

Estrategias de reordenamiento de matrices huecas para el tratamiento óptimo de las ecuaciones de la dinámica de mecanismos

Autor: José Augusto Antón Nacimiento

Tesis doctoral UDC / 2015

Director: Jesús Cardenal Carro

Departamento de Ingeniería Industrial II

A mis padres

Agradecimientos

Al finalizar esta Tesis me gustaría expresar mi agradecimiento a todas las personas que han contribuido de alguna forma a la realización de este trabajo.

Sobre todo querría agradecer a mi director de Tesis, Jesús Cardinal, toda su ayuda y su paciencia durante estos largos años. También quiero agradecer a los profesores Alvaro Deibe y Ana Díaz el interés mostrado por el desarrollo de este trabajo y su disponibilidad.

Querría expresar mi agradecimiento al profesor Javier García de Jalón: esta tesis es deudora, de una forma u otra, de muchas de sus aportaciones y también de sus ideas, que tuve ocasión de conocer de primera mano en una estancia en el año 2007.

Finalmente, un agradecimiento especial a toda mi familia; sin ella esto no habría sido posible.

Resumen

El marco general de esta Tesis es la resolución de la dinámica de un sistema multicuerpo con elementos rígidos, tanto de cadena abierta como cerrada. La finalidad es buscar un algoritmo de reordenamiento de matrices huecas que permita mejorar la eficiencia en la resolución de la dinámica. Para lograr este objetivo se hace un estudio de los modelos y formulaciones más utilizados en el ámbito de los métodos globales, y de los reordenamientos más comunes de matrices huecas. Estos reordenamientos se han aplicado al problema de resolución de mecanismos consiguiéndose una mejora significativa en la eficiencia de la simulación. Finalmente se ha propuesto una modificación de uno de estos reordenamientos, que tiene en cuenta algunas particularidades de los sistemas que se estudian. Se ha validado este nuevo reordenamiento y se ha contrastado su eficiencia utilizando diferentes sistemas multicuerpo y los reordenamientos estudiados.

Resumo

O marco xeral desta Tese é a resolución da dinámica dun sistema multicorpo con elementos ríxidos, tanto de cadea aberta como pechada. A finalidade é buscar un algoritmo de reordenamento de matrices ocas que permita mellorar a eficiencia na resolución da dinámica. Para lograr este obxectivo faise un estudo dos modelos e formulacións

máis utilizados no ámbito dos métodos globais, e dos reordenamentos máis comúns de matrices ocas. Estes reordenamientos aplicáronse ao problema de resolución de mecanismos conseguíndose unha mellora significativa na eficiencia da simulación. Finalmente propúxose unha modificación dun destes reordenamentos, que ten en conta algunhas particularidades dos sistemas que se estudan. Validouse este novo reordenamento e contrastouse a súa eficiencia utilizando diferentes sistemas multicorpo e os reordenamientos estudados.

Abstract

The general frame of this Thesis is the resolution of the dynamics of a multibody system with rigid elements, both in open and closed chain form. The main objective is to look for an ordering algorithm of sparse matrices that allow to improve the efficiency in the solution of the dynamics. In order to achieve this goal, an in depth study of the common multibody dynamics models and formulations is performed. A complementary study on the ordering algorithms for sparse matrices is also done. These algorithms have applied to the resolution of mechanisms, achieving a significant improvement in the efficiency of the simulation. Finally, a modification of one of these algorithms is proposed, that takes into account some peculiarities of the systems. This new algorithm has been validated, and its efficiency has been contrasted using different multibody systems and the orderings algorithms for sparse matrices already studied.

Índice general

Agradecimientos	v
Resumen	vii
1. Introducción	1
2. Sistemas multicuerpo	7
2.1. Definición de sistemas multicuerpo	7
2.2. Interés de la simulación de mecanismos	10
3. Simulación computacional de mecanismos	15
3.1. Revisión bibliográfica	16
3.2. Modelo del sistema multicuerpo	17
3.3. Las ecuaciones del movimiento	23
3.4. Espacio de velocidades posibles	24
3.5. Ecuaciones de la dinámica	27
3.5.1. Métodos globales y topológicos	29
3.5.2. Formulaciones basadas en coordenadas depen- dientes	30
3.5.3. Estabilización de las ecuaciones	36
3.5.4. Formulaciones basadas en penalizadores	39

3.5.5.	Formulaciones basadas en coordenadas independientes	42
3.5.6.	Coordenadas independientes y proyección de velocidades	44
3.5.7.	Método elegido	46
3.6.	Cuestiones numéricas	46
3.6.1.	Integradores	47
3.6.2.	Eficiencia en la integración	47
3.6.3.	Estructura de la función objetivo	48
3.6.4.	Características de la matriz jacobiana	50
3.6.5.	Estrategias de reordenamiento	51
3.6.6.	Resolución por bloques	53
4.	Estrategias de reordenamiento de matrices huecas	57
4.1.	Cuestiones generales de teoría de grafos	58
4.1.1.	Definiciones previas	58
4.1.2.	Factorización de Gauss	63
4.2.	Reordenamientos de matrices	67
4.2.1.	Disyuntiva entre <i>estrategias locales</i> frente a <i>estrategias globales</i>	69
4.2.2.	Algoritmos <i>minimum degree</i>	70
4.2.3.	Algoritmos Cuthill-McKee	71
4.2.4.	Algoritmos <i>Nested Dissection</i>	73
4.3.	Reducción a una forma triangular por bloques	74
4.3.1.	Búsqueda de un trasversal	75
4.3.2.	Permutaciones simétricas	76
4.4.	Reducción a una forma triangular por bloques bordeada	77
4.5.	Aplicación al problema de mecanismos	80
4.6.	Reordenamiento propuesto	81

5. Resultados	85
5.1. Consideraciones generales	85
5.2. Péndulo de cinco barras	87
5.3. Mecanismo <i>Theter</i>	89
5.4. Mecanismo de Bricard	93
5.5. Modelo de vehículo completo	99
6. Conclusiones	105
6.1. Conclusiones	105
6.2. Futuras líneas de investigación	106

Índice de figuras

2.1. Ejemplo de sistema multicuerpo	8
2.2. Ejemplos de cadena abierta y cerrada	9
3.1. Ejemplo de matriz jacobiana	50
3.2. Efecto de la factorización de Gauss	52
3.3. Reordenamiento Cuthill-McKee inverso	53
3.4. Reordenamiento P^4	53
4.1. Grafos dirigidos y no dirigidos	59
4.2. Grafo y matriz asociada	60
4.3. Eliminación de Gauss: Ejemplo con una matriz simétrica	65
4.4. Proceso de eliminación de Gauss: matriz simétrica . .	66
4.5. Algoritmo Minimum degree	71
4.6. Rejilla M de un problema típico de elementos finitos con los nodos asociados. (a) Etiquetado original. (b) Re-etiquetado de acuerdo con el criterio de Cuthill- MacKee modificado (Reverse Cuthill-MacKee)	73
4.7. Ejemplo de búsqueda de transversal	76
4.8. Matriz y su grafo asociado	77
4.9. Matriz general y grafo asociado después de haber re- conocido las componentes <i>strong</i> existentes y realizar el reordenamiento	78

4.10. Reordenamiento P^4	78
4.11. Paso intermedio de P^5	79
5.1. Péndulo de cinco barras	87
5.2. Péndulo. Matriz $\Phi_{\mathbf{q}}^d$ inicial y reordenada	88
5.3. Péndulo. Restricciones	90
5.4. Mecanismo <i>Theter</i>	90
5.5. Mecanismo <i>Theter</i> . Matriz $\Phi_{\mathbf{q}}^d$	91
5.6. Mecanismo <i>Theter</i> . Restricciones	92
5.7. Mecanismo de Bricard	93
5.8. Matriz $\Phi_{\mathbf{q}}^d$ inicial	94
5.9. Matriz $\Phi_{\mathbf{q}}^d$ reordenada	95
5.10. Matriz $\Phi_{\mathbf{q}}^d$ reordenada fijando las coordenadas relativas	96
5.11. Matriz $\Phi_{\mathbf{q}}^d$ reordenada fijando las coordenadas relativas e indicando las ecuaciones de cierre	96
5.12. Matriz $\Phi_{\mathbf{q}}^d$ inicial y reordenada (2)	97
5.13. Matriz $\Phi_{\mathbf{q}}^d$ reordenada indicando coordenadas relativas (2)	98
5.14. Bricard: Restricciones	99
5.15. Bricard: Energía total	99
5.16. Suspensión multibrazo de cinco barras	100
5.17. Suspensión delantera	100
5.18. Automóvil: matriz $\Phi_{\mathbf{q}}^d$ y reordenamiento P^5	101
5.19. Automóvil: sistema abierto y reordenamiento P^5	102
5.20. Automóvil: reordenamiento propio	103
5.21. Automóvil: restricciones	104
5.22. Automóvil: energía total	104

Índice de tablas

3.1. Número de evaluaciones	48
4.1. Coste de la factorización: matrices llenas	64
4.2. Coste de la factorización: matrices huecas	65
5.1. Péndulo de cinco barras: características	88
5.2. Tabla de tiempos para el péndulo de cinco barras . . .	89
5.3. Mecanismo de <i>Theter</i> : características	91
5.4. Tabla de tiempos para el mecanismo de <i>Theter</i> . . .	92
5.5. Mecanismo de Bricard: características (1)	94
5.6. Mecanismo de Bricard: características (2)	97
5.7. Tabla de tiempos para el mecanismo de Bricard . . .	98
5.8. Vehículo completo: características	101
5.9. Tabla de tiempos para el vehículo completo	103

Capítulo 1

Introducción

En cualquier campo de la ingeniería, la simulación computacional ha ido adquiriendo a lo largo de las últimas décadas una gran importancia porque permite acelerar los procesos de diseño, entrenamiento, etc; los hace más económicos, posibilita además dotar de cierta capacidad *inteligente* a ciertos dispositivos, etc.

En este sentido, en el campo de los mecanismos o de los sistemas multicuerpo, también ha habido una revolución. Desde los análisis gráficos anteriores a los años 70, extraordinariamente complejos e ingeniosos en la mayor parte de los casos pero también tremendamente limitados en cuanto a la cantidad y calidad de los resultados, se ha pasado en la actualidad a la capacidad de simular computacionalmente sistemas complejos con un coste moderado tanto en el tiempo requerido para definir el mecanismo como en la parte de la obtención de resultados. Además las características gráficas de los sistemas computacionales modernos hacen más atractivos e intuitivos estos resultados.

Esta revolución ha sido posible por un lado gracias a la evolución constante de los ordenadores en términos de potencia de cálculo (número de operaciones por segundo), en capacidad gráfica, en coste tanto del equipo como de su mantenimiento e instalación, en tamaño, ... Pero además, la revolución de la simulación se sustenta en la fecundidad de las últimas décadas en cuanto a propuestas de alternativas sobre el modelo matemático de los sistemas multicuerpo

y las formulaciones de los problemas cinemático y dinámico a que conducen.

El resultado es que, tal como se señalaba más arriba, es posible llevar a cabo simulaciones complejas con resultados contrastables en plazos relativamente ajustados. Y el *tiempo real* que en los años 90 se consideraba un reto es hoy en día a la vez una realidad y un reto sostenido. Es una realidad porque es posible efectuar la simulación del comportamiento dinámico de un sistema complejo en un tiempo igual o inferior al simulado: la integración en el tiempo consume menos tiempo real que el que se está simulando, con tal de que la potencia de cálculo sea suficiente. Esta *potencia de cálculo suficiente* está al alcance de cualquiera en estos momentos porque la proporciona un PC común, mientras que en los años 90 requería un ordenador sofisticado y generalmente caro.

Pero, a la vez que es una realidad, continúa siendo un reto. El avance en los algoritmos numéricos de la formulación y la potencia de cálculo de los procesadores modernos ofrece nuevas posibilidades: lo que en los años 90 se hacía con un *mainframe* o una estación de trabajo ahora se consigue con un PC. Y, ¿no sería posible hacerlo con un teléfono móvil o con un sistema embebido?

Al comienzo de esta introducción se hace referencia a dotar de cierta *inteligencia* a los dispositivos. En muchos casos esto requiere que sean capaces de efectuar una simulación. Piénsese, por ejemplo en los sistemas de control con *hardware in the loop*. El reto del tiempo real continúa, entonces, solo que adquiere otras formas y matices.

De ahí que todos los pasos que se puedan dar en el sentido de hacer más eficientes los algoritmos de simulación tienen cabida en el ámbito de la investigación. Si se consiguen mejoras sustanciales en este ámbito, los procesadores necesarios para efectuar ciertas simulaciones podrán ser más sencillos, asequibles, económicos, ecológicos, etc. o, visto desde otra perspectiva, será posible encarar la solución de problemas *más grandes* o complejos sin perder ese *tiempo real*.

El trabajo que se desarrolla en la presente tesis doctoral se encuadra en estas tareas de investigación que persiguen mejorar la eficiencia de la simulación; en este caso concreto, en la simulación de la dinámica de mecanismos.

El esfuerzo computacional para integrar las ecuaciones de la dinámica depende fundamentalmente del modelo matemático elegido para definir el mecanismo, de la propia formulación de la dinámica y de que se exploten al máximo las técnicas numéricas para sacar partido de las peculiaridades del problema que se pretende resolver.

Es en este último aspecto donde se encuadran las contribuciones del presente trabajo: se emplean técnicas de matrices dispersas para matrices no simétricas con objeto de mejorar la eficiencia en términos de rendimiento o tiempo de ejecución del proceso de integración de las ecuaciones de la dinámica.

Objetivos y Metodología

El objetivo es pues, discutir la ventaja de aplicar técnicas de matrices dispersas en la solución de las ecuaciones de la dinámica de mecanismos. Así, de una forma general se puede considerar que no es un tema completamente novedoso. Sin embargo, aunque hay trabajos en la bibliografía que tratan cuestiones de este tipo, por regla general, debido a que estas técnicas de matrices dispersas están desarrolladas fundamentalmente para matrices simétricas, no se aplican a las formulaciones en las que la matriz característica no tiene esta estructura o ni siquiera es cuadrada.

Ese objetivo de determinar el beneficio que se deriva de emplear reordenamientos ideados para matrices no simétricas, se complementa con el desarrollo de una nueva estrategia, fundamentada en un procedimiento conocido, pero acondicionado para la estructura específica de las matrices características que aparecen en la dinámica, que conduce a resultados mejores en términos de eficacia computacional, generalidad y estabilidad.

Para alcanzar este objetivo, se hace un repaso de las cuestiones que afectan a la modelización matemática de los mecanismos, se estudian las formulaciones de la dinámica dejando constancia de las alternativas que existen y remarcando las ventajas e inconvenientes de usar uno u otro procedimiento. A la vista de estos planteamientos, se justifica la elección de un modelo y una formulación concreta.

Hecha esta elección, se pormenorizan los aspectos numéricos de su implementación, tratando de identificar aquellas partes del proceso que, supuesto que fuesen mejoradas, redundarían de manera más acusada en beneficio de la eficiencia global. Es aquí donde se señala como operación más costosa la resolución sucesiva de un sistema de ecuaciones. Se ponen de manifiesto las peculiaridades de la matriz característica y se concluye que es susceptible de un tratamiento específico que debe mejorar la eficiencia del proceso.

El estudio de los procedimientos para el tratamiento de matrices huecas, conduce a elegir los más adecuados para la estructura del problema. Se analizan las ventajas, inconvenientes e incluso la viabilidad de aplicar estos procedimientos para tomar una decisión acerca de los más adecuados.

Determinada la estrategia más ventajosa, se adaptan los métodos al problema concreto y se validan los resultados mediante la aplicación del procedimiento ideado a una serie de problemas tipo que abarcan la generalidad de los casos que se pueden estudiar.

Así se puede establecer el rango de aplicación de los métodos, el incremento en el rendimiento que producen y los posibles inconvenientes que aparecen. Estas cuestiones permiten explicitar una serie de conclusiones de la investigación.

Estructura del documento

Tal como se detalla en el apartado anterior, el contenido del documento se refiere a los modelos matemáticos de los mecanismos, a las formulaciones de las ecuaciones de la dinámica y su tratamiento numérico, a repasar las técnicas para matrices dispersas y su posible aplicación al problema concreto de la dinámica de mecanismos y a estudiar su efecto en el estudio de ciertos casos.

En concreto, después del primer capítulo, en que se hace una introducción general al problema del que trata la tesis, sigue otro, el dos, dedicado a introducir ciertos conceptos sobre los sistemas multi-cuerpo, también generales, pero imprescindibles para los desarrollos posteriores.

En el capítulo tres se describe sistemáticamente la simulación de mecanismos, comenzando por el modelo matemático de los sistemas multicuerpo y sus enlaces y después analizando las diferentes formulaciones de la dinámica tanto en su aspecto teórico como en sus implicaciones numéricas. A la vista de las alternativas posibles, se justifica la elección de una de estas formulaciones para los desarrollos posteriores y se discuten los pormenores numéricos de su implementación.

El capítulo cuatro se dedica a explicar las técnicas de reordenamiento de matrices huecas. En ese estudio se deja constancia de cuáles son aplicables al problema que se pretende resolver. Se analizan las ventajas e inconvenientes de cada método y, a la vista de éstas, se desarrolla una variante del algoritmo más eficaz, que resulta especialmente apropiado para la estructura de la matriz característica de la formulación de la dinámica escogida.

En el capítulo cinco se analizan cuatro problemas tipo. Se recogen los resultados numéricos que justifican la ventaja de emplear las técnicas de matrices huecas propuestas en la tesis para resolver el problema de la dinámica de mecanismos.

Por último el capítulo seis compendia las conclusiones a las que se llega tras el desarrollo planteado y las pruebas efectuadas y propone además nuevas líneas de investigación que quedan abiertas con el trabajo desarrollado en esta tesis doctoral.

Capítulo 2

Sistemas multicuerpo

2.1. Definición de sistemas multicuerpo

Se define un *sistema multicuerpo* como un conjunto dos o más sólidos rígidos (también llamados elementos) parcialmente unidos entre sí, permitiendo la posibilidad de un movimiento relativo entre ellos. Cada una de estas uniones entre sólidos rígidos que constituyen el sistema multicuerpo se llaman pares cinemáticos o simplemente pares. Todos los pares permiten ciertos movimientos y restringen o previenen otros. En el ejemplo de la figura 2.1 se representa un par de revolución (R) que permite una rotación relativa entre los dos elementos del sistema y restringe los demás.

Puede ocurrir que dos sólidos rígidos de un sistema multicuerpo no estén unidos entre sí directamente sino mediante otros elementos que transmiten fuerzas: como muelles, amortiguadores, etc.

La mecánica clásica considera que un sólido rígido puede desplazarse libremente en un movimiento que se puede descomponer, por ejemplo, en tres rotaciones y tres traslaciones geométricas independientes (respecto a una base en el espacio). En un sistema multicuerpo con elementos unidos mecánicamente entre sí (mediante pares cinemáticos), tal como se ha explicado, algunos de estos movimientos elementales quedan restringidos: los *grados de libertad* de un sistema son los movimientos independientes que permanecen; es el número mínimo de parámetros que necesitamos especificar para

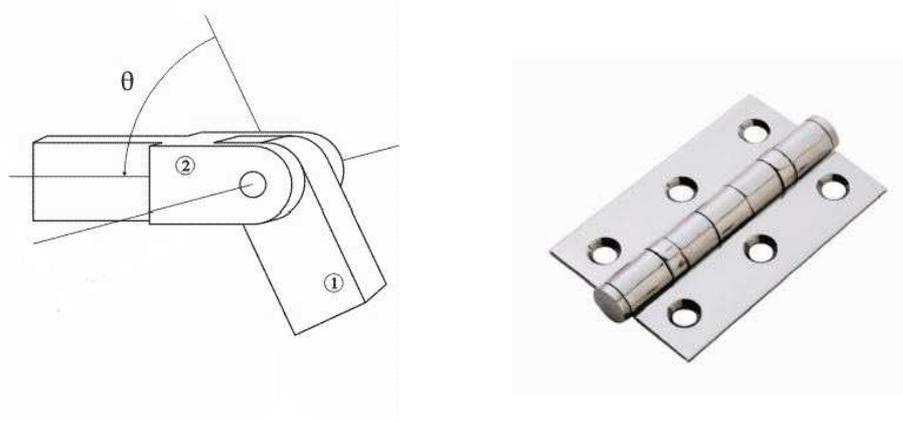


Figura 2.1: Ejemplo de sistema multicuerpo

determinar completamente la velocidad del sistema multicuerpo. Se suele considerar al primer elemento como un fijo, esto es, sin capacidad de movimiento.

En el ejemplo anterior (una bisagra) se representan dos sólidos rígidos y un par de revolución que permite una rotación relativa entre ambos y restringe por tanto los otros cinco movimientos del segundo elemento. El número de grados de libertad es uno (considerando fijo el primer cuerpo).

El *criterio de Grübler-Kutzbach* permite determinar los grados de libertad de un sistema multicuerpo. Para sistemas en tres dimensiones establece que:

$$GDL = 6n - \sum_{i=1}^j (6 - f_i) = 6(n - 1 - j) + \sum_{i=1}^j f_i$$

donde GDL son los grados de libertad; n es el número de elementos móviles del sistema; j es el número de pares cinemáticos y f_i es el número de movimientos relativos que permite cada uno de los pares. Podría ocurrir, como en uno de los ejemplos que se presentarán en el capítulo de resultados, que en un sistema multicuerpo, por una particularidad (típicamente geométrica) no se cumpliese el criterio de Grübler-Kutzbach.

Los sistemas multicuerpo se pueden clasificar en *sistemas de cadena abierta* o *de cadena cerrada*. Una única cadena abierta consiste

en n elementos móviles unidos de extremo a extremo por n pares cinemáticos con un único elemento fijo. Si un sistema está formado únicamente por cadenas abiertas se llama de cadena abierta. Por el contrario si en una cadena abierta fijamos el extremo que quedaba libre se forma un cadena cerrada (un lazo). Un sistema con lazos se llama de cadena cerrada.

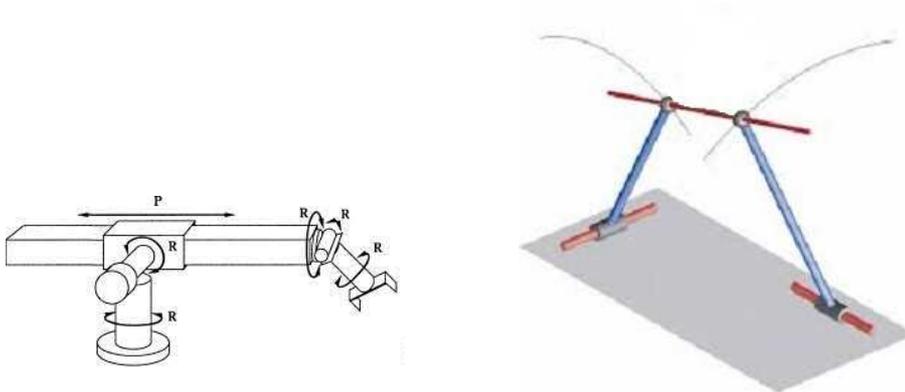


Figura 2.2: Ejemplos de cadena abierta y cerrada

Un brazo robótico es un ejemplo típico de cadena abierta: una serie de elementos conectados por 6 pares de revolución (R) o prismáticos (P) que permiten un movimiento cada uno. En el ejemplo de la figura 2.2 hay seis elementos móviles unidos por cinco pares R y un par P: seis grados de libertad. En la misma figura se puede ver un ejemplo típico de mecanismo de cadena cerrada: un cuadrilátero articulado con dos pares R en los extremos y otros dos pares esféricos (S): en este caso existen dos elementos móviles y cinco restricciones en cada par de revolución: dos grados de libertad.

El concepto de sistema multicuerpo que se ha expuesto es muy similar a una cierta clase de mecanismo. El concepto de *mecanismo* es más general y se utiliza para designar cualquier tipo de sistema mecánico con elementos móviles, como un engranaje o una leva. Los sistemas multicuerpo, tal como se han definido más arriba, son algo más restringido: se podrían llamar *mecanismos de barras articuladas* y su inclusión como una parte específica de la Mecánica en el plano teórico se podría establecer en 1977, durante el congreso sobre *Dynamics of Multibody Systems* organizado por K. Magnus en Berlín [52] bajo el patrocinio de la IUTAM (*International Union of Theoretical*

and *Applied Mechanics*).

A lo largo de esta tesis, se utilizará en ocasiones la palabra mecanismo como sinónimo de sistema multicuerpo, es decir, aceptando un sentido más restringido para la primera.

2.2. Interés de la simulación de mecanismos

Antes de tratar el interés de la simulación del comportamiento de los sistemas multicuerpo desde un punto de vista histórico, se van a exponer someramente los diferentes aspectos que comprende esta simulación. Estos tipos de estudios se conocen como *problemas*; en este sentido se habla de problemas cinemáticos, dinámicos, etc. Son siempre posteriores a la definición del mecanismo. La descripción siguiente de los *problemas* que se afrontan en la simulación no es exhaustiva:

- **Problemas cinemáticos:** se estudia la posición o el movimiento del mecanismo independientemente de las fuerzas que actúan sobre él, o las reacciones que provocan. Es un problema geométrico en el que tampoco interviene la masa ni los momentos de inercia. Ejemplos de problemas cinemáticos son: el cálculo de la *posición inicial*, cuando se conoce la posición de algunos elementos del mecanismo y se necesita calcular la de los demás; el problema de *desplazamientos finitos*, que parte de una posición válida y variando la posición de algunos elementos se calcula la posición de los demás; el *análisis de velocidades y aceleraciones* de determinados elementos de interés dadas las condiciones iniciales de posición y velocidad de algunos elementos del mecanismo; y por último, *la simulación cinemática*, que simula todo el rango de posibles posiciones de los elementos del mecanismo.
- **Problemas dinámicos:** La característica principal es que aparecen en el análisis las fuerzas que actúan sobre el mecanismo y las características inerciales de este. Los estudios más típicos en este ámbito de la solución de la dinámica son:

- **Posición de equilibrio estático:** consiste en determinar la posición en la que todas las fuerzas que actúan sobre el sistema están en equilibrio.
 - **Modos naturales de vibración:** muy relacionado con el anterior. Es necesario calcular las frecuencias y modos de vibración de pequeñas oscilaciones en torno a la posición de equilibrio.
 - **Problema dinámico inverso:** consiste en calcular los esfuerzos motores que producen un determinado movimiento así como las reacciones que aparecen en cada uno de los pares cinemáticos.
 - **Simulación dinámica:** estudia el movimiento del sistema multicuerpo en un intervalo determinado de tiempo, dadas unas fuerzas y unas condiciones iniciales. Permite calcular el comportamiento del sistema.
 - **Deformaciones elásticas:** en sistemas en los que existen *deformaciones elásticas* es necesario afrontar un problema diferente que complica sobremanera tanto los problemas de dinámica inversa como la simulación.
 - **Estudios de impactos y percusiones:** existen situaciones en las que se producen *impactos* sobre el mecanismo, con un cambio repentino en las velocidades y posibles deformaciones. En estos casos es necesario plantear el estudio de la simulación dinámica de una forma diferente; son similares los casos en los que se *percute* el mecanismo con una fuerza conocida en un tiempo muy corto donde también se producen cambios bruscos en las velocidades.
- A todos estos estudios hay que añadir los que afectan al *diseño* de los propios mecanismos que requiere un análisis de las dimensiones adecuadas y la optimización del diseño orientado al objetivo que se pretende.

Todos estos problemas aparecen históricamente en la medida que evolucionan las aplicaciones industriales de sistemas multicuerpo. Hasta la década de 1950 solo existían algunos *métodos gráficos*, como los publicados por Wittenbauer [83] en 1923 para la resolución de

algunos mecanismos especialmente sencillos. En lo demás se recurría a la experimentación y a determinadas fórmulas experimentales en cada mecanismo concreto.

El interés por la cinemática y la dinámica de los sistemas multicuerpo adquiere particular importancia a partir de la década de 1960, en la que se dan dos factores: por una parte en esa década irrumpe con enorme fuerza *la carrera espacial*. Empiezan a requerirse cálculos muy precisos y fiables de la simulación cinemática y dinámica de sistemas multicuerpo imposibles de experimentar en condiciones de ingravidez. Por otra parte comienzan a extenderse los ordenadores digitales, que aunque muy limitados en potencia de cálculo, marcaron un punto de inflexión en la resolución de estos sistemas. Lo que hasta ese momento había sido un reto inasequible, empieza a ser una necesidad alcanzable y perentoria. Además, *los vehículos espaciales* ofrecían con frecuencia una estructura topológica de cadena abierta, más sencilla de tratar, y muy similar a la de los robots que empiezan a introducirse en la industria.

En las décadas de 1980 y 1990 las aplicaciones se extienden a otras áreas de la ingeniería, principalmente al campo de la *automoción*, donde la presencia de cadenas cerradas es la tónica habitual en los sistemas de suspensión y dirección. Son particularmente importantes problemas de equilibrio estático y vibraciones.

En esos años, otro campo de aplicación y donde surgieron importantes aportaciones a los métodos de análisis de sistemas multicuerpo fue *la robótica*. Los problemas que plantea la robótica se parecen a los que atañen al estudio de los vehículos espaciales en que fundamentalmente considera sistemas de cadena abierta, pero se diferencia de ellos en que, por ser el control y no la simulación el principal objetivo de los análisis, aparecen requerimientos especiales en lo que a eficiencia se refiere. Para el *control de los actuadores* es fundamental resolver de un modo muy eficiente el problema dinámico inverso.

En las últimas décadas además de desarrollarse los campos anteriores también han aparecido nuevas aplicaciones donde la simulación de sistemas multicuerpo juega un papel importante.

En las *aplicaciones de realidad virtual* los métodos eficientes de dinámica de sistemas multicuerpo, al proporcionar el comportamiento físico real de las entidades virtuales, se han usado para dotarlas de

unas prestaciones que hacen posible su utilización eficiente en las distintas etapas del ciclo de vida de cualquier producto industrial: diseño, análisis, experimentación, fabricación, montaje, mantenimiento, y fin de la vida del producto (reutilización, reciclaje, etc.). Una revisión reciente de estas aplicaciones se puede encontrar en Cuadrado y Schiehlen [14].

En *biomecánica* las técnicas de dinámica multicuerpo se han aplicado al estudio del movimiento humano, ya que el cuerpo humano puede verse como un sistema multicuerpo compuesto por elementos rígidos (los huesos) conectados por articulaciones y actuados por músculos. Mediante un análisis dinámico inverso se pueden proporcionar los esfuerzos articulares que se producen en el movimiento, tal como han mostrado recientemente Cuadrado y Masarati [13].

Capítulo 3

Simulación computacional de mecanismos

Hoy en día la simulación cinemática y dinámica de sistemas multicuerpo es una parte de lo que se ha llamado *mecánica computacional*. Aunque la aplicación del método de los elementos finitos a problemas de medios continuos en Ingeniería Mecánica es una parte mucho más amplia de esta, la simulación de sistemas multicuerpo también tiene sus propios problemas teóricos, numéricos e informáticos: los sistemas multicuerpo suelen implicar grandes desplazamientos, con importantes cambios de geometría; suelen ser altamente no lineales; y la interactividad e incluso el tiempo real es un aspecto más importante que en los cálculos de elementos finitos.

En la simulación computacional de sistemas multicuerpo hay que tener presente las siguientes etapas del problema que han de contemplarse de una manera integral, ya que cada una de ellas influye en las demás, como ha sido expuesto por Cuadrado et al. [11].

- Modelización matemática del problema
- Formulación de las ecuaciones dinámicas
- Integración numérica de las ecuaciones

En los apartados siguientes se tratará cada uno de esos aspectos por separado justificando en cada caso las opciones elegidas en esta

tesis. Previamente y en el siguiente apartado se va a hacer una revisión bibliográfica de las principales contribuciones a la simulación computacional de los sistemas multicuerpo.

3.1. Revisión bibliográfica

El la década de los años 60 se pueden destacar las contribuciones de Uicker [79], quien estudió la dinámica de sistemas tridimensionales usando las transformaciones de Denavit y Hartenberg [18], publicadas en 1955; Hooker y Margulies [38], y Roberson y Wittenburg [62], resolvieron el problema del control de satélites artificiales formados por varios sólidos unidos en cadena abierta.

En los años 70 aparecen las primeras formulaciones que van a dar lugar a los primeros programas comerciales de propósito general. Así, Sheth y Uicker [74] presentaron en 1972 el programa IMP, basado en coordenadas relativas y en las transformaciones de Uicker [79]. Estas mismas coordenadas habían sido utilizadas por Paul y Krajcinovic [60] en 1970 y volvieron a serlo poco después por Smith et al. [76]. En 1975, Paul [59] publicó el primer artículo de revisión del estado del arte y poco después, en 1977 apareció el primer libro sobre el tema, debido a Wittenburg [84].

En 1977, Orlandea et al. [58] realizaron una gran contribución a los métodos globales, al aplicar técnicas de matrices huecas a las ecuaciones diferenciales y restricciones, con coordenadas de puntos de referencia y ángulos de Euler para determinar la orientación. De ese trabajo surgió poco tiempo después el programa MSC.Adams, que mantiene desde entonces el liderazgo del mercado.

Al comienzo de la década de 1980, Wehage y Haug [82] y Nikravesh et al. [57] pusieron los fundamentos teóricos de lo que sería el programa DADS. En la actualidad el programa DADS es comercializado por la empresa LMS con el nombre de LMS Virtual.Lab Motion.

En los últimos años de dicha década y en los primeros años de la década de los 90, aparecieron varios textos sobre el tema, entre los que se puede citar a Nikravesh [56], Roberson y Schwertassek

[63], Friberg [29], Haug [35], Huston [39] y Amirouche [2]. En 1990, Schiehlen [66] presentó un catálogo con las principales características de todos los programas existentes.

En la última parte de la década de los 90 y durante este siglo se ha dedicado más atención a las extensiones de formulaciones básicas y a la resolución de problemas interdisciplinarios. En estos años han aparecido algunos nuevos libros como los de Angeles et al. [3], Pfeiffer y Glocker [61], Eich-Soellner y Fährer [24], Schwerin [68], y Samin y Fisetite [65]. El centro de los desarrollos se ha desplazado de EEUU a Europa. En algunos congresos monográficos (como Seabra Pereira y Ambrósio [69] o Ambrósio [1]) se han presentado buena parte de las nuevas aportaciones, así como la nueva revista *Multibody System Dynamics*, publicada a partir de 1997 por Kluwer. Un estudio reciente, de 2014, en el campo teórico de la dinámica de mecanismos se puede encontrar en Cuadrado [10].

3.2. Modelo del sistema multicuerpo

Para estudiar un sistema multicuerpo y simular su comportamiento se necesita un modelo matemático que describirán sus propiedades físicas exactas y permitirá describir la posición y movimiento del sistema. El modelo matemático consiste en establecer relaciones entre los parámetros que se emplean para definir la posición de cada elemento. Así, por ejemplo, un elemento rectilíneo indeformable podría estar caracterizado por las coordenadas cartesianas de sus extremos. Si esto se hace también para los elementos adyacentes y se consideran, por un lado, la indeformabilidad de los elementos y por otro la naturaleza de los enlaces entre ellos, se podrán escribir una serie de relaciones entre los parámetros que describen los elementos (distancias constantes, puntos comunes,...).

En definitiva, el modelo matemático que describe el mecanismo consiste en establecer una serie de relaciones entre parámetros variables elegidos de manera que describan la geometría de los elementos y la naturaleza de los enlaces. A estos parámetros se les va a llamar *coordenadas* y a las relaciones entre ellos *ecuaciones de restricción*.

Lógicamente hay varias alternativas en la forma de elegir los pa-

rámetros que, a su vez, condicionarán la forma, número, complejidad, etc., de las ecuaciones de restricción. Sin ánimo de ser exhaustivos se hace a continuación una breve descripción de las alternativas más típicas en cuanto a la elección del sistema de coordenadas que permitirán describir el mecanismo y su movimiento (posición, velocidad y aceleración) de forma unívoca: en primer lugar se distingue entre un sistema de coordenadas dependientes e independientes:

- *Sistema de coordenadas independientes*: en este caso el número de coordenadas coincide con el de grados de libertad del mecanismo. En general los estudios en este campo apuntan a la conclusión de que esta elección no es óptima para un análisis de carácter general ya que un sistema de este tipo no define de forma unívoca la posición de todos los elementos del mecanismo y es necesario un análisis adicional para este propósito. Además, existen casos en los que la resolución de la posición inicial da como resultado varias soluciones posibles. Dicho esto, se puede añadir que solo en algunos casos, con variaciones muy pequeñas en la posición, un conjunto de coordenadas de este tipo puede ser útil y llevar a una alta eficiencia computacional.
- Si se usa un número mayor de coordenadas que los grados de libertad del sistema el conjunto de coordenadas forma un *sistema de coordenadas dependientes*, cuya característica principal es precisamente esa *dependencia*, es decir, que existe un conjunto de ecuaciones algebraicas que las relacionan que se llaman *ecuaciones de restricción del sistema*. El número de ecuaciones de restricción independientes tiene que ser igual a la diferencia entre el número de coordenadas dependientes y los grados de libertad. Estas restricciones juegan un papel determinante en el planteamiento de las ecuaciones de la dinámica.

A su vez hay varias alternativas en la elección de un sistema de coordenadas dependiente. Típicamente se consideran:

- *Coordenadas relativas*: han sido las primeras coordenadas usadas en programas de carácter general, en trabajos clásicos como los de Smith et al. [76]. Las coordenadas relativas definen la

posición de cada elemento, en una determinada cadena cinemática, usando las coordenadas correspondientes a los grados de libertad relativos permitidos por el par cinemático que une ese elemento con el elemento anterior en esa cadena. En un problema de cadena abierta ese conjunto de coordenadas dependientes coincide con los grados de libertad del mecanismo, por lo que no habrá ecuaciones de restricción y serán coordenadas independientes. Se mencionan a continuación algunas ventajas e inconvenientes de la utilización de este tipo de coordenadas:

- Es ventajoso el que se reduzca el número de coordenadas a un mínimo; una ventaja más importante es que en los mecanismos con actuadores, al coincidir las coordenadas usadas con los grados de libertad permitidos por cada par cinemático, se facilita el control del movimiento en esos pares.
 - Es un inconveniente que la formulación matemática debe tener en cuenta que la posición y velocidad de cada elemento solo se puede resolver conociendo las de los elementos anteriores. En los mecanismos de cadena cerrada las ecuaciones de restricción se generan al cerrar vectorialmente cada uno de los lazos del mecanismo: se requiere por tanto un preproceso adicional para determinar correctamente qué ecuaciones de restricción son independientes, eligiendo siempre lazos que no se crucen entre sí (muchas veces recurriendo a teoría de grafos).
- *Coordenadas de puntos de referencia:* en este caso se define la posición absoluta de cada uno de los elementos del sistema. Para cada elemento se determina la posición de un punto del mismo usando coordenadas cartesianas (un punto de referencia). Ese punto suele estar en el centro de masas del elemento. A continuación se añaden los ángulos necesarios para fijar la orientación del elemento en el espacio en relación a una base inercial.
- La ventaja fundamental de este sistema de coordenadas frente a las relativas es que se conoce la posición de cada uno de los elementos del sistema y no es necesario

un proceso posterior en la simulación para determinarla; esto hace que sea muy sencilla la representación de los mecanismos planos. En el caso de mecanismos en tres dimensiones esta sencillez no se puede extrapolar, por la existencia de más tipos de pares cinemáticos y el hecho de que la orientación del sólido rígido en el espacio también es más compleja.

- Otra ventaja es que las matrices que aparecen al formular las ecuaciones de la dinámica son altamente huecas, circunstancia que se aprovecha en la simulación del movimiento.
 - Otra característica importante es que en las ecuaciones de restricción asociadas a un par cinemático solo intervienen las coordenadas de los elementos relacionados mediante ese par. Esto significa que son establecidas a un *nivel local*, y no es necesario tener en cuenta todo el mecanismo para determinar el conjunto de restricciones, como en las coordenadas relativas.
- **Coordenadas naturales:** Se puede considerar como una evolución de las coordenadas de punto de referencia. En los mecanismos planos cada elemento del sistema multicuerpo está determinado por las coordenadas cartesianas de al menos dos puntos, llamados *puntos básicos* y los pares cinemáticos se definen mediante un subconjunto de los mismos puntos básicos. En mecanismos de tres dimensiones se define el sistema mediante las coordenadas cartesianas de un conjunto de puntos básicos a lo largo del mecanismo y las componentes cartesianas de un conjunto de vectores unitarios. Cada elemento debe tener el suficiente número de puntos básicos y vectores unitarios para definir su posición y orientación. Los pares cinemáticos comparten algunos de esos puntos o vectores dependiendo de sus características. Se mencionan a continuación algunas ventajas:
- El número de coordenadas naturales necesarias tiende a estar entre las utilizadas en un sistema de coordenadas relativas y un sistema de puntos de referencia: por una parte se eliminan los ángulos con los que se orientaban los elementos y por otra los puntos básicos o vectores

unitarios se pueden compartir entre elementos diferentes.

- Al eliminar los ángulos se eliminan también las funciones trigonométricas de las ecuaciones de restricción. Generalmente las restricciones que aparecen en coordenadas naturales son lineales o cuadráticas, lo que simplifica el cálculo.
 - Una vez que se conocen la posición, velocidad o aceleración de los puntos básicos la representación del mecanismo es prácticamente inmediata.
 - Una de las ventajas más importantes es la facilidad con la que se puede modelar e implementar cualquier mecanismo. Esto es particularmente importante en programas de propósito general.
- **Coordenadas mixtas:** en este conjunto de coordenadas se parte de un sistema de coordenadas naturales o de punto de referencia y se añaden ángulos, distancias o otro tipo de variables relacionadas con los grados de libertad que describen el movimiento relativo que permiten los pares cinemáticos. Estas coordenadas añadidas al sistema permiten la introducción directa en la simulación de los efectos de motores y actuadores en esos pares cinemáticos tal como ocurría en las coordenadas relativas.

Las coordenadas naturales nacieron a comienzos de los años 1980 en la Escuela Superior de Ingenieros Industriales de Bilbao (García de Jalón y Avilés [41], Serna et al. [70]), pero alcanzaron su madurez a partir de 1983 en la Escuela Superior de Ingenieros Industriales de San Sebastián y en el CEIT. Las referencias más características se encuentran en García de Jalón [43, 44], y sobre todo en el libro de García de Jalón y Bayo [42], que las explica con gran detalle. Las coordenadas mixtas, cuya base es en muchos casos un sistema de coordenadas naturales, permiten un gran avance en la modelización y la resolución de mecanismos y desde entonces han sido utilizadas por numerosos grupos de investigación.

En esta tesis se descartan las coordenadas independientes como un modelo válido: se estudian diferentes mecanismos con pocos grados de libertad y generalmente muchos elementos. Por otra parte

entre las llamadas coordenadas dependientes, tampoco se consideran las de puntos de referencia: la modelización y representación es muy compleja en los mecanismos que se estudian y se introducen numerosos ángulos de orientación en los sistemas que no se corresponden a ninguna característica de los pares cinemáticos. Se ha optado consecuentemente por un sistema de coordenadas mixtas basadas en un sistema de coordenadas naturales a las que se añaden coordenadas relativas.

En la notación que se utilizará en esta tesis el vector de coordenadas \mathbf{q} representa al conjunto de todas las coordenadas que se utilizan en la modelización del mecanismo (puntos básicos, vectores unitarios y coordenadas relativas). Las ecuaciones de restricción que relacionan las coordenadas dependientes se agruparán en el vector Φ donde cada fila corresponde a una restricción. En un sistema holónomo, en el que las restricciones no dependen de las velocidades, se establece de un modo general como modelo matemático el siguiente sistema de ecuaciones algebraicas que representan las ecuaciones de restricción:

$$\Phi(\mathbf{q}, t) = 0 \quad (3.1)$$

Si el mecanismo tiene f grados de libertad y hay n coordenadas dependientes agrupadas en el vector \mathbf{q} , el sistema anterior tendrá $m = n - f$ ecuaciones de restricción independientes. Como se ve en la notación las restricciones pueden depender explícitamente del tiempo.

Derivando con respecto al tiempo la ecuación 3.1 se obtienen ecuaciones de restricción para las velocidades $\dot{\mathbf{q}}$ y aceleraciones $\ddot{\mathbf{q}}$ dependientes:

$$\Phi_{\mathbf{q}}(\mathbf{q}, t) \dot{\mathbf{q}} = -\Phi_t \equiv \mathbf{b} \quad (3.2a)$$

$$\Phi_{\mathbf{q}}(\mathbf{q}, t) \ddot{\mathbf{q}} = -\ddot{\Phi}_t - \dot{\Phi}_{\mathbf{q}} \dot{\mathbf{q}} \equiv \mathbf{c} \quad (3.2b)$$

Se ha llamado $\Phi_{\mathbf{q}}$ a la matriz jacobiana (o jacobiano), formada por las derivadas parciales con respecto a cada una de las variables

dependientes del vector de restricciones; \mathbf{b} es la derivada parcial respecto al tiempo (cambiada de signo) de las ecuaciones de restricción; en \mathbf{c} aparecen las derivadas totales con respecto al tiempo de la matriz jacobiana y el término \mathbf{b} . A estos sistemas multicuerpo, donde las restricciones dependen del tiempo, se les llama reónomos. Si no existe esa dependencia son sistemas esclerónomos. En este último caso se simplifican las ecuaciones.

3.3. Las ecuaciones del movimiento

A continuación se van a revisar las formulaciones más extendidas para el estudio de la dinámica de sistemas multicuerpo formados por elementos rígidos. No se pretende hacer una clasificación exhaustiva sino crear un marco adecuado para la discusión en los capítulos posteriores. Para una revisión más general y profunda se pueden consultar los artículos de Schiehlen [67, 71].

Antes de abordar el estudio de las ecuaciones de la dinámica, se considera conveniente hacer alguna precisión e introducir ciertos conceptos que, aunque se pueden considerar dentro del marco del *análisis cinemático*, son imprescindibles para el desarrollo de la formulación de la dinámica. En concreto se van a describir los espacios de posiciones, velocidades y aceleraciones posibles.

Tal como se ha detallado al hablar del modelo matemático, para definir el mecanismo es preciso escoger una serie de coordenadas agrupadas en un vector \mathbf{q} . Estas coordenadas deben cumplir un conjunto de restricciones formuladas como las ecuaciones algebraicas escritas en la expresión 3.1. El conjunto de vectores \mathbf{q} que cumplen todas las restricciones $\Phi(\mathbf{q}) = 0$ constituye un subespacio del espacio vectorial de dimensión n , número de coordenadas en \mathbf{q} . En algunos ámbitos (robótica), el conjunto de posiciones posibles es el *espacio de trabajo* del mecanismo.

3.4. Espacio de velocidades posibles

El vector $\dot{\mathbf{q}}$ de las velocidades de un sistema sujeto a restricciones se puede considerar como perteneciente a un espacio vectorial llamado *espacio de velocidades*. El estudio de este espacio vectorial y la habilidad de encontrar una base para él, permitirá mejorar la eficiencia de la solución de las ecuaciones de la dinámica. Por claridad en la introducción de los conceptos más importantes se van a tratar primero los sistemas esclerómicos y después se generalizan los resultados a los sistemas reónomos.

Consideramos un sistema con m ecuaciones de restricción que no dependen explícitamente del tiempo, n coordenadas y $f = n - m$ grados de libertad. La ecuación 3.1 y las ecuaciones 3.2a y 3.2b se transforman en:

$$\Phi(\mathbf{q}) = 0 \quad (3.3a)$$

$$\Phi_{\mathbf{q}}(\mathbf{q}) \dot{\mathbf{q}} = 0 \quad (3.3b)$$

$$\Phi_{\mathbf{q}}(\mathbf{q}) \ddot{\mathbf{q}} = -\dot{\Phi}_{\mathbf{q}} \dot{\mathbf{q}} \quad (3.3c)$$

La ecuación 3.3b indica que el vector de velocidades pertenece al núcleo de la matriz jacobiana de las ecuaciones de restricción. Teniendo en cuenta que las restricciones son independientes, este núcleo es el subespacio vectorial de dimensión $f = n - m$ de las *velocidades permitidas* o *velocidades posibles* del mecanismo. Cualquier velocidad compatible con las ecuaciones de restricción pertenece a este espacio.

Esta consideración de que cualquier vector de velocidades compatible con la naturaleza del mecanismo pertenece a un espacio o subespacio vectorial de dimensión $f = n - m$, supuesto que se pudiese encontrar una base de vectores generadores, permitiría expresar cualquier velocidad posible con un número de parámetros f , netamente inferior a n , dimensión de \mathbf{q} . Bastaría con elegir la base para el espacio de velocidades y expresar cualquier velocidad $\dot{\mathbf{q}}$ como una combinación lineal de los vectores de la base.

Si se considera que los vectores $\{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots, \mathbf{r}_f\}$ de dimensión n

forman una base del núcleo de la matriz jacobiana, cualquier vector $\dot{\mathbf{q}}$ se podrá expresar como:

$$\dot{\mathbf{q}} = \sum_{i=1}^f \mathbf{r}_i \dot{z}_i \quad (3.4)$$

Los coeficientes \dot{z}_i son las componentes de un nuevo vector $\dot{\mathbf{z}}$ de f componentes que se llama *vector de velocidades independientes*. Si se llama \mathbf{R} a la matriz de n filas y m columnas cuyas columnas son los elementos \mathbf{r}_i de la expresión anterior se puede escribir como:

$$\dot{\mathbf{q}} = \mathbf{R} \dot{\mathbf{z}} \quad (3.5)$$

donde la matriz \mathbf{R} verifica la ecuación:

$$\Phi_{\mathbf{q}} \mathbf{R} = \mathbf{0} \quad (3.6)$$

En esta ecuación se pone de manifiesto que la matriz \mathbf{R} depende del vector de posiciones y por lo tanto existe una matriz \mathbf{R} diferente para cada posición del mecanismo. El resultado obtenido se puede extender a las aceleraciones derivando con respecto al tiempo la ec. 3.5. En este caso intervienen tanto la matriz \mathbf{R} como su derivada:

$$\dot{\mathbf{q}} = \mathbf{R} \ddot{\mathbf{z}} + \dot{\mathbf{R}} \dot{\mathbf{z}} \quad (3.7)$$

Para los sistemas reónomos se pueden extender los conceptos anteriores definiendo las velocidades independientes como la proyección del conjunto de velocidades (dependientes) sobre las filas de una matriz \mathbf{B} . La matriz \mathbf{B} es una matriz constante donde el número de filas coincide con los grados de libertad del sistema y tiene tantas columnas como el número de coordenadas. Además se asume que sus filas son linealmente independientes entre sí e independientes de las filas de la matriz jacobiana.

$$\dot{\mathbf{z}} = \mathbf{B} \dot{\mathbf{q}} \quad (3.8)$$

Esta última expresión, junto con la ecuación de restricción de velocidades 3.2b conduce al sistema matricial de n ecuaciones y n incógnitas siguiente:

$$\begin{pmatrix} \Phi_q \\ \mathbf{B} \end{pmatrix} \dot{\mathbf{q}} = \begin{pmatrix} \mathbf{b} \\ \dot{\mathbf{z}} \end{pmatrix} \quad (3.9)$$

La matriz de este sistema es invertible gracias a las condiciones impuestas a la matriz \mathbf{B} , y se puede despejar el vector de velocidades dependientes:

$$\dot{\mathbf{q}} = \begin{pmatrix} \Phi_q \\ \mathbf{B} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{b} \\ \dot{\mathbf{z}} \end{pmatrix} = (\mathbf{S} \quad \mathbf{R}) \begin{pmatrix} \mathbf{b} \\ \dot{\mathbf{z}} \end{pmatrix} = \mathbf{S} \mathbf{b} + \mathbf{R} \dot{\mathbf{z}} \quad (3.10)$$

donde \mathbf{S} es la matriz constituida por las primeras m filas de la matriz inversa (m es el numero de restricciones) y \mathbf{R} se define con las últimas f columnas de esa matriz (f son los grados de restricción). Se verifica también la siguiente ecuación matricial:

$$\begin{aligned} \begin{pmatrix} \Phi_q \\ \mathbf{B} \end{pmatrix} \begin{pmatrix} \Phi_q \\ \mathbf{B} \end{pmatrix}^{-1} &= \begin{pmatrix} \Phi_q \\ \mathbf{B} \end{pmatrix} (\mathbf{S} \quad \mathbf{R}) \\ &= \begin{pmatrix} \Phi_q \mathbf{S} & \Phi_q \mathbf{R} \\ \mathbf{B} \mathbf{S} & \mathbf{B} \mathbf{R} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \end{aligned} \quad (3.11)$$

Se puede observar que al igual que en el caso de los sistemas esclerónomos las columnas de la matriz \mathbf{R} son generadores del núcleo de la matriz jacobiana y se puede concluir que el sistema lineal de ecuaciones 3.2a tiene una solución homogénea ($\mathbf{R} \dot{\mathbf{z}}$) y una solución particular ($\mathbf{S} \mathbf{b}$) que constituyen la solución general del sistema.

Al igual que en el caso esclerónimo, el resultado obtenido se puede extender al estudio de las aceleraciones con solo añadir a la ecuación 3.2b la derivada con respecto al tiempo de la expresión 3.8.

$$\begin{pmatrix} \Phi_q \\ \mathbf{B} \end{pmatrix} \ddot{\mathbf{q}} = \begin{pmatrix} \mathbf{c} \\ \ddot{\mathbf{z}} \end{pmatrix} \quad (3.12)$$

De donde se puede obtener explícitamente el vector de aceleraciones como:

$$\ddot{\mathbf{q}} = \begin{pmatrix} \Phi_{\mathbf{q}} \\ \mathbf{B} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{c} \\ \ddot{\mathbf{z}} \end{pmatrix} = (\mathbf{S} \ \mathbf{R}) \begin{pmatrix} \mathbf{b} \\ \ddot{\mathbf{z}} \end{pmatrix} = \mathbf{S} \mathbf{c} + \mathbf{R} \ddot{\mathbf{z}} \quad (3.13)$$

3.5. Ecuaciones de la dinámica

En este apartado se van a describir las ecuaciones diferenciales del movimiento de un sistema constituido por varios sólidos rígidos unidos mediante pares cinemáticos (expresados matemáticamente en forma de ecuaciones de restricción), tal como se plantean en la mecánica clásica. Ese conjunto de ecuaciones diferenciales son la base para los distintos métodos de resolución que se pueden formular en función del modelo elegido.

Históricamente, las ecuaciones de Newton [55] establecen el modelo del problema de traslación del sólido rígido. Fue Euler [26], sin embargo, el que desarrolló propiamente el concepto de sólido rígido y abordó el problema de definir el modelo matemático de la rotación de este. De esa forma quedaron establecidas las ecuaciones necesarias para estudiar la dinámica de un cuerpo rígido con una distribución de masas arbitraria y libre de moverse en el espacio sin restricción alguna.

D'Alembert estudió por primera vez la dinámica de varios sólidos rígidos con restricciones [16] distinguiendo entre fuerzas aplicadas y fuerzas de reacción a las que llamó fuerzas perdidas (intuyendo el principio de los trabajos virtuales). Unos años más tarde Lagrange [50] aportó un tratamiento matemático riguroso a todas estas ideas, aplicando un principio variacional a las energías cinéticas y potenciales: las ecuaciones algebraico-diferenciales (DAE, por sus siglas en inglés) para coordenadas dependientes, y ecuaciones diferenciales ordinarias (ODE) para coordenadas independientes (sin restricciones).

En el siglo XX se extendieron estas ecuaciones a los sistemas no holónomos, aportación debida a Jordain [45] y sobre todo a Kane y Levinson [47]. El que hayan pasado más de dos siglos desde

que se formularon estas ecuaciones ha puesto de manifiesto la enorme dificultad que entraña encontrar una solución analítica para el problema —se puede consultar, por ejemplo, el tratado clásico de Goldstein [33]—.

Como se ha visto al estudiar la modelización del mecanismo, en un sistema modelado con coordenadas dependientes las ecuaciones de restricción adoptan la forma:

$$\Phi(\mathbf{q}, t) = 0 \quad (3.14)$$

Para formular las ecuaciones diferenciales del movimiento hay que tener en cuenta, por una parte, las *fuerzas de inercia* debidas a las aceleraciones; por otra, las *fuerzas debidas a las restricciones* o *fuerzas de los enlaces* y, por último, las *fuerzas exteriores* y las *dependientes de las velocidades*. Esto se expresa en la siguiente ecuación:

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \lambda = \mathbf{Q} \quad (3.15)$$

En esta ecuación las fuerzas debidas a las restricciones son proporcionales a las filas de la matriz Jacobiana mediante el correspondiente multiplicador de Lagrange, λ . Hay que tener en cuenta que los multiplicadores de Lagrange introducen más incógnitas en el sistema. Por eso, para resolver la dinámica se tienen que considerar conjuntamente las ecuaciones 3.14 y 3.15 que constituyen un sistema de ecuaciones diferenciales y algebraicas (DAE) cuya integración numérica supone la solución del problema.

La solución del sistema de DAE's no es trivial y depende en gran medida de las coordenadas dependientes utilizadas y la forma de tratar tanto las ecuaciones dadas en 3.14 como las de 3.15. De ahí que haya diferentes formulaciones en el estudio de la dinámica de sistemas multicuerpo.

En los siguientes apartados se van a presentar algunas de esas formulaciones (aquellas que más importancia tienen para el desarrollo de esta tesis) y se van a abordar distintas cuestiones que pueden ser relevantes:

- En el primer apartado se mencionará la diferencia que se suele establecer entre métodos globales y topológicos, ya que todos los métodos que se van a abordar son globales.
- En el siguiente apartado se van a derivar las ecuaciones de la dinámica que se han mostrado más arriba. Se utilizarán dos métodos diferentes que dan lugar a dos formulaciones distintas en coordenadas dependientes.
- En el apartado siguiente se presentarán los problemas de estabilidad que presentan algunas formulaciones y las distintas propuestas que intentan resolver ese problema.
- En el cuarto apartado se presentarán dos formulaciones especialmente eficientes en las que se evita la inestabilidad de las anteriores.
- En el último apartado se presentarán las formulaciones basadas en la integración de un conjunto de coordenadas independientes.

3.5.1. Métodos globales y topológicos

Los métodos para formular las ecuaciones de la dinámica se suelen clasificar en *métodos globales* y *métodos topológicos*.

Los *métodos globales* plantean las ecuaciones siempre de la misma forma tanto si el mecanismo es de cadena abierta como si es de cadena cerrada. En general, estos métodos utilizan coordenadas cartesianas (de puntos de referencia o naturales) para poder expresar la posición de cada sólido de forma absoluta e independiente.

Los métodos globales son fáciles de implementar en un ordenador y conducen a formulaciones generales. Esta característica ha provocado que históricamente hayan sido los preferidos para los programas de propósito general. El origen de las formulaciones matemáticas de estos métodos se remonta al final de la década de los 70 del siglo pasado, para consolidar su desarrollo a lo largo de los 80. Gran parte de las formulaciones estaban encaminadas a resolver problemas de diseño en la industria del automóvil (donde los sistemas de cadena cerrada son predominantes).

En relación a los autores o revisores de los métodos globales, sin ser exhaustivos, se puede citar a Orlandea et al. [58], Wehage y Haug [82] y García de Jalón et al. [43]. Una revisión más completa se puede encontrar en Schiehlen [66] y García de Jalón y Bayo [42] o la más reciente de Cuadrado [10].

Por su parte, los *métodos topológicos* tratan de aprovechar al máximo la topología o la conectividad de los elementos del sistema. En campos como la robótica y los vehículos espaciales se han desarrollado formulaciones muy eficientes que aprovechan precisamente las propiedades especiales de los sistemas que se estudian. Aunque existe una gran variedad de métodos se podrían mencionar por ejemplo el método de la inercia articulada de Featherstone [27, 28] modificado por Bae y Haug, para introducir las coordenadas naturales o los métodos semirecursivos. En García de Jalón et al. [40] se recoge de un modo más actual los últimos avances en este tipo de métodos.

El campo de aplicación de los algoritmos desarrollados en esta tesis es el de los mecanismos de carácter general; es decir, de cadena abierta o cerrada, en los que la formulación más sencilla es la de los métodos globales. Por eso, todas las formulaciones que se van a presentar en los siguientes apartados corresponden a métodos globales. Una vez formuladas las ecuaciones con métodos globales, se introduce la topología del sistema al establecer la estrategia de resolución de las ecuaciones.

3.5.2. Formulaciones basadas en coordenadas dependientes

Las ecuaciones diferenciales de la dinámica se pueden derivar de varias formas equivalentes (ecuaciones de Newton, de Lagrange, de Hamilton, ...). En lo que sigue se va a utilizar el método de las potencias virtuales por su sencillez. Con este método las ecuaciones diferenciales de la dinámica se plantean mediante la siguiente ecuación:

$$\dot{\mathbf{q}}^{*T} (\mathbf{M} \ddot{\mathbf{q}} - \mathbf{Q}) = 0 \quad (3.16)$$

En esta ecuación las fuerzas de inercia dependientes de la acelera-

ción aparecen en el término de la matriz de masas, y en el término \mathbf{Q} se agrupan las fuerzas exteriores y las fuerzas de inercia dependientes de las velocidades.

Hay que tener en cuenta que las velocidades virtuales que aparecen en esta ecuación no son, en general, independientes: están relacionadas como mínimo por m ecuaciones de restricción independientes como se escribió en la ecuación 3.2a. Esto implica que la cantidad entre paréntesis de la expresión 3.16, $(\mathbf{M} \ddot{\mathbf{q}} - \mathbf{Q})$, no tiene que anularse. Para hallar los n elementos del vector de coordenadas dependientes se precisa, por tanto, la información que proporcionan las restricciones. Mediante esas m ecuaciones y considerando los f grados de libertad que aparecen en las ecuaciones de la dinámica se dispone de información suficiente para hallar los $n = m + f$ elementos del vector de coordenadas dependientes.

Existen dos formulaciones básicas de la dinámica (la forma en la que se introducen las ecuaciones de restricción en la ecuación 3.16) que permiten determinar las aceleraciones dependientes: multiplicadores de Lagrange y matriz \mathbf{R} . En los dos apartados que siguen se describen estos métodos así como sus problemas fundamentales.

Multiplicadores de Lagrange

Las velocidades virtuales que aparecen en la expresión 3.16 deben satisfacer las m ecuaciones de restricción del mecanismo (ec. 3.2a). Además, como las ecuaciones de restricción no dependen explícitamente del tiempo para las velocidades virtuales, se verifica la ecuación:

$$\Phi_{\mathbf{q}}(\mathbf{q}, t) \dot{\mathbf{q}}^* = 0 \quad (3.17)$$

Donde t se considera constante. Trasponiendo y postmultiplicando por un vector de coeficientes indeterminados (m multiplicadores de Lagrange) se obtiene para estas velocidades virtuales:

$$\dot{\mathbf{q}}^{*T} \Phi_{\mathbf{q}}^T(\mathbf{q}, t) \lambda = 0 \quad (3.18)$$

Si se suma a esta ecuación la expresión 3.16 se tiene:

$$\dot{\mathbf{q}}^{*T} (\mathbf{M}\ddot{\mathbf{q}} - \mathbf{Q} + \Phi_{\mathbf{q}}^T(\mathbf{q}, t) \lambda) = 0 \quad (3.19)$$

Los m multiplicadores de Lagrange que se han introducido siempre se podrán elegir, sean cuales sean los valores del vector de velocidades virtuales, de forma que el paréntesis en esta ecuación sea nulo. Así se llega a la ecuación de la expresión 3.15 que se repite a continuación; en la que, tal como se expuso, están presentes las fuerzas de inercia dependientes de la aceleración, las fuerzas exteriores y las fuerzas de inercia dependientes de las velocidades, y por último las fuerzas internas debidas a las restricciones impuestas al mecanismo.

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \lambda = \mathbf{Q} \quad (3.20)$$

En estas últimas los multiplicadores representan la magnitud de las fuerzas debidas a cada restricción. Las derivadas con respecto a las coordenadas dependientes \mathbf{q} de cada una de las ecuaciones de restricción, que son las filas de la matriz jacobiana, definen las direcciones de esas fuerzas.

Si se combina esta última expresión con la del sistema de ecuaciones algebraicas que representan el conjunto de las restricciones (ecuación 3.14), se obtiene:

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \lambda &= \mathbf{Q} \\ \Phi(\mathbf{q}, t) &= 0 \end{aligned} \quad (3.21)$$

En la expresión 3.21, que representa el equilibrio dinámico de las fuerzas del mecanismo, hay n ecuaciones que junto con las m ecuaciones de restricción iniciales forman un sistema de $n + m$ ecuaciones (tantas ecuaciones como grados de libertad).

Nótese sin embargo que este sistema de ecuaciones algebraico-diferenciales es de índice 3 (en el siguiente apartado se hablará muy brevemente del índice de estas ecuaciones). Es conocida la dificultad de integrar este tipo de sistemas. Por eso, se pueden sustituir las m restricciones por la segunda derivada de estas, obteniéndose entonces un sistema de índice 1, con $n + m$ incógnitas (coordenadas y multiplicadores):

$$\begin{pmatrix} \mathbf{M} & \Phi_{\mathbf{q}}^T \\ \Phi_{\mathbf{q}}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{q}} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{Q} \\ \mathbf{c} \end{pmatrix} \quad (3.22)$$

De este sistema se pueden obtener tanto las aceleraciones dependientes como los m multiplicadores de Lagrange que representan las magnitudes de las fuerzas. La matriz del sistema es simétrica y con frecuencia altamente hueca.

Se plantean dos inconvenientes al resolver el problema expresado por la ecuación 3.22: el primero es el gran número de incógnitas (solo sería necesario calcular los valores de los multiplicadores si interesase conocer exactamente las magnitudes de las fuerzas debidas a las restricciones). El segundo, más grave, tiene su origen en el hecho de emplear las segundas derivadas de las ecuaciones de restricción (motivado por la reducción del sistema DAE desde *índice 3* a *índice 1*) en vez de usar las propias restricciones. Esto plantea un problema de estabilidad en la solución: en el proceso de integración numérica las ecuaciones de restricción tienden a dejar de cumplirse. Su violación crece de forma lineal con el tiempo. Posteriormente se estudiará este efecto con más detalle.

El índice de una ecuación diferencial algebraica

Según Ascher y Petzold [4], las ecuaciones o sistemas de ecuaciones diferenciales algebraicos se pueden clasificar en dos grandes grupos: implícitos y semiexplícitos. Estos últimos en realidad son un conjunto de ecuaciones ordinarias con restricciones y se puede expresar de la siguiente forma (en este caso una ecuación de segundo orden en la forma llamada de *Hessenberg*):

$$\begin{aligned} x' &= \mathbf{F}(t, x, y, z) \\ y' &= \mathbf{G}(t, x, y) \\ 0 &= \mathbf{H}(t, y) \end{aligned}$$

Donde el producto de las derivadas parciales $\mathbf{H}_y \cdot \mathbf{G}_x \cdot \mathbf{F}_z$ es no singular. Este es el caso del problema de las restricciones de los

sistemas multicuerpo (donde se puede definir $x = \dot{\mathbf{q}}$, $y = \mathbf{q}$ y z como los multiplicadores de Lagrange) y el conjunto de ecuaciones anteriores se puede escribir como:.

$$\mathbf{M}(y) x' = \mathbf{Q}(y, x) - \Phi_{\mathbf{q}}^T(y) \cdot z \quad (3.23a)$$

$$y' = x \quad (3.23b)$$

$$\Phi(y, t) = \mathbf{0} \quad (3.23c)$$

Se llama índice de la ecuación algebraica al número de veces que debe ser derivada hasta obtener un sistema de ecuaciones ordinarias. El índice de la ecuación también es el número de veces que es necesario derivarla para resolver y' (aquí se incluyen *todas las variables*) de forma única en función de y y t . En este caso, si se deriva dos veces la ecuación de la expresión 3.23c se obtiene:

$$\begin{aligned} \mathbf{M}(y) x' &= \mathbf{Q}(y, x) - \Phi_{\mathbf{q}}^T(y) \cdot z \\ y' &= x \end{aligned} \quad (3.24)$$

$$\Phi_{\mathbf{q}}(y)x' = -\Phi'_{\mathbf{q}} x - \Phi''$$

que es un sistema de ecuaciones ordinarias en \mathbf{q} y $\dot{\mathbf{q}}$. Para tener ecuaciones diferenciales ordinarias en todas las variables (incluido λ) se debe derivar una vez más: de ahí que se hable de índice 3.

Método basado en la matriz \mathbf{R}

Otra forma de obtener las ecuaciones de la dinámica se basa en la consideración de que la ecuación 3.17 establece que las velocidades virtuales deben pertenecer al núcleo de la matriz jacobiana $\Phi_{\mathbf{q}}$. Se puede generar una matriz \mathbf{R} cuyas columnas sean una base de dicho subespacio como se vio en el apartado 3.4. Esta matriz tendrá por tanto n filas y f columnas (la dimensión del núcleo de la matriz jacobiana es precisamente el número f de grados de libertad) y verifica la expresión:

$$\Phi_{\mathbf{q}} \mathbf{R} = \mathbf{0} \quad (3.25)$$

Como las columnas de la matriz son una base para el núcleo de $\Phi_{\mathbf{q}}$, cualquier velocidad virtual deberá poder expresarse como combinación lineal de dicha base; esto es, existirán f elementos $\dot{\mathbf{z}}^*$ tal que:

$$\dot{\mathbf{q}}^* = \mathbf{R} \dot{\mathbf{z}}^* \quad (3.26)$$

donde $\dot{\mathbf{z}}^*$ es un vector de velocidades (virtuales) *independientes*. Esto hace posible que el factor que las multiplica en la siguiente ecuación (sustituyendo en 3.16) deba ser igual a cero:

$$\dot{\mathbf{z}}^{*T} \mathbf{R}^T (\mathbf{M} \ddot{\mathbf{q}} - \mathbf{Q}) = 0 \quad (3.27)$$

Es decir:

$$\mathbf{R}^T \mathbf{M} \ddot{\mathbf{q}} = \mathbf{R}^T \mathbf{Q} \quad (3.28)$$

Esta última expresión, junto con las ecuaciones de restricción, constituyen un sistema de n ecuaciones de *índice 3*. Usando la segunda derivada de las restricciones (igual que se ha hecho más arriba en el caso del método de los multiplicadores de Lagrange) se puede reducir el sistema a *índice 1*:

$$\begin{pmatrix} \Phi_{\mathbf{q}}^T \\ \mathbf{R}^T \mathbf{M} \end{pmatrix} \ddot{\mathbf{q}} = \begin{pmatrix} \mathbf{c} \\ \mathbf{R}^T \mathbf{Q} \end{pmatrix} \quad (3.29)$$

El sistema anterior es un sistema de n ecuaciones frente al sistema de $n + m$ ecuaciones de los multiplicadores. Además la propia factorización de la matriz jacobiana utilizada en el cálculo de la base del núcleo de esta matriz (que son las columnas de \mathbf{R}) se puede utilizar en la resolución del sistema.

En general, la ecuación 3.29 es preferible cuando el mecanismo tiene pocos grados de libertad, mientras que el método de los multiplicadores de Lagrange será preferible cuando el número de grados de libertad sea grande y cuando sea necesario calcular simultáneamente la magnitud de las fuerzas de restricción.

En cuanto a los *problemas de estabilidad* apuntados en el método anterior aquí ocurre exactamente lo mismo: en la integración numérica, las ecuaciones van dejando de cumplirse poco a poco, debido a la derivación (doble) hecha en las ecuaciones de restricción.

Formulaciones de este tipo fueron introducidas por Kamman y Huston [46].

3.5.3. Estabilización de las ecuaciones

Como se ha visto en los apartados anteriores en el cálculo de las aceleraciones dependientes se ha utilizado la segunda derivada de las ecuaciones de restricción dando lugar a formulaciones de *índice 1*. En general esta segunda derivada se puede expresar de la siguiente forma:

$$\ddot{\Phi}(\mathbf{q}, t) = \Phi_{\mathbf{q}} \ddot{\mathbf{q}} + \dot{\Phi}_{\mathbf{q}} \dot{\mathbf{q}} + \Phi_t \quad (3.30)$$

La solución general tendría la forma siguiente, con dos constantes de integración:

$$\Phi(\mathbf{q}, t) = \mathbf{c}_1 t + \mathbf{c}_2 \quad (3.31)$$

Las condiciones iniciales (de cumplimiento de las ecuaciones de restricción) de posición y velocidad implican un valor nulo para las constantes de integración. No obstante, en la práctica, los errores numéricos hacen que cada instante de tiempo se comporte como un instante inicial con constantes de integración no nulas: la ecuación anterior y la experiencia indican que numéricamente el valor de las restricciones puede crecer indefinidamente con el tiempo y hacerse inestable, salvo en simulaciones muy cortas. Las técnicas que se mencionan en los apartados siguientes han sido propuestas para estabilizar las ecuaciones sin renunciar a la sencillez y eficacia de las formulaciones de *índice 1*.

Estabilización de Baumgarte

En 1972 Baumgarte [5] introdujo un método que consiste en añadir a las ecuaciones de la dinámica una modificación de la ecuación 3.30 en vez del valor de la segunda derivada de las restricciones. Se introducen dos parámetros de modo que los errores numéricos se amortigüen y desaparezcan en las etapas siguientes.

Sin embargo muchos autores, como Haug [35], han indicado que este método no siempre consigue resolver el problema satisfactoriamente debido, principalmente, a la dificultad de elegir adecuadamente los parámetros de Baumgarte y las no-linealidades de la formulación.

Estabilización basada en la proyección de posiciones, velocidades y aceleraciones

Estos métodos consisten en proyectar las variables cuyas ecuaciones de restricción no se han exigido en la integración numérica sobre los subespacios en los que dichas ecuaciones se satisfacen exactamente. El método fue introducido en 1991 por Lubich [51]. Posteriormente ha sido desarrollada y utilizada por Bayo y Ledesma [6], Cuadrado et al. [11], Dehombreux et al. [17] y otros autores. Otra referencia importante es el libro de Schwerin [68].

La proyección que se menciona se plantea como la resolución del problema de encontrar un mínimo sujeto a restricciones: si \mathbf{q} es el vector que cumple exactamente las ecuaciones de restricción y \mathbf{q}^* el resultado aproximado resultante de la integración numérica se podría establecer las siguientes igualdades (linealizando el problema):

$$\Phi(\mathbf{q}) = \Phi(\mathbf{q}^* + \delta\mathbf{q}) = \mathbf{0} \Rightarrow \Phi_{\mathbf{q}}\delta\mathbf{q} = -\Phi(\mathbf{q}^*) \quad (3.32)$$

El problema consiste en minimizar el cuadrado de la norma siguiente, sometida a las restricciones anteriores.

$$\|\delta\mathbf{q}\|_{\mathbf{W}}^2 = \delta\mathbf{q}^T \mathbf{W} \delta\mathbf{q}$$

Utilizando multiplicadores de Lagrange:

$$\begin{pmatrix} \mathbf{W} & \Phi_{\mathbf{q}}^T \\ \Phi_{\mathbf{q}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta \mathbf{q} \\ \mu \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -\Phi(\mathbf{q}^*) \end{pmatrix} \quad (3.33)$$

Si en esta expresión se toma como matriz de ponderación (en la norma) la matriz de inercia, se obtiene un sistema de ecuaciones lineales con la misma matriz que las ecuaciones diferenciales del movimiento: se puede utilizar la misma factorización en ambas matrices. Téngase en cuenta que la matriz del sistema dado en 3.33 está evaluada en la mejor posición conocida hasta ese momento proporcionada por el integrador.

Es posible utilizar también la matriz identidad. En ese caso, la solución es la de mínima norma euclídea del sistema de ecuaciones lineales 3.32, donde el rango coincide con el número de filas:

$$\delta \mathbf{q} = -\Phi_{\mathbf{q}}^T (\Phi_{\mathbf{q}} \Phi_{\mathbf{q}}^T)^{-1} \Phi(\mathbf{q}^*) \quad (3.34)$$

De forma análoga (no es necesario linealizar) se pueden obtener expresiones similares para corregir las velocidades y aceleraciones.

$$\begin{pmatrix} \mathbf{W} & \Phi_{\mathbf{q}}^T \\ \Phi_{\mathbf{q}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta \dot{\mathbf{q}} \\ \nu \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -\Phi_{\mathbf{q}} \dot{\mathbf{q}}^* \end{pmatrix} \quad (3.35)$$

$$\begin{pmatrix} \mathbf{W} & \Phi_{\mathbf{q}}^T \\ \Phi_{\mathbf{q}} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta \ddot{\mathbf{q}} \\ \kappa \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ -\Phi_{\mathbf{q}} \ddot{\mathbf{q}}^* - \dot{\Phi}_{\mathbf{q}} \dot{\mathbf{q}}^* \end{pmatrix} \quad (3.36)$$

Estas proyecciones, como forma de obligar al cumplimiento de las ecuaciones de restricción, son bastante eficientes y consiguen resolver el problema, como se puede ver en las referencias aportadas. Sin embargo, la experiencia de muchos investigadores, como se verá más adelante, es que los métodos basados en la integración de un conjunto de coordenadas independientes son casi siempre superiores.

Estabilización mediante la imposición periódica de las ecuaciones de restricción

Otra forma de evitar la acumulación de errores por incumplimiento de las ecuaciones de restricción consiste en vigilar la norma de las

violaciones de estas restricciones y detener la integración para resolver los problemas de posiciones y velocidades cuando la violación de las restricciones supera un cierto valor. Con una tolerancia pequeña para el error en las ecuaciones de restricción (por ejemplo 10^{-6}), el ajuste de posiciones y velocidades es relativamente frecuente, lo que indica que los métodos de resolución en coordenadas dependientes no son buenos a la hora de garantizar las restricciones en posición y velocidad.

Integración de sistemas de ecuaciones de *índice 2*

Otra posibilidad de aliviar, o incluso eliminar completamente, los problemas de inestabilidad en la integración fue sugerida por Paul [59] y consiste en integrar un vector formado por todas las velocidades dependientes y solo las aceleraciones independientes.

$$\{\ddot{\mathbf{z}}, \dot{\mathbf{q}}\}_t \xrightarrow{\text{Integración}} \{\dot{\mathbf{z}}, \mathbf{q}\}_{t+\Delta t}$$

Este método implica la integración de la ecuación de restricción de velocidades que es 3.2a es mucho menos crítica que la de aceleraciones: de hecho no es inestable en el sentido de que las restricciones no dejan de cumplirse de forma lineal con el tiempo, aunque los errores numéricos debidos a la integración siguen influyendo. Si se quiere mejorar la precisión de los resultados puede utilizarse conjuntamente con una proyección de posiciones o con la técnica de imponer periódicamente las ecuaciones de restricción. En este caso el ajuste en las posiciones es mucho menos frecuente. Este método puede ser muy eficiente y en esta tesis se ha utilizado, como se verá, en el contexto de las formulaciones basadas en la integración de coordenadas independientes.

3.5.4. Formulaciones basadas en penalizadores

Una alternativa a las formulaciones dinámicas generales vistas en las secciones anteriores es el método de los penalizadores, propuesto por Bayo et al. [7]. En este método las ecuaciones de restricción 3.14

no se imponen directamente si no que se *penaliza* el no cumplimiento de dichas ecuaciones con fuerzas que se oponen a su violación:

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \alpha (\ddot{\Phi} + 2\Omega\mu\dot{\Phi} + \Omega^2\Phi) = \mathbf{Q} \quad (3.37)$$

Físicamente es como si cada restricción se sustituyera por un resorte, un amortiguador y una inercia que tienden a obligar a su cumplimiento. Los vectores Φ , $\dot{\Phi}$ y $\ddot{\Phi}$ representan las violaciones de las ecuaciones de restricción en posición, velocidad y aceleración. Los coeficientes de estos vectores se eligen para que exista amortiguamiento crítico en el sistema vibratorio que se introduce para cada restricción. Los coeficientes α deben ser lo suficientemente grandes para que se satisfagan con precisión las restricciones.

Si se sustituye la derivada segunda de las restricciones según la expresión 3.30 se obtiene:

$$\begin{aligned} & (\mathbf{M} + \Phi_{\mathbf{q}}^T \alpha \Phi_{\mathbf{q}}) \ddot{\mathbf{q}} = \\ \mathbf{Q} - \Phi_{\mathbf{q}}^T \alpha (\dot{\Phi}_{\mathbf{q}}\dot{\mathbf{q}} + \dot{\Phi}_t + 2\Omega\mu\dot{\Phi} + \Omega^2\Phi) \end{aligned} \quad (3.38)$$

La característica más importante de este sistema es que las coordenadas dependientes se tratan *como si fuesen* independientes, por lo que no hay problemas de estabilización de las ecuaciones de restricción. El problema fundamental es determinar el equilibrio necesario en la magnitud de los penalizadores para evitar que aparezcan problemas de condicionamiento numérico. La utilización del *método de la Lagrangiana aumentada* introducido por Vanderplaats [80] resuelve en gran manera ese problema y mediante un proceso iterativo mejora la precisión. El método de la Lagrangiana aumentada consiste en introducir en la ecuación de los penalizadores una corrección en forma de un vector de multiplicadores de Lagrange:

$$\mathbf{M}\ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \alpha (\ddot{\Phi} + 2\Omega\mu\dot{\Phi} + \Omega^2\Phi) + \Phi_{\mathbf{q}}^T \lambda^* = \mathbf{Q} \quad (3.39)$$

Si se compara la ecuación 3.37, basada en penalizadores, con la ecuación 3.21, basada en los multiplicadores de Lagrange y se tiene en

cuenta la expresión 3.30 se puede llegar al siguiente proceso iterativo (cfr. García de Jalón y Bayo [42]) que permite mejorar el valor de las aceleraciones:

$$\begin{aligned} & (\mathbf{M} + \mathbf{\Phi}_q^T \alpha \mathbf{\Phi}_q) \ddot{\mathbf{q}}_{i+1} = \\ & \mathbf{M} \ddot{\mathbf{q}}_i - \mathbf{\Phi}_q^T \alpha \left(\dot{\mathbf{\Phi}}_q \dot{\mathbf{q}} + \dot{\mathbf{\Phi}}_t + 2\Omega \mu \dot{\mathbf{\Phi}} + \Omega^2 \mathbf{\Phi} \right)_i \end{aligned} \quad (3.40)$$

Método global de Bayo y Cuadrado basado en penalizadores, métodos de proyección y regla trapezoidal

Cuadrado et al. [11, 12] y Cardenal et al. [8] han utilizado un método especialmente eficiente basado en penalizadores con Lagrangiana aumentada y proyección de velocidades y aceleraciones ya que se utiliza un único término de penalización (para las restricciones en la posición). Se integra la siguiente ecuación de *índice* 3:

$$\mathbf{M} \ddot{\mathbf{q}} + \mathbf{\Phi}_q^T \alpha \mathbf{\Phi} + \mathbf{\Phi}_q^T \lambda^* = \mathbf{Q} \quad (3.41)$$

Este método tiene la ventaja de utilizar coordenadas naturales, que como se ha visto, consigue una matriz de inercia constante, evita la aparición de fuerzas de inercia que dependan de la velocidad y tiene ecuaciones de restricción lineales o cuadráticas. También, al introducir las ecuaciones de restricción en posición el problema se hace menos sensible al valor del penalizador.

Método de von Schwerin

La formulación que se describe a continuación ha sido desarrollada en el *Interdisciplinary Center for Scientific Computing*, en la universidad de Heidelberg. Se describe en una monografía publicada por von Schwerin [68] en 1999. Las características más importantes de este trabajo se pueden resumir en los puntos siguientes:

- Se ha tratado de optimizar la resolución del problema dinámico directo de sistemas multicuerpo en coordenadas naturales:

$$\begin{pmatrix} \mathbf{M} & \Phi_{\mathbf{q}}^T \\ \Phi_{\mathbf{q}}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \ddot{\mathbf{q}} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{Q} \\ \mathbf{c} \end{pmatrix} \quad (3.42)$$

- Se utilizan coordenadas naturales tratando de aprovechar sus principales características: ecuaciones de restricción cuadráticas o lineales, matriz de masas constante y ausencia de fuerzas de inercia dependientes de la velocidad.
- Se estabiliza las ecuaciones de restricción y de su primera derivada por medio de proyecciones ortogonales de posiciones y velocidades.
- Se estudia a fondo la resolución numérica del sistema de ecuaciones lineales 3.42 mediante dos variantes: La RSM (*Range Space Method*) y la NSM (*Null Space Method*) comparando la eficiencia para distintos valores de la relación entre el número de ecuaciones de restricción m y el número de coordenadas dependientes n .

Del estudio presentado por von Schwerin se concluye que el método basado en el subespacio nulo, NSM, es casi siempre superior. Este método está basado en la determinación de una base del subespacio nulo de la matriz jacobiana de las ecuaciones de restricción. La determinación de esta base se hace mediante un método de partición de variables en dependientes e independientes. Este método es sustancialmente idéntico al método de proyección de velocidades mediante la matriz R que se verá en los apartados siguientes y que implica la integración de un subconjunto de coordenadas independientes.

3.5.5. Formulaciones basadas en coordenadas independientes

Una forma de eliminar cualquier inestabilidad en la integración de las ecuaciones diferenciales del movimiento en coordenadas dependientes, es la integración de un subconjunto de *coordenadas independientes*, que no estén relacionadas por ninguna ecuación de restricción.

Una solución para lograr este objetivo es la partición de coordenadas propuesta por Wehage y Haug [82]. En ese mismo año Serna et al. [70] propusieron de modo independiente un método de partición de coordenadas similar haciendo uso de la matriz \mathbf{R} , cuyas columnas constituyen una base del núcleo de $\Phi_{\mathbf{q}}$. El vector de coordenadas dependientes y las ecuaciones de restricción de velocidades se particionan de la siguiente forma:

$$\mathbf{q}^T = (\mathbf{q}_d^T \quad \mathbf{q}_i^T) \quad (3.43)$$

$$\begin{pmatrix} \Phi_{\mathbf{q}}^d & \Phi_{\mathbf{q}}^i \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_d \\ \dot{\mathbf{q}}_i \end{pmatrix} = \mathbf{b} \quad (3.44)$$

donde los subíndices d e i hacen referencia respectivamente al carácter dependiente o independiente. Como las coordenadas dependientes se han de poder expresar en función de las independientes, la matriz $\Phi_{\mathbf{q}}^d$ debe ser invertible, es decir, tener rango completo.

Para realizar la partición de coordenadas expuesta (entre coordenadas dependientes o independientes) en los artículos mencionados, se propuso el propio proceso de pivotamiento en la factorización de Gauss de la matriz jacobiana: no es difícil, utilizando pivotamiento total, descomponer las columnas de $\Phi_{\mathbf{q}}$ en aquellas en las que han aparecido pivots y en las que no han aparecido. Los elementos de \mathbf{q} relacionados con las primeras son las coordenadas dependientes, mientras que los relacionados con las segundas son las independientes.

Posteriormente, se ha propuesto para esta partición operaciones algebraicas mucho más sofisticadas: la descomposición en valores singulares en Singh y Likins [75] y Mani et al. [53]; también la descomposición QR en Kim y Vanderploeg [48].

Una vez realizada la partición de coordenadas, esta debe ser mantenida tanto tiempo como se pueda, por motivos de eficiencia en la integración numérica. Esto deja de ser posible cuando la matriz $\Phi_{\mathbf{q}}^d$ se hace singular o casi singular. El valor de los pivots, por ejemplo, en la triangularización de $\Phi_{\mathbf{q}}$ permite detectar fácilmente esta situación.

Es importante señalar que en este proceso se integran las aceleraciones y velocidades independientes: una vez determinadas en un instante t la integración numérica procede a calcular las posiciones y velocidades en el instante siguiente.

$$\{\ddot{\mathbf{z}}, \dot{\mathbf{z}}\}_t \xrightarrow{\text{Integración}} \{\dot{\mathbf{z}}, \mathbf{q}_i\}_{t+\Delta t}$$

A partir de estas se calculan las velocidades dependientes mediante la ecuación 3.44. El cálculo de las coordenadas \mathbf{q}_d es más costoso: debe resolverse de modo iterativo el sistema de ecuaciones no lineales $\Phi = 0$ correspondiente al problema de posición.

3.5.6. Coordenadas independientes y proyección de velocidades

Como se ha mencionado en el apartado anterior esta formulación fue introducida por Serna, Avilés y García de Jalón [70]. A los métodos de esta familia se les suele conocer con el nombre de *métodos de proyección de velocidades*, porque las velocidades independientes son la proyección de las velocidades dependientes sobre las filas de una matriz constante: tal como se vio en la sección 3.4 se puede encontrar una expresión general que relaciona las velocidades dependientes e independientes mediante la utilización de una matriz \mathbf{B} constante y cuyas filas son linealmente independiente de las filas de la matriz jacobiana:

$$\begin{pmatrix} \Phi_{\mathbf{q}} \\ \mathbf{B} \end{pmatrix} \dot{\mathbf{q}} = \begin{pmatrix} \mathbf{b} \\ \dot{\mathbf{z}} \end{pmatrix} \quad (3.45)$$

$$\dot{\mathbf{q}} = \mathbf{S} \mathbf{b} + \mathbf{R} \dot{\mathbf{z}} \quad (3.46)$$

Estas dos expresiones también indican la forma en la que se puede calcular la matriz \mathbf{R} y el factor $\mathbf{S}\mathbf{b}$. La matriz \mathbf{R} se podría calcular triangularizando la matriz del sistema 3.45 y realizando f vueltas atrás (el número de columnas de \mathbf{R}) con las correspondientes columnas de la matriz identidad \mathbf{I} como términos independientes. Por

otra parte, el término \mathbf{Sb} se puede calcular a partir del mismo sistema teniendo en cuenta que dicho término es igual a las velocidades dependientes $\dot{\mathbf{q}}$ cuando las velocidades independientes $\dot{\mathbf{z}}$ son cero.

De forma similar para las aceleraciones se desarrollaron en la sección 3.4 las siguientes expresiones:

$$\begin{pmatrix} \Phi_{\mathbf{q}} \\ \mathbf{B} \end{pmatrix} \ddot{\mathbf{q}} = \begin{pmatrix} \mathbf{c} \\ \ddot{\mathbf{z}} \end{pmatrix} \quad (3.47)$$

$$\ddot{\mathbf{q}} = \mathbf{S} \mathbf{c} + \mathbf{R} \ddot{\mathbf{z}} \quad (3.48)$$

De forma análoga a lo visto para el término \mathbf{Sb} en esta última expresión el término \mathbf{Sc} puede calcularse como las aceleraciones dependientes $\ddot{\mathbf{q}}$ cuando las aceleraciones independientes son cero.

Sustituyendo esta última expresión en la ecuación 3.28 de la dinámica en coordenadas dependientes, se llega al siguiente sistema:

$$\mathbf{R}^T \mathbf{M} \mathbf{R} \ddot{\mathbf{z}} = \mathbf{R}^T (\mathbf{Q} - \mathbf{M} \mathbf{S} \mathbf{c}) \quad (3.49)$$

Se trata de un sistema de ecuaciones diferenciales ordinarias con tantas ecuaciones como grados de libertad, que podrán ser integradas numéricamente de una forma estándar. Para definir completamente esta formulación solo falta por describir cómo se elige la matriz \mathbf{B} , necesaria para el cálculo de la matriz \mathbf{R} .

El procedimiento más sencillo está basado en la factorización de Gauss tal como se describe en el método de partición de coordenadas introducido por Wehage y Haug [82] o Serna et al. [70]. En estos métodos se forma la matriz \mathbf{B} como la combinación de unos y ceros que define las velocidades independientes como un subconjunto de las dependientes:

$$\mathbf{B} \dot{\mathbf{q}} = \begin{pmatrix} \mathbf{0} & \mathbf{I} \end{pmatrix} \dot{\mathbf{q}} = \dot{\mathbf{z}} \quad (3.50)$$

Usando esta partición de coordenadas, la ecuación 3.45 se puede expresar de la siguiente forma:

$$\begin{pmatrix} \Phi_{\mathbf{q}} \\ \mathbf{B} \end{pmatrix} \dot{\mathbf{q}} = \begin{pmatrix} \Phi_{\mathbf{q}}^d & \Phi_{\mathbf{q}}^i \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \dot{\mathbf{q}}_d \\ \dot{\mathbf{q}}_i \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ \dot{\mathbf{z}} \end{pmatrix} \quad (3.51)$$

Si se considera la ecuación 3.11 de la página 26:

$$\begin{pmatrix} \Phi_{\mathbf{q}} \\ \mathbf{B} \end{pmatrix} \mathbf{R} = \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \quad (3.52)$$

Se obtiene finalmente una ecuación que nos permite calcular la matriz \mathbf{R} . Es importante señalar que la matriz $\Phi_{\mathbf{q}}^d$ debe ser invertible:

$$\begin{pmatrix} \Phi_{\mathbf{q}}^d & \Phi_{\mathbf{q}}^i \\ \mathbf{0} & \mathbf{I} \end{pmatrix} (\mathbf{R}) = \begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \Rightarrow \mathbf{R} = \begin{pmatrix} -(\Phi_{\mathbf{q}}^d)^{-1} \Phi_{\mathbf{q}}^i \\ \mathbf{I} \end{pmatrix} \quad (3.53)$$

3.5.7. Método elegido

En esta tesis se ha optado por una formulación en coordenadas independientes y partición de coordenadas: en concreto la propuesta por Serna, Avilés y García de Jalón [70], basada en la proyección de velocidades, que se ha desarrollado en el apartado anterior.

3.6. Cuestiones numéricas

En los apartados anteriores se han descrito y finalmente se ha elegido una forma determinada de modelar el mecanismo. En concreto, dentro del ámbito de las coordenadas dependientes se ha optado por un sistema de coordenadas mixtas basadas en un sistema de coordenadas naturales a las que se añaden coordenadas relativas. También se ha elegido un método concreto para formular la dinámica: se integra solo un subconjunto de coordenadas independientes usando una formulación basada en la proyección de velocidades.

Se dedica este capítulo a la consideración de algunas cuestiones numéricas relevantes en la implementación de los métodos propuestos para la integración de las ecuaciones de la dinámica, con el ob-

jetivo de fundamentar la ventaja de usar los métodos desarrollados en el marco de este trabajo.

3.6.1. Integradores

Según menciona Vidal en sus tesis doctoral [81], en el análisis dinámico de mecanismos sigue siendo habitual el utilizar integradores de propósito general, desarrollados por especialistas en análisis numérico e integrados en librerías de dominio público o en librerías comerciales. Para esta tesis se han utilizado algunos integradores de la librería de Matlab. Esta librería ha sido desarrollada por Shampine y Reichelt [73]. El primero de estos autores es uno de los investigadores de más prestigio en esta área.

Estos integradores de librería suponen que los sistemas de ecuaciones diferenciales son de primer orden. Así pues, sistemas de segundo orden o superiores se tendrán que transformar en sistemas de primer orden a través de la utilización de un *vector de estado*, donde aparecen las posiciones y las derivadas. La derivada de ese vector es la función que debe ser calculada en cada paso de integración.

De alguna manera la eficiencia del integrador en términos coste computacional está relacionada con el coste de evaluar esa función, puesto que tendrá que hacerse muchas veces a lo largo del proceso de integración.

En esta tesis se han seleccionado dos integradores explícitos. Por una parte, el integrador *ode113* de Matlab, que es una versión actualizada del integrador escrito por Shampine y Gordon [72]. Se trata de un integrador de orden y paso variable, capaz de iniciar la integración sin recurrir a otras fórmulas, y basado en las fórmulas PECE de Adams-Bashforth-Moulton. Por otra parte se ha recurrido al integrador *ode45*, basado en una fórmula explícita de Runge-Kutta de órdenes 4 y 5 desarrollada por Dormand y Prince [19].

3.6.2. Eficiencia en la integración

En esta tesis cuando se habla de mejora de la eficiencia en la integración se hace referencia concretamente al conjunto de métodos

que permiten evaluar la derivada del vector de estado del integrador en un menor tiempo posible. En algunos procedimientos numéricos que emplean fórmulas de recurrencia, la aplicación sucesiva de la fórmula correspondiente se conoce como *evaluación de función*. En el caso del integrador, por cada paso de tiempo, es preciso evaluar varias veces el vector de derivadas de las funciones. De ahí que, en términos de rendimiento, sea crítico el coste de cada una de esas evaluaciones.

En la tabla 3.1 se resume el número necesario de evaluaciones de la derivada cuando se integran las ecuaciones de la dinámica de un mecanismo relativamente pequeño, con 5 grados de libertad y 35 coordenadas dependientes para conocer su evolución en el tiempo a lo largo de 10 segundos.

Integrador	ode45	ode113
pasos de integración	586	1452
evaluaciones de la derivada	3881	2937

Tabla 3.1: Número de evaluaciones

Es importante establecer las propiedades fundamentales de esa función objetivo y antes de abordar las diferentes estrategias que se han propuesto para mejorar la eficiencia y las que se proponen en esta tesis.

3.6.3. Estructura de la función objetivo

En el método propuesto, la estructura básica de la integración es la siguiente:

$$\{\ddot{\mathbf{z}}, \dot{\mathbf{q}}\}_{i+1} \xleftarrow{\text{Integración}} df(\{\dot{\mathbf{z}}, \mathbf{q}\}_i)$$

En la siguiente enumeración se muestra la estructura básica de la función (siguiendo el método elegido) y se indica el coste numérico de cada paso:

- Se calcula el valor de la matriz jacobiana $[\Phi_{\mathbf{q}}]$ y se hace la partición de coordenadas $\{\Phi_{\mathbf{q}}^d, \Phi_{\mathbf{q}}^i\}$.

- Se resuelve el siguiente sistema de ecuaciones y se calcula la matriz \mathbf{R} :

$$\Phi_{\mathbf{q}}^d \mathbf{R}^* = -\Phi_{\mathbf{q}}^i \Rightarrow \mathbf{R} = \begin{pmatrix} \mathbf{R}^* \\ \mathbf{I} \end{pmatrix}$$

Como se vio más arriba la matriz $\Phi_{\mathbf{q}}^d$ debe ser no singular durante todo el proceso de integración. Esto se garantiza mediante la partición de coordenadas (si se eligen un conjunto de coordenadas relativas adecuadas como independientes) salvo posiciones singulares del mecanismo.

- Se calcula el vector de velocidades: $\dot{\mathbf{q}} = \mathbf{R} \dot{\mathbf{z}}$. En el caso de un sistema holónomo $\dot{\mathbf{q}} = \mathbf{R} \dot{\mathbf{z}} + \mathbf{Sb}$ con \mathbf{Sb} calculado como $\Phi_{\mathbf{q}}^d [\mathbf{Sb}] = -\Phi_{\mathbf{t}}$. *En este paso se imponen las ecuaciones de restricción de las velocidades.*
- Se calcula el término $\mathbf{c} = -\dot{\Phi}_{\mathbf{q}} \dot{\mathbf{q}}$ o en su caso $\mathbf{c} = -\dot{\Phi}_{\mathbf{q}} \dot{\mathbf{q}} - \dot{\Phi}_{\mathbf{t}}$ que aparece en las ecuaciones de restricción de las aceleraciones.
- Se determina el valor de \mathbf{Sc} mediante el sistema:

$$\Phi_{\mathbf{q}}^d [\mathbf{Sc}] = \mathbf{c}$$

- Finalmente se calculan las aceleraciones independientes utilizando las ecuaciones de la dinámica. *En este paso se imponen también las ecuaciones de restricción de las aceleraciones.*

$$\mathbf{R}^T \mathbf{M} \mathbf{R} \ddot{\mathbf{z}} = \mathbf{R}^T (\mathbf{Q} - \mathbf{M} \mathbf{S} \mathbf{c})$$

- Si se quiere mejorar la precisión se puede recurrir a la imposición de las restricciones de posición cuando la norma de estas supere un determinado valor. Esto obligaría a resolver cada cierto tiempo el problema de posición mediante un proceso iterativo. En este proceso se tendría que resolver el sistema de ecuaciones dado por:

$$\Phi_{\mathbf{q}}^d (\mathbf{q}_{i+1} - \mathbf{q}_i) = -\Phi$$

En el proceso se resuelven un mínimo de cuatro veces un sistema de ecuaciones donde la matriz del sistema es la partición dependiente de la matriz jacobiana $\Phi_{\mathbf{q}}^d$. Es de particular importancia, por tanto, mencionar sus características.

3.6.4. Características de la matriz jacobiana

Las dimensiones de esta matriz son del orden del número de coordenadas (en un mecanismo típico el número de grados de libertad es pequeño). En la figura 3.1 se ve la forma que tendría esta matriz en un mecanismo con 10 grados de libertad y 197 coordenadas dependientes.

Como se puede ver en el ejemplo de la figura 3.1 la característica más importante de la matriz jacobiana es el gran número de elementos nulos (hay que pensar que en cada ecuación de restricción solo intervienen un pequeño número de coordenadas). Se trata de matrices altamente huecas (*sparse*). En el ejemplo de la figura 3.1 los elementos no nulos representan un 2,6 %.

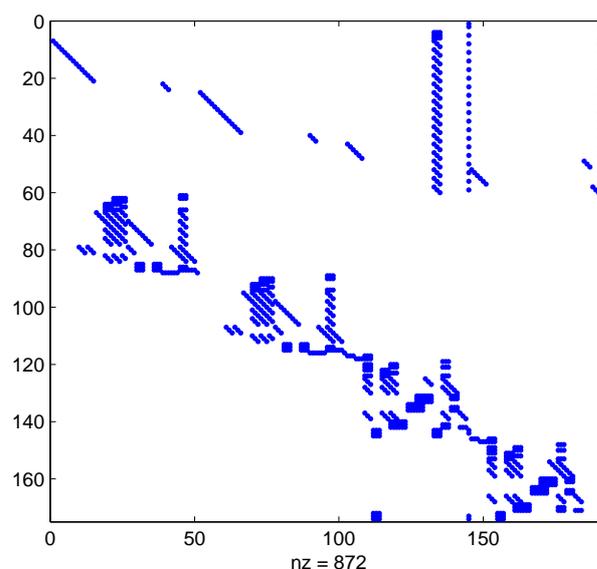


Figura 3.1: Ejemplo de matriz jacobianaa

Sin hablar todavía de la resolución de los sistemas, al que se dedicará el apartado siguiente, se pueden mencionar algunas técnicas básicas que aprovechan la naturaleza hueca de estas matrices:

- Se pueden calcular y almacenar solo los elementos no nulos del Jacobiano: la primera vez que se genera el jacobiano (fuera de

la función objetivo) se almacena la información de la posición de todos los elementos que pueden ser no nulos y utilizarla en futuras referencias. De esta forma el tiempo que se emplea para el cálculo de la matriz jacobiana es mucho menor: en el ejemplo de la figura es del orden de un 40 % del tiempo original.

- No se opera con los elementos que son nulos: por ejemplo, en una supuesta factorización, triangularización u otra técnica del álgebra numérica, se evitan las operaciones con elementos nulos. Esto exige, tradicionalmente, realizar un preproceso simbólico del problema que se pretende resolver.

3.6.5. Estrategias de reordenamiento

En general, las estrategias de resolución de sistemas *sparse* buscan reducir el número de operaciones aritméticas para mejorar la eficiencia. Por otra parte, en la resolución de un sistema es necesario factorizar la matriz del mismo y en ese proceso se pueden generar nuevos elementos no nulos: en general las matrices factor tienen más elementos no nulos que la que ha sido factorizada. A este fenómeno se le llama llenado de la matriz o *fill-in*. Lógicamente, cuantos más elementos no nulos se generen durante la factorización, mayor será el número de operaciones aritméticas necesarias con la consiguiente degradación de la eficiencia en el cálculo. Una buena parte de las técnicas de reordenamiento de matrices dispersas tienen como objetivo minimizar este *fill-in*.

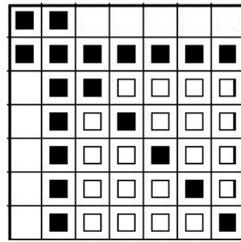
Siguiendo la clasificación de Duff et al. [21] existen varias estrategias para conseguir este objetivo que se verán con detalle en el capítulo siguiente:

- Las estrategias de pivotamiento local consisten en reordenar la matriz del sistema de forma que al hacer la factorización *de toda la matriz* se produzca el mínimo llenado posible. Por reordenamiento se entiende las permutaciones que se hacen en las filas y las columnas de la matriz.
- Otra posibilidad es reordenar la matriz para conseguir *confinar* el llenado a una determinada parte de la misma. En este caso

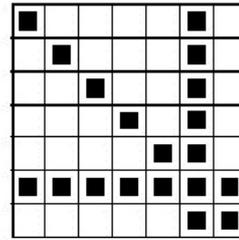
se busca que después del reordenamiento la matriz tenga una determinada forma. Un caso típico es conseguir una matriz banda o una matriz diagonal por bloques. En estos casos el llenado quedaría confinado a los bloques de la diagonal.

- Por último están los métodos en los que se busca una forma que permita resolver el sistema por bloques de forma que solo es necesario factorizar los bloques de la diagonal, que suelen ser mucho más pequeños.

Cada una de estas estrategias se desarrollará con detalle en el siguiente capítulo. De forma esquemática, se muestran algunos ejemplos de cada una de ellas:



(a) Matriz original



(b) Matriz reordenada

Figura 3.2: Efecto de la factorización de Gauss

En la figura 3.2 se puede ver el efecto que tiene la factorización de Gauss en una matriz según el reordenamiento utilizado. En 3.2a está la matriz original: los puntos negros son los elementos no nulos y los que están marcados los que aparecen en el llenado. En 3.2b se puede ver que después de hacer un reordenamiento adecuado no se produce ningún llenado en la matriz.

En la figura 3.3 se ha aplicado un reordenamiento para obtener una forma tridiagonal por bloques. El relleno, que sí se produce, está confinado a esos elementos.

Por último, en la figura 3.4 se ve el resultado final del reordenamiento a una forma triangular por bloques bordeada. Las zonas grises son partes de la matriz donde podrían aparecer elementos no

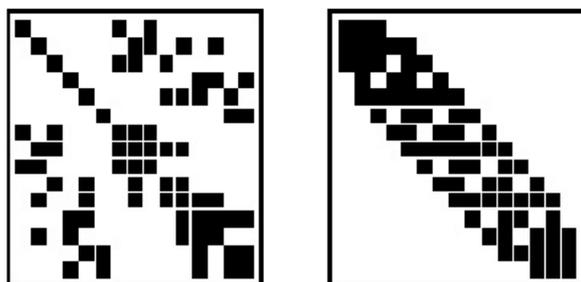
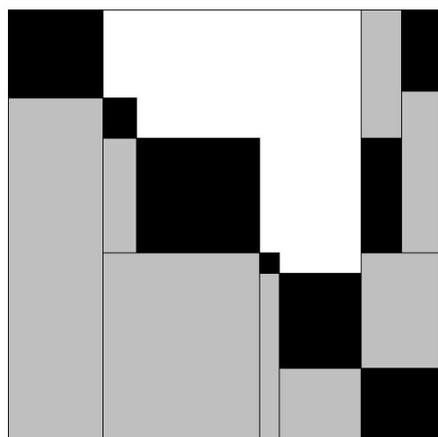


Figura 3.3: Reordenamiento Cuthill-McKee inverso

nulos si se factorizase la matriz completa. En cualquier caso la estrategia para resolver ese sistema no es la factorización de toda la matriz sino resolverla siguiendo los bloques de la diagonal.

Figura 3.4: Reordenamiento P^4

3.6.6. Resolución por bloques

Se ha mencionado en el apartado anterior la resolución por bloques de un sistema de ecuaciones después de reordenar la matriz del sistema. Se va a explicitar ahora el motivo por el que estos sistemas se pueden reducir a matrices triangulares por bloques (con borde o sin borde) y la forma de realizar esa resolución, siguiendo las indicaciones de Funes et al. [30], y distinguiendo los mecanismos de cadena

abierta de los mecanismos de cadena cerrada.

- En los mecanismos de cadena abierta siempre se pueden seleccionar las coordenadas en un determinado orden: comenzando por los puntos y vectores que están relacionados con elementos fijos, se puede recorrer los elementos del mecanismo siguiendo cada cadena abierta. A medida que se van escogiendo esas coordenadas se pueden seleccionar las restricciones en el orden en el que van apareciendo. Como la matriz jacobiana representa las derivadas de las restricciones con respecto a cada una de las coordenadas se puede ver fácilmente que siempre se podrá reducir a una forma triangular por bloques.
- Por otra parte un mecanismo de cadena cerrada siempre se puede *abrir* en un conjunto de cadenas abiertas. Para esas cadenas se puede aplicar lo dicho en el párrafo anterior y crear una submatriz con la forma indicada. Por otra parte, para *abrir* las cadenas cerradas es necesario imponer ciertas restricciones de *cierre de lazo* que generarán el borde. Esta técnica ya ha sido usada, entre otros, por Cardenal et al. [9] en la simulación cinemática de mecanismos.

Una vez garantizado que se pueden reordenar las matrices según los patrones indicados, la resolución por bloques es muy sencilla:

En cadena abierta:

$$\begin{pmatrix} \mathbf{D}_1 & & & & \\ \mathbf{E}_{21} & \mathbf{D}_2 & & & \\ \mathbf{E}_{31} & \mathbf{E}_{32} & \mathbf{D}_3 & & \\ \vdots & \vdots & \vdots & \cdot & \\ \mathbf{E}_{n1} & \mathbf{E}_{n2} & \mathbf{E}_{n3} & \dots & \mathbf{D}_n \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \vdots \\ \mathbf{x}_n \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \vdots \\ \mathbf{b}_n \end{pmatrix} \quad (3.54)$$

Se resuelve como:

$$\mathbf{D}_i \mathbf{x}_i = \mathbf{b}_i - \sum_{j=i}^{i-1} \mathbf{E}_{ij} x_j \quad i = 1, \dots, n \quad (3.55)$$

Para un mecanismo de cadena cerrada:

$$\begin{pmatrix} \mathbf{T} & \mathbf{B} \\ \mathbf{C} & \mathbf{A} \end{pmatrix} \begin{pmatrix} \mathbf{x}_T \\ \mathbf{x}_A \end{pmatrix} = \begin{pmatrix} \mathbf{b}_T \\ \mathbf{b}_A \end{pmatrix} \quad (3.56)$$

Se pueden resolver las ecuaciones:

$$(\mathbf{A} - \mathbf{C}\mathbf{T}^{-1}\mathbf{B}) \mathbf{x}_A = \mathbf{b}_A - \mathbf{C}\mathbf{T}^{-1} \mathbf{b}_T \quad (3.57)$$

$$\mathbf{x}_T = \mathbf{T}^{-1} (\mathbf{b}_T - \mathbf{B}\mathbf{x}_A) \quad (3.58)$$

Es importante señalar que en el último caso la dimensión del borde (su número de columnas) coincide con la dimensión de la banda inferior (su número de filas). En un reordenamiento óptimo esa dimensión es igual al número de restricciones de cierre de lazo que se hayan impuesto.

Capítulo 4

Estrategias de reordenamiento de matrices huecas

En objetivo principal de este capítulo es encontrar un reordenamiento óptimo para el problema de la resolución *por bloques* del sistemas de ecuaciones que se ha descrito en el apartado anterior, con la matriz jacobiana como matriz del sistema. Esto supone reducirla a una forma triangular por bloques –en los mecanismos de cadena abierta– o una forma triangular por bloques bordeada –si el mecanismo es de cadena cerrada–, con unas determinadas características para los bloques de la diagonal y el borde debido a la naturaleza del sistema (grados de libertad, etc.).

En las primeras secciones se va a presentar el estado del arte en lo que afecta a los reordenamientos más frecuentes, tanto para matrices simétricas como no simétricas. Primero se verán algunos conceptos de teoría de grafos que se pondrán en relación con la factorización de Gauss. Esto es necesario debido a que la mayoría de los reordenamientos utilizados, también los que se utilizarán como base en esta tesis, están pensados, como se indicó en el capítulo anterior, para reducir el llenado de las matrices en esa factorización.

En una segunda parte se abordarán los problemas que se presentan al intentar utilizar estos reordenamientos para una resolución por

bloques del sistema y se propondrá un método para solucionarlos.

4.1. Cuestiones generales de teoría de grafos

4.1.1. Definiciones previas

Se van a dar algunas definiciones de conceptos que se utilizan en teoría de grafos para aclarar el sentido en el que se van a utilizar en este capítulo.

Grafo $G(V, E)$ asociado a una matriz Dada una matriz se le puede asociar un grafo de la siguiente forma: se define un conjunto de *vértices*, uno por cada elemento de la diagonal etiquetado por el número i de la fila-columna a_{ii} —este conjunto se denota como $V(G)$; un conjunto de *aristas* $-E(G)$ -que unen el vértice i con el vértice j si y solo si el elemento $a_{ij} \neq 0$. Si se trata de una matriz simétrica en la que $a_{ij} = a_{ji}$ las aristas no tienen una dirección definida y el grafo se llama *grafo no dirigido*. En este caso se considera que para cada par de entradas en la matriz a_{ij} y a_{ji} sólo existe una arista (v_i, v_j) , siendo, por ejemplo $i < j$. En el caso de matrices no simétricas el grafo se llama *grafo dirigido* o también *digrafo*. En este caso existirá una arista por cada elemento de la matriz a_{ij} .

En la figura 4.1 se puede ver un ejemplo de cada uno de ellos.

Algunas propiedades que conviene señalar:

- $B = P^T A P$, donde P es una matriz de permutación, tiene el mismo grafo asociado (salvo la numeración de los vértices). Se habla en este caso de permutaciones simétricas.
- Se considera que $a_{ii} \neq 0$. Esto equivale a suponer que existe una arista que une cada vértice consigo mismo. En inglés *selfedge* (v_i, v_i) . En las representaciones de los grafos no se dibujan tales aristas.

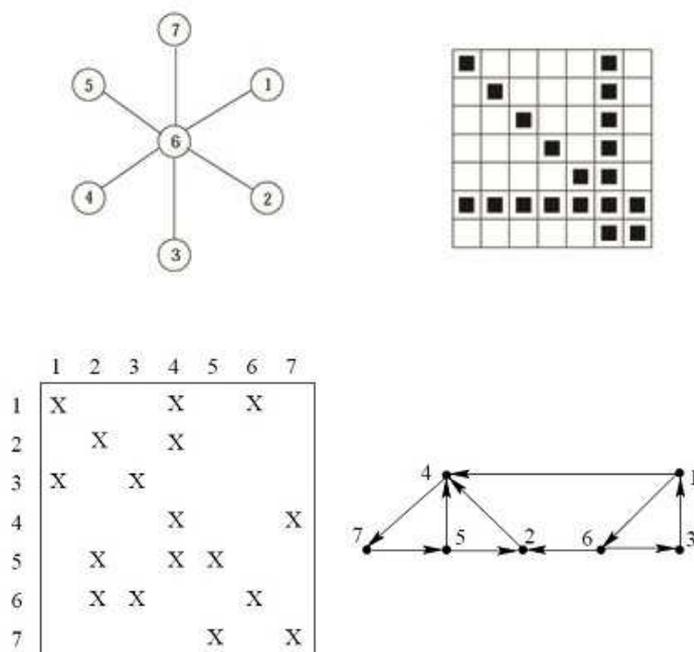


Figura 4.1: Grafos dirigidos y no dirigidos

Subgrafo Se define un *subgrafo* $G' = (V', E')$ de $G = (V, E)$ al grafo formado por un subconjunto de vértices y de aristas del grafo original. Como se verá no tiene por que ser adyacente.

Adyacencia Dos vértices unidos por una arista son *adyacentes*. En el caso de grafos no dirigidos se considera que esta arista es *incidente* en los dos vértices.

Grado El *grado* de un vértice es el número de aristas que inciden en él. También se puede definir como el cardinal del conjunto de adyacencia, como se verá.

En la figura 4.2 se puede ver que los vértices $\{11, 8, 5, 3\}$, junto con las aristas que los unen, forman una sección del grafo original. El vértice $\{8\}$ es adyacente a los vértices $\{11, 5\}$. El grado del vértice $\{8\}$ es 2, mientras que el grado del vértice $\{11\}$ es 5.

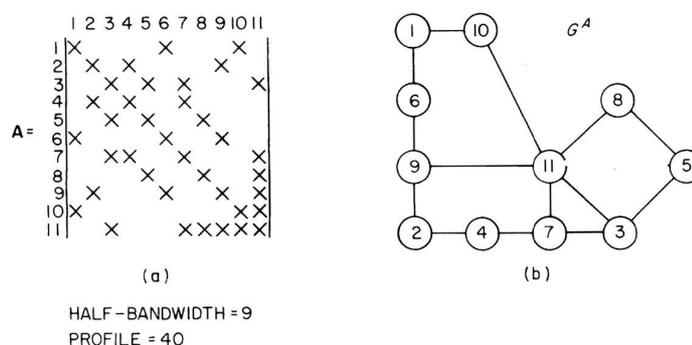


Figura 4.2: Grafo y matriz asociada

Conjunto de adyacencia Dado $W \subset V$ de $G(V, E)$ se define la *conjunto de adyacencia* de la siguiente forma:

$$Adj(W) = \{u \in V - W | \exists v \in W | (u, v) \in E\}$$

Esto es, el conjunto de los vértices que son adyacentes a W . En la figura 4.2, $Adj(\{1,6\}) = \{9,10\}$

Subgrafo completo Un subgrafo es *completo*, en inglés, *clique*, si cada par de vértices son adyacentes entre sí. En la figura 4.2 los vértices $V = \{3, 7, 11\}$ con las aristas que los unen entre sí, forman una sección "completa".

Camino Un *camino* entre dos vértices v_1 y v_m es un conjunto ordenado de vértices $V' = \{v_1, v_2, \dots, v_m\}$ de forma que v_i es adyacente a v_{i+1} .

Distancia Se llama *distancia* entre dos vértices $d(u, v)$ a la longitud del camino más corto entre ellos. Sólo se cuentan las aristas que unen los vértices. En la figura 4.2, la distancia entre el vértice 1 y el 3 es 3 y la distancia entre el 1 y el 5 es 4.

Excentricidad Se llama *excentricidad* de un vértice, $e(v)$, a la mayor distancia entre v y cualquier otro vértice del grafo. Se puede ver en la figura 4.2 que la excentricidad del vértice 1 es $e(1) = 4$.

Diámetro Se llama *diámetro* de un grafo a la mayor excentricidad entre todos los vértices. En el ejemplo anterior, correspondiente a la figura 4.2, el diámetro es 4.

Periféricos Un vértice es *periférico* si su excentricidad es igual al diámetro. En la figura 4.2 los vértices $\{1, 2, 4, 5, 6\}$ son periféricos.

Grafo conexo Se dice que un grafo es *conexo* si para cada par de vértices del grafo existe un camino que los une. Si llamamos *conectividad* a la relación " u está conectado con v ", esta relación es de equivalencia y las clases de equivalencia son las *componentes conexas* del grafo. Un grafo conexo sólo tiene una componente conexa.

Conectividad Se dice que un grafo no-completo tiene *conectividad* k , si es k -conexo y no es $(k+1)$ -conexo. Un grafo completo de $k + 1$ vértices tiene conectividad k .

Separadores Un *separador* es un conjunto de vértices que al desaparecer del grafo, junto con todas las aristas que inciden en ellos, dejan a éste dividido en dos grafos o más desconectados entre sí.

Extensión Se define el conjunto de vértices *extensión* de W ,

$$Span(W) = W \cup Adj(W)$$

Alcanzables Se dice que un vértice v es *alcanzable* desde u a través de S , si existe un camino de u a v , que contiene exclusivamente vértices que pertenecen a S o son adyacentes a u o v . A todos los vértices v alcanzables desde u a través de S , se les llama $Reach(u, S)$.

Se verifican las siguientes propiedades:

- I) Si S está vacío $\Rightarrow Reach(u, S) = Adj(u)$
- II) Si $u \notin Adj(S) \Rightarrow Reach(u, S) = Adj(u)$

Pseudo-periféricos Un vértice u es *pseudo-periférico* si existe un vértice v alcanzable desde u , cuya excentricidad coincide con la de u y esta es igual a la distancia. Esta definición tendrá importancia cuando busquemos vértices de gran excentricidad.

Matriz de Incidencia Dado un grafo no-dirigido $G(V, E)$, se define la *matriz de incidencia* de la siguiente forma. Sea $m = |V|$, el número de vértices y $n = |E|$ el número de aristas. La matriz de incidencia $M(G)$ es una matriz de m filas y n columnas con entradas $\{0, 1, -1\}$ de forma que para cada columna j , $m_{ij} = 1$ si la arista $e_j = (v_i, v_k)$, $m_{ij} = -1$ si $e_j = (v_k, v_i)$ y cero en el resto. Por ejemplo: para el grafo completo de cuatro vértices, se tiene $|V| = 4$ y $|E| = 6$, por tanto:

$$M(G) = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 0 & -1 & -1 \end{pmatrix}$$

Matriz de adyacencia Se define la *matriz de adyacencia aumentada* como el producto matricial $M(G) \cdot M(G)^T = Aug(G)$. La *matriz de adyacencia* A , es esta última, reemplazando los elementos de la diagonal por ceros. Para el ejemplo anterior, las matrices de adyacencia aumentada y de adyacencia son respectivamente:

$$Aug(G) = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{pmatrix}$$

$$A(G) = \begin{pmatrix} 0 & -1 & -1 & -1 \\ -1 & 0 & -1 & -1 \\ -1 & -1 & 0 & -1 \\ -1 & -1 & -1 & 0 \end{pmatrix}$$

Es interesante notar la relación entre la matriz de adyacencia (aumentada o no) y la matriz simétrica asociada al grafo no dirigido.

Estructura en niveles de adyacencia En diversas operaciones se utiliza una estructura que se denominará *estructura en niveles de adyacencia*. Esta estructura se construye comenzando en un nodo \mathbf{v} que constituye el primer nivel, el siguiente nivel contiene a todos los vecinos de \mathbf{v} , y en los sucesivos, cada nivel \mathbf{l}_{i+1} constará del conjunto de vecinos de nodos en nivel

\mathbf{l}_i que no están ya incluidos en el propio \mathbf{l}_i , ni han sido ya anotados en \mathbf{l}_{i-1} .

4.1.2. Factorización de Gauss

En este apartado se pretende mostrar la relación entre la factorización de Gauss y algunos conceptos de la teoría de grafos que se han expuesto en apartados anteriores.

No se explica la factorización de Gauss, por ser un procedimiento bien conocido en la solución de sistemas de ecuaciones. Para aclarar la notación baste decir que en el paso k – *esimo* de la eliminación se eliminan todos los elementos no nulos de la columna k por debajo de la diagonal. Esto es:

$$A^{(k)} = M_k A^{(k-1)} \text{ con } a_{kk}^{(k-1)} \neq 0$$

donde M_k es la llamada transformación gaussiana dada por :

$$\mathbf{M}_k := \mathbb{I}_n - \mathbf{m}_k \mathbf{e}_k^T \text{ con}$$

$$\mathbf{m}_k = \left(\underbrace{0, 0, \dots, 0}_k, -\frac{a_{k+1,k}^{(k-1)}}{a_{kk}^{(k-1)}}, -\frac{a_{k+2,k}^{(k-1)}}{a_{kk}^{(k-1)}}, \dots, -\frac{a_{n,k}^{(k-1)}}{a_{kk}^{(k-1)}} \right)^T$$

siendo $A^{(0)} = A$. Al cabo de n eliminaciones se tiene:

$$\mathbf{A}^{(n)} = \mathbf{M}_n \mathbf{M}_{n-1} \dots \mathbf{M}_1 \mathbf{A} = \mathbf{L} \mathbf{A} = \mathbf{U}$$

con \mathbf{L} y \mathbf{U} matrices triangulares (inferior y superior respectivamente).

El coste en operaciones elementales con matrices genéricas es del orden de $\mathcal{O}(n^3)$, tal como se aprecia en la tabla 4.1.

En la figura 4.3 se va a mostrar de forma gráfica lo que le ocurre a un grafo cuando se aplica este proceso de factorización en el estado k . Se trata de *una matriz simétrica* lo que implica que el grafo es no dirigido. La primera matriz que se muestra es la matriz activa en la factorización en un determinado punto del proceso, que se ha llamado A^k . El primer grafo se corresponde con esa submatriz, esto

Matriz llena	no simétrica	simétrica
Multiplicaciones	$\frac{1}{3}n^3 - \frac{1}{3}n$	$\frac{1}{6}n^3 + \frac{1}{2}n^2 - \frac{2}{3}n$
Sumas	$\frac{1}{3}n^3 - \frac{1}{2}n^2 + \frac{1}{6}n$	$\frac{1}{6}n^3 - \frac{1}{6}n$
Almacenamiento	n^2	$\frac{1}{2}n^2 + \frac{1}{2}n$

Tabla 4.1: Coste de la factorización: matrices llenas

es, el grafo asociado $G^{(k)}$. El segundo grafo de esta figura muestra la submatriz activa una vez hecha la eliminación, es decir $G^{(k+1)}$.

Viendo la ilustración se observa que el grafo asociado a $G^{(k+1)}$ es el fruto de eliminar del grafo $G^{(k)}$ el nodo k y de conectar de todas las formas posibles los nodos adyacentes al nodo eliminado: es decir, *completar el subgrafo de adyacencia*. De aquí se desprende una primera conclusión:

- En cada paso de la eliminación se pueden producir nuevas aristas, fruto del propio proceso de eliminación, se llaman *aristas adicionales*. Estas se manifiestan en la matriz factor como nuevos elementos distintos de cero.
- Estas aristas adicionales, o nuevos elementos no nulos extra-diagonales, son el conocido relleno de la matriz factor o *fill-in*.
- Un menor grado en el vértice que se ha de eliminar en el estado k , produce menos *aristas adicionales* y por lo tanto menos elementos distintos de cero en la matriz factorizada.

Poniendo la atención en el número de operaciones requeridas para la factorización, se observa la dependencia de esta con el *grado* de cada vértice. Los valores de *relleno* de la matriz están incluidos como sumas en la tabla 4.2. r_i coincide con el grado de cada vértice después de la factorización. En el caso no simétrico r_i es el número de elementos distintos de cero fuera de la diagonal en la matriz U en cada fila y c_i lo mismo en L por columnas.

En la figura 4.4 se muestran los primeros tres pasos en el proceso de factorización en una matriz simétrica.

Matriz sparse	simétrica	no simétrica
Multiplicaciones	$\sum r_i(r_i + 3)/2$	$\sum (r_i + 1)c_i$
Sumas	$\sum r_i(r_i + 1)/2$	$\sum r_i c_i$
Almacenamiento	$n + \sum r_i$	$n + \sum (r_i + c_i)$

Tabla 4.2: Coste de la factorización: matrices huecas

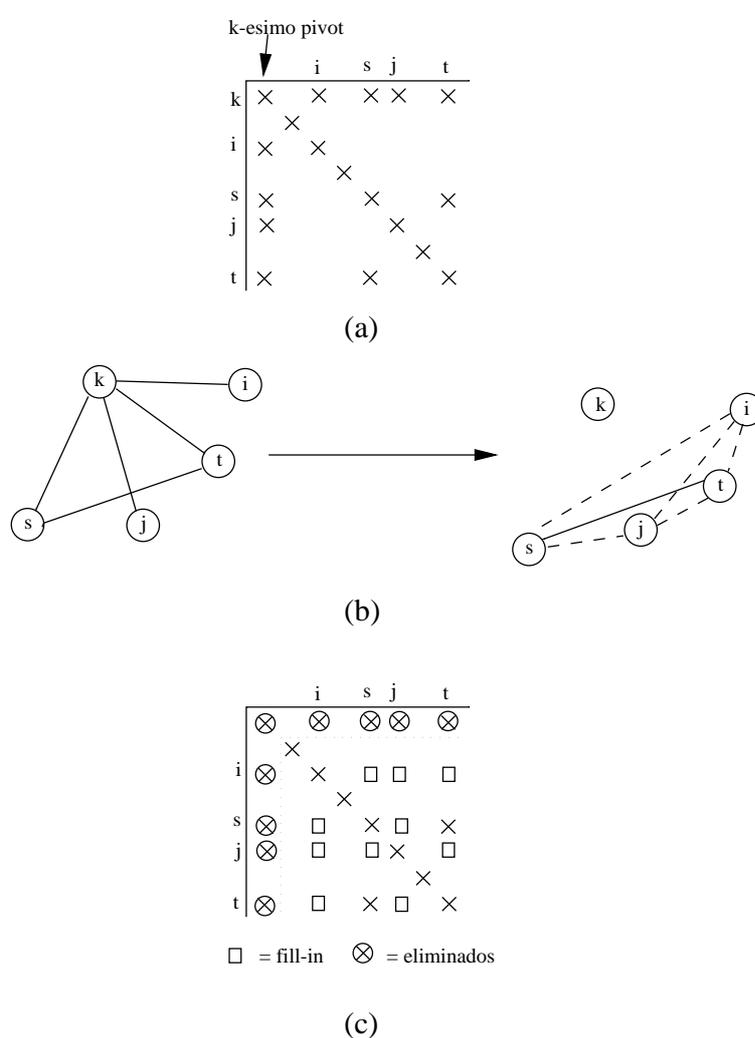


Figura 4.3: Eliminación de Gauss: Ejemplo con una matriz simétrica

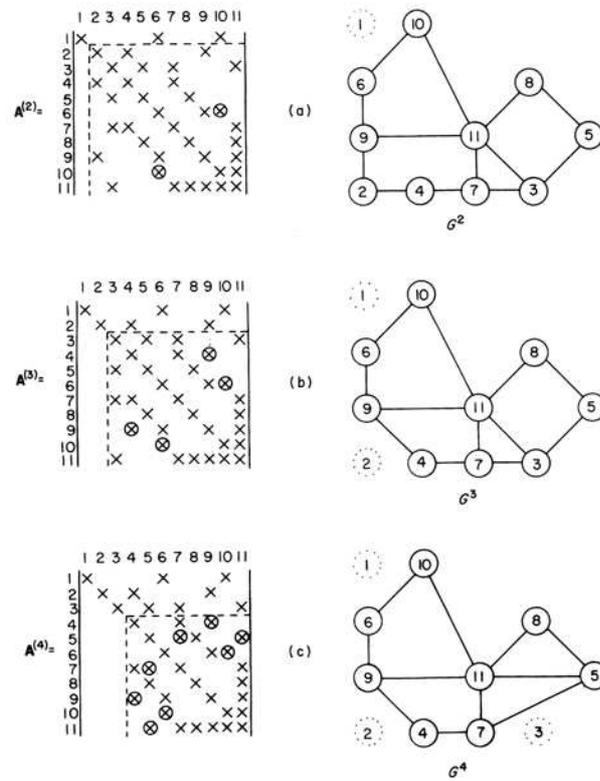


Figura 4.4: Proceso de eliminación de Gauss: matriz simétrica

4.2. Reordenamientos de matrices

La teoría de grafos tiene un campo importante de aplicación en el reordenamiento de matrices huecas. El reordenamiento en sí surge como una técnica muy recurrida en el álgebra de matrices huecas. Son varios los motivos que pueden implicar la necesidad, o cuando menos la idoneidad, de introducir una etapa previa de reordenamiento de una matriz dispersa antes de la realización de ciertas operaciones sobre ella, bien sean algebraicas o de otro tipo.

El motivo que se expuso en el apartado 3.6.5, y más importante, es que el número de operaciones que implique el procedimiento posterior se vea disminuido por el efecto del reordenamiento. Intuitivamente, desde el punto de vista computacional, esto implicará una reducción importante en el tiempo de cálculo.

Este apartado ha sido enfocado hacia el problema de resolver un sistema de ecuaciones con una matriz dispersa asociada: el problema de la factorización y la eliminación. El problema de la factorización es un problema con unas características especiales. Para la resolución del sistema, bien sea realizada en su modalidad directa o en su modalidad iterativa, un algoritmo de factorización modifica la estructura de la matriz problema conforme avanza. En una matriz dispersa, como se mostró en el apartado anterior, esto implica la introducción de nuevas entradas en posiciones que antes estaban ocupadas por ceros y que por tanto no implicaban coste computacional.

Como se vio, será muy provechoso el estudio de algoritmos que, mediante un reordenamiento previo de las filas y columnas de la matriz problema, lleven a fijar estas entradas nuevas en posiciones o zonas concretas de las matrices durante el proceso de factorización o eliminación, o que, ya en sentido más general, disminuyan el número de las mismas. Este último objetivo puede tener una repercusión muy importante sobre el coste computacional del siguiente paso de solución del sistema de ecuaciones disperso, ya que los algoritmos dispersos se diseñan para que su coste guarde una relación lo más directa posible con el número de entradas no nulas, y no con las dimensiones de las matrices. Esto se puede lograr si se realiza una primera factorización simbólica tras un algoritmo de reordenamiento para reducir el *llenado*. Así se fijan las posiciones de las nuevas entra-

das, y se preparan las estructuras de datos adecuadas para recoger los valores resultado cuando se realice la factorización numérica.

Las ventajas de reducir el *llenado* van más allá del coste del número de operaciones y del almacenamiento, éste último cada vez menos importante.

- Operar pocas veces con un elemento de una matriz es una forma de reducir la acumulación de errores en la computación. La existencia de mucho *llenado* en una matriz durante la resolución de un sistema, por ejemplo, puede implicar la necesidad de un método iterativo para preservar la estabilidad numérica.
- En el campo de la computación paralela la reducción de estas entradas toma un significado aún más interesante. Por una parte no sólo se disminuye el número de operaciones numéricas de la factorización, sino también el de las asociadas a las comunicaciones entre procesadores. Pero todavía puede ir más allá en el interés de estos algoritmos. En sistemas multicomputadores con distribución estática de datos, y con un modelo de programación basado en el pase de mensajes, las matrices usualmente son preprocesadas y reordenadas para distribuir de forma adecuada el trabajo entre los procesadores, y minimizar las necesidades posteriores de mensajes en la red de interconexión. Algunos reordenamientos constituyen métodos eficaces para que los algoritmos paralelos sean optimizados en lo que se refiere a distribución de las operaciones, y el consiguiente balanceo de la carga computacional en cada uno de los procesadores.

La cantidad de planteamientos del procedimiento de reordenamiento es muy extensa, afianzada sobre todo por la diversidad de parámetros que se optimizan en cada uno de ellos, y por la aplicación muy específica que se ha dado a determinadas técnicas. Por ello, se ha intentado reflejar en la selección de algoritmos siguiente el carácter diferencial de la estrategia que ese algoritmo implementa dentro del amplio número de alternativas posibles.

El proceso de reordenamiento se ha planteado en todo momento como un paso previo al algoritmo que se va a aplicar a la matriz para

resolver el sistema. En general, el tiempo empleado en este reordenamiento será pequeño. Aún así, como se ha comentado anteriormente, en algunos casos, un paso de reordenamiento con una duración algo elevada puede interesar cuando lo que se consigue es una mejora en la eficiencia en el problema que se pretende resolver.

4.2.1. Disyuntiva entre *estrategias locales* frente a *estrategias globales*

En los siguientes apartados se presentarán varios métodos de reordenamiento para matrices simétricas y no simétricas. Ningún método puede ser considerado el mejor para todos los problemas, e incluso en cada método, no puede decirse siempre que el algoritmo alcance la mejor solución para optimizar el parámetro en el que focaliza su esfuerzo. Por poner un ejemplo, se puede considerar el parámetro del llenado como uno de los objetivos a resolver. El problema de encontrar la solución a este objetivo, incluso en el caso de que los *pivots* sean escogidos sólo de la diagonal, se ha demostrado que es NP completo (Rose et al. [64] para el caso no simétrico y Yannakakis [85] para el caso simétrico). Por ello, los métodos de ordenamiento para obtener mínimo llenado pueden no ser rentables desde el punto de vista computacional.

Además la definición de *mejor* es a menudo relativa por otros motivos. Además del coste que implica, reducir el *llenado* puede implicar inestabilidad numérica, o disminuir la explotación eficaz del problema en computadores de arquitecturas vectoriales o paralelas. Por todo ello, la mayor parte de los algoritmos utilizan una heurística que permite alcanzar una *buena* solución en sentido amplio, antes que optimizar parámetros muy concretos.

Para una mayor estructuración del contenido se han dividido los algoritmos en dos estrategias: en *las estrategias locales*, un determinado parámetro se va optimizando en cada paso del algoritmo; en *las estrategias globales* se busca la obtención de una forma determinada de la matriz. A las primeras pertenecen para las matrices simétricas el método del *minimum degree* mientras que a las segundas el algoritmo *Cuthill-McKee* o los algoritmos *nested dissection*, también para matrices simétricas. Para las matrices no simétricas interesa presen-

tar con detalle los algoritmos que buscan una forma triangular por bloques, como se ha visto, con o sin borde.

4.2.2. Algoritmos *minimum degree*

El algoritmo, introducido por Tinney y Walker [78] basándose en los trabajos de Markowitz [54] se basa en una idea simple. Se pretende minimizar localmente el *llenado* introducido simulando una eliminación en la que, en cada paso, se elige como pivot aquella fila y columna que implican un menor número nuevos elementos no nulos en la matriz que va resultando de la factorización, operación que constituye el núcleo del algoritmo.

En la figura 4.3 se analizaba la visualización sobre el grafo de adyacencias de un paso de eliminación gaussiana, cuanto esta se realizaba fila a fila sobre una matriz. El algoritmo *minimum degree*, en su versión más directa, realiza exactamente esta operación escogiendo en cada paso de la eliminación un nodo de mínimo grado del subgrafo que va quedando tras las eliminaciones.

Esta sencillez, sin embargo, no se corresponde con un bajo coste computacional. Las operaciones de mantenimiento sobre el grafo (que representa la matriz) de la información de cómo va evolucionando la factorización simbólica son muy costosas, sobre todo cuando la vecindad entre los nodos es elevada.

En la figura 4.5 se muestra el procedimiento para los primeros pasos de una matriz de dimensión $n = 11$. Los símbolos X indican las entradas originales de la matriz. Las entradas nuevas, junto con las actualizaciones de valor por el procedimiento de factorización de entradas que ya eran no nulas son las señaladas con cuadrados en negro. En este caso, la disposición actual de las filas y columnas se adecua al reordenamiento que la heurística *minimum degree* en su versión más simplificada sugiere.

El reordenamiento *minimum degree* es muy adecuado para problemas simétricos sin una configuración definida, los cuales dan lugar a grafos de carácter irregular, a diferencia de otros que trabajan mejor con grafos de formas bien definidas, como se verá posteriormente. Sin embargo, se convierte en muy poco eficiente, en cuanto al esfuer-

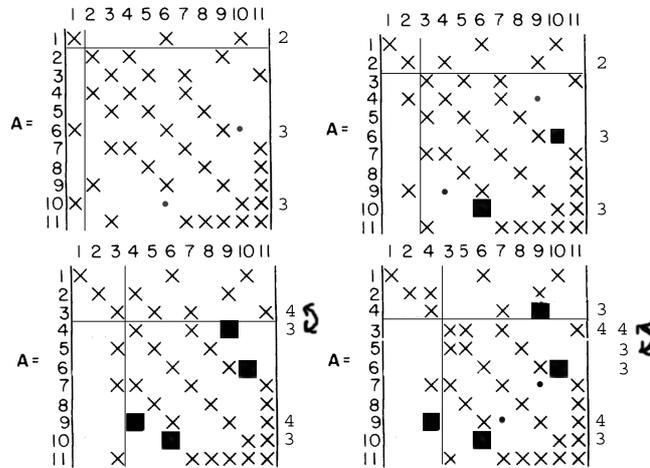


Figura 4.5: Algoritmo Minimum degree

zo computacional que exige, cuando las dimensiones del grafo son muy elevadas, no siéndolo el grado de dispersión de la matriz. Esto es debido al alto coste que implica la actualización de vecinos cuando un nodo se elimina del grafo.

Otras estrategias locales han sido estudiadas en detalle por Duff [23].

4.2.3. Algoritmos Cuthill-McKee

Es el primer algoritmo que se analizará dentro del grupo que se han llamado de *estrategias globales*. Este grupo lo constituyen una serie de procedimientos que, a diferencia del mostrado en el apartado anterior y algunos otros, consideran el grafo asociado al problema de forma global. Estos métodos no se basan tanto, como los ya comentados, en actualizaciones de vecindades locales, y en la aplicación reiterativa del propio algoritmo de reordenamiento al subgrafo resultado. En particular, los algoritmos que se muestran en las siguientes secciones imprimirán sobre los *patrones de entradas de las matrices* ciertas formas. Esta estructura especial, en cada caso, puede hacer las matrices más adecuadas para un procesamiento posterior, bien sea minimizando el *llenado* al factorizar, o adecuándose a algún algoritmo en especial como la *resolución por bloques*.

El algoritmo Cuthill-McKee fue introducido por Cuthill y McKee [15] y una modificación del mismo por George [31]. Para matrices simétricas se modifica la forma de la matriz para obtener una de banda reducida, esto es, una matriz para la que los valores distintos de cero se concentren en un entorno cercano a la diagonal.

Para una matriz simétrica el ancho de banda puede ser reducido mediante permutaciones también simétricas de sus filas y columnas. El algoritmo que aquí se plantea explora el grafo no dirigido de la matriz, encuentra una lista de etiquetas adecuada, y después permuta las filas y las columnas de la matriz de la misma forma para conservar la simetría.

Por consiguiente, el algoritmo de Cuthill-MacKee se basa en reetiquetar el grafo de la matriz intentando definir un grafo con estructura *estirada*, en la que las vecindades entre nodos sean reducidas. Este tipo de grafos, con una numeración adecuada de sus nodos, tienen asociados matrices de banda estrecha.

En la figura 4.6a se muestra un ejemplo de rejilla de un problema de elementos finitos. La matriz asociada al sistema de ecuaciones de elementos finitos $Ax = b$, una vez ensamblada, tiene la propiedad de ser una matriz definida positiva. Se muestra la estructura del grafo. Ha sido dibujado de forma que en la dirección vertical se presenta la visión más *estirada* posible del grafo, siendo equivalente a la subestructuración del grafo en niveles de adyacencia, uno por cada fila vertical, y con el nodo 1 *pseudo-periférico*, raíz de la estructura.

Si se reenumeran los nodos de arriba a abajo, como se muestra en la figura 4.6(b) de forma consecutiva se consigue que en el grafo la vecindad sea entre nodos que se numeran próximos. Esto, pensado en la matriz asociada al nuevo grafo, es equivalente a hacer que todos los nodos se aglutinen en la medida de lo posible en torno a la diagonal.

El primer paso del algoritmo consiste en la búsqueda de esta estructura de niveles óptima. La más usual se obtiene por medio de la estrategia, también utilizada en otras técnicas de disección de grafos, de encontrar un nodo pseudo-periférico a partir del cual comenzar la exploración; ir construyendo la estructura de niveles y reenumerando los nodos conforme van siendo encontrados. La elección de este nodo es de importancia transcendental en la calidad del reordenamiento obtenido. Existen estrategias planteadas en las que se elige más de un

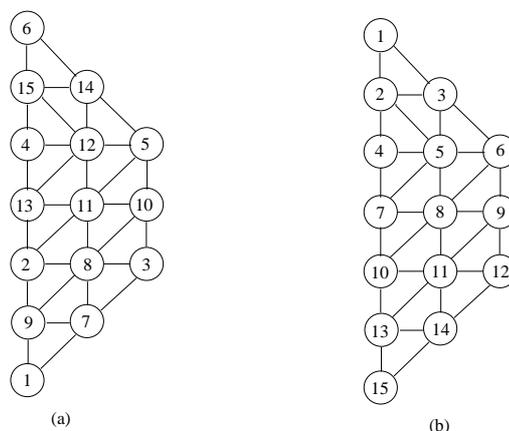
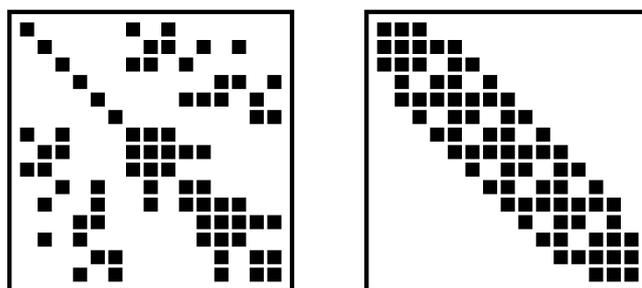


Figura 4.6: Rejilla M de un problema típico de elementos finitos con los nodos asociados. (a) Etiquetado original. (b) Re-etiquetado de acuerdo con el criterio de Cuthill-MacKee modificado (Reverse Cuthill-MacKee)

nodo para formar el primer nivel, siendo el resto del procedimiento análogo.

4.2.4. Algoritmos *Nested Dissection*

El término *nested dissection* fue introducido por George [32]. Surgió como un método apropiado para la subestructuración en los problemas de elementos finitos, incluso con mallas irregulares. En general, se puede aplicar a métodos que impliquen subdivisión de problemas de gran dimensión en problemas más pequeños, para los que los algoritmos existentes para operaciones dispersas son altamente eficaces, pero que, con problemas de tamaño muy elevado, pierden

su efectividad. En algunos casos, esta estrategia de particionamiento puede resultar adecuada cuando el problema en sí es lo suficientemente pequeño para ser tratado directamente, pero, mediante esta subdivisión, ganará eficiencia computacional.

La idea central del algoritmo es realizar sobre el grafo de adyacencias de la matriz una operación de *partición*, buscando un conjunto pequeño de nodos tales que su eliminación del grafo de adyacencias deje a este dividido en dos partes, entre las que no existen vías de conexión. Posteriormente, se reetiquetan los nodos de cada grupo consecutivamente, dejando para el final los nodos que pertenecen al separador. Una vez realizada la primera *dissección* el proceso se repite con cada uno de los subgrafos.

4.3. Reducción a una forma triangular por bloques

Los algoritmos se han descrito en las últimas tres secciones han sido desarrollados principalmente para los casos simétricos aunque en algunas circunstancias son generalizables a problemas no simétricos. En los próximos apartados se van a considerar métodos desarrollados de forma específica para matrices no simétricas. En este primer apartado se busca reducir la matriz a la siguiente forma triangular por bloques:

$$\mathbf{P} \mathbf{A} \mathbf{Q} = \begin{pmatrix} \mathbf{B}_{11} & & & & \\ \mathbf{B}_{21} & \mathbf{B}_{22} & & & \\ \mathbf{B}_{31} & \mathbf{B}_{32} & \mathbf{B}_{33} & & \\ \vdots & \vdots & \vdots & \ddots & \\ \mathbf{B}_{n1} & \mathbf{B}_{n2} & \mathbf{B}_{n3} & \dots & \mathbf{B}_{nn} \end{pmatrix} \quad (4.1)$$

Cuando existen las permutaciones que permiten este reordenamiento la matriz se llama *reducible*. Por supuesto, existen problemas cuyas matrices son irreducibles y en las que este método no será aplicable. Como se ha visto, esto permitiría resolver el sistema $\mathbf{A} \mathbf{x} = \mathbf{b}$ de forma iterativa usando los bloques de la diagonal, que serían los únicos en ser factorizados:

$$\mathbf{B}_{ii} \mathbf{y}_i = (\mathbf{Pb})_i - \sum_{j=i}^{i-1} \mathbf{B}_{ij} y_j \quad i = 1, \dots, n \quad (4.2)$$

$$\mathbf{x} = \mathbf{Qy} \quad (4.3)$$

Las técnicas que existen para lograr este reordenamiento se suelen dividir en dos etapas:

- Primero se deben permutar entradas sobre la diagonal para obtener una diagonal libre de ceros. Este proceso se denomina búsqueda de un transversal.
- En una segunda etapa se utiliza un proceso de permutaciones simétricas para encontrar la forma triangular por bloques propiamente dicha. Las permutaciones simétricas no destruirán la diagonal libre de ceros obtenida por el procedimiento anterior.

4.3.1. Búsqueda de un transversal

Se pretende completar la diagonal de la matriz mediante permutaciones hasta que esté libre de ceros. Esto es coherente con cualquier representación de la matriz en forma de grafo ya que cada entrada en la diagonal se corresponde con un *autolazo* que no se representa nunca. La carencia de entradas en la diagonal necesitaría un tratamiento con grafos especiales, para diferenciar cuándo un nodo se tiene de vecino a sí mismo y cuándo no.

Existen varias alternativas: se puede ver, por ejemplo, los trabajos de Kuhn [49], Duff [20] y Gustavson [34], que han discutido estos algoritmos en detalle. El algoritmo más utilizado pretende obtener las entradas en la diagonal mediante una permutación solo de filas y utiliza una estrategia de búsqueda en profundidad.

El algoritmo procede de la siguiente forma: se recorren todas las filas a partir de la primera, intentando que la diagonal de la fila en curso se complete, al tiempo que se van preservando las diagonales de las anteriores, ya completadas en permutaciones asociadas a pasos previos del algoritmo. Esto se ilustra en la figura 4.7.

Una fila con un cero en su diagonal se intentaría reemplazar por una inferior en la matriz con entrada en la misma columna en la que está la diagonal nula (a). De no ser posible se buscarían alternativas entre las filas anteriores ya revisadas (b), siempre y cuando, ante la permutación de una fila superior que pueda implicar dejar una diagonal nula, esta diagonal pueda ser preservada mediante una tercera permutación (c), y así sucesivamente. Es en esta forma de proceder en la que se pone de manifiesto la búsqueda primero en profundidad del algoritmo.

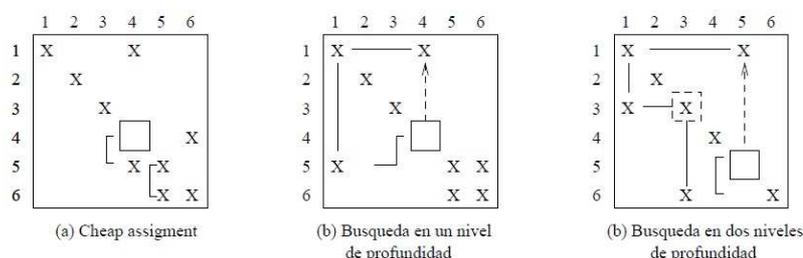


Figura 4.7: Ejemplo de búsqueda de transversal

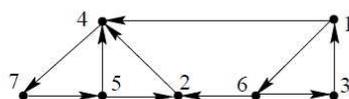
4.3.2. Permutaciones simétricas

Como se mencionó se busca una permutación simétrica que logre la forma deseada. Mediante el primer paso se ha garantizado que se puede hacer un tratamiento correcto usando el grafo asociado y como la permutación es simétrica sólo se necesita hacer una reenumeración de los nodos. En la figura 4.8 se puede ver una matriz y su grafo asociado antes de aplicar esta parte del reordenamiento.

El algoritmo más utilizado fue desarrollado por Tarjan [77] basándose en las ideas de Sargent y Westerberg. Puede verse un estudio en profundidad en Duff y Reid [22]. Sin entrar en muchos detalles la idea básica es encontrar las componentes del grafo que se denominan *fuertemente conectadas* o *strong*. Para cada una de ellas existe un camino cerrado que pasa por todos los nodos de esa componente y no hay ninguno que conecte a esos y a otros nodos del grafo. Cada una de esas componentes *fuertemente conectadas* representa un bloque en la matriz, una vez que se reenumeran los nodos.

	1	2	3	4	5	6	7
1	X			X		X	
2		X		X			
3	X		X				
4				X			X
5		X		X	X		
6		X	X			X	
7					X		X

(a)



(b)

Figura 4.8: Matriz y su grafo asociado

En la figura 4.9 se puede ver el ejemplo mostrado en la figura 4.8 una vez encontradas estas componentes:

4.4. Reducción a una forma triangular por bloques bordeada

El algoritmo que se describe a continuación ha sido utilizado fundamentalmente para problemas no simétricos irreducibles, es decir, aquellos en los que la matriz que se pretende reordenar no se puede reducir a una forma triangular por bloques. En ese caso se puede generar un borde en las últimas filas de la matriz, como se aprecia en la figura 3.4 de la sección 3.6.5, y que se reproduce en la figura 4.10.

El reordenamiento que consigue este objetivo fue introducido por Hellerman y Rarick [36] y llamado P^3 por sus siglas: *Preassigned Pivot Procedure*. Al año siguiente en Hellerman y Rarick [37] los mismos autores añadieron a su reordenamiento un paso inicial que permuta la matriz a una forma triangular por bloques y a continuación aplica

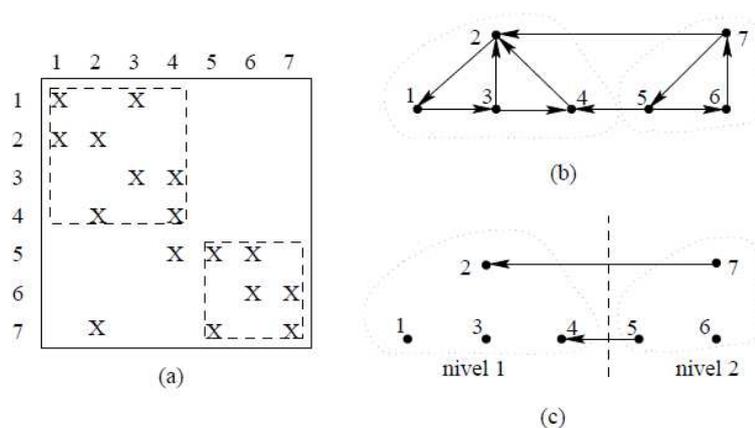


Figura 4.9: Matriz general y grafo asociado después de haber reconocido las componentes *strong* existentes y realizar el reordenamiento

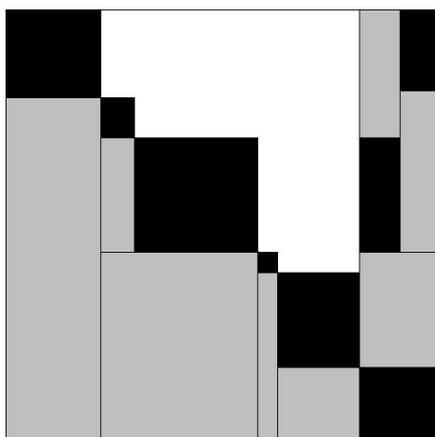


Figura 4.10: Reordenamiento P^4

el reordenamiento P^3 a cada uno de esos bloques (que son necesariamente irreducibles). A este procedimiento se le llamó P^4 (*Precautionary Preassigned Pivot Procedure*). Posteriormente en Erisman et al. [25] se propuso una variación del algoritmo P^3 que se llamó P^5 también por sus siglas: *Precautionary Partitioned Preassigned Pivot Procedure*.

En cualquiera de las variantes, las columnas con elementos por encima de la diagonal de la matriz se les suele denominar *spikes*. La

diferencia fundamental entre los reordenamientos P^3 y su variante P^5 es que en el primero todas estas columnas se encuentran en el borde. En propiedad, la forma triangular por bloques bordeada corresponde solo al reordenamiento P^5 .

El algoritmo explora una submatriz (que en el primer paso es la toda la matriz) de la que se van extrayendo filas y columnas al extremo izquierdo y columnas al borde derecho, y algunas filas se permutan al borde inferior. En cada iteración la matriz que va quedando se denomina matriz activa. En la figura 4.11 se ve la estructura de submatrices de un paso intermedio del proceso.

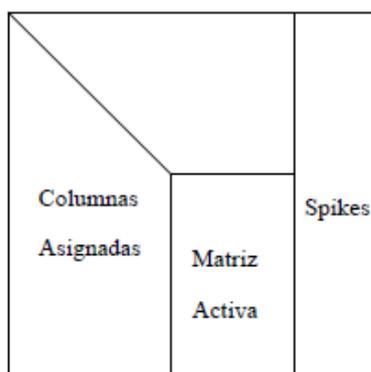


Figura 4.11: Paso intermedio de P^5

El algoritmo busca una permutación en filas y columnas, no necesariamente simétrica, tal que se obtenga un conjunto de bloques diagonales lo más pequeños posibles (si es posible de orden uno) y un borde muy estrecho.

Con la intención de formar bloques siempre pequeños, en primer lugar se buscan aquellas filas que tienen un menor número de entradas, m , constituyendo un conjunto de f filas. Se escoge además un conjunto de m columnas de las que tengan el máximo número de entradas en las f filas anteriores. Las columnas se van escogiendo por orden de mayor número de entradas, permutándolas al frente de la matriz activa y eliminándolas de ésta. Cada vez que se elimina una columna se actualiza el conjunto de filas que tienen un menor número de entradas.

Cuando la eliminación de columnas implique que algunas de las filas del conjunto anterior se quedan sin ninguna entrada las s filas a las que le sucede esto son también permutadas al principio de la matriz activa. De esta manera se promociona el traslado de bloques pequeños y densos a la diagonal. Las $m - s$ columnas restantes se trasladan al final.

Hay que notar que s debe ser menor que m para no contradecir el supuesto de que la matriz es irreducible, ya que en otro caso se estaría determinando un bloque diagonal separable del resto de la matriz. Hecho esto la matriz activa se ha reducido en s filas superiores m columnas por la izquierda y $m - s$ columnas a la derecha. El proceso se repite sobre la nueva matriz activa hasta su extinción.

4.5. Aplicación al problema de mecanismos

Debido al sistema de coordenadas utilizado, el vector \mathbf{q} de coordenadas mixtas estará formado por las *tres* componentes de cada punto o de cada vector unitario empleados y por las coordenadas relativas que se hayan elegido en determinados pares cinemáticos.

Como se ha indicado en la sección 3.6.6 en un mecanismo de cadena abierta la matriz jacobiana siempre se podrá reducir a una forma diagonal por bloques. Las columnas de la matriz jacobiana están directamente relacionadas con las coordenadas, y en todos los sistemas que se pretenden resolver aparece la parte que se ha llamado *dependiente* de esta matriz jacobiana $\Phi_{\mathbf{q}}^d$. Esto influye en la naturaleza de los bloques de la diagonal: casi todos deberán estar formados por matrices llenas 3×3 . Solo aquellas correspondientes a coordenadas relativas deberían tener dimensiones menores dependiendo de los grados de libertad del par cinemático. En los mecanismos de cadena cerrada en la parte correspondiente a la submatriz triangular por bloques ocurre lo mismo, salvo algunos términos que aparecen en determinadas restricciones (cuyas derivadas, respecto a algunas coordenadas, siempre son constantes y distintas de cero).

Si se aplican los reordenamientos convencionales descritos en el

apartado anterior surgen uno o varios de los inconvenientes que se relacionan a continuación:

- Interesa poder aplicar un método lo más general posible tanto para los mecanismos de cadena abierta como cerrada. En los reordenamientos existentes no existe ningún algoritmo ideado para matrices irreducibles que se pueda aplicar también a problemas reducibles. En el caso visto del reordenamiento P^4 se opera en dos fases buscando primero la estructura por bloques y después aplicando necesariamente P^3 a cada bloque de la diagonal. Este último paso no sería necesario en un mecanismo de cadena abierta.
- Otro inconveniente de los reordenamientos expuestos es que no siempre se respetan las dimensiones de los bloques de la diagonal, llegando a extremos donde casi todos los bloques están formados por matrices de un único elemento. Existen ecuaciones de restricción en las que esto no plantea ningún problema, pero en estos reordenamientos no se plantea esa diferencia. Al no respetarse estas dimensiones puede ocurrir que en determinadas posiciones del mecanismo (sin ser singulares) la resolución por bloques plantee un problema.
- Por último en un reordenamiento óptimo de cadena cerrada las dimensiones del borde de la matriz está relacionada con la geometría del sistema, como se vio. En los algoritmos existentes al intentar reducir al mínimo los bloques de la diagonal aparecen bordes que suelen ser mayores que la dimensión óptima.

Para tratar de subsanar en gran medida estos problemas, se ha diseñado un algoritmo de reordenamiento propio que se detalla en el siguiente apartado.

4.6. Reordenamiento propuesto

El reordenamiento propuesto es una variante del P^5 particularizado para el caso en que los bloques de la diagonal son de dimensión una, dos o tres.

Las características fundamentales del reordenamiento propuesto son las siguientes:

- No se diferencia entre mecanismos de cadena abierta y cerrada. El algoritmo de reordenamiento es siempre el mismo: en los casos de cadena abierta no se genera ningún borde. Si el mecanismo es de cadena cerrada, una vez aplicado el reordenamiento, se genera automáticamente el borde. Esto representa una gran ventaja tanto en el diseño de mecanismos como en su resolución.
- Se va a particularizar en el propio reordenamiento los casos para bloques de dimensión una, dos o tres. No se van a considerar más posibilidades. Esto reduce la generalidad del método, pero intenta paliar en cierta medida la creación de bloques que puedan ser singulares.
- Como se verá en los resultado del capítulo siguiente se suelen obtener bordes más reducidos que en los reordenamientos existentes.

Al igual que los reordenamientos en los que se basa este algoritmo, en cada iteración se explora una submatriz que se llamará, en lo que sigue, *matriz activa*. Al comienzo, lógicamente, la matriz activa es la toda la matriz. De esta matriz se irán extrayendo filas y columnas al extremo izquierdo (al frente de la matriz) y columnas al borde derecho. Algunas filas se permutan al borde inferior. Cada vez que se mueve una fila o columna la matriz activa se actualiza.

La estructura del código es la siguiente:

- En el primer paso se seleccionan de la matriz activa las filas que contengan el menor número de entradas: a ese valor se le llamará m . Al mismo tiempo se seleccionan todas las columnas con entradas en las filas anteriores. Estos dos conjuntos serán en el proceso los conjuntos de filas y columnas activas, respectivamente. (No confundir con la matriz activa).
- En el caso de que $m = 0$ las filas activas seleccionadas se envían al borde inferior de la matriz y se vuelve al paso anterior.

- Como se ha indicado la máxima dimensión para los bloques de la diagonal será 3, debido a la naturaleza de la matriz. Como además m representa el máximo número de columnas que se pueden mover al frente, solo se considerarán los casos con $m \leq 3$.
- Con $m = 1$: se selecciona la columna con un mayor número de entradas en la matriz activa y se envía al frente de la matriz. A continuación se selecciona una de las filas con entradas en esa columna (que llamaremos en adelante la fila correspondiente) y también se envía al frente de la matriz. Como siempre que se muevan filas o columnas, se actualiza la matriz activa eliminando la fila y la columna seleccionadas.
- Con $m = 2$:
 - Se comienza modificando el número de filas y columnas activas en un proceso de tres pasos: se eliminan todas las columnas con un número de entradas mayor que m ; una vez eliminadas esas columnas del conjunto de columnas activas, las filas con un número de entradas menor que m . Por último, una vez eliminadas esas filas, se seleccionan las columnas con un recuento igual o mayor que m en las nuevas filas activas.
 - Si no ha quedado ninguna columna activa se vuelve al primer paso seleccionando en este caso las filas con $m = 3$.
 - En caso contrario se mueven al frente un número de columnas igual al número de filas activas, y sus filas correspondientes. Las columnas que no se hayan movido formarán parte del borde.
- Con $m = 3$: en este caso se seleccionan las columnas de máximo número de entradas en las filas activas. Si ese número es menor que 3, una de las columnas y su fila correspondiente pasarán al frente y las otras columnas al borde. En caso contrario se seleccionan tres columnas con sus filas para moverlas al frente.

Capítulo 5

Resultados

5.1. Consideraciones generales

El programa general que se utiliza para definir el mecanismo está programado en el entorno Matlab con llamadas a funciones externas escritas en lenguaje C o Fortran. Se han compilado funciones MEX¹ para la comunicación entre ambos programas.

En un primer paso se define el modelo del mecanismo, incluidas las restricciones sobre los elementos y sus propiedades dinámicas (matriz de inercia, etc.) y las características de los pares cinemáticos (constantes de rigidez, etc.). Para generar de una forma sencilla el modelo matemático de los elementos y sus características inerciales, los pares cinemáticos, etc; en definitiva, el modelo del mecanismo, se han empleado unos programas parcialmente desarrollados en el grupo de investigación del Prof. García de Jalón en la UPM.

La evaluación del Jacobiano tiene dos partes. En primer lugar, una función calcula la posición de los elementos no nulos de la matriz, según las expresiones algebraicas de las ecuaciones de restricción. Este cálculo se hace una sola vez al inicio de la simulación. Se ha tenido en cuenta para esta fase que determinadas ecuaciones de restricción que afectan a coordenadas angulares pueden estar alter-

¹Estas funciones permiten comunicar el entorno de Matlab con otras funciones exteriores.

nativamente escritas en forma de seno o coseno². Esta primera fase permite obtener un Jacobiano *simbólico* que es utilizado tanto para los reordenamientos como para la evaluación propiamente dicha. Esta evaluación del Jacobiano para cada instante de tiempo que se hace a demanda del integrador constituye la segunda fase. Tanto el proceso simbólico como el numérico se llevan a cabo en una función programada en C porque de esta manera se mejora la eficiencia del programa. La matriz jacobiana se devuelve como matriz *dispersa* para su utilización posterior en el código.

También se ha implementado en C el código de la función que permite resolver los sistemas de ecuaciones *por bloques*. La función recibe la matriz almacenada de forma *sparse*, y va resolviendo, bloque a bloque, el sistema de ecuaciones con uno o más términos independiente utilizando un pivotamiento parcial.

Para los integradores, como se ha indicado, se han utilizado los integradores de librería de Matlab *ode45* y *ode113*.

Los reordenamientos que se van a utilizar son el P^5 de Erisman et al. [25] y el diseñado para este trabajo. Ambos reordenamientos se aplican antes de comenzar la integración sobre el Jacobiano *simbólico* y utilizando solo las coordenadas dependientes (Φ_q^d).

Se van a presentar los resultados correspondientes a cuatro ejemplos muy comunes en la bibliografía. Se trata de los siguientes mecanismos:

- Un péndulo de cinco barras unidas por pares de revolución,
- Un mecanismo de atadura típico en las estructuras desplegadas en las misiones espaciales que se llama genéricamente *theter*,
- El mecanismo de Bricard,
- Un modelo de completo automóvil.

²Es conocido que los errores numéricos derivados de emplear las funciones trigonométricas seno y coseno tienen comportamientos complementarios en torno a 0 , $\pi/2$, π y $3\pi/2$.

5.2. Péndulo de cinco barras

Lo que se ha llamado *péndulo de cinco barras* es un mecanismo de cadena abierta con cinco elementos móviles y cinco pares cinemáticos. La primera barra está anclada a un punto fijo mediante un par de revolución y las demás barras también están unidas entre sí con otros cuatro pares de revolución no colineales. El mecanismo posee 5 grados de libertad y se ha modelado usando coordenadas mixtas: se han considerado como coordenadas independientes cinco ángulos definidos como coordenadas relativas. Todas las barras son de masa y longitud unidad y no se han aplicado fuerzas exteriores distintas a la gravedad.

En la figura 5.1 se ve una representación del mecanismo en su posición inicial. Solo hay un punto y dos vectores fijos (un vector vertical para el eje de rotación del primer par de revolución, y otro para definir la primera coordenada relativa).

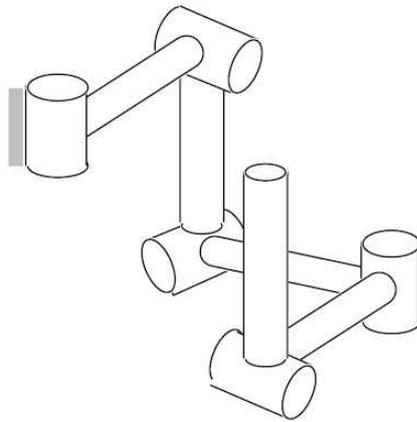


Figura 5.1: Péndulo de cinco barras

En la tabla 5.1 se sintetiza la descripción del mecanismo y su modelo matemático:

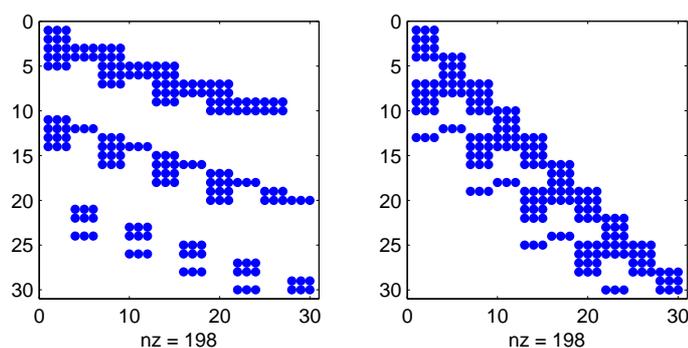
La matriz jacobiana se puede ver en la figura 5.2, con todos los posibles elementos no nulos. Como en todos los ejemplos, no se incluyen las coordenadas independientes. En esa figura aparece la matriz

Características del modelo

Tipología:	cadena abierta
Coordenadas:	mixtas (35 libres, 15 fijas)
Puntos:	6
Vectores:	7
Coordenadas relativas:	5 ángulos
Grados de libertad:	5
Restricciones:	30
Matriz característica	dimensión 30x35
Elementos no nulos:	198 (6,7%)

Tabla 5.1: Péndulo de cinco barras: características

inicial y la matriz reordenada, en este caso los reordenamientos P^5 y el propio de este trabajo son idénticos. Como se puede ver, la matriz es reducible y el caso de este último reordenamiento no ha sido necesario ninguna etapa de reducción como en el algoritmo P^4 ni se ha aplicado ningún reordenamiento (P^5) a cada bloque de la diagonal.

Figura 5.2: Péndulo. Matriz Φ_q^d inicial y reordenada

La simulación se ha hecho con el integrador *ode45* de Matlab con una tolerancia de 10^{-6} y una formulación de índice dos, con mejora en las posiciones mediante Newton-Raphson. El tiempo de la simulación son 10 segundos y el integrador ha llamado 3673 veces a la función derivada del vector de estado en 590 pasos de integración.

En la tabla 5.2 se ven los tiempos que se han obtenido en las siguientes simulaciones: en la primera simulación se ha usado la matriz llena; En la segunda la matriz reordenada con el reordenamiento propio. Se dan los datos del tiempo total que corresponde al integrador; el dato del tiempo empleado en la resolución de los sistemas de ecuaciones con la matriz jacobiana y el dato del tiempo total de simulación. El tiempo total en términos absolutos no es significativo (incluye representaciones gráficas, etc.), pero sí en términos relativos ya que todas las simulaciones contienen esas partes.

	Matriz llena	Matriz reordenada
Integración	5,82 s	2,80 s
Resolución de los sistemas	3,20 s	0,24 s
Tiempo total	6,41 s	4,06 s

Tabla 5.2: Tabla de tiempos para el péndulo de cinco barras

Como se puede observar se produce una mejora de un poco más de un 90 % en los tiempos de resolución de los sistemas de ecuaciones. Trasladado al tiempo de integración la mejora es de cerca del 50 %. Esto es debido a los procesos adicionales de partición de matrices que se realizan.

Por último, se muestran en la figura 5.3 los valores de las restricciones a lo largo de la simulación.

5.3. Mecanismo *Theter*

Este mecanismo es un medio de amarre o atadura que se puede utilizar para modelar sistemas más complejos y con aplicaciones, por ejemplo, en la ingeniería espacial. Como se observa en la figura 5.4 está formado por un conjunto de 10 sólidos rígidos unidos por pares esféricos. El primer elemento está fijo y en cada par esférico

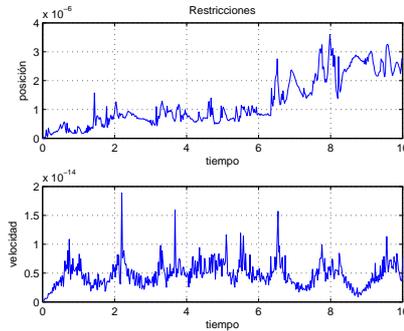


Figura 5.3: Péndulo. Restricciones

se definen tres ángulos que son las 28 coordenadas independientes: mediante esas coordenadas relativas se condiciona el movimiento, al introducir un resorte en cada par cinemático, que se modeliza mediante tres componentes. Una para el movimiento de torsión y otras dos para cada movimiento de flexión. Las constantes de elasticidad son independientes.

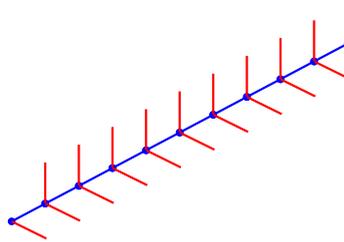


Figura 5.4: Mecanismo *Theter*

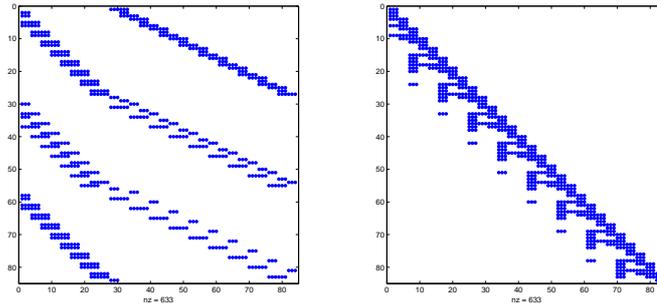
Las características del modelo matemático aparecen en la tabla 5.3:

La matriz inicial del sistema y la matriz reordenada se pueden ver en la figura 5.5. Al igual que en el caso anterior, y por las mismas razones, solo se ha utilizado el reordenamiento que se ha implementado en este trabajo. Se han formado 28 bloques triangulares de matrices de dimensión tres. Si se hace un cálculo aproximado del orden de operaciones necesarias para resolver un sistema como este,

Características del modelo	
Tipología:	cadena abierta
Coordenadas:	mixtas (112 libres, 9 fijas)
Puntos:	11
Vectores:	20
Coordenadas relativas:	28 ángulos
Grados de libertad:	28
Restricciones:	84
Matriz característica	dimensión 84x112
Elementos no nulos:	661 (7%)

Tabla 5.3: Mecanismo de *Theter*: características

mediante una factorización de gauss o por bloques, se puede ver que la diferencia entre los dos métodos es de dos órdenes de magnitud.

Figura 5.5: Mecanismo *Theter*. Matriz $\Phi_{\mathbf{q}}^d$

En este caso se ha hecho la simulación con el integrador *ode113* de Matlab con una tolerancia de 10^{-5} y una formulación de índice tres, con el cálculo de las posiciones mediante Newton-Raphson. El tiempo de la simulación es de 3 segundos. El integrador evalúa 17,620 veces a la función derivada del vector de estado en 8809 pasos de integración. Con uno números tan alto se puede esperar una gran diferencia en los tiempos de integración entre considerar la matriz característica llena o reordenada. No se considera la fuerza de la gravedad y al

último elemento se la aplica una rotación y un impulso inicial. Los resultados se presentan en de la tabla 5.4.

	Matriz llena	Matriz reordenada
Integración	124,29 s	31,33 s
Resolución de los sistemas	101,31 s	9,87 s
Tiempo total	125,24 s	31,33 s

Tabla 5.4: Tabla de tiempos para el mecanismo de *Theter*

En los tiempos que se muestran la mejora en la eficiencia en la resolución de los sistemas es similar, pero en tiempo de integración la mejora llega a un 75 %, debido a las dimensiones del sistema.

En la figura 5.6 se muestra el valor de las restricciones a lo largo de la simulación.

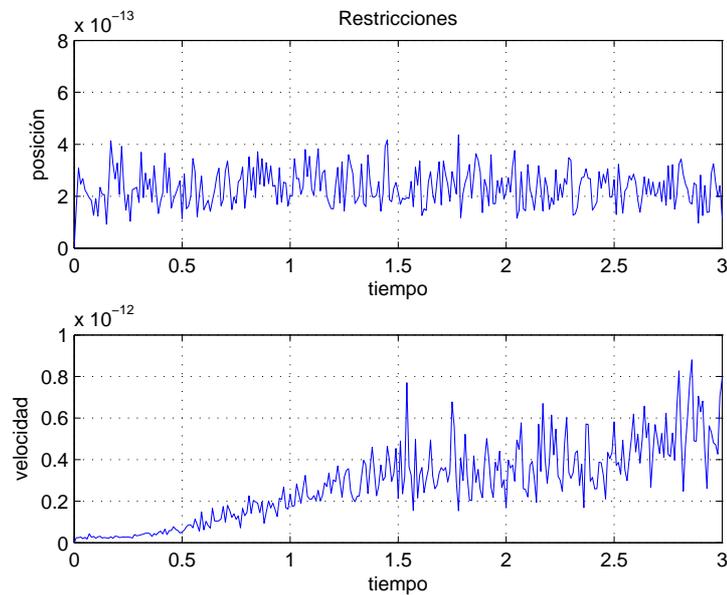


Figura 5.6: Mecanismo *Theter*. Restricciones

5.4. Mecanismo de Bricard

Este es el primer mecanismo de cadena cerrada que se ha estudiado. Se trata de un mecanismo muy similar al péndulo de cinco barras ya estudiado. En este caso el último elemento está unido a un punto fijo, mediante otro par de revolución. Se puede ver en la figura 5.7. Este mecanismo tiene restricciones redundantes (por una particular orientación de los ejes le permiten tener un grado de libertad). Es un mecanismo pequeño que permite estudiar los diferentes problemas que se afrontan.

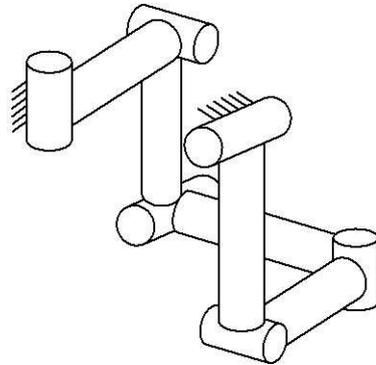


Figura 5.7: Mecanismo de Bricard

El mecanismo se ha modelado de dos formas posibles. En el primer caso se ha utilizado una modelización similar a la que se puede encontrar en métodos de tipo topológico: se han duplicado un punto y un vector, y se han añadido seis ecuaciones de restricción adicionales o ecuaciones de cierre.

Las características de este modelo se muestran en la tabla 5.5.

La matriz jacobiana se puede ver en la figura 5.8. Se puede observar que las restricciones de cierre aparecen al fondo de la matriz y las coordenadas relativas (de las que se ha eliminado una coordenada independiente) a la derecha. El hecho de que ya salga una banda natural de 4 elementos se debe a la forma de introducir las coordenadas iniciales.

Una vez reordenada la matriz mediante el reordenamiento que se ha implementado, aparecen bloques en la diagonal y una banda a la derecha. El resultado se puede observar en la figura 5.9.

Características del modelo	
Tipología:	cadena cerrada
Coordenadas:	mixtas (35 libres, 15 fijas)
Puntos:	7
Vectores:	8
Coordenadas relativas:	5 ángulos
Grados de libertad:	1
Restricciones:	36
Matriz característica	dimensión 36x35
Elementos no nulos:	209 (16 %)

Tabla 5.5: Mecanismo de Bricard: características (1)

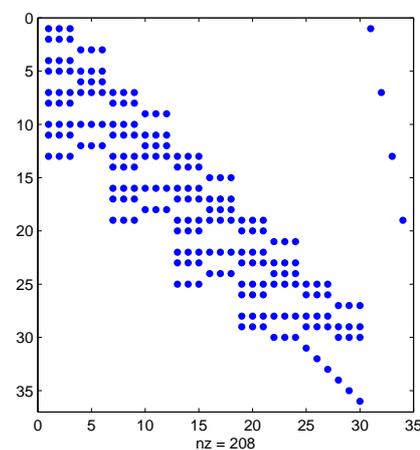


Figura 5.8: Matriz Φ_q^d inicial

Tal como se observa se ha generado un borde de 3 elementos y una banda inferior de 5. En este caso particular el reordenamiento P^5 da un resultado muy semejante e idéntico en cuanto a las dimensiones del borde y la banda. En todos los mecanismos probados las bandas generadas por P^5 son siempre mayores o iguales a la generada por este reordenamiento.

El problema con esta matriz es que los bordes de la diagonal pueden ser singulares en determinadas posiciones de la matriz. Esto no ocurre con las entradas de las primeras filas, que se corresponden con las restricciones de cierre introducidas (y que son siempre dife-

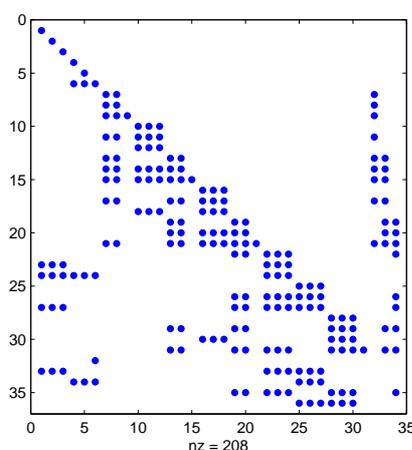


Figura 5.9: Matriz $\Phi_{\mathbf{q}}^d$ reordenada

rentes de cero), pero sí puede ocurrir con las coordenadas relativas que aparecen en la diagonal o a los bloques de dos elementos. Una forma de solucionar este problema sería modificar ligeramente la resolución por bloques que se propuso: en el proceso de factorización de cada bloque, si se comprueba que es singular, se puede ampliar ese bloque con el siguiente y continuar la factorización. Esto hace que en algunos casos, durante la resolución, haya bloques de más de tres elementos y se reduzca un poco la eficacia de la resolución.

Si en la modelización del mecanismo se indicase qué coordenadas son relativas, el hecho de tener ecuaciones de cierre en el modelo hace posible que ocurra lo mismo que en el caso anterior, tal como se observa en la figura 5.10. Ahora las posiciones singularidades (de los bloques diagonales) son mucho menores.

Una forma de evitar completamente el problema, sería indicar también qué ecuaciones de cierre son las que se han añadido al modelizar el mecanismo. No es una solución óptima ya que se perdería la capacidad de poder resolver el sistema de forma general, independientemente de las restricciones introducidas. Se puede ver lo que ocurre en la figura 5.11.

Otra posibilidad para este mecanismo sería utilizar un modelo diferente, sin ecuaciones de cierre. Se muestran las características en la tabla 5.6.

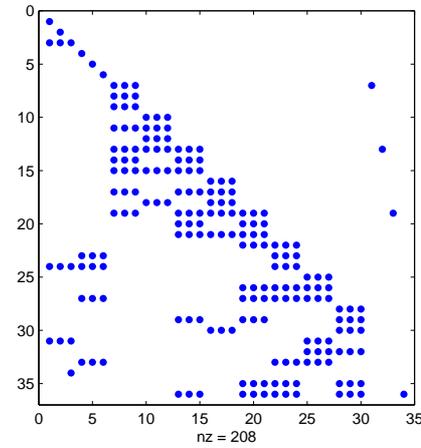


Figura 5.10: Matriz Φ_q^d reordenada fijando las coordenadas relativas

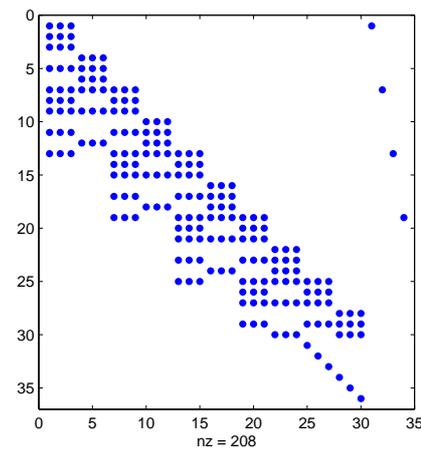


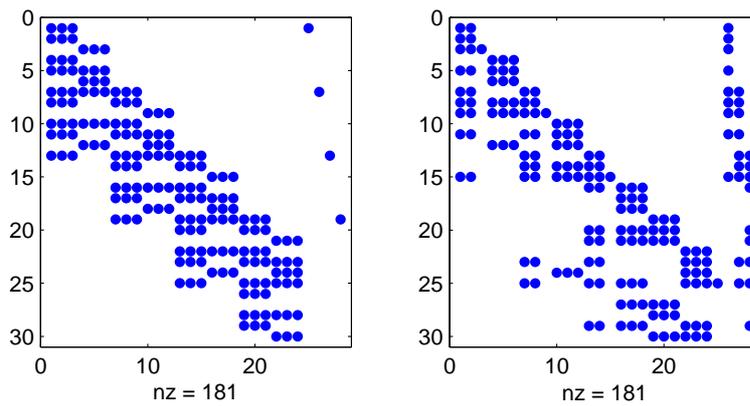
Figura 5.11: Matriz Φ_q^d reordenada fijando las coordenadas relativas e indicando las ecuaciones de cierre

En la figura 5.12 se puede ver la matriz inicial y la matriz reordenada. Como se observa se dan los mismos problemas que en el caso anterior. Las coordenadas relativas del mecanismo y determinados bloques de dos elementos aparecen en la diagonal.

Indicando en el modelo las coordenadas relativas sí se puede obtener una forma reducida a bloques en la diagonal como se observa en la figura 5.13.

Características del modelo	
Tipología:	cadena cerrada
Coordenadas:	mixtas (29 libres, 15 fijas)
Puntos:	6
Vectores:	7
Coordenadas relativas:	5 ángulos
Grados de libertad:	1
Restricciones:	30
Matriz característica	dimensión 30x29
Elementos no nulos:	181 (20 %)

Tabla 5.6: Mecanismo de Bricard: características (2)

Figura 5.12: Matriz Φ_q^d inicial y reordenada (2)

Como se ha indicado al comienzo, en el Jacobiano están representados todos los posibles elementos no nulos en cualquier posición del mecanismo. Esto implica que en un caso general no se pueda garantizar que no aparezcan bloques singulares.

Como se ha mencionado más arriba, en este ejemplo el reordena-

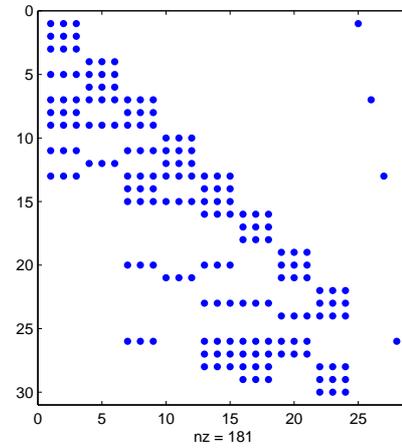


Figura 5.13: Matriz Φ_q^d reordenada indicando coordenadas relativas (2)

miento P^5 genera un borde de la misma dimensión al procedimiento propuesto. Este último no presenta, por tanto, ninguna mejora en cuanto a la eficiencia sobre el primero. En la tabla 5.7 se dan los datos obtenidos para la resolución de la matriz llena y la matriz reordenada con el método propuesto. Se ha utilizado el integrador *ode45* de Matlab con una tolerancia de 10^{-6} y una formulación de índice dos, con mejora en las posiciones mediante Newton-Raphson. El tiempo de la simulación son 15 segundos y el integrador ha llamado 865 veces a la función derivada del vector de estado en 136 pasos de integración.

En este caso, la dimensión del mecanismo, y sobre todo el número pequeño de evaluaciones que hace el integrador de la derivada del vector de estado (comparándolo, por ejemplo, con el péndulo de cinco barras) hacen que se note muy poco la mejora en la eficiencia.

	Matriz llena	Matriz reordenada
Integración	1,72 s	1,25 s
Resolución de los sistemas	0,76 s	0,56 s
Tiempo total	2,19 s	2,46 s

Tabla 5.7: Tabla de tiempos para el mecanismo de Bricard

En la figuras 5.14 y 5.15, se puede ver el cumplimiento de las

restricciones y la conservación de la energía total en el mecanismo.

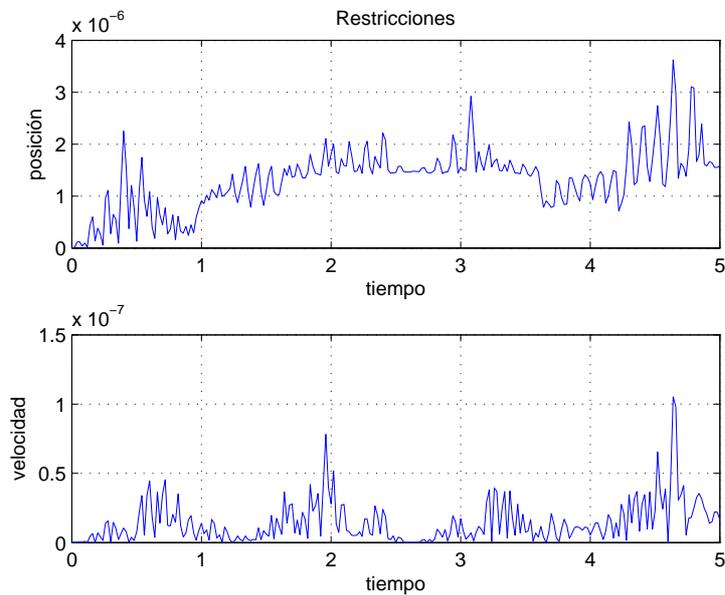


Figura 5.14: Bricard: Restricciones

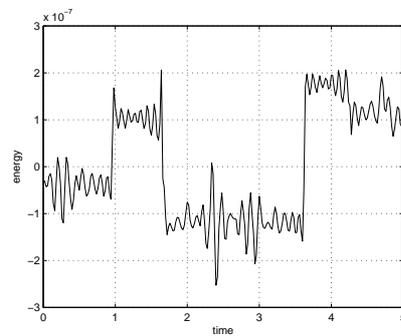


Figura 5.15: Bricard: Energía total

5.5. Modelo de vehículo completo

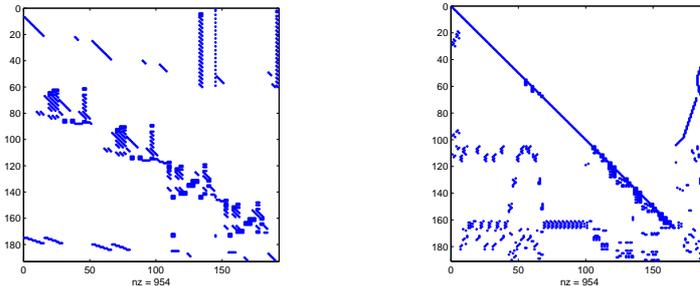
Se presenta el modelo de un vehículo de cuatro ruedas en el que se han tenido en cuenta suspensiones independientes, el mecanismo de dirección y el movimiento de las ruedas.

Características del modelo

Tipología:	cadena cerrada
Coordenadas:	mixtas (202 libres, 5 fijas)
Puntos:	54
Vectores:	12
Coordenadas relativas:	4 ángulos y 5 distancias variables
Grados de libertad:	10
Restricciones:	192
Matriz característica	dimensión 192x202
Elementos no nulos:	954 (2,4%)

Tabla 5.8: Vehículo completo: características

matriz, como se ha indicado en las características del sistema con un 2,4 % de elementos diferentes de cero (en todas estas matrices los elementos nulos reales pueden ser mayores, ya que se está considerando, como se indicó, la matriz jacobiana *simbólica*).

Figura 5.18: Automóvil: matriz $\Phi_{\mathbf{q}}^d$ y reordenamiento P^5

El coche se ha modelado, como en el primer ejemplo que se ha usado para el péndulo de Bricard, añadiendo puntos adicionales y seleccionando unas ecuaciones de *cierre de lazo*. Se ha optado por esta modelización porque permite estudiar simultáneamente un sistema de cadena abierta, eliminadas esas ecuaciones de cierre (en este caso algunas de esas restricciones ya aparecen en el propio mecanismo de cadena cerrada, debido al tratamiento que se hace de las barras biarticuladas). Además, permite comparar el reordenamiento propuesto, o el P^5 , con lo que podría ser un *reordenamiento natural*: si se eliminasen las nuevas coordenadas *independientes* de ese sistema abierto

la matriz resultante debería poder reducirse a una forma triangular por bloques. En cualquier caso, la resolución del mecanismo no debería ser independiente de la modelización.

En este caso el reordenamiento P^5 produce resultados muy poco satisfactorios. La matriz reordenada de forma automática por P^5 se puede ver en la figura 5.18b. Como se observa el borde que se produce es de 27 elementos. Hay que indicar que el sistema abierto del que se habló en el párrafo anterior tendría 18 coordenadas independientes adicionales. Al elegirse de forma automática una banda tan grande los problemas derivados de la existencia de bloques singulares crece sustancialmente.

Si se estudia el sistema de cadena abierta (eliminadas las ecuaciones de restricción de cierre y las nuevas coordenadas independientes), el reordenamiento P^5 fracasa al intentar reducirlo a un sistema por bloques en la diagonal. En la figura 5.19 se ve la matriz del sistema abierto y reordenamiento mediante P^5 . Como se observa aparece una nueva banda de 5 elementos.

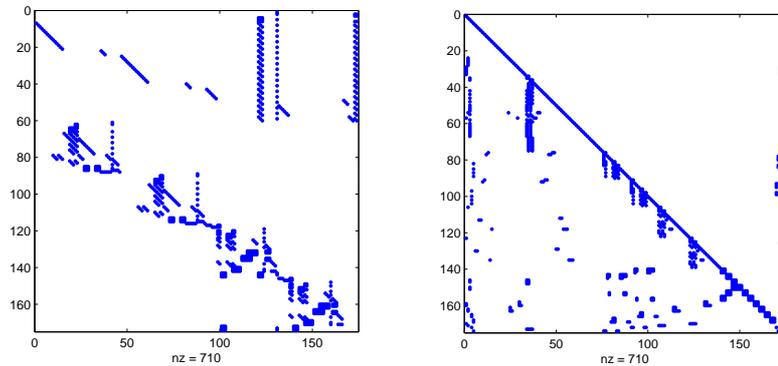


Figura 5.19: Automóvil: sistema abierto y reordenamiento P^5

Con el reordenamiento propio se consiguen los dos objetivos propuestos: la banda se reduce (en este caso aparecen 17 elementos) y el reordenamiento del sistema de cadena abierta se reduce totalmente a una forma triangular por bloques. Los dos objetivos se pueden ver en la figura 5.20

Hay que señalar que persiste el problema, en ambos casos, de que se den posiciones singulares en la diagonal.

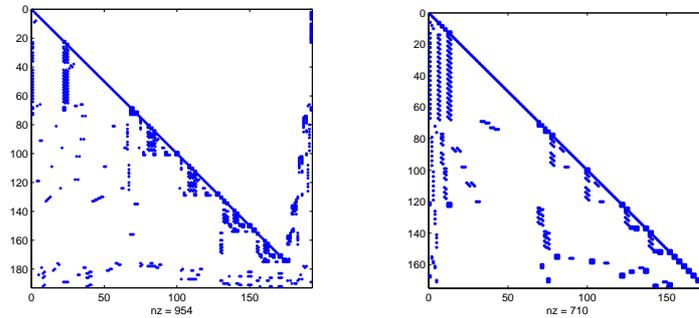


Figura 5.20: Automóvil: reordenamiento propio

Finalmente se presentan los datos numéricos de la simulación en la tabla 5.9, para la resolución de la matriz llena y la matriz reordenada con el método propuesto. Con el reordenamiento P^5 fracasa la integración. Se ha simulado el problema de encontrar la posición de equilibrio dinámico, modificando —en la posición inicial— la altura del terreno.

	Matriz llena	Matriz reordenada
Integración	33,25 s	12,09 s
Resolución de los sistemas	26,47 s	4,01 s
Tiempo total	34,30 s	13,06 s

Tabla 5.9: Tabla de tiempos para el vehículo completo

Se ha utilizado el integrador *ode113* de Matlab con una tolerancia de 10^{-5} y una formulación de índice tres, con cálculo de las posiciones dependientes mediante Newton-Raphson. El tiempo de la simulación es de 1 segundo y el integrador ha llamado 2509 veces a la función derivada del vector de estado en 1182 pasos de integración. Los tiempos son similares a los mostrados en los casos anteriores.

Se presentan también las restricciones a lo largo de la simulación y la energía total del sistema en las figuras 5.21 y 5.22

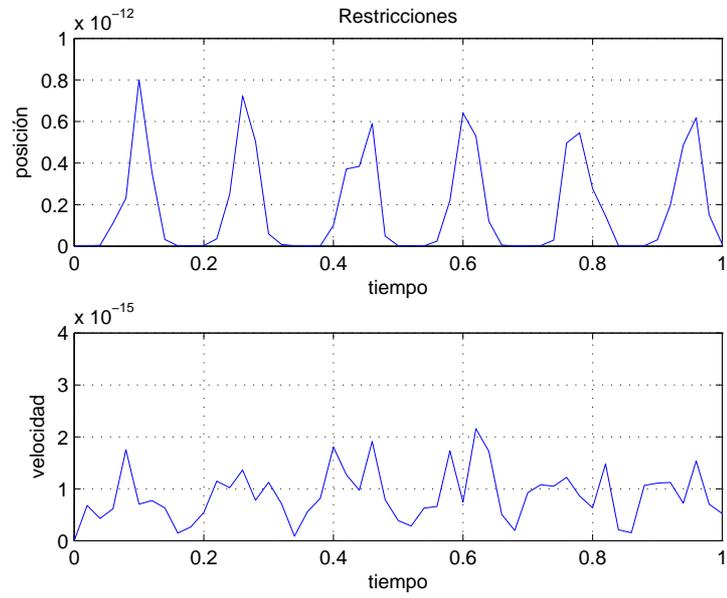


Figura 5.21: Automóvil: restricciones

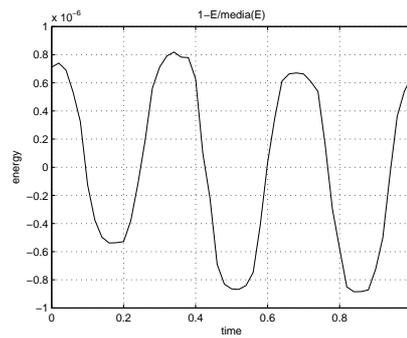


Figura 5.22: Automóvil: energía total

Capítulo 6

Conclusiones y futuras líneas de investigación

Al finalizar, en este capítulo se presentan una serie de conclusiones sobre los resultados obtenidos en la tesis y una serie de posibles líneas de investigación.

6.1. Conclusiones

En este trabajo se ha hecho un estudio de los modelos matemáticos más utilizados que permiten estudiar la simulación de un sistema multicuerpo. También se han revisado las formulaciones dinámicas más importantes en el ámbito de los métodos globales. Teniendo en cuenta estos estudios, se ha optado por una formulación donde la matriz característica tiene una dimensión mínima y es altamente hueca, aunque no es simétrica, ni cuadrada en la mayor parte de los casos. Se ha estudiado la base teórica de los métodos generales de reordenamiento más frecuentes y se han podido seleccionar aquellos que se han considerado mejores para obtener ventaja de la estructura de la matriz características a que conduce la formulación.

Con los reordenamientos seleccionados, y mediante la simulación dinámica de un conjunto de mecanismos de diferentes tipologías, se ha demostrado la ventaja que supone la reducción a formas que pre-

sentan bloques diagonales. También se ha podido aplicar un conjunto de técnicas de matrices huecas a un problema donde las matrices características no son simétricas.

Por último, se ha modificado el reordenamiento propuesto por Hellermann y Rarick. Mediante esta modificación, en cierta medida, se ha introducido en la resolución del sistema la topología del mismo, mediante el proceso de permutar las coordenadas y las ecuaciones de restricción que realiza el reordenamiento. Con esta modificación se ha creado un nuevo reordenamiento que podrá servir de base a futuros desarrollos. Este reordenamiento se ha contrastado y validado mediante la simulación de un conjunto de mecanismos y también se ha comparado su eficiencia con otros reordenamientos conocidos. Se ha demostrado como un método válido en algunos casos, donde fracasan los métodos generales analizados.

Se han comentado los problemas que siguen apareciendo con este tipo de reordenamientos: por ejemplo, los problemas numéricos derivados de los bloques formados. Este problema se ha tratado de aliviar en el proceso de resolución de las ecuaciones.

6.2. Futuras líneas de investigación

A partir del trabajo expuesto en esta Tesis Doctoral se pueden apuntar algunas líneas de investigación que lo complementen. Entre ellas, quizá las más destacables son:

- El estudio exhaustivo de las formulaciones topológicas para poder optimizar el reordenamiento propuesto, contrastándolo y relacionándolo con estas formulaciones. Como se ha indicado, sería óptimo que este tipo de reordenamiento tuviese en cuenta todas las características topológicas del sistema de forma automática.
- Muy ligado al punto anterior, la mejora del reordenamiento desarrollado para evitar (cuando sea posible) los bloques singulares que pueden aparecer una vez modificada la matriz.

- El estudio de los mecanismos que se encuentran en posiciones singulares, para determinar cuáles podrían ser las modificaciones al método propuesto en estos casos.

Bibliografía

- [1] JORGE A. C. AMBRÓSIO, ed. *Advances in computational multi-body systems*. Número 2 en *Computational methods in applied sciences*. Springer, Dordrecht (2005).
- [2] FARID M. L. AMIROUCHE. *Computational methods in multi-body dynamics*. Prentice Hall, Englewood Cliffs, N.J (1992).
- [3] J. ANGELES, A. KECSKEMÉTHY, y INTERNATIONAL CENTRE FOR MECHANICAL SCIENCES, eds. *Kinematics and dynamics of multi-body systems*. Número 360 en *Courses and lectures / International Centre for Mechanical Sciences*. Springer, Wien (1995).
- [4] U.M. ASCHER y L.R. PETZOLD. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics (1998).
- [5] J. BAUMGARTE. Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 1(1):1–16 (1972).
- [6] E. BAYO y R. LEDESMA. Augmented lagrangian and mass-orthogonal projection methods for constrained multibody dynamics. *Nonlinear Dynamics*, 9(1-2):113–130 (1996).
- [7] EDUARDO BAYO, JAVIER GARCÍA DE JALON, y MIGUEL ANGEL SERNA. A modified lagrangian formulation for the dynamic analysis of constrained mechanical systems. *Computer Methods in Applied Mechanics and Engineering*, 71(2):183–195 (1988).

-
- [8] J. CARDENAL, J. CUADRADO, P. MORER, y E. BAYO. A multi-index variable time step method for the dynamic simulation of multibody systems. *International Journal for Numerical Methods in Engineering*, 44(11):1579–1598 (1999).
- [9] JESÚS CARDENAL, IAIN S DUFF, y JOSÉ M JIMÉNEZ. Solution of sparse quasi-square rectangular systems by gaussian elimination. *IMA journal of numerical analysis*, 18(2):165–177 (1998).
- [10] J. CUADRADO. Special issue: Multibody dynamics formulations. *Journal of Computational and Nonlinear Dynamics*, 9(4) (2014).
- [11] J. CUADRADO, J. CARDENAL, y E. BAYO. Modeling and solution methods for efficient real-time simulation of multibody dynamics. *Multibody System Dynamics*, 1(3):259–280 (1997).
- [12] J. CUADRADO, J. CARDENAL, P. MORER, y E. BAYO. Intelligent simulation of multibody dynamics: Space-state and descriptor methods in sequential and parallel computing environments. *Multibody System Dynamics*, 4(1):55–73 (2000).
- [13] J. CUADRADO y P. MASARATI. Special issue: Application of multibody dynamics to biomechanics. *Journal of Multi-body Dynamic*, 227(4) (2013).
- [14] J. CUADRADO y W. SCHIEHLEN. Special issue: Real-time simulation and virtual reality applications of multibody systems. *Multibody System Dynamics*, 17(2-3) (2007).
- [15] E. CUTHILL y J. MCKEE. *Reducing the bandwidth of sparse symmetric matrices*. ACM Press (1969).
- [16] J. D’ALEMBERT. *Traité de Dynamique*. Paris (1743).
- [17] PIERRE DEHOMBREUX, OLIVIER VERLINDEN, y CALOGERO CONTI. An implicit multistage integration method including projection for the numerical simulation of constrained multibody systems. *Multibody System Dynamics*, 1(4):405–424 (1997).

-
- [18] J. DENAVIT y R.S. HARTENBERG. A kinematic motion of lower pair mechanisms based on matrices. *Journal of Applied Mechanics*, (222):215–221 (1955).
- [19] J. R. DORMAND y P. J. PRINCE. A family of embedded Runge-Kutta formulae. *J. Comp. Appl. Math.*, 6:19–26 (1980).
- [20] I. S. DUFF. On Algorithms for Obtaining a Maximum Transversal. *ACM Transactions on Mathematical Software*, 7(3):315–330 (1981).
- [21] I. S. DUFF, ALBERT ERISMAN, y JOHN REID. *Direct Methods for Sparse Matrices*. Oxford University Press, Inc., New York, NY, USA (1986).
- [22] I. S. DUFF y J. K. REID. An Implementation of Tarjan's Algorithm for the Block Triangularization of a Matrix. *ACM Transactions on Mathematical Software*, 4(2):137–147 (1978).
- [23] I.S. DUFF. A survey of sparse matrix research. *Proceedings of the IEEE*, 65(4):500–535 (1977).
- [24] EDDA EICH-SOELLNER y CLAUS FÄHRER. *Numerical methods in multibody dynamics*. European Consortium for Mathematics in Industry. Teubner, Stuttgart (1998).
- [25] A. M. ERISMAN, R. G. GRIMES, J. G. LEWIS, y W. G. POOLE, JR. A Structurally Stable Modification of Hellerman Rarick's p4 Algorithm for Reordering Unsymmetric Sparse Matrices. *SIAM Journal on Numerical Analysis*, 22(2):369–385 (1985).
- [26] EULER. *Nova Methodus Motum Corporum Rigidarum Determinandi, Novi Comentariorum Academiæ Scientiarum Petropolitanae*, 20 (1776).
- [27] R. FEATHERSTONE. The Calculation of Robot Dynamics Using Articulated-Body Inertias. *The International Journal of Robotics Research*, 2(1):13–30 (1983).
- [28] ROY FEATHERSTONE. *Robot dynamics algorithms*. Número SECS 22 en The Kluwer international series in engineering and computer science ; Robotics. Kluwer, Boston (1987).

- [29] OLOF FRIBERG. Dynamics of multibody systems. *International Journal for Numerical Methods in Engineering*, 29(8):1797–1797 (1990).
- [30] FRANCISCO JAVIER FUNES, JOSÉ MANUEL JIMÉNEZ, JOSÉ IGNACIO RODRÍGUEZ, y JAVIER GARCÍA DE JALÓN. A new sparse solver for kinematics simulation of multibody systems in natural coordinates based on tearing methods (2001).
- [31] JOHN ALAN GEORGE. *Computer Implementation of the Finite Element Method*. Tesis Doctoral, Stanford, CA, USA (1971).
- [32] JOHN ALAN GEORGE. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10(2):345–363 (1973).
- [33] HERBERT GOLDSTEIN, CHARLES P. JR. POOLE, y JOHN L. SAFKO. *Classical Mechanics, 3th Ed.* Addison-Wesley, Reading (Mass.), Menlo Park (Calif.), Amsterdam (2001).
- [34] F. G. GUSTAVSON. Finding the block lower triangular form of a sparse matrix (1976).
- [35] EDWARD J. HAUG. *Computer aided kinematics and dynamics of mechanical systems*. Allyn and Bacon series in engineering. Allyn and Bacon, Boston (1989).
- [36] ELI HELLERMAN y DENNIS RARICK. Reinversion with the preassigned pivot procedure. *Mathematical Programming*, 1(1):195–216 (1971).
- [37] ELI HELLERMAN y DENNIS C. RARICK. The partitioned preassigned pivot procedure (p4). En DONALD J. ROSE y RALPHA. WILLOUGHBY, eds., *Sparse Matrices and their Applications*, The IBM Research Symposia Series, páginas 67–76. Springer US (1972).
- [38] W HOOKER y G MARGULIES. The dynamical attitude equations for an n-body satellite. *The Journal of Astronautical Sciences*, 12(4):123–128 (1965).
- [39] RONALD L. HUSTON. *Multibody Dynamics*. Butterworth-Heinemann (1990).

- [40] J. GARCÍA DE JALÓN, E. ÁLVAREZ, F.A. DE RIBERA, I. RODRÍGUEZ, y F.J. FUNES. A fast and simple semi-recursive formulation for multi-rigid-body systems. En JORGE A.C. AMBRÓSIO, ed., *Advances in Computational Multibody Systems*, tomo 2 de *Computational Methods in Applied Sciences*, páginas 1–23. Springer Netherlands (2005).
- [41] J. GARCÍA DE JALÓN y R. AVILÉS. A computer method for kinematic analysis of lower-pair mechanisms. Second part: position problems. *Mechanism and Machine Theory*, 16:557–566 (1981).
- [42] J. GARCÍA DE JALÓN y E. BAYO. *Kinematic and Dynamic Simulation of Multibody Systems: The Real-Time Challenge*. Springer, New York (1993).
- [43] J. GARCÍA DE JALÓN, J. UNDA, y A. AVELLO. Natural coordinates for the computer analysis of multibody systems. *Computer Methods in Applied Mechanics and Engineering*, 56(3):309–327 (1986).
- [44] J. GARCÍA DE JALÓN, J. UNDA, A. AVELLO, y J. M. JIMÉNEZ. Dynamic Analysis of Three-Dimensional Mechanisms in Natural Coordinates. *Journal of Mechanisms Transmissions and Automation in Design*, 109(4):460 (1987).
- [45] PH. E. B. JOURDAIN. Note on an analogue of Gauss' principle of least constraint. *The Quarterly Journal of Pure and Applied Mathematics*, 40:153–157 (1908).
- [46] J. W. KAMMAN y R. L. HUSTON. Dynamics of Constrained Multibody Systems. *Journal of Applied Mechanics*, 51(4):899 (1984).
- [47] THOMAS R. KANE y DAVID A. LEVINSON. *Dynamics, theory and applications*. McGraw-Hill series in mechanical engineering. McGraw-Hill, New York (1985).
- [48] S. S. KIM y M. J. VANDERPLOEG. Qr decomposition for state space representation of constrained mechanical dynamic systems. *Journal of Mechanisms, Transmissions, and Automation in Design*, 108(2):183–188 (1986).

- [49] HAROLD W. KUHN. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97 (1955).
- [50] J.L. LAGRANGE. *Mécanique Analytique*. L'Academie Royal des Sciences, Paris (1788).
- [51] CH. LUBICH. Extrapolation integrators for constrained multi-body systems. *{IMPACT} of Computing in Science and Engineering*, 3(3):213 – 234 (1991).
- [52] K MAGNUS. *Dynamics of Multibody Systems Symposium Munich/Germany August 29-September 3, 1977*. Springer Berlin Heidelberg, Berlin, Heidelberg (1978).
- [53] N. K. MANI, E. J. HAUG, y K. E. ATKINSON. Application of singular value decomposition for analysis of mechanical system dynamics. *Journal of Mechanisms, Transmissions, and Automation in Design*, 107(1):82–87 (1985).
- [54] H. M. MARKOWITZ. The elimination form of the inverse and its application. *Management Science*, 3:257–269 (1957).
- [55] I. NEWTON. *Philosophiae Naturalis Principia Mathematica*. Royal Society, London (1687).
- [56] P. E. NIKRAVESH. *Computer-aided analysis of mechanical systems*. Prentice-Hall, Englewood Cliffs, N.J (1988).
- [57] P. E. NIKRAVESH, R. A. WEHAGE, y O. K. KWON. Euler Parameters in Computational Kinematics and Dynamics. Part 1. *Journal of Mechanisms Transmissions and Automation in Design*, 107(3):358 (1985).
- [58] N. ORLANDEA, M.A. CHACE, y D.A. CALAHAN. A sparse oriented approach to the dynamic analysis and design of mechanicals systems, part i and ii. *ASME Journal of Engineering for Industry*, 99:773–784 (1977).
- [59] B. PAUL. Analytical dynamics of mechanisms- a computer oriented overview. *Mechanism and Machine Theory*, 10:481–507 (1975).

-
- [60] B. PAUL y D. KRAJČINOVIC. Computer analysis of machines with planar motion- part i: Kinematics; part ii dynamics. *Journal of Applied Mechanics*, 37:697–712 (1970).
- [61] FRIEDRICH PFEIFFER y CHRISTOPH GLOCKER. *Multibody dynamics with unilateral contacts*. Wiley series in nonlinear science. Wiley, New York (1996).
- [62] R. E. ROBERSON y J. WITTENBURG. A dynamical formalism for an arbitrary number of interconnected rigid bodies, with reference to the problem of satellite attitude control. En *3rd IFAC Congress*, páginas 46D.2, 46D.9 (1967).
- [63] ROBERT E. ROBERSON y RICHARD SCHWERTASSEK. *Dynamics of Multibody Systems*. Springer-Verlag (1988).
- [64] DONALD J. ROSE, R. ENDRE TARJAN, y GEORGE S. LUEKER. Algorithmic Aspects of Vertex Elimination on Graphs. *SIAM Journal on Computing*, 5(2):266–283 (1976).
- [65] JEAN-CLAUDE SAMIN y PAUL FISETTE. *Symbolic modeling of multibody systems*. Número v. 112 en Solid mechanics and its applications. Kluwer Academic Publishers, Dordrecht ; Boston (2003).
- [66] W. SCHIEHLEN. *Multibody Systems Handbook*. Springer Berlin Heidelberg, Berlin, Heidelberg (1990).
- [67] W. SCHIEHLEN. Multibody system dynamics: Roots and perspectives. *Multibody system dynamics*, 1(2):149–188 (1997).
- [68] REINHOLD VON SCHWERIN. *Multibody system simulation: numerical methods, algorithms, and software*. Número 7 en Lecture notes in computational science and engineering. Springer-Verlag, Berlin ; New York (1999).
- [69] MANUEL F. O SEABRA PEREIRA y JORGE A. C AMBRÓSIO. *Computer-Aided Analysis of Rigid and Flexible Mechanical Systems*. Springer Netherlands : Imprint : Springer, Dordrecht (1994).

- [70] MIGUEL ANGEL SERNA, RAFAEL AVILÉS, y JAVIER GARCÍA DE JALÓN. Dynamic analysis of plane mechanisms with lower pairs in basic coordinates. *Mechanism and Machine Theory*, 17(6):397–403 (1982).
- [71] AHMED A. SHABANA. Flexible Multibody Dynamics: Review of Past and Recent Developments. *Multibody System Dynamics*, 1(2):189–222 (1997).
- [72] L. F. SHAMPINE y M. K. GORDON. *Computer solution of ordinary differential equations: the initial value problem*. W. H. Freeman, San Francisco (1975).
- [73] LAWRENCE F. SHAMPINE y MARK W. REICHEL. The MATLAB ODE Suite. *SIAM Journal on Scientific Computing*, 18(1):1–22 (1997).
- [74] P. N. SHETH y J. J. UICKER. IMP (Integrated Mechanisms Program), A Computer-Aided Design Analysis System for Mechanisms and Linkage. *Journal of Engineering for Industry*, 94(2):454 (1972).
- [75] R. P. SINGH y P. W. LIKINS. Singular value decomposition for constrained dynamical systems. *Journal of Applied Mechanics*, 52(4):943–948 (1985).
- [76] D. A. SMITH, M. A. CHACE, y A. C. RUBENS. The Automatic Generation of a Mathematical Model for Machinery Systems. *Journal of Engineering for Industry*, 95(2):629 (1973).
- [77] ROBERT TARJAN. Depth first search and linear graph algorithms. *SIAM Journal on Computing* (1972).
- [78] W.F. TINNEY y J.W. WALKER. Direct solutions of sparse network equations by optimally ordered triangular factorization. *Proceedings of the IEEE*, 55(11):1801–1809 (1967).
- [79] J UICKER. *On the Dynamic Analysis of Spatial Linkages Using 4 by 4 Matrices*. Tesis Doctoral, Northwestern University (1965).

-
- [80] G.N. VANDERPLAATS. *Numerical optimization techniques for engineering design: with applications*. McGraw-Hill Series in Mechanical Engineering. McGraw-Hill Ryerson, Limited (1984).
- [81] JESÚS VIDAL. *Un Método General, Sencillo y Eficiente, para la Definición y Simulación Numérica de Sistemas Multicuerpo*. Tesis doctoral, Universidad Politécnica de Madrid, Madrid (2006).
- [82] R. A. WEHAGE y E. J. HAUG. Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems. *Journal of Mechanical Design*, 104(1):247 (1982).
- [83] FERDINAND WITTENBAUER. *Graphische Dynamik: ein Lehrbuch für Studierende und Ingenieure mit zahlreichen Anwendungen und Aufgaben*. Julius Springer (1923).
- [84] J. WITTENBURG. *Dynamics of systems of rigid bodies*. Leitfäden der angewandten Mathematik und Mechanik. Teubner (1977).
- [85] MIHALIS YANNAKAKIS. Computing the Minimum Fill-In is NP-Complete. *SIAM Journal on Algebraic Discrete Methods*, 2(1):77–79 (1981).