1,1)).PVA(4:6);
V2p = VEC(MGrupos(ng).vec(2,1)).PVA(4:6);
V3p = VEC(MGrupos(ng).vec(3,1)).PVA(4:6);
V5p = VEC(MGrupos(ng).vec(5,1)).PVA(4:6);

```

Cuadro 3.42: Problema de velocidad del grupo {7,9}. Cuarta parte del fichero de grupo.

```

%% Problema de aceleración.
% Fihhs, vector producto de Fih por el vector aceleraciones independientes de GE (hs).
% Fiqp, matriz derivada temporal de la Jacobiana de las mFi respecto a todas las coordenadas
de GE.
% Fiqpqp, vector producto de Fiqp por el vector derivada temporal de todas las coordenadas de
GE.

%% Triedro sólido 9.
% formamos Fiqpqp:
Fiqpqp(id:id+5,1) = FiqpqpTrdPffPVV(P3p,V1p,V2p);
id = 7;
%% Triedro sólido 7.
% formamos Fiqpqp:
Fiqpqp(id:id+4,1) = FiqpqpTriedro1P3VC(V4p,V5p,V3p);
id = 12;
%% Cardan 7-9.
Fiqpqp(id,1) = FiqpqpCardan(V1p,V4p);
id = 13;
%% CL punto P2.
% formamos Fiqpqp:
Fiqpqp(id:id+2,1) = FiqpqpCCL1P3V;

P2s = POI(MGrupos(ng).poi(2,1)).PVA(7:9);
V4s = VEC(MGrupos(ng).vec(4,1)).PVA(7:9);

hs = [P2s V4s]';
Fihhs = Fih * hs;
qsgrupo = -jacobDep \ (Fiqpqp + Fihhs);

% Reescribimos las aceleraciones calculadas en POI.PVA real, el valor obtenido en su id de GE.
% Ej. POI(MGrupos(ng).poi(3,1)).PVA(7:9) -> punto 18 del modelo = P3 del GE.

POI(MGrupos(ng).poi(3,1)).PVA(7:9) = qsgrupo(1:3);
VEC(MGrupos(ng).vec(1,1)).PVA(7:9) = qsgrupo(4:6);
VEC(MGrupos(ng).vec(2,1)).PVA(7:9) = qsgrupo(7:9);
VEC(MGrupos(ng).vec(3,1)).PVA(7:9) = qsgrupo(10:12);
VEC(MGrupos(ng).vec(5,1)).PVA(7:9) = qsgrupo(13:15);

%% Grabamos resultados:
sólido7 = zeros(4,4);
sólido9 = zeros(4,4);

sólido7(1:4,4) = [POI(MGrupos(ng).poi(3,1,1)).PVA(1:3) 1];
sólido7(1:3,1:3) = MGrupos(ng).matTrans(:,1);
sólido9(1:4,4) = [POI(MGrupos(ng).poi(1,1,1)).PVA(1:3) 1];
sólido9(1:3,1:3) = MGrupos(ng).matTrans(:,2);

save('C:\ Users\ Bernardino\ Desktop\ PFC\ Simulacion\ src_cpp\ Debug\
Suspension.out','sólido7','-ascii','-double','-append')
save('C:\ Users\ Bernardino\ Desktop\ PFC\ Simulacion\ src_cpp\ Debug\
Suspension.out','sólido9','-ascii','-double','-append')

```

Cuadro 3.43: Problema de velocidad del grupo {7,9}. Cuarta parte del fichero de grupo.

```

function [POI,VEC,DIST,MGrupos]=SolucionGrupo1213_b(GdI,ng,POI,VEC,DIST,MGrupos)
%% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %% %%
%% %% Este script se encarga de resolver el GE{12,13}.
%% %%
%% Se reciben:
%% -z y sus derivadas, coordenadas independientes globales.
%% -ng, número de grupo estructural.
%% -POI, matriz puntos de interés.
%% -VEC, matriz de vectores.
%% -MGrupos, matriz de matrices.

%% Identificación de constantes y de variables de este grupo:

%% Son variables dependientes las coordenadas X Y Z de las entidades:
%% V1, V2, V3, V4, V5 = 3*5 = 15.
%% En total 15 variables dependientes.

%% Son variables independientes las coordenadas X Y Z de las entidades:
%% P2, V6 = 3*2 = 6.
%% Hay que considerar que aquí, en el propio grupo, los V1, V2, etc tienen la nomenclatura propia
del grupo, según su esquema, independientemente de cuántos grupos como éste haya. Eso ya se
define en MBDatos.

%% Las coordenadas de GE quedan ordenadas: P2, V1, V2, V3, V4, V5, V6.

%% Posiciones.

P1 = POI(MGrupos(ng).poi(1,1)).PVA(1:3); % PVA del punto 5 del modelo. No contabiliza
a coordenadas independientes porque es punto de bastidor.
P2 = POI(MGrupos(ng).poi(2,1)).PVA(1:3); % PVA del punto 11 del modelo. Es coordenada
independiente de GE, perteneciente al GE 2345.

%% Vectores dependientes.
V1 = VEC(MGrupos(ng).vec(1,1)).PVA(1:3); % PVA del vector 16 del modelo.
V2 = VEC(MGrupos(ng).vec(2,1)).PVA(1:3); % PVA del punto 15 del modelo.
V3 = VEC(MGrupos(ng).vec(3,1)).PVA(1:3); % PVA del punto 17 del modelo.
V4 = VEC(MGrupos(ng).vec(4,1)).PVA(1:3); % PVA del punto 19 del modelo.
V5 = VEC(MGrupos(ng).vec(5,1)).PVA(1:3); % PVA del punto 18 del modelo.

%% Vectores independientes.
V6 = VEC(MGrupos(ng).vec(6,1)).PVA(1:3); % PVA del vector 8 del modelo.

%% qgrupo, coordenadas dependientes del GE.
qgrupo(1:15,1) = [V1, V2, V3, V4, V5];
Fig = zeros(15,21);

%% Índices de coordenadas de GE.
idpt2 = 1; idvc1 = 4; % Cuenta los 21 huecos.
idvc2 = 7; idvc3 = 10; idvc4 = 13; idvc5 = 16; idvc6 = 19;

```

Cuadro 3.44: Identificación de constantes y variables del grupo {12,13}. Primera parte del fichero de grupo.

```

%% Problema de posición.
% mFi, vector de funciones de restricción.
% FiqG, matriz jacobiana de mFi respecto a las coordenadas dependientes de grupo (qG).
id = 1; tol = 1e-8; n = 0; Res = 20; % Criterio de parada de solución aceptable.
while Res > tol
    n = n+1;
    %% Triedro solido 13.
    [mFi, Jacob] = Triedro1P3V(P1, V3, V2, V1); % Sólido 13 (1:6);
    FiqG(id:id+5, 1) = mFi;
    % formamos la jacobiana:
    Fiq(id:id+5, idvc3: idvc3+2) = Jacob(:, 1:3);
    Fiq(id:id+5, idvc2: idvc2+2) = Jacob(:, 4:6);
    Fiq(id:id+5, idvc1: idvc1+2) = Jacob(:, 7:9);
    id = 7;
    %% Triedro solido 12.
    [mFi, Jacob] = Triedro1P2V(P2, V5, V4, V1); % Sólido 12 (7:11);
    FiqG(id:id+4, 1) = mFi;
    % formamos la jacobiana:
    Fiq(id:id+4, idvc5: idvc5+2) = Jacob(:, 1:3);
    Fiq(id:id+4, idvc4: idvc4+2) = Jacob(:, 4:6);
    Fiq(id:id+4, idvc1: idvc1+2) = Jacob(:, 7:9);
    id = 12;
    %% Par prismático.
    [mFi, Jacob] = Prismatico(P1, P2, V1, V2, V5); % (12:15); Prismático compartiendo V1.
    Son 5secs-2=3.
    FiqG(id:id+3, 1) = mFi; % metemos las tres de producto vectorial->redundante
    % formamos la jacobiana:
    Fiq(id:id+3, idpt2: idpt2+2) = Jacob(:, 1:3);
    Fiq(id:id+3, idvc1: idvc1+2) = Jacob(:, 4:6);
    Fiq(id:id+3, idvc2: idvc2+2) = Jacob(:, 7:9);
    Fiq(id:id+3, idvc5: idvc5+2) = Jacob(:, 10:12);
    id = 16;
    %% Cardan 12-2.
    [mFi, Jacob] = Cardan(V6, V5); % (16);
    FiqG(id, 1) = mFi;
    % formamos la jacobiana:
    Fiq(id, idvc6: idvc6+2) = Jacob(:, 1:3);
    Fiq(id, idvc5: idvc5+2) = Jacob(:, 4:6);

% Obtenemos la Jacobiana respecto a las variables dependientes (jacobDep).
% Las coordenadas de GE quedan ordenadas: P2, V1, V2, V3, V4, V5, V6.

jacobDep(:, 1:15) = Fiq(:, idvc1: idvc5+2);
JTJ = jacobDep'*jacobDep;

```

Cuadro 3.45: Problema de posición del grupo {12,13}. Segunda parte del fichero de grupo.

```

% Solución mediante Newton-Raphson.

qgrupo1 = qgrupo - J TJ \ (jacobDep'*FiG);
qgrupo = qgrupo1;

V1(1,1:3) = qgrupo(1:3);
V2(1,1:3) = qgrupo(4:6);
V3(1,1:3) = qgrupo(7:9);
V4(1,1:3) = qgrupo(10:12);
V5(1,1:3) = qgrupo(13:15);

Res = norm(FiG);
id=1;
end

% Ahora se calcula S:
S=norm(POI(MGrupos(ng).poi(1,1)).PVA(1:3)-POI(MGrupos(ng).poi(2,1)).PVA(1:3));

% Reescribimos posiciones calculadas en POI.PVA real, el valor obtenido para su id de GE:
% Ejemplo: POI(MGrupos(ng).poi(3,1)).PVA(1:3) ->18 del modelo = P3 del GE.

VEC(MGrupos(ng).vec(1,1)).PVA(1:3) = V1;
VEC(MGrupos(ng).vec(2,1)).PVA(1:3) = V2; % Escribe en el VEC(2) valores de V2.
VEC(MGrupos(ng).vec(3,1)).PVA(1:3) = V3; % Escribe en el VEC(8) valores de V3.
VEC(MGrupos(ng).vec(4,1)).PVA(1:3) = V4; % Escribe en el VEC(8) valores de V3.
VEC(MGrupos(ng).vec(5,1)).PVA(1:3) = V5;

DIST(MGrupos(ng).dist(1,1)).PVA(1) = S;

% Matriz de transformación del triedro matTrans = X \ Xloc.

MGrupos(ng).matTrans(1:3,1:3,1) = zeros(3,3);
MGrupos(ng).matTrans(1:3,1:3,2) = zeros(3,3);

MGrupos(ng).triedroGlo(1:3,1:3,1) = [VEC(MGrupos(ng).vec(5,1)).PVA(1:3)
    VEC(MGrupos(ng).vec(4,1)).PVA(1:3)
    VEC(MGrupos(ng).vec(1,1)).PVA(1:3)]; % La del sólido 7.

MGrupos(ng).triedroGlo(1:3,1:3,2) = [VEC(MGrupos(ng).vec(3,1)).PVA(1:3)
    VEC(MGrupos(ng).vec(2,1)).PVA(1:3)
    VEC(MGrupos(ng).vec(1,1)).PVA(1:3)]; % La del sólido 9.

for j=1:2
MGrupos(ng).matTrans(:,j) = MGrupos(ng).triedroGlo(:,j)*
    MGrupos(ng).invTriedroLoc(:,j); %A= X * XLoc^-1
end

```

Cuadro 3.46: Problema de posición del grupo {12,13}. Tercera parte del fichero de grupo.

```

%% Problema de velocidad.
% Fih, matriz Jacobiana de las mFi respecto a las coordenadas independientes de GE (h).
% Fihhp, vector producto de Fih por el vector velocidades independientes de GE hp.
P2p = POI(MGrupos(ng).poi(2,1)).PVA(4:6);
V6p = VEC(MGrupos(ng).vec(6,1)).PVA(4:6);
Fih(:,1:3) = Fiq(:,idpt2:idpt2+2); % De la Jacobiana sacamos las respecto h.
Fih(:,4:6) = Fiq(:,idvc6:idvc6+2);
hp = [P2p V6p]';
Fihhp= Fih*hp;
qpgrupo = JTJ\ (jacobDep'*Fihhp) - jacobDep

% Escribimos las velocidades calculadas en POL.PVA real, el valor obtenido en su id de GE.
% Ej. POI(MGrupos(ng).poi(2,1)).PVA(1:3) -> punto 11 del modelo = P2 del GE.
VEC(MGrupos(ng).vec(1,1)).PVA(4:6) = qpgrupo(1:3);
VEC(MGrupos(ng).vec(2,1)).PVA(4:6) = qpgrupo(4:6);
VEC(MGrupos(ng).vec(3,1)).PVA(4:6) = qpgrupo(7:9);
VEC(MGrupos(ng).vec(4,1)).PVA(4:6) = qpgrupo(10:12);
VEC(MGrupos(ng).vec(5,1)).PVA(4:6) = qpgrupo(13:15);

V1p = VEC(MGrupos(ng).vec(1,1)).PVA(4:6);
V2p = VEC(MGrupos(ng).vec(2,1)).PVA(4:6);
V3p = VEC(MGrupos(ng).vec(3,1)).PVA(4:6);
V4p = VEC(MGrupos(ng).vec(4,1)).PVA(4:6);
V5p = VEC(MGrupos(ng).vec(5,1)).PVA(4:6);

Sp=norm(POI(MGrupos(ng).poi(1,1)).PVA(4:6)-POI(MGrupos(ng).poi(2,1)).PVA(4:6));
DIST(MGrupos(ng).dist(1,1)).PVA(2) = Sp;

```

Cuadro 3.47: Problema de velocidad del grupo {12,13}. Cuarta parte del fichero de grupo.



```

%% Problema de aceleración.
% Fihhs, vector producto de Fih por el vector aceleraciones independientes de GE (hs).
% Fiqp, matriz derivada temporal de la Jacobiana de las mFi respecto a todas las coordenadas
de GE.
% Fiqqp, vector producto de Fiqp por el vector derivada temporal de todas las coordenadas de
GE.

%% Triedro sólido 13.
% formamos Fiqqp:
Fiqqp(id:id+5,1)=FiqqpTriedro1P3V(V1p,V2p,V3p);
id = 7;
%% Triedro sólido 12.
% formamos Fiqqp:
Fiqqp(id:id+4,1)=FiqqpTriedro1P2V(V4p,V5p,V1p);
id = 12;
%% Par prismático.
% formamos Fiqqp:
Fiqqp(id:id+3,1)=FiqqpPrismatico(P2p,V1p,V2p,V5p);
id = 16;
%% Cardan 12-2.
% formamos Fiqqp:
Fiqqp(id,1)=FiqqpCardan(V6p,V4p);

P2s = POI(MGrupos(ng).poi(2,1)).PVA(7:9);
V6s = VEC(MGrupos(ng).vec(6,1)).PVA(7:9);

hs = [P2s V6s]';
Fihhs = Fih * hs;
qsgrupo = -JTJ\jacobDep*(Fiqqp+Fihhs));

% Reescribimos las aceleraciones calculadas en POI.PVA real, el valor obtenido en su id de GE.
% Ej. POI(MGrupos(ng).poi(2,1)).PVA(1:3) -> punto 11 del modelo = P2 del GE.

VEC(MGrupos(ng).vec(1,1)).PVA(7:9) = qsgrupo(1:3);
VEC(MGrupos(ng).vec(2,1)).PVA(7:9) = qsgrupo(4:6);
VEC(MGrupos(ng).vec(3,1)).PVA(7:9) = qsgrupo(7:9);
VEC(MGrupos(ng).vec(4,1)).PVA(7:9) = qsgrupo(10:12);
VEC(MGrupos(ng).vec(5,1)).PVA(7:9) = qsgrupo(13:15);

%% Grabamos resultados:

Solido10(1:4,4) = [POI(MGrupos(ng).poi(2,1)).PVA(1:3) 1];
Solido10(1:3,1:3) = MGrupos(ng).matTrans(:,1);
Solido11(1:4,4) = [POI(MGrupos(ng).poi(1,1)).PVA(1:3) 1];
Solido11(1:3,1:3) = MGrupos(ng).matTrans(:,2);

save('C:\ Users\ Bernardino\ Desktop\ PFC\ Simulacion\ src_cpp\ Debug\
Suspension.out','Solido10','-ascii','-double','-append')
save('C:\ Users\ Bernardino\ Desktop\ PFC\ Simulacion\ src_cpp\ Debug\
Suspension.out','Solido11','-ascii','-double','-append')

```

Cuadro 3.48: Problema de aceleración del grupo {12,13}. Quinta parte del fichero de grupo.

```
function [mFi, Jacob] = Triedro1P3V(P1, V1, V2, V3)
```

%Función que devuelve las ecuaciones de restricción y contribución a la Jacobiana, derivada respecto de todas las variables de GE, de sólido rígido 3D (triedro) formado por un punto y tres vectores unitarios.

```
mFi = [dotProduct(V1, V3); %V1(1)*V3(1) + V1(2)*V3(2) + V1(3)*V3(3)
dotProduct(V1, V2); %V1(1)*V2(1) + V1(2)*V2(2) + V1(3)*V2(3)
dotProduct(V2, V3); %V2(1)*V3(1) + V2(2)*V3(2) + V2(3)*V3(3)
NormCont(V1); %V1(1)^2 + V1(2)^2 + V1(3)^2 - 1
NormCont(V2); %V2(1)^2 + V2(2)^2 + V2(3)^2 - 1
NormCont(V3);]; %V3(1)^2 + V3(2)^2 + V3(3)^2 - 1
```

```
Jacob = zeros(6, 9);
Jacob(1, 1:9) = [V3 0 0 0 V1];
Jacob(2, 1:6) = [V2 V1];
Jacob(3, 4:9) = [V3 V2];
Jacob(4, 1:3) = 2*V1;
Jacob(5, 4:6) = 2*V2;
Jacob(6, 7:9) = 2*V3;
```

Cuadro 3.49: Función auxiliar Triedro1P3V.

```
function [mFi, Jacob] = Triedro1P3VC(P1, V1, V2, V3)
```

%Función que devuelve las ecuaciones de restricción y contribución a la Jacobiana, derivada respecto de todas las variables de GE, de sólido rígido 3D (triedro) formado por un punto y tres vectores unitarios. En este caso, uno de los vectores es compartido (V1), no necesariamente fijo.

```
mFi = [dotProduct(V1, V3); %V1(1)*V3(1) + V1(2)*V3(2) + V1(3)*V3(3)
dotProduct(V1, V2); %V1(1)*V2(1) + V1(2)*V2(2) + V1(3)*V2(3)
dotProduct(V2, V3); %V2(1)*V3(1) + V2(2)*V3(2) + V2(3)*V3(3)
NormCont(V2); %V2(1)^2 + V2(2)^2 + V2(3)^2 - 1
NormCont(V3);]; %V3(1)^2 + V3(2)^2 + V3(3)^2 - 1
```

```
Jacob = zeros(5, 9);
Jacob(1, 1:9) = [V3 0 0 0 V1];
Jacob(2, 1:6) = [V2 V1];
Jacob(3, 4:9) = [V3 V2];
Jacob(4, 4:6) = 2*V2;
Jacob(5, 7:9) = 2*V3;
```

Cuadro 3.50: Función auxiliar Triedro1P3VC.

```

funcion [mFi, Jacob] = Triedro2PF2VF(P1, P2, V1, V2, dist)
%Función que devuelve las ecuaciones de restricción y contribución a la Jacobiana, derivada
respecto de todas las variables de GE, de sólido rígido 3D (triedro) formado por dos puntos y
dos vectores, donde:

% P1 es el fijo; P2 el móvil.
% V1 es el fijo; V2 el móvil.

V3 = P2-P1;
mFi = [dotProduct(V1, V3); %V1(1)* V3(1) + V1(2)* V3(2) + V1(3)* V3(3)
dotProduct(V1, V2); %V1(1)* V2(1) + V1(2)* V2(2) + V1(3)* V2(3)
dotProduct(V2, V3); %V2(1)* V3(1) + V2(2)* V3(2) + V2(3)* V3(3)
NormCont(V2); %V2(1)^ 2 + V2(2)^ 2 + V2(3)^ 2 - 1
RigCont(V3, dist); %V3(1)^ 2 + V3(2)^ 2 + V3(3)^ 2 - dist^ 2

Jacob = zeros(5,6);
Jacob(1,1:3) = V1; % [V1(1) V1(2) V1(3)];
Jacob(2,4:6) = V1; % [V1(1) V1(2) V1(3)];
Jacob(3,:) = [V2 V3]; % [ V2(1) V2(2) V2(3) V3(1) V3(2) V3(3)];
Jacob(4,4:6) = 2*V2; % [2*V2(1) 2*V2(2) 2*V2(3)];
Jacob(5,1:3) = 2*V3; % [2*V3(1) 2*V3(2) 2*V3(3)];

```

Cuadro 3.51: Función auxiliar Triedro2PF2VF.

```

funcion [mFi, Jacob] = TrdFPVV(P1, P2, V1, V2, dist)
%Función que devuelve las ecuaciones de restricción y su contribución a la Jacobiana respecto
de todas las variables de GE, de sólido rígido 3D (triedro) formado por dos puntos y dos vectores
unitarios, donde:

% Primer punto P1, fijo.
%Segundo punto y vectores móviles.

V3 = P2-P1;

mFi = [dotProduct(V1, V3); %V1(1)* V3(1) + V1(2)* V3(2) + V1(3)* V3(3)
dotProduct(V1, V2); %V1(1)* V2(1) + V1(2)* V2(2) + V1(3)* V2(3)
dotProduct(V2, V3); %V2(1)* V3(1) + V2(2)* V3(2) + V2(3)* V3(3)
NormCont(V2); %V2(1)^ 2 + V2(2)^ 2 + V2(3)^ 2 - 1
RigCont(V3, dist); %V3(1)^ 2 + V3(2)^ 2 + V3(3)^ 2 - norm(V3)
NormCont(V1); %V1(1)^ 2 + V1(2)^ 2 + V1(3)^ 2 - 1

Jacob = zeros(6,9);
Jacob(1,1:6) = [V1 V3];
Jacob(2,4:9) = [V2 V1];
Jacob(3,1:9) = [V2 0 0 0 V3];
Jacob(4,7:9) = 2*V2;
Jacob(5,1:3) = 2*V3; % (1) -2*V3(2) -2*V3(3)];
Jacob(6,4:6) = 2*V1;

```

Cuadro 3.52: Función auxiliar TrdFPVV.

```

function [mFi, Jacob] = CCLIP3V(P2,P1,V1,V2,V3,coef)
%Función que devuelve las ecuaciones de restricción de combinación lineal de un punto, P2,
respecto a un sólido rígido definido por P1 y el trietro V1,V2 y V3.

V4 = P2-P1;
B = [V4(1)
     V4(2)
     V4(3)];
A = [V1(1) V2(1) V3(1)
     V1(2) V2(2) V3(2)
     V1(3) V2(3) V3(3)];
mFi = B-A*coef;
% V4(1) - coef(1,1)*V1(1) - coef(2,1)*V2(1) - coef(3,1)*V3(1)
% V4(2) - coef(1,1)*V1(2) - coef(2,1)*V2(2) - coef(3,1)*V3(2)
% V4(3) - coef(1,1)*V1(3) - coef(2,1)*V2(3) - coef(3,1)*V3(3)

Jacob = zeros(3,15);
Jacob(1,1:15) = [1 0 0 -1 0 0 -coef(1,1) 0 0 -coef(2,1) 0 0 -coef(3,1) 0 0];
Jacob(2,1:15) = [0 1 0 0 -1 0 0 -coef(1,1) 0 0 -coef(2,1) 0 0 -coef(3,1) 0];
Jacob(3,1:15) = [0 0 1 0 0 -1 0 0 -coef(1,1) 0 0 -coef(2,1) 0 0 -coef(3,1)];

```

Cuadro 3.53: Función auxiliar CCLIP3V.

```

function [mFi, Jacob] = Cardan(V1, V2)
%Función que devuelve ecuaciones de restricción de cardan y las contribuciones a la Jacobiana
respecto de todas las variables de GE.

mFi = dotProduct(V1,V2); %V1(1)*V2(1) + V1(2)*V2(2) + V1(3)*V2(3)
Jacob(1,1:6) = [V2 V1];

```

Cuadro 3.54: Función auxiliar Cardan.

```
function [mFi,Jacob] = Prismatico(P1,P2,V1,V2,V3)
```

% Función que devuelve las ecuaciones de restricción y contribuciones a la Jacobiana, respecto de todas las coordenadas de GE, del par prismático cuando el par está definido entre P1 (fijo sobre bastidor) y P2. V1 define esa dirección, y dos vectores, V2 y V3, pertenecientes a P1 y P2 respectivamente, son perpendiculares. Dará un sistema redundante. En el caso de querer un sistema LI se deberá eliminar por configuración una de las tres primeras ecuaciones.

```
V4 = P2-P1;
mFi = [V4(1)*V1(2) - V4(2)*V1(1)
       V4(3)*V1(1) - V4(1)*V1(3)
       V4(2)*V1(3) - V4(3)*V1(2)
       dotProduct(V3,V2)]; %V3(1)*V2(1) + V3(2)*V2(2) + V3(3)*V2(3)
Jacob = zeros(4,12);
Jacob(1,1:6) = [V1(2) -V1(1) 0 -V4(2) V4(1) 0];
Jacob(2,1:6) = [-V1(3) 0 V1(1) V4(3) 0 -V4(1)];
Jacob(3,1:6) = [0 V1(3) -V1(2) 0 -V4(3) V4(2)];
Jacob(4,7:12) = [V3 V2];
```

Cuadro 3.55: Función auxiliar Prismatico.

```
function Scalar = dotProduct(V1,V3)
```

%Función que devuelve el producto escalar de dos vectores.

```
%Scalar = V1(1)*V3(1) + V1(2)*V3(2) + V1(3)*V3(3);
Scalar = V1*V3';
```

Cuadro 3.56: Función secundaria dotProduct.

```
function Scalar = NormCont(V1)
```

%Función que devuelve el resultado de evaluar la ecuación de restricción de normalidad de un vector.

```
Scalar = V1(1)^ 2 + V1(2)^ 2 + V1(3)^ 2 - 1;
```

Cuadro 3.57: Función secundaria NormCont.

```
function Scalar = RigCont(V1,dist)
```

%Función que devuelve el resultado de evaluar la ecuación de restricción de sólido rígido entre dos puntos formados por un vector.

```
Scalar = V1(1)^ 2 + V1(2)^ 2 + V1(3)^ 2 - dist^ 2;
```

Cuadro 3.58: Función secundaria RigCont.

```
function [F1qpqp] = F1qpqpTriedro1P3V(V1p,V2p,V3p)
```

%Función que devuelve la contribución de un triedro formado por un punto y tres vectores, a F1qpqp.

```
F1qpqp = [2*(V3p(1)*V1p(1) + V3p(2)*V1p(2) + V3p(3)*V1p(3))
          2*(V2p(1)*V1p(1) + V2p(2)*V1p(2) + V2p(3)*V1p(3))
          2*(V3p(1)*V2p(1) + V3p(2)*V2p(2) + V3p(3)*V2p(3))
          2*(V1p(1)^ 2 + V1p(2)^ 2 + V1p(3)^ 2)
          2*(V2p(1)^ 2 + V2p(2)^ 2 + V2p(3)^ 2)
          2*(V3p(1)^ 2 + V3p(2)^ 2 + V3p(3)^ 2)];
```

Cuadro 3.59: Función auxiliar F1qpqpTriedro1P3V.

```

function [Ficqpp] = FicqppTriedro1P3VC(V1p,V2p,V3p)
%Función que devuelve la contribución de un triedro formado por punto y dos vectores, a Ficqpp.

Ficqpp = [2*(V3p(1)*V1p(1) + V3p(2)*V1p(2) + V3p(3)*V1p(3))
          2*(V2p(1)*V1p(1) + V2p(2)*V1p(2) + V2p(3)*V1p(3))
          2*(V3p(1)*V2p(1) + V3p(2)*V2p(2) + V3p(3)*V2p(3))
          2*(V1p(1)^2 + V1p(2)^2 + V1p(3)^2)
          2*(V2p(1)^2 + V2p(2)^2 + V2p(3)^2)];

```

Cuadro 3.60: Función auxiliar FicqppTriedro1P3VC.

```

function [Ficqpp] = FicqppTriedro2PF2VF(P2p,V2p)
%Función que devuelve la contribución de un triedro formado por dos puntos y dos vectores,
donde un punto y vector son fijos, a Ficqpp. Dado que uno de los vectores es fijo, las ecuaciones
de sólido rígido son sólo cinco.

Ficqpp = [0
          0
          2*(P2p(1)*V2p(1) + P2p(2)*V2p(2) + P2p(3)*V2p(3))
          2*(V2p(1)^2 + V2p(2)^2 + V2p(3)^2)
          2*(P2p(1)^2 + P2p(2)^2 + P2p(3)^2)];

```

Cuadro 3.61: Función auxiliar FicqppTriedro2PF2VF.

```

function [Ficqpp] = FicqppTrdPFPVW(P2p,V1p,V2p)
%Función que devuelve la contribución de un triedro formado por dos puntos y dos vectores,
donde un punto es fijo, a Ficqpp.

Ficqpp = [2*(V1p(1)*P2p(1) + V1p(2)*P2p(2) + V1p(3)*P2p(3))
          2*(V2p(1)*V1p(1) + V2p(2)*V1p(2) + V2p(3)*V1p(3))
          2*(V2p(1)*P2p(1) + V2p(2)*P2p(2) + V2p(3)*P2p(3))
          2*(V2p(1)^2 + V2p(2)^2 + V2p(3)^2)
          2*(P2p(1)^2 + P2p(2)^2 + P2p(3)^2)
          2*(V1p(1)^2 + V1p(2)^2 + V1p(3)^2)];

```

Cuadro 3.62: Función auxiliar FicqppTrdPFPVW.

```

function [Ficqpp] = FicqppCCLIP3V
%Función que devuelve la contribución de la combinación lineal a Ficqpp.

Ficqpp = zeros(3,1);

```

Cuadro 3.63: Función auxiliar FicqppCCLIP3V.

```
function [Fiqppq] = FiqppqCardan(V1p,V2p)
%Función que devuelve la contribución del cardan a Fiqppq.
Fiqppq = 2*(V2p(1)*V1p(1) + V2p(2)*V1p(2) + V2p(3)*V1p(3));
```

Cuadro 3.64: Función auxiliar FiqppqCardan.

```
function [Fiqppq] = FiqppqPrismatico(P2p,V1p,V2p,V3p)
%Función que devuelve la contribución de un par prismático formado a Fiqppq.
Fiqppq = [2*(V1p(2)*P2p(1) - V1p(1)*P2p(2))
          2*(-V1p(3)*P2p(1) + V1p(1)*P2p(3))
          2*(-V1p(2)*P2p(3) + V1p(3)*P2p(2))
          2*(V3p(1)*V1p(1) + V3p(2)*V1p(2) + V3p(3)*V1p(3))];
```

Cuadro 3.65: Función auxiliar FiqppqPrismatico.



# Capítulo 4

## Resultados

En este capítulo se analizan los resultados obtenidos en el Capítulo 3. A través de ellos, se muestra la simulación del movimiento de los dos mecanismos escalables (plano y espacial), además de la evolución de variables características. También se estudia la eficiencia de la formulación implementada, comparando distintas formulaciones.

### 4.1. Simulación cinemática de los sistemas analizados

En el capítulo anterior se llevó a cabo la implementación en Matlab de la resolución cinemática para los dos casos de estudio. Resulta interesante estudiar la evolución de las posiciones, velocidades y aceleraciones de los POI's más representativos. Por ello, para cada problema se ha elegido un punto característico, lo suficientemente alejados de la entrada del movimiento, para estudiar su cinemática. También es curioso ver el movimiento real que adquirirían estos mecanismos, ya que es propiamente el objetivo de este análisis. En este apartado también se mostrará la simulación, plana o espacial, del cuadrilátero escalable y del sistema de suspensión de un camión, respectivamente.

#### 4.1.1. Cuadrilátero escalable

Un punto interesante de estudio para este mecanismo plano es el punto de unión entre cuadrilátero base ( $k = 1$ ) con el resto de diadas  $3R$  que se van adicionando.

Las condiciones de trabajo tanto para la cinemática como para la simulación son:

- Tratamos con cinco grupos  $3R$  unidos a la manivela  $1RF$  ( $k = 5$ ).
- La posición inicial de la entrada  $\theta$  es de  $90^\circ$ .
- La velocidad angular de la entrada  $\dot{\theta}$  es de  $1 \text{ rad/s}$ .
- La manivela recorre dos vueltas completas durante la simulación de movimiento.

A partir del programa que aparece en el apartado (3.3.1) del Capítulo 3 se obtienen las posiciones, velocidades y aceleraciones del punto  $P_3$  respecto del ángulo de entrada. Estas evoluciones vienen representadas en las gráficas 4.1, 4.2 y 4.3, respectivamente.

Su simulación plana se realiza en Matlab, usando un fichero específico que recrea el movimiento una vez resuelta la cinemática. En las imágenes de la Figura 4.4 se muestra el resultado; las dos vueltas que da la manivela y el movimiento de sus cinco diadas.

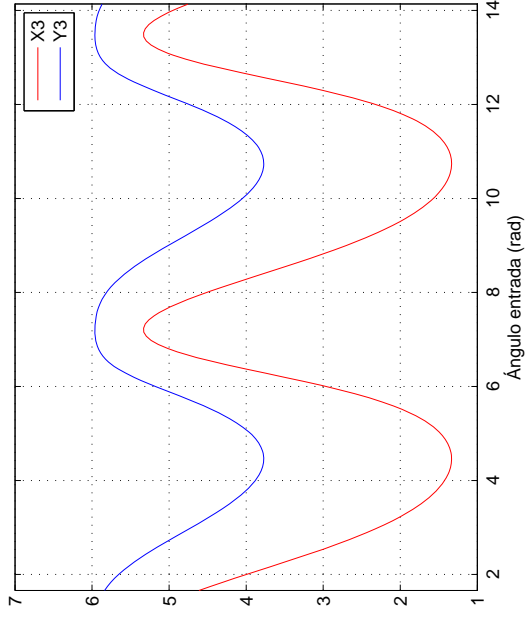


Figura 4.1: Mecanismo cuadrilátero escalable: Posición del punto  $P_3$  respecto del ángulo de entrada  $\theta$ .

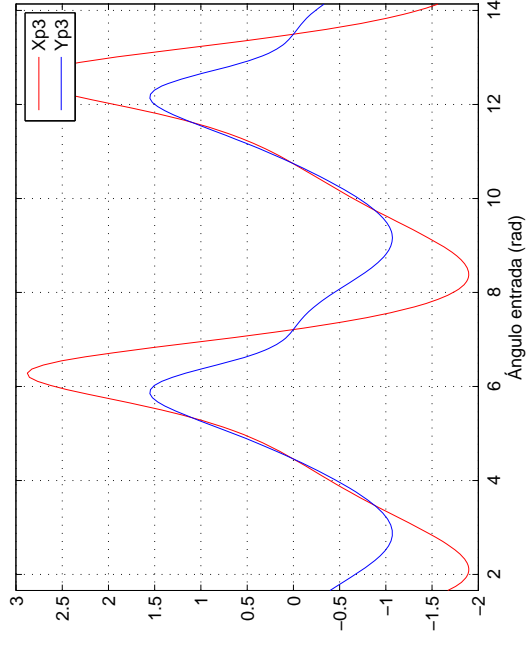


Figura 4.2: Mecanismo cuadrilátero escalable: Velocidad del punto  $P_3$  respecto del ángulo de entrada  $\theta$ .

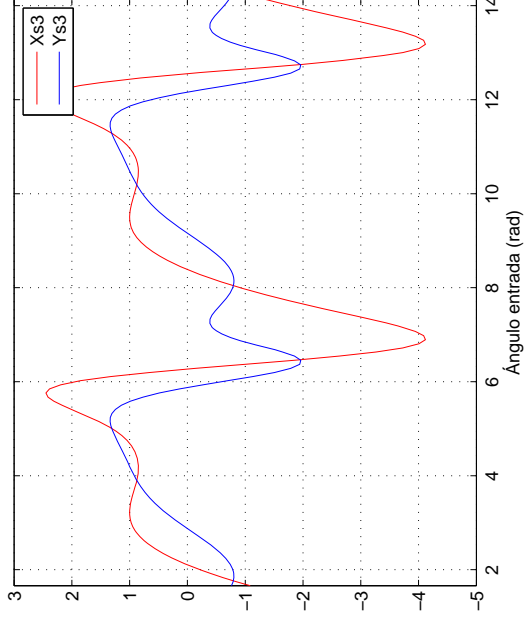


Figura 4.3: Mecanismo cuadrilátero escalable: Aceleración del punto  $P_3$  respecto del ángulo de entrada  $\theta$ .

#### 4.1.2. Sistema de suspensión de un camión

El mecanismo de suspensión analizado consta de 12 sólidos rígidos, modelizado con 114 coordenadas naturales, de las cuales 2 son movimientos de entrada. La extensión a 5 ejes hace un total de 60 sólidos rígidos más chasis, con 570 coordenadas. En este mecanismo espacial, en vez de estudiar la evolución de un punto de interés se analizarán tres variables características. Se estudia la distancia relativa en un par cilíndrico, la velocidad asociada al mismo y el giro relativo entre los extremos de la barra a torsión 6 – 7. A partir de la implementación en Matlab, apartado (3.3.2.1), se obtienen, para el cuarto eje del camión, las distancias del par cilíndrico del lado derecho, sus velocidades y ángulos de torsión de la barra 6 – 7, todo respecto a la elevación vertical de la rueda de ese mismo lado. Los resultados vienen representados en las Figuras 4.6, 4.7 y 4.5.

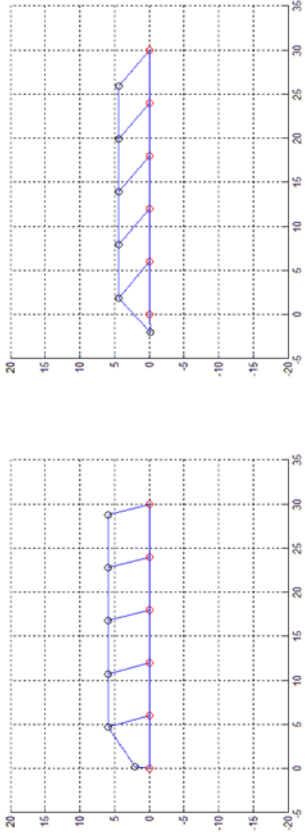
Las condiciones de trabajo tanto para la cinemática como para la simulación son las siguientes:

- Se ha contado con el sistema de suspensión del camión de cinco ejes ( $m = 5$ ).
- La posición inicial de todas las entradas son mulas y sus velocidades varían dependiendo del eje, ya que se cuenta con entradas distintas para cada uno de ellos. En la Tabla 4.1 se recogen las citadas velocidades en mm/s.
- El tiempo de simulación es de 20 segundos, con paso de cálculo  $h = 0,05$  segundos.

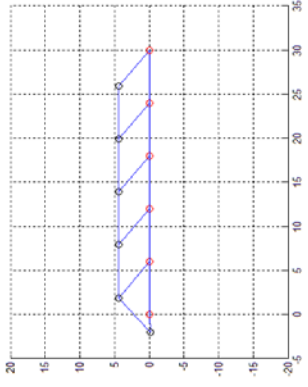
Ejes	1	2	3	4	5
Rueda izquierda (mm/s)	0	0	0.008	0	0.01
Rueda derecha (mm/s)	0.01	0.006	0	0.009	0.005

Cuadro 4.1: Velocidades de las entradas al sistema de suspensión.

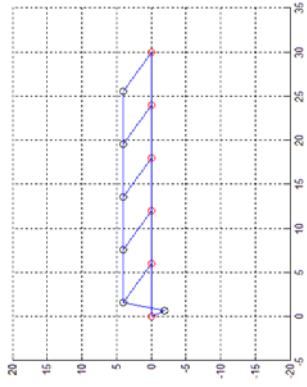
Se presenta a continuación dos simulaciones espaciales, una en detalle asociada al cuarto eje y otra para el sistema completo. Para ello se utiliza la OpenSceneGraph, que requiere del fichero *Suspension.out* y los eslabones en formato *.obj*. El fichero recoge los triedros locales y su punto de referencia para cada eslabón a lo largo de la simulación, que son propiamente los resultados de la ejecución de Matlab para cinco ejes. En las Figuras 4.8 y 4.9 se muestran las imágenes inicial y final para cada una de las simulaciones de movimiento.



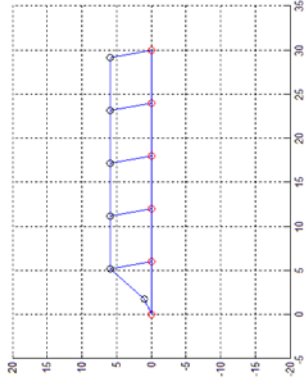
1



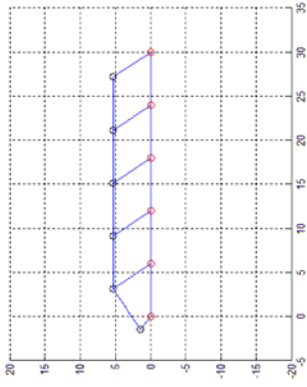
2



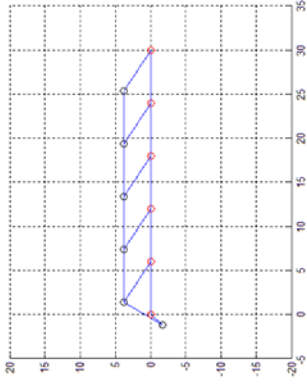
3



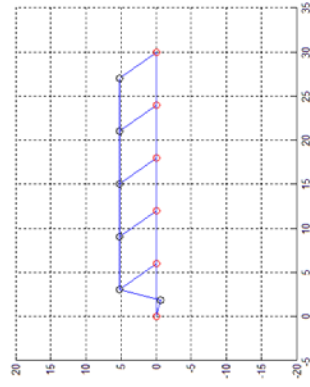
4



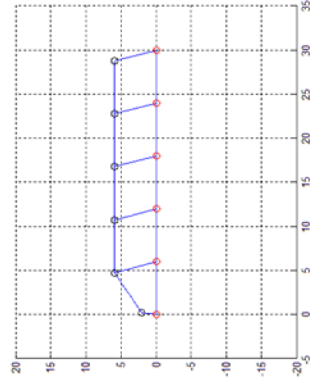
5



6



7



8

Figura 4.4: Simulación cinemática del cuadrilátero escalante.

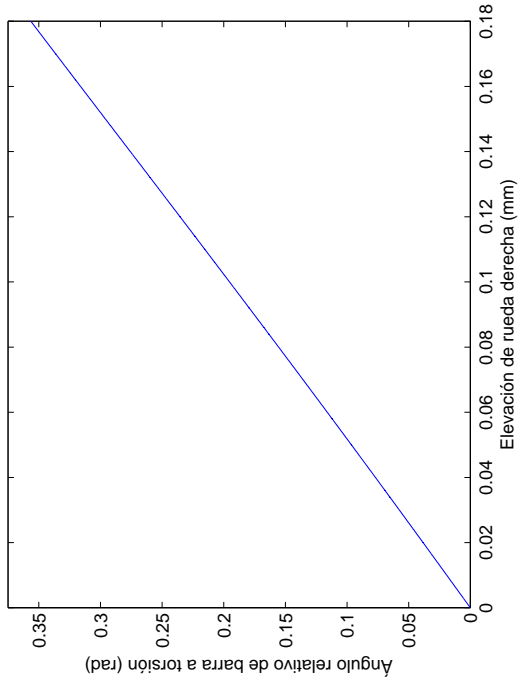


Figura 4.5: Ángulo de torsión de la barra 6 – 7, para el cuarto eje del sistema, en función de la elevación vertical de la rueda derecha.

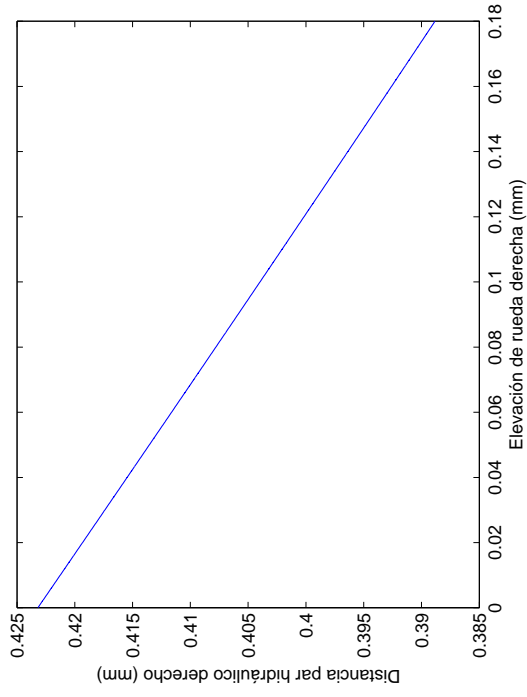


Figura 4.6: Distancia asociada al par hidráulico derecho, para el cuarto eje del sistema, en función de la elevación vertical de la rueda del mismo lado.

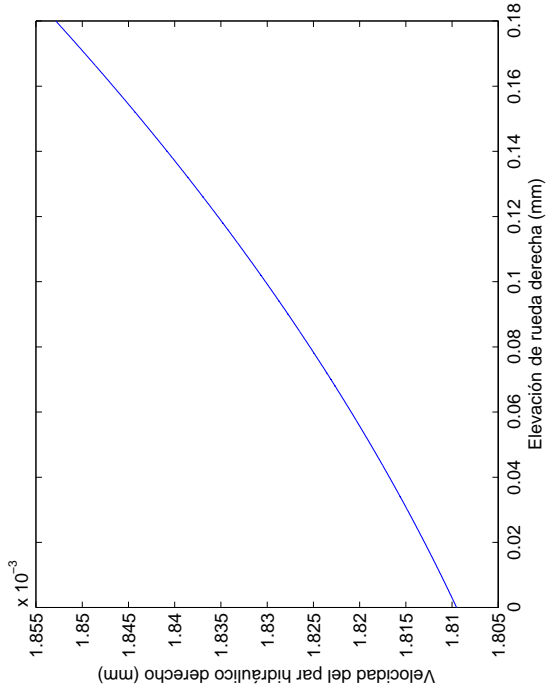


Figura 4.7: Velocidad asociada al par hidráulico derecho, para el cuarto eje del sistema, en función de la elevación vertical de la rueda del mismo lado.

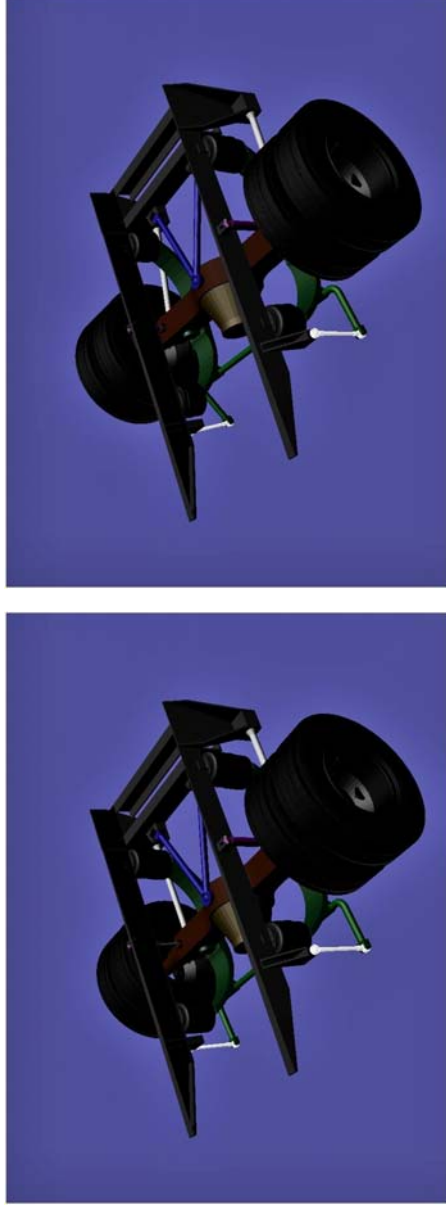


Figura 4.8: Simulación cinemática en detalle del sistema de suspensión.

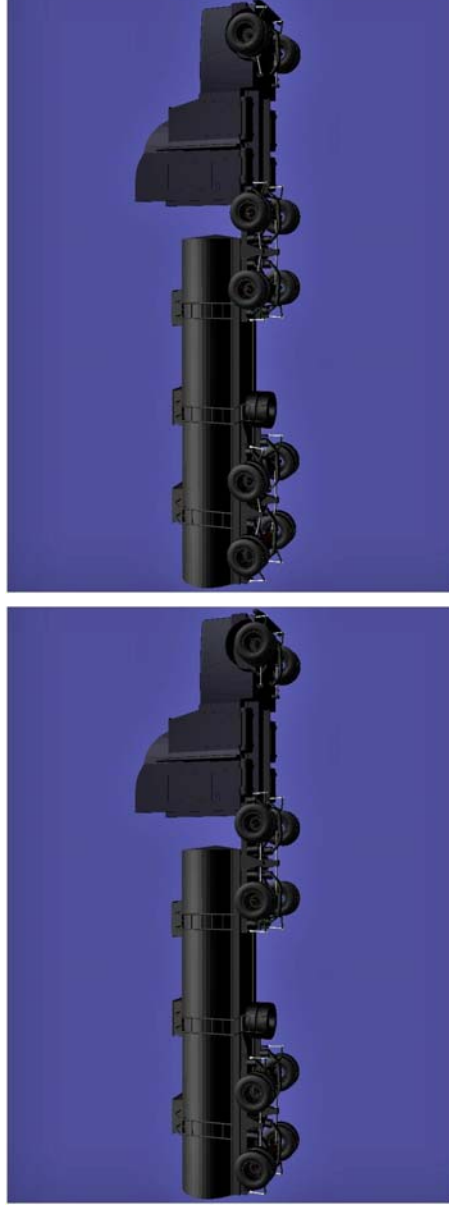


Figura 4.9: Simulación cinemática del sistema de suspensión del camión.

## 4.2. Eficiencia de las diferentes implementaciones

En este apartado se hace la comparativa del uso de distintas formulaciones para el caso del cuadrilátero escalable.

### 4.2.1. Formulación global vs la topológica

Aprovechando la facilidad que ofrece el mecanismo cuadrilátero escalable, para evaluar la eficiencia de formulaciones cinemáticas, en este apartado se comparan cuatro métodos diferentes para el análisis cinemático de sistemas multicuerpo planos: I) Formulación global con coordenadas mixtas, II) Formulación topológica basada en grupos estructurales con coordenadas mixtas y derivadas temporales (sec.2.2.1), III) Formulación topológica con coordenadas punto de referencia y tensor de tercer orden (sec.2.2.2). Además, debido a la dispersión de las matrices en la solución global (I), se introduce una cuarta solución: IV) Formulación global sparse, que es lo mismo que (I) pero los sistemas lineales y no lineales de ecuaciones son resueltos por la opción sparse de Matlab.

Para evaluar la eficiencia de los cuatro métodos se va a usar el cuadrilátero escalable, que como vimos consta de una manivela (eslabón 2) y un número  $k$  de diadas: (los eslabones  $(2k + 1)$ ,  $(2k + 2)$ ), junto con la articulación de rotación, en la Figura 3.7. Este método nos permite controlar el número de ecuaciones de restricción y evaluar cómo influye en el tiempo de cálculo de los cuatro métodos.

Tanto las soluciones de los métodos globales (I,IV) como del topológico (II) definen el sistema multicuerpo con el mismo número de ecuaciones de restricción  $m = 2 \times k + 2$  y de coordenadas  $n = 2 \times k + 3$ . En el método (III) se reducen en dos tanto las ecuaciones de restricción  $m = 2 \times k$  como las coordenadas  $n = 2 \times k + 1$ .

Para llevar a cabo el análisis cinemático se define el movimiento de entrada, la rotación de la manivela con velocidad constante  $\theta_1 = 1 \text{ rad/s}$ . Todas las longitudes se fijan a 6, excepto  $A_1B_1 = 2$ . Para conseguir que los resultados sean comparados para un número reducido de diadas, los cálculos se mantendrán hasta que la manivela complete las 500 rotaciones.

**I) Solución global:** Las  $n = 2 \times k + 3$  coordenadas para esta formulación corresponden al ángulo de rotación  $\theta_1$  y a las dos coordenadas cartesianas del punto  $B_1$  y los  $k$  puntos  $C_k$ . El Algoritmo 1 del apartado 2.3, muestra las llamadas a las funciones en el programa principal, *mFi*, *Jacob* y *Fzqpp*, que automáticamente forman los vectores y matrices correspondientes  $\Phi$ ,  $\Phi_q$ ,  $\dot{\Phi}$ ,  $\dot{\Phi}_q$ , cuyo tamaño depende de las diadas consideradas.

**II) Solución topológica:** Esta solución usa las mismas coordenadas que el método previo. Los pasos que sigue este método se muestran en el Algoritmo 2, apartado 2.3, que es el mismo para los métodos II y III, basados en la descomposición de grupos estructurales. La diferencia entre la solución I y II radica en la formulación cinemática usada para resolver cualquier grupo estructural. Como se muestra, para cada paso de cálculo, cada grupo estructural forma la estructura cinemática del sistema multicuerpo que es analizado (Algoritmo 2). Dependiendo del tipo del grupo estructural se llama a la subrutina específica correspondiente: (*IR\_SG*) para calcular  $\varphi_I = (x_B y_B)$  para un valor dado de  $\mathbf{h}_I = z$ , o (*3R\_SG*) para calcular  $\varphi_k = (x_{C_k} y_{C_k})$  dados los valores de  $\mathbf{h}_k = (x_{C_{k-1}} y_{C_{k-1}} x_{D_{k-1}} y_{D_{k-1}})$ . En la opción II, cualquiera de estas subrutinas sigue la formulación introducida en el apartado 2.2.1, basado en el método de partición de coordenadas, que es similar al mostrado en el algoritmo 1 (líneas 4-22) para la aproximación global. Sin embargo, el tamaño de las matrices que intervienen es ahora mucho más reducido.

**III) Solución topológica:** Para implementar este método, se selecciona un conjunto de coordenadas de rotación  $\theta_{nb}$  de los eslabones  $nb$ , junto a las coordenadas cartesianas de los puntos  $B_1$  y  $C_k$ . Se ha incluido este método en la comparativa para demostrar que puede elegir cualquier tipo de coordenadas, siempre y cuando se definan adecuadamente las ecuaciones de restricción. El programa raíz tiene la misma estructura que el método II (Algoritmo 2), pero el código necesario para resolver cada clase de grupo estructural es, en este caso, bastante más complejo.

**IV) Solución global sparse:** Debido a la geometría del cuadrilátero escalable, las ecuaciones de res-

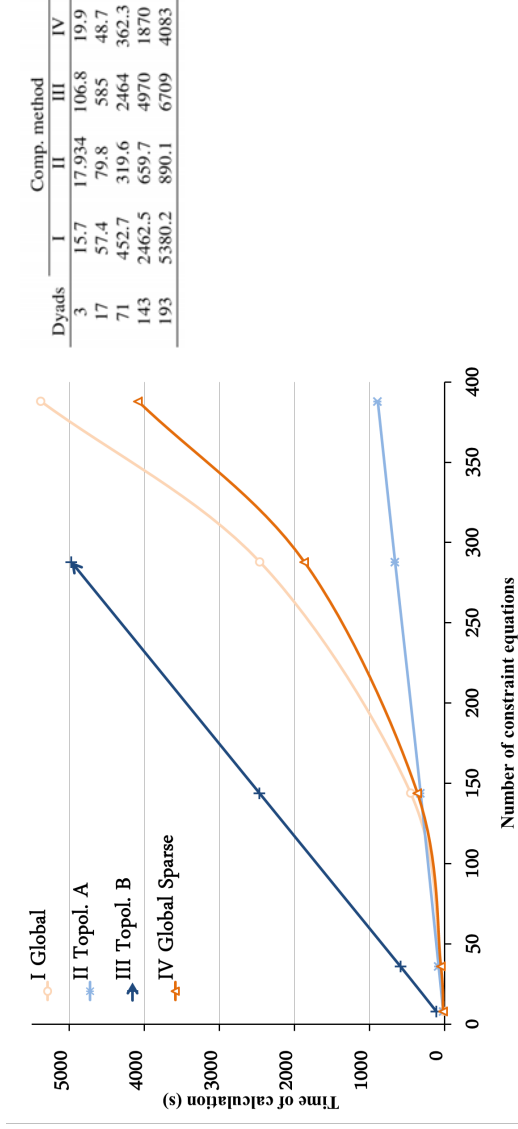


Figura 4.10: Evolución del cálculo de tiempo requerido para las soluciones del cuadrilátero escalable, dependiendo de a) el número de ecuaciones de restricción, b) número de diadas.

tricción de cualquier diada  $k$  dependen solamente de las coordenadas cartesianas de los puntos  $C_k$  y  $C_{k-1}$ . Esto convierte a la matriz Jacobiana  $\Phi_q$  en dispersa, y por tanto se pueden considerar funciones especiales para mejorar la eficiencia de los cálculos. Es posible considerar esta opción en el entorno de programación de Matlab, que ha sido implementada como se mostró en el algoritmo 3.

#### 4.2.1.1. Cálculo de tiempo para diferentes aproximaciones

La Figura 4.10 muestra los segundos empleados para cada método computacional (derecha) y su evolución gráfica (izquierda) como función del número de ecuaciones de restricción. Mientras que los métodos globales (I, IV) evolucionan exponencialmente, los topológicos (II,III) lo hacen linealmente. Excepto el método III, que muestra muy baja eficiencia, los métodos experimentan un desarrollo muy similar en el rango de diadas 3-17 (8-144 ecuaciones de restricción). Concretamente, el método global I es más rápido por el reducido número de ecuaciones de restricción y, sólo para un número cercano a 36, la función *sparse* muestra alguna ventaja. Entorno a 71 diadas (144 ecuaciones de restricción) y por encima, el comportamiento lineal del método II cambia, siendo el más eficiente de los cuatro métodos considerados.

Para estudiar más en profundidad la eficiencia de estos métodos, se ha usado la herramienta profile de Matlab, que permite ver cómo se distribuye el tiempo de cálculo entre todas las operaciones matemáticas. La Figura 4.11 muestra el porcentaje que los cuatro métodos emplean en sus operaciones más importantes: cálculo de la matriz Jacobiana  $\Phi_q^d$ , el vector de ecuaciones de restricción  $\Phi$ , el término independiente para las velocidades  $\Phi_q^b \dot{\mathbf{q}}^b$ , uno de los términos involucrados en el cálculo de aceleraciones  $\ddot{\Phi}_q \dot{\mathbf{q}}^b$  y la solución del sistema no lineal de ecuaciones para las posiciones  $\mathbf{q}^d$ , y el sistema lineal para las velocidades  $\dot{\mathbf{q}}^d$  y aceleraciones  $\ddot{\mathbf{q}}^d$ . El ítem “otros” es usado para las líneas de código en las que cada método desarrolla “otras” operaciones matemáticas que emplean tiempo. El método topológico III no calcula  $\Phi_q^b \dot{\mathbf{q}}^b$  ni  $\ddot{\Phi}_q \dot{\mathbf{q}}^b$ , así que en su lugar se representan los cálculos de  $\mathbf{K}$  y  $\mathbf{L}$ , como se definieron en (2.12) y (2.18) respectivamente.

Hay varios resultados interesantes de este análisis. El primero es que, mientras que los métodos globales gastan cerca de 45% en la evaluación de los términos de la matriz Jacobiana, el método topológico II gasta solamente un 12%, y donde los primeros tardan cerca de un 17% en la evaluación del vector de ecuaciones de restricción, el método topológico III gasta un 7%. Sin embargo, mientras que los métodos globales gastan sólo un 7% en “otros” operaciones, el método II le dedica sobre un 42% de su tiempo accediendo a los datos de los grupos estructurales, que están siendo o han sido ya analizados. Otro resultado interesante es que el método III tarda casi un 70% de su tiempo evaluando  $\mathbf{L}$  y los ítems “otros”. Estos resultados confirman la ineficacia del método III, revelando que la eficiencia de los métodos globales está condicionada por la formación de la matriz Jacobiana, y abren paso a una mejora del método topológico II, tan pronto como se sea capaz de reducir el tiempo invertido en el desarrollo de “otros” cálculos.



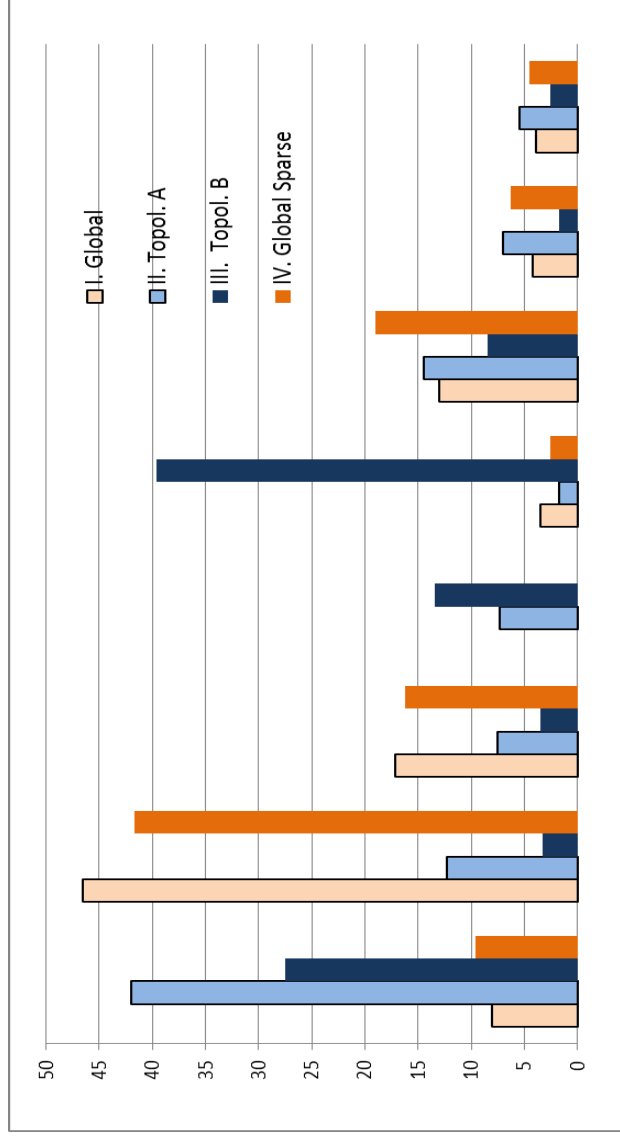


Figura 4.11: Cálculo de la distribución de tiempo (%) para las diferentes soluciones cinemáticas entre las principales operaciones, en el análisis cinemático del cuadrilátero escalable con 17 diadas (144 ecuaciones de restricción).

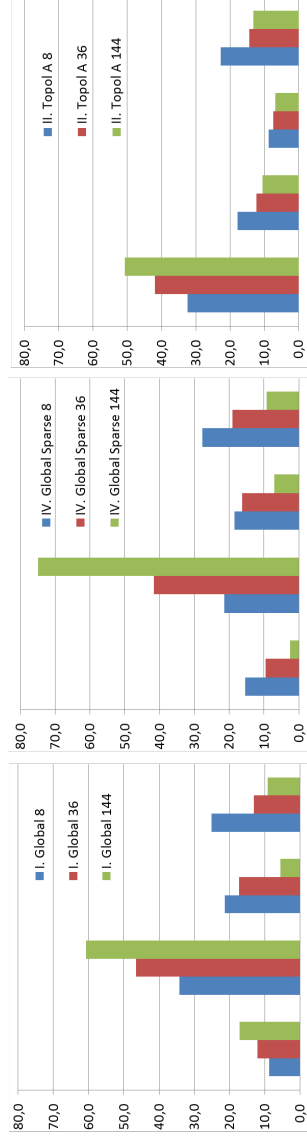


Figura 4.12: Distribución del porcentaje de tiempo entre los items más relevantes, dependiendo del número de ecuaciones de restricción para las tres formulaciones: I. Global (izquierda), II. Topological (derecha) and III. Topological (centro)

Todos los métodos (excepto el III) invierten un bajo porcentaje de su tiempo en la resolución de las velocidades y aceleraciones; esto nos permite estudiar en profundidad cómo es distribuido el tiempo en sus funciones principales, dependiendo del número de coordenadas (Figure 4.12).

Aunque el número de coordenadas influye en los tres métodos, el topológico muestra una reducción similar en el porcentaje de tiempo dedicado a realizar todas las operaciones, distintas del item “otros”. Sin embargo, este método es el único que muestra un aumento importante del tiempo empleado en “otras” operaciones cuando el número de restricciones se incrementa. Los métodos globales, por otro lado, muestran una fuerte sensibilidad a el número de ecuaciones de restricción, que es más importante para el global sparse (IV). Conforme se incrementa el número de ecuaciones de restricción, el porcentaje de tiempo empleado en evaluar la Jacobiana del sistema multicuerpo crece exponencialmente. Se puede observar que los métodos globales podrían mejorar considerando técnicas de matriz dispersa si el número de ecuaciones de restricción es relevante (por encima de 36) y si se usan métodos eficientes para almacenar los elementos de la matriz Jacobiana diferentes de cero.



## Capítulo 5

# Conclusiones y desarrollos futuros

En este capítulo se presentan las conclusiones sobre los materiales y modelos utilizados, así como sus resultados, para el análisis estructural y cinemático de mecanismos planos y espaciales. También se citan una serie de mejoras para una futura continuación del estudio iniciado en este proyecto.

### 5.1. Conclusiones

#### 5.1.1. Conclusiones generales

- Se ha empleado una formulación matemática (solución basada en grupos estructurales) para la simulación cinemática de sistemas multicuerpo planos y espaciales, basada en los resultados del análisis estructural.
- La formulación presentada se ha implementado en software de análisis matemático y los resultados han sido validados.
- Ha sido posible la reproducción real del movimiento de los sistemas multicuerpo planteados.

#### 5.1.2. Conclusiones específicas

##### Formulación global vs topológica

En este proyecto se ha introducido un método topológico, basado en la estructura cinemática del sistema multicuerpo, que automáticamente modela y resuelve su análisis cinemático. El método permite usar cualquier clase de coordenadas  $\mathbf{q}$ , aunque se ha de definir el conjunto de ecuaciones de restricción adecuado  $\Phi$ . Se presentó en el Capítulo 2 dos formulaciones para el método topológico: una en la que la matriz Jacobiana de las ecuaciones de restricción ( $\Phi_{\mathbf{q}}$ ) es derivada respecto al tiempo (2.2.1), y otra en la que esta matriz se deriva respecto a la coordenada independiente de todo el sistema  $\mathbf{z}$  (2.2.2). La eficiencia de esas formulaciones ha sido comparada con otras dos formulaciones en el Capítulo 4 (apartado 4.2.1) - globales (método I) y global sparse (método IV) - haciendo uso del cuadrilátero escalable, que nos permite controlar el número de ecuaciones de restricción. Los resultados obtenidos permiten numerar algunas ventajas de los métodos topológicos, basados en estructura cinemática presentada (II y III), respecto a los métodos globales (I y IV) usados para propósitos comparativos:

- La estructura cinemática se puede obtener computacionalmente en forma de matriz, la cuál puede ser usada para modelar y resolver la cinemática del sistema multicuerpo. Esto simplifica considerablemente la fase de modelado y evita cometer errores en esta primera etapa de análisis.
- Los métodos comparados no están limitados a una clase específica de coordenadas: (II) y (III) usan coordenadas naturales y mixtas, respectivamente. Sólo se necesita programar las subrutinas específicas que resuelven las diferentes clases de grupos estructurales, usando el tipo de coordenadas seleccionado.

- Las dos formulaciones diferentes, de análisis cinemático de un grupo estructural genérico, han sido introducidas de una forma sistemática, útil como guía para implementar nuevas subrutinas para otra clase de grupos estructurales.
- El método topológico III es mucho más complicado de programar en subrutina que el método topológico II. Esto no es a causa del tipo de coordenadas elegidas, sino por la evaluación de todos los coeficientes de velocidad  $\mathbf{K}$  y derivadas de coeficientes de velocidad  $\mathbf{L}$ . Aunque la matriz  $\mathbf{K}$  y el tensor de tercer orden  $\mathbf{L}$  podrían ser útiles para dinámica, análisis de sensibilidad o problemas de optimización, el tiempo necesitado para evaluarlos hace esta solución altamente ineficiente, incluso para un sistema con sólo un grado de libertad.
- Los métodos topológicos pueden ser extendidos a la cinemática y dinámica de sistemas multicuerpo espaciales, como hacen los métodos globales.
- El tiempo computacional crece literalmente en los métodos topológicos II y III, y exponencialmente en los métodos globales I y IV. La razón principal es el gran tamaño de las matrices de cada método: los métodos globales tratan con  $Dim(\Phi_{\mathbf{q}}) = m \times n$ , mientras los topológicos con  $Dim(\Phi_{\mathbf{q}}) = 2 \times 2$ , considerando el grupo estructural usado en este estudio.
- Cuando las técnicas de matriz dispersa son usadas en formulaciones globales (IV), el método es un poco más rápido que (I) para problemas por debajo de 100 restricciones. Sin embargo, desde este momento, las ventajas del método topológico II aumenta de forma lineal.

### Formulación empleada

En cuanto a la formulación matemática empleada para la simulación cinemática de sistemas multicuerpo, se pueden concretar las siguientes conclusiones específicas:

1. Permite el análisis cinemático de mecanismos escalables.
2. Sistematiza la solución del análisis estructural mediante criterios topológicos, que permiten una gran modularidad en la definición de las rutinas de análisis de grupo estructural.
3. Se ha aplicado a la modelización de un número determinado de grupos estructurales planos y espaciales.
4. Se ha dividido en cuatro pasos muy bien diferenciados, lo que facilita la inclusión de otros tipos de grupos estructurales a la librería creada.
5. Se puede aplicar a problemas de síntesis estructural y dimensional, así como al análisis de mecanismos espaciales y que incluyan sólidos deformables.

### Programación

En cuanto a su implementación en software de análisis matemático, se exponen las siguientes conclusiones específicas:

1. El método implementado se ha aplicado a dos mecanismos de distinta complejidad, contrastando la sencillez del cuadrilátero escalable con la dificultad del sistema de suspensión del camión de 5 ejes.
2. La implementación del análisis cinemático logra con éxito la obtención del movimiento real de mecanismos planos y espaciales.
3. El conjunto de funciones auxiliares y secundarias, desarrolladas en este proyecto para la resolución cinemática de grupos estructurales (en mecanismos espaciales), permite automatizar el proceso. Con estas funciones se generan las ecuaciones de restricción, Jacobiana y la matriz derivada temporal de la Jacobiana de todas las coordenadas de grupo, multiplicadas por sus velocidades. De una forma rápida y segura, bajo condiciones de par o de sólido rígido, se evita tener que introducir las matrices de resolución completas. A continuación se muestra la comparativa de tiempos de ejecución del programa, entre la resolución directa y el uso de funciones auxiliares. Se puede observar que no hay una gran variación de tiempos entre ambos métodos, por tanto es una ventaja el uso de las funciones.

Número de ejes	1	2	3	4	5
Solución directa	136.268s	253.283s	375.601	543.629s	652.489s
Solución f. auxiliares	149.224s	285.588s	419.806s	583.674s	694.120s

Cuadro 5.1: Comparativa de tiempos entre la resolución directa y mediante uso de funciones auxiliares.

### Sobre los resultados

1. No se observan errores acumulativos en ningún caso estudiado. Esto confirma, dado que ambos programas se basan en un análisis topológico, la eficacia de la formulación implementada, método de derivada temporal de la matriz Jacobiana.

## 5.2. Desarrollos futuros

Para este proyecto se ha analizado la cinemática de mecanismos planos y espaciales. Quedan para futuros desarrollos una amplia relación de aspectos, que continúan con el trabajo aquí iniciado y que no han podido ser abarcados en este proyecto por quedar fuera del alcance.

### Ampliación del campo de trabajo

1. Análisis dinámico de mecanismos espaciales: realización del análisis dinámico del mecanismo espacial presentado.
2. Ampliar la formulación cinemática del mecanismo espacial a métodos globales.
3. Elección del tipo de coordenadas: en este proyecto se han introducido las coordenadas naturales. Se puede implementar una nueva formulación basada en otro tipo de coordenadas.
4. Ampliar la biblioteca de grupos estructurales: deberá ampliarse la formulación que obtenga la contribución, a la ecuación del movimiento, de un mayor número de tipos de grupos estructurales que permitan analizar otros mecanismos. Así mismo, se programará esta formulación con el fin de ampliar la biblioteca de grupos estructurales.
5. Ampliar la formulación del método y su implementación a mecanismos con eslabones flexibles.
6. Ampliar la formulación a la solución de problemas de síntesis estructural y dimensional.
7. Incluir el cálculo de las fuerzas de ligadura, así como la hipótesis de rozamiento no nulo.
8. Posibilitar la selección del tipo de análisis, distinguiendo entre cinemático o dinámico, y dentro de este, estático, problema inverso o simulación dinámica.

### Programación

1. Modelización automática del mecanismo espacial.
2. Análisis dinámico computacional: implementar un programa de análisis dinámico para el mecanismo espacial y hacer una comparativa con otras formulaciones para estudiar la eficiencia.
3. Introducción de datos: crear un entorno gráfico en el que el usuario introduzca la información requerida por el programa, de una manera sencilla, fácil, visual y sin necesidad de la modificación de ningún fichero.
4. Integrador externo: emplear un integrador externo a Matlab que emplee un lenguaje de programación (C++, Java,...) más potente y trabaje con expresiones simbólicas.



## Capítulo 6

### Anexo de matrices

En este anexo se muestran las matrices necesarias para la resolución de la cinemática de los grupos estructurales  $2C3E1R$ ,  $1C1E1R$  y  $1C1E1P$ .

$$\Phi_{\varphi}(:, 1 : 11) = \begin{bmatrix}
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 2 \cdot (x_1 - x_{10}) & 2 \cdot (y_1 - y_{10}) & 2 \cdot (z_1 - z_{10}) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 v_{1x} & v_{1y} & v_{1z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 v_{2x} & v_{2y} & v_{2z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 2 \cdot (x_2 - x_{12}) & 2 \cdot (y_2 - y_{12}) & 2 \cdot (z_2 - z_{12}) & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & v_{8x} & v_{8y} & v_{8z} & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & v_{9x} & v_{9y} & v_{9z} & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 2 \cdot (x_3 - x_{11}) & 2 \cdot (y_3 - y_{11}) & 2 \cdot (z_3 - z_{11}) & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & v_{6x} & v_{6y} & v_{6z} & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & v_{7x} & v_{7y} & v_{7z} & 0 & 0 \\
 -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
 \end{bmatrix}$$

(6.1)

















$$\Phi_{\varphi} = \begin{bmatrix}
2 \cdot (x_3 - x_1) & 2 \cdot (y_3 - y_1) & 2 \cdot (z_3 - z_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 \cdot v_{1x} & 2 \cdot v_{1y} & 2 \cdot v_{1z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 \cdot v_{2x} & 2 \cdot v_{2y} & 2 \cdot v_{2z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_{1x} & v_{1y} & v_{1z} & (x_3 - x_1) & (y_3 - y_1) & (z_3 - z_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_{2x} & v_{2y} & v_{2z} & 0 & 0 & 0 & (x_3 - x_1) & (y_3 - y_1) & (z_3 - z_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & v_{2x} & v_{2y} & v_{2z} & v_{1x} & v_{1y} & v_{1z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_{4x} & v_{4y} & v_{4z} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_{4x} & v_{4y} & v_{4z} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_{5x} & v_{5y} & v_{5z} & v_{3x} & v_{3y} & v_{3z} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \cdot v_{5x} & 2 \cdot v_{5y} & 2 \cdot v_{5z} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \cdot v_{3x} & 2 \cdot v_{3y} & 2 \cdot v_{3z} & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0,00003 & 0 & 0 & -0,3606 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0,00003 & 0 & 0 & -0,3606 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0,00003 & 0 & 0 & -0,3606 & 0 & 0 \\
0 & 0 & 0 & v_{4x} & v_{4y} & v_{4z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_{6x} & v_{6y} & v_{6z} & 0 & 0 & 0 & 0 & \sin \theta
\end{bmatrix}$$

(6.9)



$$\Phi_{\varphi} = \begin{bmatrix}
2 \cdot (x_3 - x_1) & 2 \cdot (y_3 - y_1) & 2 \cdot (z_3 - z_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 \cdot v_{1x} & 2 \cdot v_{1y} & 2 \cdot v_{1z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 \cdot v_{2x} & 2 \cdot v_{2y} & 2 \cdot v_{2z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_{1x} & v_{1y} & v_{1z} & (x_3 - x_1) & (y_3 - y_1) & (z_3 - z_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_{2x} & v_{2y} & v_{2z} & 0 & 0 & 0 & (x_3 - x_1) & (y_3 - y_1) & (z_3 - z_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & v_{2x} & v_{2y} & v_{2z} & v_{1x} & v_{1y} & v_{1z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_{4x} & v_{4y} & v_{4z} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_{4x} & v_{4y} & v_{4z} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_{5x} & v_{5y} & v_{5z} & v_{3x} & v_{3y} & v_{3z} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \cdot v_{5x} & 2 \cdot v_{5y} & 2 \cdot v_{5z} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \cdot v_{3x} & 2 \cdot v_{3y} & 2 \cdot v_{3z} & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0,00003 & 0 & 0 & -0,3606 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0,00003 & 0 & 0 & -0,3606 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0,00003 & 0 & 0 & -0,3606 & 0 \\
0 & 0 & 0 & v_{4x} & v_{4y} & v_{4z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_{6x} & -v_{6y} & 0 & 0 & 0 & -v_{4x} \cdot \cos \theta
\end{bmatrix}$$

(6.10)



$$\Phi_{\mathbf{h}} = \begin{bmatrix} -2 \cdot (x_3 - x_1) & -2 \cdot (y_3 - y_1) & -2 \cdot (z_3 - z_1) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -v_{1x} & -v_{1y} & -v_{1z} & 0 & 0 & 0 & 0 & 0 & 0 \\ -v_{2x} & -v_{2y} & -v_{2z} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & v_{3x} & v_{3y} & v_{3z} \\ 0 & 0 & 0 & 0 & 0 & 0 & v_{5x} & v_{5y} & v_{5z} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -0,199 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -0,199 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -0,199 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\sin \theta & 0 & 0 \end{bmatrix} \quad (6.12)$$



$$\dot{\Phi}_{\mathbf{q}_G}(:, 12 : 25) = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 \cdot v_{1z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 \cdot v_{2x} & 2 \cdot v_{2y} & 2 \cdot v_{2z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
(\dot{z}_3 - \dot{z}_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & (\dot{x}_3 - \dot{x}_1) & (\dot{y}_3 - \dot{y}_1) & (\dot{z}_3 - \dot{z}_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_{2z} & v_{1x} & v_{1y} & v_{1z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & v_{4x} & v_{4y} & v_{4z} & v_{3x} & v_{3y} & v_{3z} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & v_{5x} & v_{5y} & v_{5z} & v_{4x} & v_{4y} & v_{4z} & 0 \\
0 & 0 & 0 & 0 & v_{5x} & v_{5y} & v_{5z} & 0 & 0 & 0 & v_{3x} & v_{3y} & v_{3z} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \cdot v_{5x} & 2 \cdot v_{5y} & 2 \cdot v_{5z} & 0 \\
0 & 0 & 0 & 0 & 2 \cdot v_{3x} & 2 \cdot v_{3y} & 2 \cdot v_{3z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_{4z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & v_{6x} & v_{6y} & v_{6z} & 0 & 0 & 0 & 0 & 0 & 0 & \cos \theta \cdot \dot{\theta}
\end{bmatrix}$$

(6.14)



$$\dot{\Phi}_{\mathbf{q}_G}(:, 12 : 25) = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 \cdot \dot{v}_{1z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 \cdot \dot{v}_{2x} & 2 \cdot \dot{v}_{2y} & 2 \cdot \dot{v}_{2z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
(\dot{z}_3 - \dot{z}_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & (x_3 - x_1) & (y_3 - y_1) & (z_3 - z_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\dot{v}_{2z} & \dot{v}_{1x} & \dot{v}_{1y} & \dot{v}_{1z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \dot{v}_{4x} & \dot{v}_{4y} & \dot{v}_{4z} & \dot{v}_{3x} & \dot{v}_{3y} & \dot{v}_{3z} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \dot{v}_{5x} & \dot{v}_{5y} & \dot{v}_{5z} & \dot{v}_{4x} & \dot{v}_{4y} & \dot{v}_{4z} & 0 \\
0 & 0 & 0 & 0 & \dot{v}_{5x} & \dot{v}_{5y} & \dot{v}_{5z} & 0 & 0 & 0 & \dot{v}_{3x} & \dot{v}_{3y} & \dot{v}_{3z} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \cdot \dot{v}_{5x} & 2 \cdot \dot{v}_{5y} & 2 \cdot \dot{v}_{5z} & 0 \\
0 & 0 & 0 & 0 & 2 \cdot \dot{v}_{3x} & 2 \cdot \dot{v}_{3y} & 2 \cdot \dot{v}_{3z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\dot{v}_{4z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \dot{v}_{6x} & -\dot{v}_{6y} & -\cos \theta \cdot \dot{\theta} & 0 & 0 & 0 & 0 & 0 & -\dot{v}_{4x} \cdot \cos \theta + \dot{v}_{4x} \cdot \sin \theta \cdot \dot{\theta}
\end{bmatrix} \quad (6.16)$$

$$\Phi_{\varphi} = \begin{bmatrix}
2 \cdot v_{1x} & 2 \cdot v_{1y} & 2 \cdot v_{1z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2 \cdot v_{2x} & 2 \cdot v_{2y} & 2 \cdot v_{2z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 \cdot v_{3x} & 2 \cdot v_{3y} & 2 \cdot v_{3z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_{2x} & v_{2y} & v_{2z} & v_{1x} & v_{1y} & v_{1z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
v_{3x} & v_{3y} & v_{3z} & 0 & 0 & 0 & v_{1x} & v_{1y} & v_{1z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & v_{3x} & v_{3y} & v_{3z} & v_{2x} & v_{2y} & v_{2z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \cdot v_{4x} & 2 \cdot v_{4y} & 2 \cdot v_{4z} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \cdot v_{5x} & 2 \cdot v_{5y} & 2 \cdot v_{5z} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_{5x} & v_{5y} & v_{5z} & v_{4x} & v_{4y} & v_{4z} & 0 & 0 \\
v_{4x} & v_{4y} & v_{4z} & 0 & 0 & 0 & 0 & 0 & 0 & v_{1x} & v_{1y} & v_{1z} & 0 & 0 & 0 & 0 & 0 \\
v_{5x} & v_{5y} & v_{5z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_{1x} & v_{1y} & v_{1z} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \cdot \mathbf{s} \\
(y_1 - y_2) & (x_2 - x_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
(z_2 - z_1) & 0 & (x_1 - x_2) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & (z_1 - z_2) & (y_2 - y_1) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & v_{5x} & v_{5y} & v_{5z} & 0 & 0 & 0 & 0 & 0 & 0 & v_{2x} & v_{2y} & v_{2z} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & v_{6x} & v_{6y} & v_{6z} & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

(6.17)



$$\Phi_{\mathbf{h}} = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-2 \cdot (x_2 - x_1) & -2 \cdot (y_2 - y_1) & -2 \cdot (z_2 - z_1) & 2 \cdot (x_2 - x_1) & 2 \cdot (y_2 - y_1) & 2 \cdot (z_2 - z_1) & 0 & 0 & 0 \\
-v_{1y} & v_{1x} & 0 & v_{1y} & -v_{1x} & 0 & 0 & 0 & 0 \\
v_{1z} & 0 & -v_{1x} & -v_{1z} & 0 & v_{1x} & 0 & 0 & 0 \\
0 & -v_{1z} & v_{1y} & 0 & v_{1z} & -v_{1y} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & v_{4x} & v_{4y} & v_{4z}
\end{bmatrix}$$

(6.18)



$$\dot{\Phi}_{\mathbf{q}_G}(:, 12 : 25) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 \cdot \dot{v}_{2z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 \cdot \dot{v}_{3x} & 2 \cdot \dot{v}_{3y} & 2 \cdot \dot{v}_{3z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \dot{v}_{1z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \dot{v}_{1x} & \dot{v}_{1y} & \dot{v}_{1z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \dot{v}_{3z} & \dot{v}_{2x} & \dot{v}_{2y} & \dot{v}_{2z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \cdot \dot{v}_{4x} & 2 \cdot \dot{v}_{4y} & 2 \cdot \dot{v}_{4z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \cdot \dot{v}_{5x} & 2 \cdot \dot{v}_{5y} & 2 \cdot \dot{v}_{5z} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dot{v}_{5x} & \dot{v}_{5y} & \dot{v}_{5z} & \dot{v}_{4x} & \dot{v}_{4y} & \dot{v}_{4z} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dot{v}_{1x} & \dot{v}_{1y} & \dot{v}_{1z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dot{v}_{1x} & \dot{v}_{1y} & \dot{v}_{1z} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \cdot \dot{s} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \dot{v}_{5z} & 0 & 0 & 0 & 0 & 0 & 0 & \dot{v}_{2x} & \dot{v}_{2y} & \dot{v}_{2z} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dot{v}_{6x} & \dot{v}_{6y} & \dot{v}_{6z} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (6.20)$$



# Bibliografía

- [1] J. GarcíaOrden, Dinámica no lineal de sistemas multicuerpo flexibles mediante algoritmos conservativos, Master's thesis, Universidad Politécnica de Madrid (1999).
- [2] J. Gómez-Cristóbal, Método de síntesis dimensional óptima de sistemas multicuerpo con restricciones dinámicas. aplicación al diseño de mecanismos planos, Master's thesis, Universidad de la Rioja (2003).
- [3] M. kolovsky, A. Evgrafov, Y. Semenov, A. Slousch, Advanced theory of mechanism and machines, Springer, 2000.
- [4] A. Shabana, Dynamics of Multibody systems, 2nd Edition, Cambridge Univ Press, 1998.
- [5] M. Gerardin, A. Cardona, Flexible Multibody Dynamics: A Finite element approach, John Wiley & Sons, 2001.
- [6] F. Amirouche, Computational methods in multibody dynamics, Prentice Hall, Englewood Cliffs, NJ, 1992.
- [7] E. Haug, Computer-Aided Kinematics and Dynamics of Mechanical Systems, Allyn and Bacon, Boston, MA, 1989.
- [8] J. G. de Jalon, E. Bayo, Kinematic and Dynamic Simulation of Multibody Systems, Springer-Verlag, New York, 1993.
- [9] P. Nikravesh, Computer Aided Analysis of Mechanical Systems, Prentice Hall, Englewood Cliffs, NJ, 1988.
- [10] R. E. Roberson, R. Schwertassek, Dynamics of Multibody Systems, Springer-Verlag, New York, 1988.
- [11] A. Shabana, Computational Dynamics, 2nd Edition, John Wiley & Sons, 2001.
- [12] J. Cuadrado, D. Dopico, A combined penalty and semi-recursive formulation for closed-loops in mbs, in: T. Huang (Ed.), Proceedings of the 11th World Congress in Mechanism and Machine Science, China Machine Press, Tianjin, China, 2004.
- [13] J. Wittenburg, Dynamics of systems of rigid bodies, BF Teubner, Stuttgart, 1977.
- [14] W. Book, Recursive lagrangian dynamics of flexible manipulator arms, Int. J. Robot. Res. 3 (3) (1984) 87–101.
- [15] T. Mruthyunjaya, Kinematic structure of mechanisms revisited, Mechanism and Machine Theory 38 (3) (2003) 279–320.
- [16] P. Shi, J. Mcphee, Dynamics of flexible multibody systems using virtual word and linear graph theory, Multibody Syst. Dyn. (4) (2000) 355–381.
- [17] J. J. Mcphee, On the use of linear graph theory in multibody systems dynamics, Nonlinear dynamics (9) (1996) 73–90.
- [18] Y. H. AA. Shabana, R. Wehage, Projection methods in flexible multibody dynamics, part i: Kinematics, partii: Dynamics and recursive projection methods, Int. J. Numer. Methods Eng. 35 (1992) 1927–1966.
- [19] S. Kim, E. Haug, A recursive formulation for flexible multibody dynamics, part ii: Closed loop systems, Comput. Methods Appl. Mech. Eng. 74 (1989) 251–269.

- [20] J. Keat, Multibody system order n dynamics formulation based on velocity transfer method, *J. Guid. Control Dyn.* 13 (2) (1990) 207–212.
- [21] Y. Hwang, A. Shabana, Dynamics of flexible multibody space cranes using recursive projection methods, *Comput. Struct.* 43 (3) (1992) 549–564.
- [22] D. J. P. Fiset, J. Samin, A fully symbolic generation of the equations of motion of multibody systems containing flexible beams, *Comput. Methods Appl. Mech. Eng.* 142 (1997) 123–152.
- [23] J. S. P. Fiset, P. Willems, Contribution to symbolic analysis of deformable multibody systems, *Int. J. Numer. Methods Eng.* 32 (1991) 1621–1635.
- [24] E. Melzer, Symbolic computations in flexible multibody systems, *Nonlinear Dyn.* 9 (1-2) (1996) 147–163.
- [25] P. Shi, J. McPhee, Symbolic programming of a graph theoretic approach to flexible multibody dynamics, *Mech. Struct. Mach.* 30 (1) (2002) 123–154.
- [26] Y. Khulief, A. Shabana, Dynamics of multibody systems with variable kinematic structure, *ASME J. Mech., Transm. Autom. Des.* (108) (1986) 167–175.
- [27] E. Haug, J. Yen, Implicit numerical integration of constrained equations of motion via generalized coordinate partitioning, *ASME J. Mech. Des.* (114) (1992) 296–304.
- [28] P. Fiset, B. Vaneghem, Numerical integration of multibody system dynamic equations using the coordinate partitioning method in an implicit newmark scheme, *Comput. Methods Appl. Mech. Eng.* 135 (1-2) (1992) 85–106.
- [29] O. Verlinden, Approach based on minimal coordinates, *Tech. rep.*, Faculté polytechnique de Mons (2010).
- [30] F. Cugnon, A. Cardona, A. Selvi, C. Paleczny, M. Pucheta, Synthesis and optimization of flexible mechanisms, in: C. L. Bottasso (Ed.), *Multibody Dynamics: Computational Methods and Applications*, Springer, ECCOMAS, 2009, pp. 81–93.
- [31] M. Leger, J. McPhee, Selection of modeling coordinates for forward dynamics multibody simulations, *Multibody System Dynamics* 18 (18) (2007) 277–297.
- [32] T. S. Mruthyunjaya, Kinematic structure of mechanisms revisited, *Mechanism and Machine Theory* 38 (4) (2003) 279–320.
- [33] O. Verlinden, G. Kouroussis, S. Datoussaid, C. Conti, Open source symbolic and numerical tools for the simulation of multibody systems, *Tech. rep.* (2003).
- [34] I. Popescu, D. B. Marghitu, Structural design of planar mechanisms with dyads, *Multibody System Dynamics* 19 (19) (1997) 407–425.
- [35] A. Kecskeméthy, T. Krupp, M. Hiller, Symbolic processing of multiloop mechanism dynamics using closed-form kinematics solution, *Multibody System Dynamics* 1 (1) (1997) 23–45.
- [36] H. Varbanov, T. Yankova, K. Kulev, S. Lilov, S&A - Expert system for planar mechanism design, *Expert systems with applications* 31 (31) (2006) 558–569.
- [37] M. Hansen, A general method for analysis of planar mechanisms using a modular approach, *Mechanism and Machine Theory* 31 (8) (1996) 1155–1166.
- [38] A. Slijoka, O. Shai, W. Whiteley, Cheking mobility and decompositions of linkages via pebble game algorithm, in: *Proceedings of the ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Washington, DC, USA, 2011, pp. 1–10.
- [39] T. Maria, S. Offer, Applying rigidity theory methods for topological decompositions and synthesis of gear trains systems, in: *Proceedings of the ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Chicago, IL, USA, 2012, pp. 1–10.
- [40] T. S. Mruthyunjaya, M. Raghavan, Computer aided analysis of the structure of kinematic chains, *Mechanism and Machine Theory* 19 (19) (1984) 357–368.

- [41] T. S. Mruthyunjaya, A computerized methodology for structural synthesis of kinematic chains: Part 1: Formulation, *Mechanism and Machine Theory* 19 (19) (1984) 487–495.
- [42] R. Ravinsankar, T. S. Mruthyunjaya, Computerized synthesis of the structure of geared kinematic chains, *Mechanism and Machine Theory* 20 (20) (1985) 367–387.
- [43] C. U. Galletti, A note on modular approach to planar linkage kinematic analysis, *Mechanism and Machine Theory* 21 (21) (1986) 385–391.
- [44] S. Li, Computer-aided structure synthesis of spatial kinematic chains, *Mechanism and Machine Theory* 25 (25) (1990) 645–653.
- [45] S. Mauskar, S. Krishnamurthy, A loop configuration approach to automatic sketching of mechanisms, *Mechanism and Machine Theory* 31 (31) (1996) 427–437.
- [46] W. M. Hwang, Y. Hwang, Computer-aided structural synthesis of planar kinematic chains with simple joints, *Mechanism and Machine Theory* 27 (27) (1992) 189–199.
- [47] N. P. Belfiore, E. Penestri, Automatic sketching of planar kinematic chains, *Mechanism and Machine Theory* 29 (29) (1994) 177–193.
- [48] E. R. Tuttle, Generation of planar kinematic chains, *Mechanism and Machine Theory* 31 (31) (1996) 729–748.
- [49] A. C. Rao, J. Anne, Topology based characteristics of kinematic chains: workspace, rigidity, input-joint and isomorphism, *Mechanism and Machine Theory* 33 (33) (1998) 625–638.
- [50] S. Mitsi, K. D. Bouzakis, G. Mansour, I. Popescu, Position analysis in polynomial form of planar mechanisms with assur groups of class 3 including revolute and prismatic joints, *Mechanism and Machine Theory* 38 (2003) 1325–1344.
- [51] J. Buskiewicz, A method of optimization of solving a kinematic problem with the use of structural analysis algorithm (sam), *Mechanism and Machine Theory* 41 (2006) 823–837.
- [52] H. Ding, Z. Huang, D. Mu, Computer-aided structure decomposition theory of kinematic chains and its applications, *Mechanism and Machine Theory* 43 (12) (2008) 1596–1609.
- [53] X. Ding, Y. Yang, J. S. Dai, Topology and kinematic analysis of color-changing ball, *Mechanism and Machine Theory* (46) (2011) 67–81.
- [54] M. Saura, Utilidad del analisis estructural en el analisis cinematico de mecanismos articulados planos, in: CIBEM (Ed.), Proceedings of the Congreso IberoAmericano de Ingenieria Mecanica, Cuba, 2005, pp. 1–11.
- [55] O. Shai, Topological synthesis of all 2d mechanisms through assur graphs, in: Proceedings of the ASME 2010 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Montreal, Quebec, Canada, 2010, pp. 1–11.
- [56] G. Z. Baranov, A Course in the Theory of Mechanisms and Machines, Mir, 1988.