



UNIVERSIDADE DA CORUÑA

ESCUELA POLITÉCNICA SUPERIOR

Aplicación de la Dinámica
Multicuerpo en Tiempo Real a
la Simulación y el Control de
Automóviles

Memoria para optar al grado de Doctor Ingeniero Industrial

Miguel Ángel Naya Villaverde

Ingeniero Industrial

Director: Javier Cuadrado Aranda

Ferrol, Marzo 2007

*A mis padres,
a mis hermanos.*

Agradecimientos

Gracias a Dios, se termina en algún momento de escribir la tesis doctoral. Dar gracias, además de una obligación, es un placer porque se tiene la oportunidad de recordar a muchos amigos y muchos buenos momentos.

Mi agradecimiento se dirige especialmente a Javier Cuadrado por su labor, no pequeña, en la dirección de esta tesis. No exagero al decir que a él se debe prácticamente la realización de este trabajo gracias a sus indicaciones, aliento, paciencia, ..., a todo.

También quiero agradecer a mis compañeros del Laboratorio de Ingeniería Mecánica en la Universidad de La Coruña, Manuel González (Lolo), Daniel Dopico, Urbano Lugrís, Alberto Luaces y Fran González, las innumerables ayudas que han prestado y lo bien que se trabaja con ellos. De entre los compañeros de la Universidad debo un agradecimiento particular a José Antonio Pérez por su ayuda y sus explicaciones sobre control.

Esta tesis habría sido mucho más difícil sin el trabajo realizado a través de algunos Proyectos Fin de Carrera y agradezco a sus autores (me gusta pensar que mis alumnos), especialmente a Abraham Prieto y, sobre todo, a Alberto Luaces su colaboración.

También debo recordar la acogida de Pablo Luque y Daniel Álvarez en la Escuela Politécnica de Ingenieros de Gijón, así como sus orientaciones sobre instrumentación. Ellos hicieron que mi estancia, ya lejana, en Gijón resultase totalmente fructífera.

Siempre es necesario el apoyo y el ánimo de la familia para realizar cualquier tarea mínimamente ardua y la realización de una tesis no es una excepción. Por eso, quiero agradecer de manera muy especial a mi madre y a mis hermanos su ayuda, su comprensión, el ánimo y, ¿cómo no?, por saber disculpar.

Resumen

La dinámica de sistemas multicuerpo permite realizar modelos detallados de sistemas mecánicos complejos, como puede ser un vehículo automóvil. Las formulaciones desarrolladas durante los últimos años permiten llevar a cabo simulaciones muy eficientes, que incluso hacen posible calcular el movimiento por debajo del tiempo real. Por este motivo, la dinámica multicuerpo es una herramienta cada vez más útil tanto en el diseño del automóvil como en el de sus sistemas de control. En esta tesis se presenta, como ejemplo de aplicación, el desarrollo de un simulador de conducción de bajo coste y su empleo en el diseño de controladores del vehículo simulado.

La formulación empleada en el presente trabajo ha sido desarrollada con anterioridad en el Laboratorio de Ingeniería Mecánica de la Universidad de La Coruña. Se trata de una formulación aumentada de Lagrange de índice 3 con proyecciones en velocidades y aceleraciones, con la regla trapezoidal como integrador, y que ha demostrado ser robusta y eficiente.

En primer lugar se describe el vehículo a simular: un prototipo formado por un chasis de tubos y elementos tomados de vehículos comerciales, construido en el Laboratorio. A continuación se detalla su modelización y la caracterización de las fuerzas que actúan sobre él: suspensión, contacto con el suelo, fuerzas de aceleración y frenado, colisiones, etc.

A partir del modelo computacional del vehículo, se desarrolla un simulador de conducción, que permite evaluar la factibilidad de las maniobras a seguir por el vehículo guiado por los controladores, y comparar su realización con las llevadas a cabo por un conductor humano. Se trata de un simulador de bajo coste, que toma sus inputs de un volante y unos pedales del tipo de los empleados en los juegos de ordenador. La salida gráfica se puede realizar a través de la pantalla del ordenador, o mediante unas gafas de realidad virtual, incorporando un sensor para detectar el movimiento de la cabeza del conductor. Se han desarrollado varios

entornos para la simulación: una recreación de un barrio de Ferrol, una representación del Circuito de Cataluña para simular varios vehículos como demostración de eficiencia, y un entorno con obstáculos (rampas, escaleras, etc.) en el que poder realizar maniobras bruscas para comprobar la robustez del modelo virtual.

Para diseñar controladores, se han estudiado las diversas posibilidades de conexión del modelo computacional con Matlab, de las que Matlab Engine ha resultado ser la más adecuada. Se ha empleado la toolbox de Lógica Borrosa de Matlab, para desarrollar unos controladores borrosos que actúen sobre el volante y los pedales del vehículo y realizar, como ejemplo, dos maniobras de forma autónoma: recorrer una distancia recta de 20 m deteniéndose al completar esa distancia, y una trayectoria sinusoidal similar a la que seguiría el vehículo al eludir un obstáculo.

Finalmente, con el fin de validar los controladores diseñados, se ha instrumentado el prototipo real y se ha desarrollado un algoritmo para la recuperación del movimiento. Este algoritmo se basa en un procedimiento odométrico que proporciona errores muy bajos en la localización del vehículo. También se ha instrumentado la actuación sobre el vehículo con vistas a su conducción autónoma, y se propone un algoritmo para la actuación.

Índice

1.	Introducción	1
1.1.	Motivación.	1
1.2.	Estado del Arte.	2
1.2.1.	Principales Proyectos ITS.	4
1.2.2.	Vehículos Inteligentes	9
1.2.3.	Esquemas de Control para la Automatización.	21
1.2.4.	Modelos Dinámicos.	30
1.2.5.	Tiempo Real.	34
1.3.	Objetivos.	37
1.4.	Contenido de la Tesis.	38
2.	El Prototipo Real	41
2.1.	Chasis.	42
2.2.	Grupo Motor.	45
2.3.	Masas, Centros de Gravedad e Inercias.	48
2.4.	Sistema de Frenos.	50
2.5.	Suspensión.	51
2.6.	Dirección.	53
3.	El Modelo Virtual del Prototipo	55
3.1.	Modelado: Coordenadas Naturales.	55
3.2.	Ecuaciones del Movimiento e Integración Numérica.	57
3.3.	Propiedades Inerciales del Chasis y del Motor.	61
3.4.	Fuerzas Transmitidas por la Suspensión.	62
3.5.	Caracterización del Contacto Neumático-Suelo.	63
3.5.1.	Fuerza Vertical y de Compresión.	63
3.5.2.	Fuerza Longitudinal.	71
3.5.3.	Fuerza Lateral y Momento Autoalineante.	74
3.5.4.	Peculiaridades de la Fuerza Lateral.	80

3.5.5.	Introducción de las Fuerzas.	81
3.5.6.	Resistencia a la Rodadura.	89
3.6.	Fuerzas de Aceleración y Frenado.	91
3.6.1.	Curvas de Par del Motor.	91
3.6.2.	Cambio Automático de Velocidades.	95
3.6.3.	Fuerzas de Frenado.	96
3.6.4.	Introducción de las Fuerzas.	97
3.7.	Colisiones.	100
3.8.	Fuerzas Aerodinámicas.	102
4.	Simulador de Conducción con Control Humano	105
4.1.	Organización del Simulador.	105
4.1.1.	Desarrollo del Programa.	110
4.1.2.	Sincronización Temporal de la Simulación.	110
4.2.	Input: Volante y Pedales, Datos Topográficos.	112
4.3.	Output: Entorno Gráfico, Gafas de Realidad Virtual, Grabación de Maniobras.	115
4.3.1.	OpenGL frente a DirectX.	115
4.3.2.	Generación de Entornos.	117
4.3.3.	Representación del Vehículo y Otros Objetos.	125
4.3.4.	Gafas de Realidad Virtual (HMD).	130
4.3.5.	Grabación de Maniobras.	140
5.	Módulo para el Diseño de Controladores	143
5.1.	Organización del Módulo.	143
5.2.	Técnica de Control Elegida.	146
5.2.1.	Justificación del Empleo de Fuzzy Logic.	146
5.2.2.	Características Principales de la Lógica Borrosa.	148
5.3.	Diseño de los Controladores: Conexión con Matlab.	152
5.3.1.	Matlab Engine.	153
5.3.2.	Ficheros MEX.	157
5.4.	Aplicación de los Controladores al Simulador de Conducción.	161
5.4.1.	Organización del Simulador.	162
5.4.2.	Matrices del Controlador.	163
5.4.3.	Control Parcial.	164
6.	Diseño de Controladores: Ejemplos	167
6.1.	Elección de las Maniobras.	167
6.2.	Primera Maniobra: Trayectoria Recta.	168
6.2.1.	Realización de la Maniobra.	173
6.3.	Segunda Maniobra: Elusión de un Obstáculo.	175
6.3.1.	Descripción de la Trayectoria a Seguir: Curvas de Bézier.	175
6.3.2.	Diseño de los Controladores.	180
6.3.3.	Realización de la Maniobra.	184

7.	Evaluación de Eficiencia y Robustez	189
7.1.	Entorno con Obstáculos.	190
7.2.	Simulación de Varios Vehículos.	193
7.2.1.	El Circuito.	194
7.2.2.	Realismo e Inmersión en el Entorno.	197
7.2.3.	Panel de Datos.	201
7.2.4.	Introducción de Varios Vehículos.	202
7.2.5.	Algoritmo de Decisión de la Velocidad Máxima en un Tramo.	202
7.2.6.	Eficiencia del Simulador.	204
8.	Instrumentación del Prototipo	207
8.1.	Motivación.	207
8.1.1.	Adquisición y Control de Datos.	208
8.2.	Determinación del Movimiento.	211
8.2.1.	Sensores Empleados.	213
8.2.2.	Algoritmo.	214
8.3.	Actuación.	217
8.3.1.	Motores.	218
8.3.2.	Algoritmo de Actuación.	220
8.4.	Pruebas y Resultados.	221
9.	Conclusiones	225
9.1.	Conclusiones.	225
9.2.	Futuras Líneas de Investigación.	228
10.	Bibliografía	233

1.

Introducción

1.1. Motivación.

En la actualidad, las técnicas de simulación de sistemas multicuerpo se encuentran cada vez más implantadas en el ámbito industrial, especialmente en grandes industrias como la de la automoción y el ferrocarril, la aeroespacial, la robótica y, más recientemente pero cada vez con más aplicaciones, la biomecánica. Se dispone de diferentes herramientas de software que permiten reducir costes de producción y disminuir tiempos en los procesos de diseño y fabricación. Sin embargo, no dejan de ser técnicas recientes cuya implantación puede extenderse mucho más. Además, están muy ligadas a los desarrollos en otros ámbitos como la informática o la realidad virtual. Por este motivo, todavía hay un gran campo para el estudio de nuevos métodos y formulaciones que resuelvan la cinemática y la dinámica de sistemas mecánicos complejos.

En el Laboratorio de Ingeniería Mecánica de la Universidad de La Coruña se ha trabajado durante largo tiempo en el desarrollo de algoritmos de simulación de sistemas multicuerpo eficientes. Estos trabajos han conducido al desarrollo de varias formulaciones que permiten conseguir la ejecución de simulaciones de la dinámica de un sistema en tiempo real.

Habitualmente, las primeras aplicaciones para probar estas formulaciones son sistemas mecánicos poco complejos, como un mecanismo de cuatro barras. Es también conveniente aplicar estas metodologías a problemas complejos, con un elevado número de variables, para comprobar que la formulación sigue comportándose de la misma manera que en el ejemplo. Particular interés tienen aquellos problemas derivados de realidades industriales, como puede ser la simulación de un automóvil, y éste es el problema que se aborda en este trabajo.

Una de las metodologías desarrolladas consiste en una formulación aumentada de Lagrange, de índice 3, con proyecciones en velocidades y aceleraciones, y que emplea como integrador la regla trapezoidal. Se trata de una formulación global y de un esquema de integración implícito. Sin embargo, en las aplicaciones orientadas a la simulación dinámica en tiempo real es común emplear formulaciones topológicas con esquemas de integración explícitos. Por ello, verificar la idoneidad de la formulación del Laboratorio para este tipo de problemas constituye un reto de gran interés.

Por otro lado, la simulación de automóviles abre la puerta a otros campos de gran actualidad como son el control y todas sus aplicaciones de seguridad activa y pasiva, la necesidad de que la simulación se desarrolle en tiempo-real o por debajo de él, la inclusión de dispositivos periféricos en la simulación que permitan interactuar con el usuario, la creación de entornos gráficos, y la realidad virtual. Todo este conjunto ha servido como motivación para el desarrollo del presente trabajo, en el que se da solución a muchos de estos problemas y, al mismo tiempo, se abre la puerta a nuevos interrogantes.

1.2. Estado del Arte.

El control y la automatización de un vehículo automóvil abarcan diversos ámbitos como son todos los relativos a las técnicas de control y automatización de maniobras, la instrumentación del vehículo y de la carretera, y la modelización del vehículo. Asimismo, los enfoques de este mismo problema son variados (seguridad vial, asistencia a la conducción, etc.) y provienen tanto de instituciones públicas como privadas, con estrategias variadas e intereses diversos. Por este motivo, la información se encuentra muy dispersa y resulta difícil llegar a una síntesis clarificadora de los avances que se han producido en este campo. Sin embargo, tres

artículos nucleares en esta revisión son los de Shladover (1995), Vahidi y Eskandarian (2003) y Shladover (2005).

En este apartado sobre el estado del arte, en primer lugar se hace un repaso de la evolución de la automatización de los automóviles. A continuación, se describen los principales programas internacionales relacionados con esta investigación. Seguidamente, bajo el epígrafe de Vehículos Inteligentes, se exponen los principales sistemas de control de vehículos desarrollados y, en el epígrafe siguiente se detallan las técnicas de control aplicadas. Finalmente, se resumen los modelos empleados en el estudio de la dinámica de vehículos y las formulaciones de sistemas multicuerpo empleadas en la actualidad.

Los principales objetivos de la investigación en este campo son mejorar la seguridad, incrementar la capacidad de tráfico en carretera, reducir el consumo de combustible y aumentar el confort y la manejabilidad por parte del conductor (Vahidi and Eskandarian 2003). Una mayor automatización del vehículo descarga al conductor de muchas de las operaciones rutinarias englobadas en la conducción y reduce la probabilidad de fallos humanos. Al mismo tiempo, al aumentar el número de vehículos en las carreteras, surge una nueva problemática que es la capacidad de las vías. Como solución para todos estos fenómenos se ha propuesto, entre otras, el desarrollo de un tipo de automatización que ayude a incrementar el flujo de tráfico de manera segura. Asimismo, una circulación de los vehículos menos brusca, guiada mediante un sistema automatizado, puede reducir el consumo de combustible y el desgaste del motor.

Teniendo presentes todos los beneficios anteriormente mencionados, la investigación para automatizar todos o algunos de los aspectos incluidos en la conducción se ha desarrollado hace ya varias décadas (Shladover 1995). No obstante, durante buena parte del siglo XX, la electrónica y la tecnología de los sensores eran todavía muy rudimentarias, y eso supuso un freno para el desarrollo y la aplicación de estas investigaciones.

Se puede decir que la automatización comienza hacia 1930, pero no es hasta finales de los años 80 y comienzos de los 90 cuando aparecen programas, tanto estatales como privados, y especialmente en Estados Unidos, Europa y Japón (French et al. 1994), con la idea de hacer realidad sistemas de transporte inteligente o automatizado. El principal interés residía en aumentar la capacidad de las carreteras y la seguridad mediante la automatización, tanto a nivel de infraestructuras como de vehículos. La

investigación desarrollada durante esta época estaba muy bien organizada aunque, en ocasiones, resultase un tanto futurista, y contribuyó, junto con los rápidos avances en electrónica y tecnología de los sensores, a una mejor comprensión de las dificultades y de las potencialidades de estos sistemas. Inicialmente, la investigación se centró más en el desarrollo de Sistemas Avanzados de Carreteras (Advanced Highway Systems, AHS) y, posteriormente, el interés pasó a la denominada Iniciativa en Vehículos Inteligentes (Intelligent Vehicle Initiative, IVI). Sin embargo, toda la investigación desarrollada en la etapa inicial sobre AHS constituyó una buena base para la aplicación en sistemas IVI. Todo el conjunto de esta investigación, tanto la relativa al equipamiento y automatización de los vehículos, como la mejora y la dotación de instrumentos de información en las infraestructuras, se agrupa en el concepto denominado Sistemas de Transporte Inteligente (Intelligent Transportation Systems, ITS).

En estos momentos, al comenzar el siglo XXI, algunos fabricantes han introducido ya en sus vehículos el Control Adaptativo de Crucero (ACC), y otros muchos se esmeran por introducir ACC y otros sistemas avanzados, como la detección y elusión de colisiones. Muchos sistemas que en el pasado parecían elucubraciones futuristas son ya realidad, pero quedan todavía por resolver muchas cuestiones antes de poder introducir ampliamente los sistemas de asistencia a la conducción. La teoría de control y sus aplicaciones se encuentra en un estadio avanzado. La sensorización plantea todavía nuevos retos, y el impacto de la automatización de la conducción sobre el conductor necesita aún un estudio más detallado, ocupando buena parte de los esfuerzos que se están llevando a cabo actualmente. Por otro lado, se han realizado diferentes propuestas para el aumento de la seguridad vial, el aumento de capacidad de las carreteras y la mejora del confort. Sin embargo, en muchas ocasiones surgen numerosas inconsistencias entre los distintos puntos de vista desde los que se abordan los diversos objetivos.

1.2.1. Principales Proyectos ITS.

En Europa, la investigación sobre sistemas ITS se ha llevado a cabo dentro de los programas DRIVE (Dedicated Road Infrastructure for Vehicle Safety in Europe), y PROMETHEUS (Program for European Traffic with Highest Efficiency and Unprecedented Safety). El programa DRIVE estaba dedicado al aumento de la seguridad en infraestructuras, mientras que PROMETHEUS tenía como objetivo la mejora de la eficiencia en el tráfico

por carretera. Sin embargo, la cooperación entre ambos proyectos fue muy frecuente para el logro de un fin común: la mejora de la seguridad en la conducción.

DRIVE y sus diversos subprogramas comenzaron en el año 1989 bajo la dirección de la Comisión de Comunidades Europeas (CEC). Tras una fase inicial de estudio, en el entorno de este programa se desarrollaron áreas relacionadas con la gestión de tráfico urbano e interurbano, asistencia en carretera, información al conductor sobre el estado del tráfico, y gestión de flotas de camiones y mercancías.

El programa PROMETHEUS comenzó en 1986 como parte de la plataforma European Research Coordination Agency (EUREKA). La finalidad de este programa era estimular el desarrollo en áreas como tecnologías de la información, telecomunicaciones, robótica y tecnologías del transporte. El proyecto se llevó a cabo por 18 compañías europeas de automóviles, autoridades estatales, y más de 40 instituciones dedicadas a la investigación.

El principal interés de PROMETHEUS se centró en el desarrollo de equipos de ayuda a la navegación en los vehículos, y su investigación se dividió fundamentalmente en las siguientes categorías:

- Mejora de la información al conductor, proporcionándole nuevos recursos mediante el uso de las nuevas tecnologías.
- Asistencia activa al conductor, de tal manera que cuando éste sea incapaz de realizar alguna tarea, el sistema pueda proporcionarle información o bien desarrollar una actuación autónoma.
- Establecimiento de una red de comunicaciones entre vehículos, que sea capaz de proporcionar al conductor información sobre el estado del tráfico en algún tramo de la ruta.
- Gestión del tráfico y flotas de vehículos.

Como se puede observar, dentro de estas áreas se encuentran cuestiones de gran interés, como pueden ser los sistemas autónomos de control de la conducción, la visión artificial, los sistemas de navegación, etc.

Para facilitar la introducción en el mercado de los desarrollos de estos programas, en 1991 se creó la European Road Transport Telematics Implementation Coordination Organization (ERTICO), que se encuentra actualmente vigente, y en la que se hallan integrados la mayoría de los países europeos y un elevado número de instituciones.

Japón fue el primer país en introducir sistemas de control del tráfico (Kuwahara 1998). Esto fue debido a la necesidad acuciante de tales sistemas debido a las limitaciones de espacio, y al elevado número de vehículos existentes ya en 1970. En muchas áreas urbanas, las velocidades de circulación eran inferiores a 16 km/h durante muchas de las horas del día. Por este motivo, el gobierno japonés ha instalado dispositivos de control de tráfico en todas las grandes ciudades y, en la mayoría de las zonas interurbanas. Estos dispositivos están orientados, fundamentalmente, a proporcionar información al conductor.

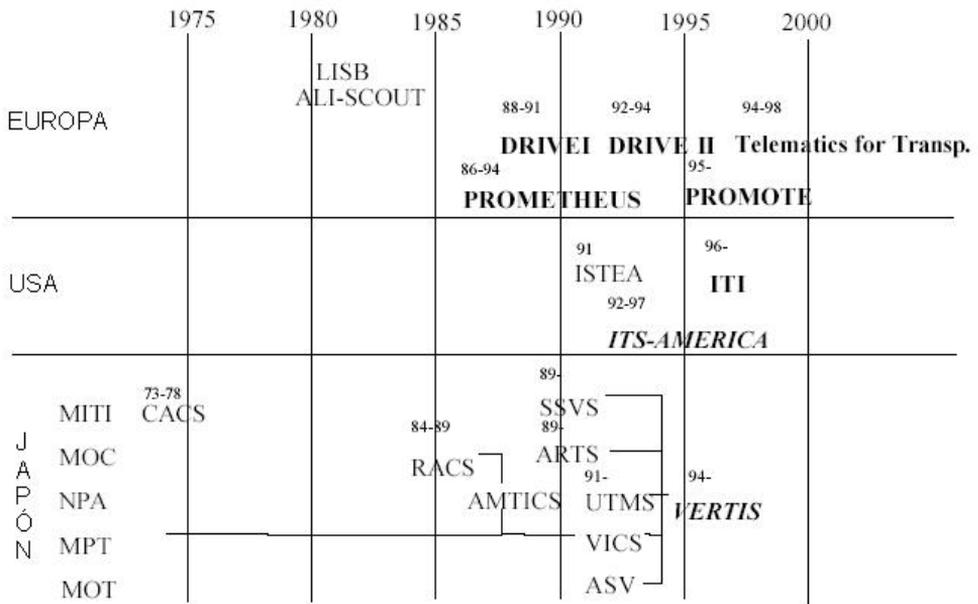


Figura 1.1 Proyectos ITS en Europa, Estados Unidos y Japón.

Ya en 1970 apareció el primer sistema de navegación, CACS, aunque con escasa aceptación en el mercado. Sin embargo, desde mediados de los años 80 se iniciaron varios proyectos ITS: RACS, AMTICS y SSVS, organizados desde instancias gubernamentales diferentes.

El programa RACS propuso un desarrollo en las infraestructuras, mediante la instalación de tres tipos de balizas en la carretera para proporcionar información y ayuda al conductor. Un tipo de baliza proporcionaba información de la posición del vehículo con respecto a un origen, de tal forma que se pudiesen corregir los errores en la navegación. Las balizas de tipo 2 proporcionaban datos sobre el estado del tráfico, y las de tipo 3 intercambian información con el vehículo (situación, deudas de peajes, etc.)

Al mismo tiempo, y bajo otros patrocinadores, el programa AMTICS empleaba un sistema con un equipamiento a bordo del vehículo similar al propuesto por RACS, pero con otro interfaz de usuario. En este sistema se procura ofrecer al conductor una amplia variedad de información, a través de un sistema de terminales fijos en estaciones de ferrocarril, hoteles, etc., así como en el propio vehículo.

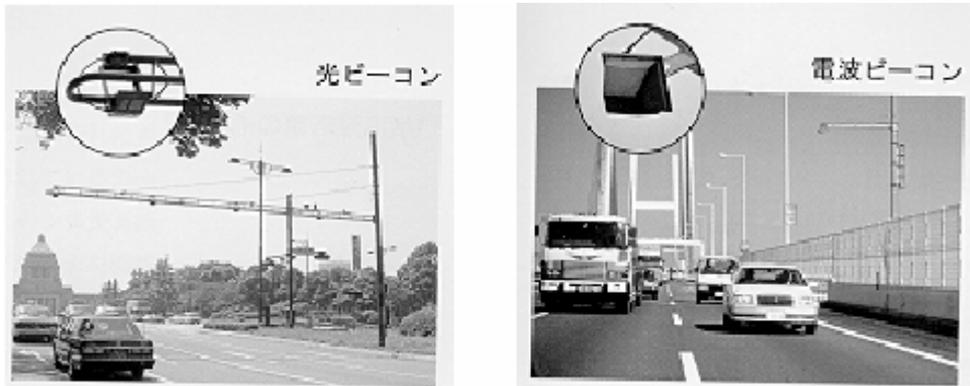


Figura 1.2 Sistemas de balizas por infrarrojos y microondas.

En el programa VICS, que comenzó en 1996, una unidad central de supervisión recibe información en tiempo real sobre el estado de cada una de las carreteras y, a su vez, proporciona información al conductor a través de balizas y de la radio. El sistema de comunicación que emplean las balizas instaladas en las autopistas son las microondas, mientras que las balizas situadas en las calles realizan sus emisiones por infrarrojos.

Sin embargo, a partir de 1994, el esfuerzo en investigación y desarrollo ha sido más armonioso entre los distintos organismos, y se ha recogido en VERTIS, que es un programa similar al ERTICO europeo. De esta manera, la investigación en ITS en Japón se ha dividido en las siguientes nueve áreas, que están actualmente en estudio y desarrollo:

1. Avances en sistemas de navegación.
2. Sistemas de telepeaje.
3. Asistencia para la seguridad en la conducción.
4. Optimización de la gestión del tráfico.
5. Aumento de la eficiencia en la gestión de carreteras.
6. Asistencia al transporte público.
7. Aumento de la eficiencia en el trabajo de vehículos comerciales.

8. Asistencia a peatones.

9. Asistencia a vehículos en situación de emergencia.

En los Estados Unidos, el programa PATH, fundado por el California Department of Transportation y el Institute of Transportation Studies de la Universidad de Berkeley, promovió los principales proyectos de investigación en vehículos avanzados y sistemas de carretera. Asimismo, la investigación llevada a cabo por el programa PATH estimuló el estudio de los sistemas de transporte a lo largo y ancho de los Estados Unidos. La idea principal en estas investigaciones consistía en sustituir un grupo de decisiones y acciones llevadas a cabo por el conductor, mediante un conjunto de tareas más sistematizadas y precisas ejecutadas por un ordenador, con el fin de alcanzar un flujo de tráfico más regular y, consiguientemente, obtener unas carreteras más seguras y capaces de albergar una mayor densidad de tráfico. Las operaciones que podrían ser automatizadas en las carreteras del futuro, así como las relaciones entre ellas, han sido pormenorizadamente descritas en Zhang et al. (1994) y Varaiya (1993). En estos artículos se proponen diferentes estructuras jerárquicas de control formadas por varios niveles de automatización y se hace una relación detallada de las operaciones incluidas en cada una de las estructuras. Por ejemplo, para una maniobra de cambio de carril, un sistema central de planificación es responsable de desarrollar y coordinar una maniobra segura y que tenga en cuenta el tiempo disponible, en función de los otros vehículos presentes en la vía, y de dar las órdenes para ejecutar los movimientos apropiados. Posteriormente, otro sistema de regulación a bordo de cada uno de los vehículos se encarga de ejecutar las órdenes dictadas por la capa anterior. Aunque una automatización tan amplia como la descrita en estos artículos resulta un tanto visionaria desde el punto de vista práctico, y requeriría nuevos elementos en las infraestructuras y en los vehículos, las claras definiciones de cada operación y las cuestiones relacionadas constituyeron una ayuda para el avance del estado del arte en la automatización parcial de los vehículos. Este estudio sirvió tanto para allanar el camino hacia la meta final de la automatización de las autovías como para aplicaciones a corto plazo.

En este mismo contexto de los sistemas de AHS, se llevó a cabo una considerable investigación en control y en tecnología de sensores para gobernar un pelotón de vehículos simultáneamente. La idea de fondo consiste en formar una hilera de vehículos que se siguen unos a otros a corta distancia, a velocidades propias de la circulación en autovía, sin

riesgo de choques o de interferencia con otros pelotones de vehículos. Se ha demostrado que, disminuyendo el espacio entre vehículos, se mejora considerablemente la capacidad de las carreteras (Swaroop and Huandra 1998 y Swaroop and Rajagopal 1999). No obstante, un pelotón compuesto por vehículos equiespaciados sólo es estable si se dispone de algún tipo de comunicación entre los distintos vehículos (Swaroop et al. 1994). Esto supone que todos los vehículos en el pelotón estén equipados con algún dispositivo de radio-comunicación. Se han realizado con éxito pruebas en carretera con pelotones de vehículos (Tan et al. 1998) y (Rajamani et al. 2000), y actualmente continúa investigación con pelotones de vehículos con vistas a mejorar la próxima generación de carreteras.

1.2.2. Vehículos Inteligentes

Como hemos visto, en los comienzos de la década de los 90, con el fin de proporcionar al conductor una mayor información, la investigación en Estados Unidos y Europa estaba más orientada hacia los AHS, que necesitan infraestructuras tanto en los vehículos como en las carreteras.

Debido a problemas económicos y a limitaciones prácticas, la investigación a corto plazo se ha desplazado desde los AHS hacia los IVI, y se ha centrado en sistemas de asistencia al conductor. Estos sistemas se pueden instalar en vehículos reales sin necesidad de realizar costosas modificaciones en infraestructuras, y proporcionan al conductor información, alarma y ayuda en algunas operaciones. Algunos sistemas que podemos emplear en los vehículos de hoy en día como los ACC, el control “stop and go”, la alarma de colisión, y sistemas para evitar colisiones, se desarrollaron dentro de esta línea de investigación. Muchos desarrollos realizados en el terreno de la automatización de carreteras también son aplicables en este campo, puesto que los requerimientos de control y sensorización son similares. Sin embargo, no conviene olvidar que existen numerosas diferencias en el planteamiento conceptual, que hacen que la incidencia de cada uno de estos sistemas en la seguridad, capacidad de las carreteras y confort de la conducción pueda ser dispar.

Lu y Hedrick (2004), al estudiar los vehículos equipados con sistemas de control que realizan algunas maniobras de forma completamente automática, o con alguna ayuda del conductor, se refieren a ellos como Autonomous Road Vehicles (ARV), y hacen una revisión de las distintas posibilidades de automatización: control longitudinal, lateral y vertical. A

continuación, se sigue esta clasificación para pasar revista a los distintos sistemas que progresivamente se han investigado para su instalación en el automóvil. En el siguiente apartado se hará una revisión más centrada en los aspectos técnicos del control.

A. Vehículos Totalmente Autónomos.

Los vehículos totalmente autónomos constituyen un apartado singular dentro del estado del arte. Su aplicación práctica, por el momento, no deja de ser un tanto inviable (se necesita mucha confianza en el vehículo y el sistema de control), pero son un banco de ensayos excelente para el desarrollo de controles parciales y de aplicaciones que posteriormente puedan llevarse a los vehículos comerciales.

El primer vehículo autónomo fue construido en Japón en 1977 dentro del proyecto CACS (Tsugawa et al. 1979). El vehículo poseía un sistema de control bastante rudimentario que podía manejar el volante del vehículo durante aproximadamente 50 m, empleaba un motor adosado a la barra de dirección, y estaba basado en visión artificial y control proporcional.

Este mismo grupo de investigación ha continuado su investigación, y ha desarrollado controladores laterales y longitudinales, basados fundamentalmente en un posicionamiento del vehículo mediante GPS y comunicación entre los vehículos (Kato et al. 2002).

En EEUU la mayoría de los primeros proyectos significativos en este campo fueron financiados por el Ministerio de Defensa. En 1986 comenzó el proyecto NavLab de la Carnegie Mellon University, en el que se desarrollaron varios vehículos prototipos capaces de circular de forma autónoma en autopistas con líneas bien marcadas en el asfalto. Entre estos vehículos se encuentran tanto turismos como autobuses, e incluso vehículos todo terreno para realizar expediciones campo a través con fines militares.

El proyecto se centró en el control automático de vehículos en autopistas (AHS). Para la conducción autónoma de un vehículo en carretera, una parte muy importante es el reconocimiento del entorno. En este trabajo se probaron diversos algoritmos para la detección de los bordes de la carretera. Uno de ellos es el denominado ALVINN (Pomerleau 1993 y Pomerleau and Jockey 1996). Está basado en el entrenamiento de una red neuronal cuya entrada es una imagen de baja resolución de 30x32, y cuya salida es directamente la posición que deben tener las ruedas del vehículo

para seguir la carretera, y consiguió conducir el vehículo de prueba Navlab 1 durante 21.2 millas con velocidades de hasta 55 millas/h, en las primeras etapas del proyecto. Posteriormente, se han conseguido trayectos de miles de kilómetros por autopista a velocidades cercanas a los 100 km/h.

Como instrumentación, el sistema funciona con una simple cámara en color de bajo coste, un radar y un ordenador personal corriente. El proyecto Navlab demuestra la viabilidad de resolución de problemas de navegación autónoma reales, con máquinas baratas y conocidas como son los ordenadores personales. La versión Navlab VIII, Figura 1.3, está controlada por un sencillo ordenador portátil. Se han desarrollado hasta once versiones de vehículos dentro del proyecto.



Figura 1.3 Vehículos desarrollados en el proyecto Navlab.

En 1997, se realizó una prueba definitiva del sistema en San Diego, patrocinada por el Nacional Automated Highway System Consortium (un grupo de corporaciones, universidades y agencias gubernamentales) de Estados Unidos. Se llevó a cabo la conducción autónoma de un autobús, con pasajeros durante varios cientos de kilómetros. El autobús, ilustrado en la figura, fue conducido de forma manual hasta la autopista y, a partir de ahí el conductor conmutó el sistema a funcionamiento automático. El conductor sólo debe volver a tomar las riendas de la conducción cuando decida salir de la autopista, en el punto que corresponda.

El sistema ofrece otras prestaciones, igualmente interesantes. Por ejemplo, se mantiene una distancia de seguridad predeterminada con el vehículo precedente, lo cual puede ahorrar vidas y aumentar la eficiencia de conducción en atascos. Por otro lado, se realizan maniobras de cambio de carril, empleando para ello el sistema basado en radar instalado en el vehículo.

Se han llevado a cabo varios intentos, por parte del gobierno norteamericano, para comercializar estos sistemas, especialmente el vehículo automático para autopistas. No obstante, su funcionamiento en entornos urbanos sería muy difícil, al depender de las líneas pintadas en las calles. Por este motivo, la investigación de Navlab se centra ahora mismo en la determinación de trayectorias en distintos entornos.

En Europa, Dickmans ha desarrollado varios prototipos (Dickmanns and Zapp 1986) capaces de guiarse autónomamente, hasta lograr guiar el vehículo en las autopistas alemanas, con tráfico real y alcanzado velocidades de hasta 130 km/h (Dickmanns et al. 1994). Este vehículo incluía un sistema compuesto por 4 cámaras color, tres sensores inerciales, un tacómetro, sensores de ángulo y un sofisticado sistema de proceso.

Finalmente, se han desarrollado dos vehículos más dentro del proyecto PROMETHEUS: VITA_2 de Daimler Benz y VaMP de UBM (Dickmanns 1997). Ambos presentan soluciones de bajo coste, empleando ordenadores personales con procesadores Pentium. Estos vehículos son capaces de “entender” parte del entorno en el que se encuentran, y reaccionar adecuadamente a las necesidades de la situación de forma completamente autónoma. En su desarrollo se han empleado los siguientes elementos:

- Modelo dinámico del movimiento del vehículo, considerando las salidas de los actuadores y los tiempos de retardo.
- Bibliotecas de formas tridimensionales para especificar los tipos de objetos que el sistema de visión puede reconocer.
- Modelos de mapeado de perspectiva.
- Realimentación del error de predicción para estimar el estado de los objetos, tanto en el espacio 3-D como en el tiempo.

Posteriormente, el mismo grupo de trabajo ha diseñado sistemas de control automático de velocidad en cooperación con la compañía de vehículos MAN (MAN 2000). El sistema desarrollado está orientado a camiones o autobuses, y apoya al conductor para conseguir un control de cruceo inteligente de velocidad, y que mantenga la distancia de seguridad apropiada con los vehículos precedentes. El sistema ofrece la posibilidad de funcionar en modo automático, aunque también puede ser anulado por el conductor en cualquier momento. El control lateral permanece bajo el control manual del conductor. El sistema alerta al conductor en caso de situaciones en las que se necesite un frenado manual de mayor fuerza que el que puede realizarse de modo automático.

Este sistema, mejorado con un control lateral que avisa al conductor en el caso de que vaya a salirse del carril a otro que ya esté ocupado, ha sido probado en simuladores de conducción. Forma parte del Copiloto MAN, y está instalado en un autobús de la MAN Nutzfahrzeuge AG.

Durante la década de los 90, en la Universidad de Parma se ha desarrollado el proyecto ARGO (Broggi et al. 1999), en el ámbito del proyecto europeo EUREKA PROMETHEUS. ARGO finalizó en 1994 con una demostración de los resultados conseguidos en París, pero la investigación continuó en la Universidad de Parma financiada por el Consejo de Investigación Nacional Italiano. Se equipó un vehículo comercial Lancia Thema 2000, con los dispositivos necesarios para realizar la conducción autónoma en autopistas. El vehículo dispone de un sistema de adquisición trinocular, empleando como procesador un sencillo PC-Pentium bajo el sistema operativo Linux, junto a un procesador hardware desarrollado específicamente para esta aplicación, denominada PAPRICA.



Figura 1.4 Vehículo autónomo ARGO

Durante la primera semana de Junio en 1998, el vehículo realizó un viaje de demostración alrededor de Italia, el MilleMiglia in Automatico (Bertozzi and Broggi 1998 y Bertozzi et al. 2000), durante el cual el vehículo condujo más de 2000 km de forma autónoma. El sistema tiene tres posibilidades de conducción:

- Manual: el sistema monitoriza la actividad del conductor y emite señales acústicas y ópticas de aviso en caso de situaciones potenciales de peligro.
- Supervisada: el sistema advierte al conductor con señales ópticas y acústicas y, en caso de peligro, toma el control del vehículo.
- Automática: el sistema conduce automáticamente, siguiendo las líneas del carril, y localizando obstáculos en condiciones de tráfico real. Es incluso capaz de realizar cambios de carril.

Debido a las características del sistema de visión artificial utilizado, el vehículo puede conducir de forma autónoma tanto en autopistas como en

algunas carreteras locales, siempre que presenten líneas pintadas sobre el asfalto.

Con posterioridad, la investigación de este equipo ha continuado en el ámbito del reconocimiento del entorno mediante sistemas de visión artificial. Recientemente, ha participado en el desarrollo del vehículo TerraMax (Braid et al. 2006), un vehículo militar, aportando el sistema de visión. El desarrollo de este vehículo se encuentra dentro del objetivo del gobierno norteamericano de que, en el año 2015, un tercio de los vehículos militares no requiera conductor.

Recientemente, en España se ha llevado a cabo el proyecto AUTOPÍA, en el que se han desarrollado varios vehículos autónomos. Inicialmente, se desarrolló un sistema de visión, integrado con GPS, para determinar el posicionamiento del vehículo en carretera, que incorporaba la capacidad de reconocimiento de intersecciones y de carreteras carentes de líneas (Sotelo 2001). La conducción de estos vehículos se realiza mediante controladores borrosos, y un controlador PID para la regulación del motor asociado a la barra del volante, (Naranjo 2005).

Además de estos vehículos, totalmente autónomos, fruto de la investigación se han logrado sistemas de diverso grado de automatización que se describen a continuación. Muchos de ellos nos resultan ya familiares, mientras que otros están dando los primeros pasos de cara a su integración en vehículos comerciales. En muchos casos, se han respetado los nombres en inglés, por encontrarse mucho más asentados que la traducción castellana.

B. Control Longitudinal.

- Cruise Control (CC).

Este sistema el control se encuentra ya muy extendido. En él, el sistema de control del vehículo se limita a regular la velocidad. El conductor debe vigilar la distancia a los vehículos precedentes y, si es necesario, reducir la velocidad o asumir el control manual de la misma.

El sistema no requiere añadir sensores adicionales a los que ya posee el vehículo, como son el sensor de velocidad del motor, sensores de velocidad de las ruedas, etc.

- Adaptive Cruise Control (ACC).

Los sistemas ACC, que pueden detectar la distancia al vehículo precedente y mantener un espaciamiento constante entre ambos (Pham et al. 1997), constituyen una aplicación del control de crucero a los vehículos actuales. El objetivo es descargar al conductor de la tarea pesada que suponen mantener la distancia con el vehículo precedente a velocidades propias de desplazamientos en autopista. Por este motivo, se oferta como un medio de mejora del confort. Al tratarse de una aplicación que puede aumentar el mercado de sus productos, los fabricantes de automóviles están dedicando un gran esfuerzo al desarrollo de sistemas robustos de ACC, e incluso algunos fabricantes han dotado de estos sistemas a sus vehículos de gama alta (Tomizuka et al. 1995). De todas formas, todavía se requiere un mayor estudio sobre cómo afectan estos sistemas a la seguridad en la conducción, y sobre su impacto en el flujo de tráfico.

El sistema ACC puede funcionar de dos maneras: sin vehículo precedente (modo autónomo), y en seguimiento de un vehículo precedente (modo de seguimiento adaptativo). Para que funcione de una forma suave debe haber un proceso de transición entre ambos modos, pero no es necesario que haya comunicación entre los dos vehículos (Lu and Hedrick 2004).

Entre los sensores que requiere se encuentra un radar Doppler que posea buena resolución, un rango de operación entre 4 y 100 m con cono de apertura entre 8° y 16° y un tiempo de refresco inferior a 65 ms para medir la distancia entre vehículos, la velocidad relativa y el ángulo de inclinación para seguimiento multiobjetivo.

Mitsubishi fue el primer fabricante en introducir ACC en su modelo Diamante en 1995. En Europa, Daimler-Chrysler lanzó el Distronic ACC S-Class en el año 2000. Los sistemas ACC no pueden funcionar con velocidades inferiores a 30 km/h y, por tanto, no resultan de utilidad para el tráfico urbano. Sin embargo, en estas situaciones en las que hay que llevar a cabo maniobras tediosas y repetitivas de arranque y frenada, la automatización tiene un interés indudable. Por este motivo, actualmente la investigación relativa al control de velocidad, ha centrado su interés en la unión de los sistemas ACC y los de control “stop and go” (Naranjo et al. 2006).

- Cooperative Adaptive Cruise Control (CACC).

Si se incorpora un sistema de comunicación en el sistema de control que determina la maniobra a desempeñar por el ACC, de tal manera que se

pueda establecer comunicación sobre la marcha (Lee et al. 2001), es posible intercambiar información con los vehículos adyacentes, e incorporarla dentro del sistema de control. De esta manera, se pueden superar los problemas debidos al muestreo y filtrado de la señal del radar. La información más importante que se intercambia entre los vehículos es la velocidad, la aceleración, el ángulo de desplazamiento del pedal del acelerador, la presión del freno, el giro del volante, una variable para el caso de que se produzca fallo, y una variable de identificación de la maniobra. Esta identificación de la maniobra representa la maniobra que el vehículo está tratando de realizar y es comprensible para todos los vehículos implicados. De esta forma, el controlador puede organizar la maniobra de forma más eficiente, y reduciendo la distancia entre vehículos. (Lu et al. 2002) ofrecen un enfoque unificado para el control de sistemas CC, ACC y CACC.

Junto con estos sistemas que incorporan la comunicación con otros vehículos, también se encuentran en desarrollo otros sistemas de control de velocidad que tienen en cuenta la geometría de la carretera (Aparicio et al. 2005). Estos sistemas incorporan una mapa de la carretera lo más preciso posible, un dispositivo de posicionamiento del vehículo y de visión para detectar las líneas de la carretera, y un modelo dinámico del vehículo para determinar la velocidad a la que el vehículo puede circular con seguridad.

- Sistemas “Stop & Go”.

Los sistemas de control “stop and go” constituyen una extensión de los sistemas ACC, que hacen que el vehículo sea capaz de acelerar y frenar automáticamente durante el tránsito en ciudad (Persson et al. 1999 y Venhovens et al. 2000). Mediante este sistema se pretende reducir la carga de trabajo en áreas suburbanas, donde los sistemas ACC prácticamente no tienen aplicación. Al mismo tiempo, los retos que plantean los sistemas de control “stop and go” son también mayores que los ACC, debido al entorno de conducción que es más complicado y a los requisitos para los sensores que son más exigentes.

En estos sistemas se necesita una toma de datos muy precisa, que es posible recibir de varios sensores como pueden ser el radar (Hoess et al. 1995), láser, sistemas de visión (Broggi et al. 1999 y Takahama et al. 2003), o una combinación de estos tres (Labayrade et al. 2003). Recientemente, en una publicación del grupo Volkswagen se indica que “la

próxima generación de sistemas ADC (automatic distance control) + F2S (F2S=follow to stop) será capaz, automáticamente, de detener por completo el vehículo tras el vehículo precedente si es necesario". Otra manera de suministrar los datos al sistema de control del vehículo es mediante una comunicación *wireless*, que puede proporcionar incluso más información que la suministrada por los sensores a bordo (Kato et al. 2002). Este es un planteamiento propio de AHS, en el que se comunican una carretera "inteligente" con un vehículo "inteligente", e intercambian información sobre velocidades máximas en el tramo, cruces e intersecciones próximas, etc.

- Automated Vehicle Following.

Una forma de disminuir el tráfico a lo largo de las carreteras es formar grupos o pelotones de vehículos totalmente automatizados, de forma que la distancia entre ellos pueda reducirse lo más posible. Para ello, los carriles previstos para la circulación de estos "pelotones" se instrumentarían con imanes en el centro del carril. En el programa PATH, anteriormente descrito, se han desarrollado y probado satisfactoriamente varias maniobras. Junto al control longitudinal, este tipo de maniobras involucran también problemas de control lateral. A su vez, los vehículos envueltos en la maniobra deben estar instrumentados con radar, magnetómetros y GPS, de tal manera que sea posible determinar la distancia entre los vehículos con fiabilidad. Junto a esto, la comunicación entre vehículos es indispensable (Lu and Hedrick 2002).

- Sistemas de Alarma y Asistencia al Conductor.

En los últimos años, se han realizado diseños de vehículos mucho más seguros gracias a una mayor preocupación en la seguridad frente a las colisiones. Debido a este esfuerzo, aunque el número de accidentes ha aumentado, se ha logrado que el número de fallecimientos en accidente de carretera haya disminuido. Sin embargo, es necesario continuar el esfuerzo por reducir el número de accidentes que, además de daños personales, producen un elevado número de pérdidas económicas. Por ejemplo, las colisiones traseras ascienden anualmente a 1,8 millones y constituyen el 28 por ciento de todos los accidentes. En 1998, las colisiones traseras produjeron 855.000 lesiones y 1.570 fallecimientos (Statistics Retrieved: www.its.dot.gov/ivi/3DC.html). Las normas de circulación y unos estándares de seguridad más exigentes pueden ayudar a prevenir accidentes pero, puesto que el conductor tiene limitaciones a la hora de reconocer situaciones peligrosas, tomar las decisiones adecuadas y actuar, los

accidentes son prácticamente inevitables. A pesar de ello, muchos accidentes son evitables si se superan algunas de las limitaciones propias del conductor, automatizando algunas de las tareas inherentes a la conducción, en este caso mediante estrategias de seguridad. Esta iniciativa ha fomentado una amplia investigación en sistemas de alarma y prevención de colisiones, que puedan aumentar la seguridad del pasajero y reducir pérdidas, al evitar accidentes que ocurrirían en el caso de que el control fuese únicamente del conductor (Kawai 1994 y Deering and Viano 1994). Muchos de los componentes de estos sistemas son similares a los estudiados en el caso de AHS, y la diferencia fundamental reside en los diferentes planteamientos.

Los Sistemas de Alarma de Colisión (CWS) pueden alertar al conductor en el caso de una colisión inminente. Datos estadísticos sobre accidentes muestran que una parte considerable de los mismos son debidos al retraso del conductor a la hora de reconocer o juzgar una situación peligrosa. En una colisión frontal, por ejemplo, se ha señalado que con medio segundo de tiempo adicional de alarma, el 60 % de las colisiones se podrían evitar, y con un segundo de tiempo adicional la proporción aumenta al 90 % (Woll 1997). Por este motivo, los fabricantes de automóviles desarrollan proyectos de investigación para implementar sistemas CWS, que puedan aumentar la seguridad y, consiguientemente, las ventas de sus vehículos (Barber and Clarke 1998). En los Estados Unidos, los sistemas CWS se encuentran en uso desde hace unos años en flotas de camiones (Woll 1997) y autobuses (Woll 1994), y han demostrado un comportamiento satisfactorio. Sin embargo, al correr los fabricantes con la responsabilidad de sus productos ralentizan el proceso de introducción de CWS en los vehículos de pasajeros. También faltan todavía por resolver algunos problemas, como pueden ser las falsas alarmas del sistema al conductor, para que los CWS se ganen la confianza del consumidor. Una medida más futurista para prevenir las colisiones estaría constituida por un sistema que pudiera detectar la situación de peligro, y asumiera automáticamente el control del vehículo para llevarlo fuera del peligro. Las pautas de control para realizar maniobras de emergencia sencillas se encuentran en un estado de desarrollo aceptable. Sin embargo, antes de que se puedan instalar en vehículos comerciales, es necesario desarrollar sistemas de detección todavía más robustos y, además, se debe tener en cuenta que la actuación de estos sistemas (actuando al margen del conductor) puede conducir el vehículo a situaciones totalmente imprevistas.

C. Control Lateral.

Como se ha visto al hablar de la formación de pelotones de vehículos, parte del control lateral está orientado a maniobras automáticas de cambio de carril y de ayuda para mantener al vehículo en el propio carril. En estas maniobras se deben considerar dos errores:

- El error en distancia entre el centro del parachoques delantero y el centro del carril.
- El error angular con respecto a la tangente de la carretera.

Como instrumentación, estos sistemas necesitan un magnetómetro colocado bajo el parachoques frontal y trasero (o un GPS), que se emplean para evaluar el ángulo de guiñada del vehículo y la velocidad de cambio de este mismo ángulo, con respecto a una línea que describa la carretera (Lu and Tomizuka 2006).

Por otro lado, en lo referente al control lateral del vehículo, se ha realizado un gran desarrollo en los sistemas de control de frenada y de estabilidad del vehículo al describir una trayectoria (Heinzl et al. 2003). Esta metodología se recoge bajo el nombre de Vehicle Dynamics Control (VDC), también conocido como VSC o EPS (Zheng et al. 2006). Los primeros trabajos en este tipo de control fueron presentados por Shibahata et al. (1992), Inagaki et al. (1994), Matsumoto et al. (1992). Es el sistema de seguridad más reciente que se ha introducido en los vehículos comerciales tras el desarrollo del ABS (Antilock Brake System) y el sistema de control de tracción (TCS, Traction Control System). Al mismo tiempo, la posibilidad de desarrollo de esta técnica es debida a la introducción masiva en los vehículos del freno electrohidrodinámico (electrohydraulic brake, EHB), (Gottwick 1999). El objetivo de los sistemas VDC es que el vehículo se comporte de acuerdo con la maniobra trazada por el conductor. Para ello, se recurre a la introducción mediante los frenos de fuerzas en una o varias de las ruedas. Esto crea un par que actúa sobre el ángulo de guiñada del vehículo. La figura recoge la instrumentación empleada en un sistema ESP, que está compuesta por los sensores de velocidad de las ruedas, un sensor de velocidad angular para medir la variación del ángulo de guiñada, un acelerómetro para medir la aceleración lateral, y un sensor (potenciómetro o encóder) para medir el desplazamiento del pedal del acelerador.

Debido a las características del neumático, la fuerza longitudinal que es capaz de proporcionar cuenta sólo con un margen hasta su saturación. Por este motivo, se ha propuesto el denominado “método β ”, para tener en

cuenta como varía el momento de guiñada creado por las fuerzas laterales en los neumáticos en función del ángulo de deriva (Shibahata et al. 1992). Cuando el ángulo de deriva del neumático es pequeño, el momento de guiñada generado por los neumáticos es capaz de reducir el deslizamiento lateral del vehículo; sin embargo, cuando los ángulos de deriva son elevados el momento de guiñada decrece. El decremento del momento autoalineante cuando el ángulo de deriva crece es la causa fundamental de que un vehículo derrape, y un momento de guiñada sobreañadido provocará la recuperación de la estabilidad del vehículo. Por este motivo, Van Zanten et al. (1995) han señalado que es necesario controlar la evolución del ángulo de deslizamiento mediante el control del ángulo de guiñada para mantener la estabilidad del vehículo. En el desarrollo de los controladores del ángulo de guiñada se han ido aplicando las diversas técnicas de control. Tahami et al. (2004) y Boada et al. (2005) proporcionan algunos ejemplos de estas aplicaciones.

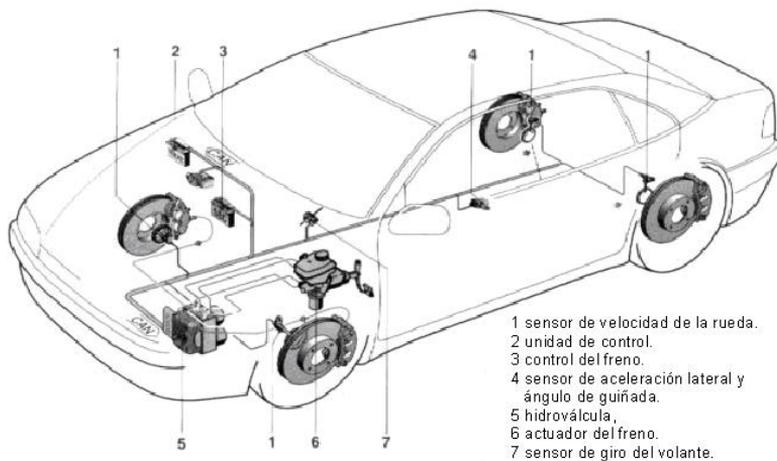


Figura 1.5 Instrumentación de un vehículo con sistema ESP.

Asumiendo las técnicas desarrolladas en el control longitudinal y lateral, algunos autores (Hayakawa et al. 2004, Tai et al. 2004) han propuesto sistemas de conducción automáticos, que son muy cercanos a los vehículos estudiados como vehículos totalmente autónomos.

D. Control vertical.

Los desarrollos en esta área tienen fundamentalmente dos direcciones:

1. Mejorar la calidad del guiado del vehículo.

2. Mejorar la estabilidad del vehículo en carreteras en malas condiciones y/o en situaciones de emergencia como son los sistemas de Frontal Collision Warning and Avoidance.

Estas dos direcciones de investigación son divergentes desde el punto de vista del diseño del control de la suspensión (Sohn et al. 2000). En este sentido, para mejorar el guiado del vehículo en situaciones normales en las que las fuerzas proporcionadas por los neumáticos no han alcanzado la saturación, se debe manipular la suspensión de tal modo que el vehículo mantenga:

- la estabilidad.
- unos valores adecuados de rigidez del muelle y de la amortiguación.
- valores mínimos de la aceleración y su derivada (sobreactuación) en las tres direcciones y los tres ángulos.

Sin embargo, en situaciones de emergencia, se debe procurar que cada rueda proporcione la adherencia necesaria, tanto longitudinal como transversal. La adherencia en cada rueda depende de la fuerza normal sobre la misma, de la rigidez del neumático y del deslizamiento del mismo. No es posible actuar sobre la rigidez del neumático, que depende de la presión del neumático y de las características establecidas por el fabricante. El deslizamiento del neumático puede ser medido en tiempo real (Lu and Hedrick 2001), y empleado para estimar el coeficiente de adherencia con la carretera y, posteriormente, la fuerza que suministra el neumático. Como se puede ver, sólo es posible actuar sobre la fuerza normal del neumático, y de esto se encargan los sistemas de suspensión activa y semiactiva.

1.2.3. Esquemas de Control para la Automatización.

Una de las áreas más estudiadas en la automatización de la conducción es la metodología de control. Una vez que se dispone de información suficiente para determinar la situación del vehículo en la carretera y con respecto a otros vehículos, se necesita un esquema de control, tanto para ayudar al conductor como para controlar autónomamente el vehículo en su totalidad. Normalmente, un controlador de más alto nivel, dotado de un modelo del vehículo, determina los valores de velocidad y aceleración bajo los que el vehículo se mantiene controlado durante el transcurso del movimiento. En los sistemas de asistencia al conductor, las indicaciones de actuación proporcionadas por el controlador se comparan con la acción del

conductor y, si es necesario, se emite una señal de alarma. En sistemas todavía más avanzados, el controlador determina también el movimiento deseado, para que sea ejecutado por otros controladores de menor nivel que controlan el acelerador, los frenos, la dirección, etc. El diseño de los controladores de alto nivel requiere un buen conocimiento del entorno del vehículo. Para diseñar los controladores de bajo nivel se necesita un buen modelo del vehículo. Se ha realizado un esfuerzo teórico y práctico considerable para desarrollar controladores y modelos de vehículos de diferentes niveles de complejidad.

A. Control de Supervisión.

- Control Longitudinal.

Como se ha señalado anteriormente, la mayoría de los estudios se han centrado en control longitudinal de vehículos, porque constituye la base para diferentes aplicaciones como la agrupación de vehículos en pelotones, ACC y sistemas de alarma y prevención de colisión frontal. Mientras que los controladores de más bajo nivel son muy similares para estos distintos problemas, las diferencias en el planteamiento del diseño del control se hacen más evidentes en los sistemas de alto nivel. Ioannou y Raza (1996 y 1997) han presentado un diseño de control de alto nivel (de supervisión), bien estructurado, aplicado al control longitudinal de vehículos y que dispone de diferentes niveles de actuación. Este control de supervisión procesa los inputs que recibe del conductor, la carretera, otros vehículos y los sensores de a bordo, y envía las señales apropiadas a los pedales de acelerador y freno.

Los sistemas ACC y los de agrupación de vehículos, se pueden entender como sistemas de seguimiento de vehículos que tienen en común diversas cuestiones. Sin embargo, en el caso del agrupamiento de vehículos, el objetivo consiste en hacer la distancia entre vehículos lo menor posible para incrementar la capacidad de la carretera, mientras que, en los sistemas ACC, se trata de mantener una distancia de seguridad que descargue al conductor de la tarea de ajuste de la distancia con el vehículo precedente. Estos distintos objetivos se reflejan en el diseño del controlador de supervisión. En un pelotón de vehículos, la aceleración del coche es calculada para asegurar la estabilidad de la hilera, de tal modo que la distancia entre automóviles no disminuya al final del pelotón (Swaroop et al. 1994, Swaroop and Huandra 1998 y Swaroop and Rajagopal 1999). Se ha demostrado que la comunicación entre vehículos (Shladover 1995,

Swaroop and Huandra 1998 y Swaroop and Rajagopal 1999), o el uso de sensores traseros (Zhang et al. 1999), es necesario para garantizar la estabilidad del pelotón. Esta necesidad de comunicación restringe la implementación de estos sistemas en los vehículos de pasajeros. Por este motivo, la aplicación de estos sistemas se ha centrado en las flotas de vehículos pesados (Heavy Duty Vehicles, HDV). En Gehring y Fritz (1997), se explica y se presentan los experimentos realizados con un diseño para el control de un pelotón de camiones. A la hora de aplicar las técnicas de control longitudinal a los HDV, el hincapié reside fundamentalmente en el incremento de la seguridad y la eficiencia de la flota, más que en el aumento de la capacidad de la carretera. Como consecuencia, el controlador de supervisión para los camiones es menos agresivo. Además, algunos problemas del control son propios de vehículos pesados, como pueden ser las variaciones en la carga o la subida de una pendiente (Vahidi et al. 2003). Una buena estimación de la masa y de la pendiente de la carretera puede mejorar la realización de la misión del control de supervisión, al reducir las posibilidades de enfrentarse con comandos de control imposibles de llevar a cabo (Vahidi et al. 2003 y Bae and Gerdes 2000).

En los sistemas ACC, el objetivo se centra en obtener de forma segura un aumento del confort, más que en incrementar la capacidad de la carretera. Por este motivo, es necesario establecer un criterio para determinar la distancia de espaciamiento con el vehículo precedente. Una vez que se ha determinado la distancia de espaciamiento, el controlador de nivel superior calcula la aceleración necesaria para que el vehículo cumpla los requisitos de forma “suave” o “rápida”. Para imitar el comportamiento humano, es posible desarrollar controladores borrosos o neuronales como sugieren Germann e Isermann (1995), Holzmann et al. (1997) y Chakroborty y Kikuchi (1999). A pesar de ello, muchos controles de supervisión se basan en modelos matemáticos más que en el comportamiento humano. Algunos ejemplos son las aplicaciones de métodos de control no lineal como el *sliding mode control* (Gerdes and Hedrick 1997), o el *optimal dynamic back-stepping control* (Lu et al. 2001), para obtener la aceleración necesaria en el nivel de supervisión. Liang y Peng (1998 y 1999) han implementado un diseño empleando técnicas de control óptimo para ajustar algunas de las variables en una maniobra de seguimiento de otro vehículo. El problema consiste en determinar si un comportamiento similar al de un conductor humano es el procedimiento más adecuado posible. Un sistema ACC

entrenado por un conductor real puede ser percibido como más “natural”. Al mismo tiempo, también hay que tener en cuenta que conductores diferentes poseen formas de comportarse distintas. Por este motivo, es prácticamente imposible realizar la maniobra de una forma satisfactoria para todos los conductores, si el controlador imita a un grupo determinado de ellos. Además, la información que proporcionan los sensores es distinta que la que posee el conductor, y que percibe tanto a través de los sentidos como mediante sensores y, por tanto, el mejor procedimiento no tiene por qué ser aquél que imita al conductor humano.

Touran et al. (1999) han estudiado el impacto global de los sistemas ACC en la seguridad de las carreteras. En su estudio, han empleado simulaciones mediante el método de Monte Carlo para determinar la probabilidad de una colisión en una hilera de vehículos. Su conclusión es que los sistemas ACC reducen significativamente la probabilidad de colisión entre el vehículo controlado y el precedente, pero la probabilidad de colisión con el coche controlado de los coches que lo siguen aumenta.

Los criterios de actuación del control de supervisión del ACC afectan secundariamente al flujo de tráfico. Liang y Peng (2000) han estudiado la influencia de los sistemas ACC en una hilera de vehículos compuesta por automóviles automatizados y otros guiados manualmente. Han demostrado que, si los sistemas están adecuadamente diseñados, los vehículos equipados con ACC contribuyen eficazmente a aumentar la velocidad media de un grupo mixto de vehículos, y también reducen los niveles de aceleración. De esta manera, se aumenta el flujo de circulación y ésta es más suave y segura y se disminuye el consumo de combustible. Bose y Ioannou (2001) ponen de relieve que se produce hasta un 60 % de disminución en la contaminación del aire si el 10 % de los vehículos están equipados con ACC.

En los sistemas ACC, el conductor debe iniciar su funcionamiento y supervisar el cumplimiento de sus funciones. Los sistemas de prevención de colisiones (Collision Avoidance, CA) y de alarma de colisión (Collision Warning, CW), en cambio, se encuentran siempre activos durante el funcionamiento del vehículo. Los sistemas CA/CW deben supervisar el comportamiento del conductor y, si es necesario, actuar adecuadamente. Esta diferencia fundamental entre los sistemas CA/CW y otros sistemas de control longitudinal, ponen de manifiesto la importancia del factor humano en el diseño de controles de supervisión robustos que sean seguros y efectivos.

La investigación a corto plazo en sistemas CA/CW se centra fundamentalmente en el control longitudinal (frenado de emergencia), o la alarma para evitar colisiones traseras. En general, cuando la distancia al vehículo precedente se hace menor que la distancia de seguridad, se alerta al conductor. En una situación más crítica, se tiene en cuenta la medida de la distancia de frenado para frenar automáticamente el vehículo. Las distancias de alarma y de frenado son los parámetros clave en el diseño de estos sistemas, y se determinan a partir de procedimientos de distinta complejidad, como pueden ser los modelos desarrollados por Honda y Mazda (Seiler et al. 1998). Wilson (Wilson 2001) ha empleado una estrategia basada en distintas superficies que envuelven al vehículo, para establecer el tiempo de frenado y alertar de una colisión trasera. Un buen modelo también debe tener en cuenta las condiciones de la carretera y del conductor. Un sensor o una metodología basada en el modelo del vehículo es necesaria para determinar la fricción en el contacto neumático-suelo. Yi et al. (1999) y Yi y Chung (2001) han propuesto un modelo para la fricción entre neumático y suelo adecuado para el diseño de controladores CA/CW. Seiler et al. (1998) han propuesto versiones modificadas de los algoritmos de Honda y Mazda, que escalan las distancias críticas a partir de la fricción en el contacto. También se ha tenido en cuenta la posibilidad de que el conductor pueda introducir las distancias de alarma o de frenado de acuerdo con sus preferencias.

- Control Lateral.

El control lateral de vehículos para mantener la trayectoria en el carril, realizar maniobras coordinadas de cambio de carril, o de prevención de colisiones, es una tarea de control más difícil, puesto que requiere un entorno bidimensional más sofisticado. Los sistemas de ayuda al conductor para mantener el carril ya están siendo comercializados. La investigación para automatizar maniobras laterales todavía debe recorrer un largo camino antes de que los métodos sean suficientemente robustos con un criterio de seguridad. La manipulación automática de la dirección sólo es posible de implementar empleando sensores en el interior del vehículo. Una combinación de sistemas de reconocimiento de imagen, sensores inerciales y DGPS hace posible, hoy en día, la definición de una trayectoria susceptible de ser seguida por el vehículo (Raharijaona et al. 2006). La cinemática de una maniobra lateral es determinada por el controlador de supervisión (Rajamani et al. 2000 b y Swaroop and Yoon 1999). Jula et al. (2000) han estudiado la cinemática de una maniobra de cambio de carril, y

las condiciones en las que los accidentes relacionados con esta maniobra se pueden evitar. Eskandarian y Thiriez (1998) han propuesto un control basado en una red neuronal, que determina la trayectoria a seguir por el vehículo en una maniobra de prevención de colisión. Todos los sistemas propuestos tienen valor, y algunos han resultado exitosos en condiciones experimentales (Rajamani et al. 2000 b), pero la complejidad de una situación real requiere sistemas redundantes y verificaciones lógicas para evitar situaciones imprevistas.

B. Controladores a nivel de vehículo.

Para diseñar controladores de alto nivel sólo es relevante la cinemática global del vehículo, pero para el diseño de controladores a nivel del vehículo se necesita un buen modelo de la dinámica vehicular.

Las aplicaciones de control longitudinal requieren un modelo del motor, la transmisión, los neumáticos y el sistema de frenos. El control lateral también precisa un modelo de dirección. Los modelos realistas de estos sistemas son altamente no lineales y difíciles de resolver, especialmente en tiempo real. Por este motivo, se han propuesto pocos modelos diferentes para el diseño de controladores.

El par motor es una función no lineal dependiente de múltiples parámetros. Los diagramas de par estático que se emplean en muchas aplicaciones de control también son funciones no lineales de la velocidad del motor y del desplazamiento del pedal del acelerador. Unos modelos muy extendidos del comportamiento del motor son los desarrollados por Cho y Hedrick (1989) y Kim y Cho (1997). Swaroop et al. (1994) han resuelto el comportamiento del motor mediante una linealización Input/Output. Gerdes y Hedrick (1997) han empleado control en modo deslizante para modelos similares de motor y freno, obteniendo una buena correlación entre la cinemática de la simulación y la de los experimentos. Un modelo longitudinal más simplificado del vehículo se presenta en Zhang et al. (1999). Se pueden encontrar detalles sobre las técnicas de control del motor en Choi y Hedrick (1995 a y b), que proporciona una comparación de diferentes modelos de motor empleados para el control longitudinal de vehículos. Schiehlen y Fritz (1999) han propuesto y comparado algunos controladores lineales y no lineales para la dinámica del motor. Mayr (1996) ha evitado modelizar el comportamiento del motor y la caja de cambios mediante un modelo simplificado, que relaciona la fuerza en las ruedas con

el desplazamiento del acelerador y la velocidad del vehículo, diseñando un controlador longitudinal basado en este modelo.

Otro reto en control longitudinal lo representan la correcta modelización y el control de los frenos. El comportamiento de los frenos depende de varios parámetros, como la temperatura y el coeficiente de fricción entre el neumático y el suelo, que no son constantes y son difíciles de modelizar. Un ejemplo de modelo no lineal de freno se puede encontrar en Raza et al. (1997). Este modelo emplea una realimentación para el diseño del controlador, y ha mostrado un buen comportamiento en las pruebas experimentales. Yi et al. (1999) y Yi y Chung (2001) proponen un controlador *sliding mode* para controlar un modelo no lineal de freno para aplicaciones CW/CA. En vehículos pesados se emplea otro mecanismo, el freno- motor, que es adecuado en aplicaciones ACC. En un motor diesel, al activar el freno-motor o el freno-compresor, se inhibe la inyección de fuel y, como resultado, el motor turboalimentado pasa a funcionar como un compresor que absorbe energía del cigüeñal. Es una forma efectiva y económica de detener un vehículo pesado, y es una posibilidad más a la hora de realizar ajustes de velocidad para control longitudinal en sistemas HDV. De todas formas, las no linealidades propias del motor se manifiestan también en el freno por compresión, y se emplean mapas estáticos para calcular el par de frenado. Druzhina et al. (2000, 2002 a y b) presentan varios diseños para sistemas de freno por compresión, para el control longitudinal de camiones de gran tonelaje. Druzhina y Stefanopoulou (2002) presentan resultados satisfactorios del uso combinado de los frenos de servicio y del freno por compresión, en control longitudinal.

La mayoría de los controladores arriba mencionados son controladores de ganancia fija, es decir, no varían en función de cambios que se puedan producir en el vehículo. No obstante, los parámetros característicos de cualquier vehículo varían a lo largo de su vida útil. Alguno de los parámetros del vehículo o de la carretera, como es la resistencia a la rodadura, pueden incluso variar durante un único desplazamiento, dependiendo del tipo de superficie. La sensibilidad a la variación de algunos parámetros es especialmente importante para los vehículos pesados. La masa de uno de estos vehículos puede variar hasta en un 400 % de un trayecto a otro. La pendiente de la carretera es otro fenómeno con influencia en el comportamiento de un HDV. Por estos motivos, resulta conveniente un enfoque adaptativo, en el que los parámetros del modelo del vehículo varíen, con un estimador implícito o explícito para obtener un

control más satisfactorio. Ejemplos de controladores adaptativos en control de vehículos se pueden encontrar en los trabajos de Liubakka et al. (1993), Ioannou y Xu (1994), Oda et al. (1991). Para el control longitudinal de HDV, Yanakiev y Kanellakopoulos (1996 y 1997) han propuesto un controlador adaptativo con adaptación directa de ganancias de un controlador PIQ (Proporcional-integral-cuadrático). Druzhinina et al. (2002 b) han diseñado un control adaptativo empleando una función de Lyapunov. Youcef-Toumi et al. (1992) han propuesto el método de *time-delay control* para el control longitudinal. El controlador *time-delay* observa la respuesta en el pasado del sistema y de la variable de control, para modificar directamente la acción de control, en lugar de ajustar la ganancia del controlador o identificar los parámetros del sistema.

Para el control en maniobras de emergencia lateral o de mantenimiento de carril es necesario controlar la dirección. La modelización de la dirección depende de la interacción entre el neumático y el suelo. Ésta depende a su vez de muchos parámetros (carga sobre la rueda, presión del neumático, estado de la superficie de la calzada y del neumático, etc.) y es difícil de modelizar. Por este motivo, la mayoría de los autores emplean un modelo simplificado para el control de la dirección. Shimakage et al. (2001) han desarrollado un sistema de control del par sobre la dirección para ayuda al conductor en el mantenimiento del carril. El esquema de actuación es el indicado en la Figura 1.6. Proponen un control LQ con un modelo de bicicleta del automóvil, y que se ha probado con éxito en vehículos reales.

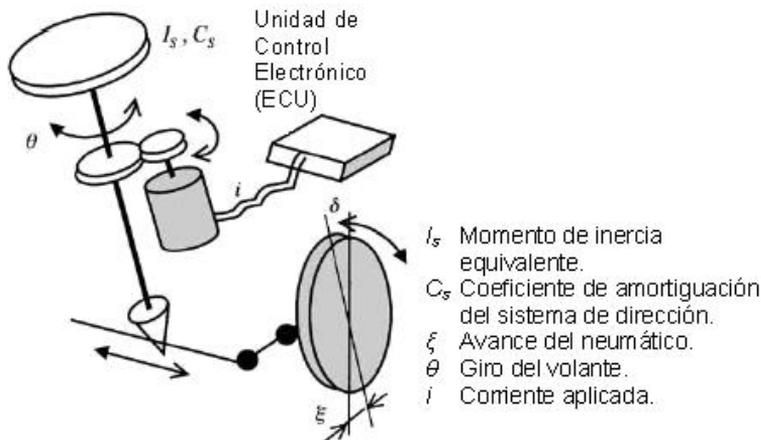


Figura 1.6 Esquema de actuación para el control de la dirección propuesto por Shimakage.

Peng y Tomizuka (1993) han desarrollado un modelo más detallado de la dirección para un vehículo 4x4, y han diseñado un controlador con anticipación (*preview*) para seguimiento de trayectoria. Han mostrado mejores resultados en el seguimiento de la trayectoria cuando se dispone de información sobre el tramo próximo de la carretera a recorrer. Chan y Tan (2001) proponen un control post-choque para estabilizar las trayectorias de los vehículos envueltos en una colisión. Tai et al. (2004) han realizado un modelo lineal de segundo orden para el control lateral de vehículos pesados en carretera. Hebden et al. (2004) han propuesto un control en modo deslizante para el control de la estabilidad en situaciones de pérdida de adherencia a la calzada.

El mayor esfuerzo en los diseños anteriormente mencionados se centra en desarrollar buenos modelos del motor, la transmisión, los frenos y la dirección. Mucha de la información relativa a estos sistemas es propiedad de los fabricantes y, por este motivo, la información para investigadores del mundo académico se encuentra limitada. Sin embargo, en los vehículos actuales, una gran parte de esta información está disponible a través de un interfaz de usuario de la unidad de control. La conexión CANBUS del vehículo proporciona información sobre el par estático del motor, el estado de la transmisión, la velocidad de giro del motor y la velocidad del vehículo. Estas señales se transmiten bajo un estándar que está a disposición del público. Para camiones pesados, el estándar más habitual es el J1939 (SAE 2005), recomendado por la Society of Automotive Engineers (SAE). Al mismo tiempo, la información proporcionada por el CANBUS se puede grabar en tiempo real durante ensayos en carretera, y los datos se pueden emplear para validar modelos de motor, transmisión y frenos. Estos datos pueden ser empleados directamente para el diseño de controladores (Holve et al. 1996).

Al desarrollar modelos para el control longitudinal se acepta generalmente que los neumáticos no deslizan. Esta hipótesis reduce la complejidad de los modelos y los controladores, y es aceptable para aplicaciones como el ACC, en las que se evitan elevados niveles de aceleración o frenado. Sin embargo, en maniobras de emergencia las ruedas seguramente deslizarán. En este contexto, se necesita modelizar correctamente el deslizamiento del neumático para asegurar un comportamiento adecuado del controlador. Este aspecto se ha tenido en cuenta en algunos artículos de investigación pero, de cara al futuro, se necesita una mayor investigación en esta dirección. La interacción neumático-suelo en diferentes entornos también

es importante para el diseño de controladores laterales. Esta información, por ejemplo, sería crucial para un sistema de prevención de colisiones que conduzca el vehículo con seguridad fuera de la zona de peligro.

1.2.4. Modelos Dinámicos.

Como se ha indicado, el control de supervisión necesita un modelo de la dinámica del vehículo para establecer las pautas de comportamiento a seguir. Lugner y Plösch (2004) dan detalle de los modelos de vehículo empleados más habitualmente, así como de sus aplicaciones. Para comprender el comportamiento del vehículo, fijándose especialmente en el comportamiento global y en cuestiones relacionadas, se han desarrollado modelos simples que consideran por separado la dinámica lateral, longitudinal y vertical (Mitschke 1997). A menudo, para emplear este tipo de modelos se recurre a linealizarlos. Para el estudio de la dinámica se suelen aplicar directamente las ecuaciones de Newton-Euler al modelo considerado. La expresión matemática resultante se emplea con frecuencia para el diseño de controladores de estabilidad, como el ESP o suspensiones activas, y otros sistemas de asistencia al conductor.

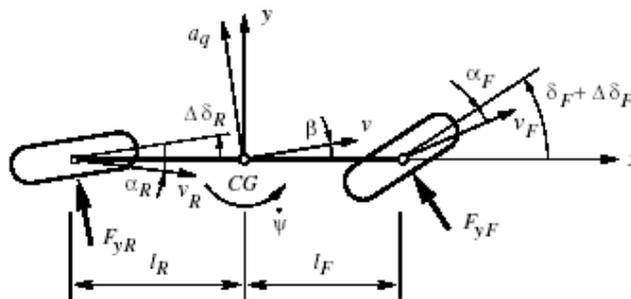


Figura 1.7 Modelo simplificado de vehículo para el estudio de la dinámica lateral.

Para investigaciones centradas en las características de giro del vehículo, la respuesta de la dirección y el análisis de estabilidad, se emplea un modelo de dos ruedas, conocido como el modelo de bicicleta. En este modelo, los dos neumáticos de un eje se reducen a uno solo, situado en el plano central del vehículo como se puede ver en la Figura 1.7, y se desprecia la altura del centro de gravedad del vehículo por encima de la superficie de la carretera, dando como resultado un modelo plano del vehículo. Es frecuente linealizar este modelo, considerando ángulos pequeños y un modelo lineal para las fuerzas laterales de los neumáticos

en estado estacionario. También suele suponerse que la velocidad es constante y, en ese caso, se desprecian las fuerzas longitudinales de los neumáticos con respecto a su influencia en la dinámica lateral.

A pesar de su simplicidad, algunos autores (Freeza et al. 2004) han propuesto el empleo de este modelo para el desarrollo de un conductor virtual. Esta simplificación es muy válida para el empleo de técnicas de control predictivo basado en el modelo.

Para la dinámica longitudinal también se emplea un modelo plano de vehículo que no tiene en cuenta el movimiento independiente de las ruedas con respecto al chasis. Por este motivo, no se incluyen las fuerzas de la suspensión o las debidas a la dinámica relativa del chasis.

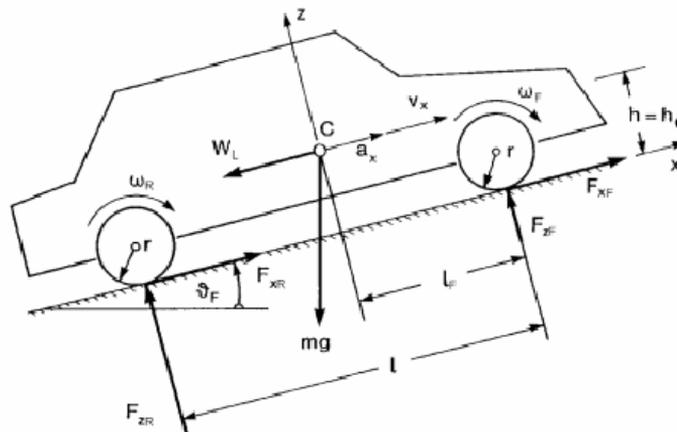


Figura 1.8 Modelo simplificado de vehículo para el estudio de la dinámica longitudinal.

Para determinar las características del vehículo al frenar o acelerar, es necesario combinar este modelo con una representación más o menos sofisticada de la transmisión y del sistema de freno, aunque para una investigación elemental es posible ignorarla (Lugner 1986).

En la Figura 1.9 se puede observar un modelo empleado para el estudio de la dinámica vertical que comprende el chasis y el movimiento relativo de las ruedas con respecto al chasis (suspensión). Este modelo está fundamentalmente orientado al estudio del confort. Para ello, se aplica una excitación correspondiente con un modelo estocástico de la calzada, $\zeta(t)$, que permite obtener la aceleración vertical de respuesta del vehículo, $a_{RMS,z}$. El empleo de un modelo linealizado de la suspensión facilita esta

evaluación. Posteriormente, para tener en cuenta la sensibilidad del conductor a las vibraciones, es necesario emplear unas tablas que relacionen las aceleraciones con el confort.

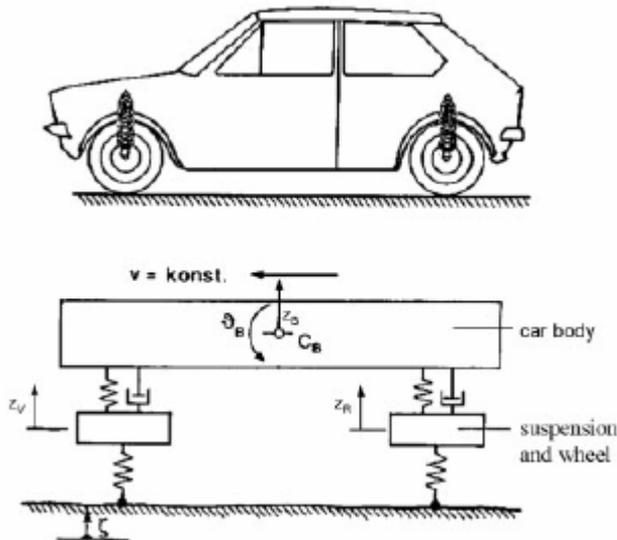


Figura 1.9 Modelo simplificado de vehículo para el estudio de la dinámica vertical.

Desde hace aproximadamente 25 años (Lugner 1982), se han diseñado modelos de vehículos que emplean un tratamiento matricial y códigos numéricos particulares, generalmente escritos en Fortran. Algunos de estos códigos y modelos se han desarrollado como programas específicos de una compañía, con la ventaja de tener unos componentes y estructuras ya ajustados (aunque en ocasiones esto también puede ser un inconveniente). Estos modelos tienen en cuenta el movimiento individual de las ruedas, y suelen emplear curvas precalculadas u obtenidas previamente para determinar el movimiento del centro de la rueda y la inclinación del plano de la misma.

Empleando estos modelos que tienen en cuenta el contacto con el suelo e incluyen una compleja modelización del comportamiento del neumático, es posible realizar el estudio del comportamiento del vehículo durante todo el rango de su comportamiento, incluyendo las no-linealidades como el deslizamiento del vehículo o la realización de trompos. También suelen tener en cuenta una descripción más detallada del comportamiento no lineal del sistema de suspensión (muelles y amortiguadores). Para que el modelo sea congruente, es necesario describir con igual nivel de detalle el

comportamiento de la dirección y de la transmisión del vehículo (características del motor, embrague, diferenciales,...) y sistema de frenos aunque, con frecuencia, el comportamiento de estos sistemas es calculado aparte. Por ejemplo, la dinámica de la dirección no suele tener en cuenta la aceleración del chasis sino sólo su movimiento relativo.

Hoy por hoy, la mayoría de las simulaciones de vehículos se realizan empleando software comercial (ADAMS 2002, RecurDyn 2004, SIMPACK 2003 y DADS 2001). Estos productos ponen especial énfasis en una modelización detallada de los componentes del vehículo y sus conexiones.

La Figura 1.10 muestra parte de las subestructuras y de los componentes empleados en la simulación de un camión mediante un software comercial. Estos programas permiten realizar una modelización previa de las subestructuras y de los componentes con gran detalle. Por ejemplo, es posible realizar un modelo flexible del chasis del camión, teniendo en cuenta sus modos principales de flexión y torsión.

El incremento del número de grados de libertad de los modelos y la necesidad de modelizar las conexiones entre los distintos elementos comporta un aumento considerable en el número de parámetros.

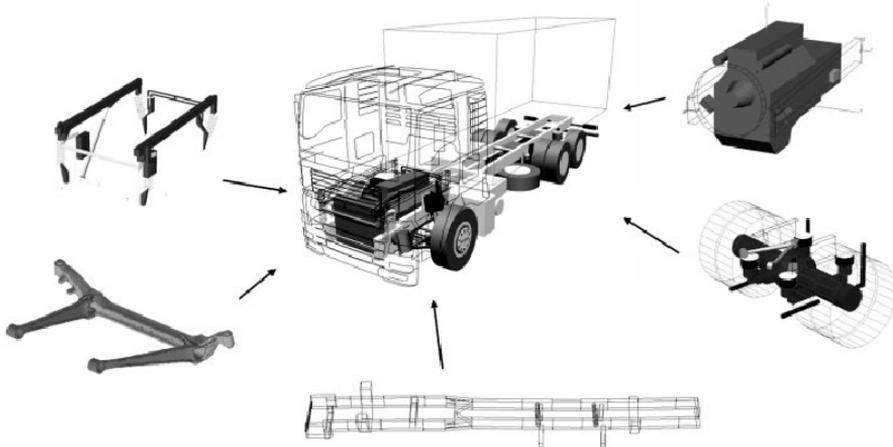


Figura 1.10 Subestructuras del modelo y montaje en Simpack.

Estos programas, conocidos como MBS (Multi-Body Systems), obtienen el movimiento de los elementos del sistema mediante sistemas de ecuaciones diferenciales e imponen las restricciones cinemáticas (uniones, longitudes constantes, etc.) mediante ecuaciones algebraicas. A la hora de plantear estas restricciones se debe escoger entre dos enfoques distintos: una formulación global o una formulación topológica. En las formulaciones

globales las restricciones plantean las características de cada elemento con respecto al conjunto. Las formulaciones topológicas tienen en cuenta la topología del sistema para seguir una estructura en árbol, en la que la conexión entre cada una de las ramas se introduce mediante restricciones. Los sistemas de ecuaciones también pueden incluir descripciones específicas para, por ejemplo, el comportamiento en régimen transitorio del neumático, un subsistema hidráulico, etc.

Por último, la simulación de la dinámica vehicular juega cada vez un papel más importante para el cálculo de tensiones en los elementos del vehículo de cara al diseño. Esto requiere conocer con la mayor exactitud posible el reparto de cargas que se produce en situaciones extremas entre los distintos elementos del sistema. Ello conlleva la necesidad de extender la modelización al entorno, de forma que sea posible representar la superficie de la calzada en contacto con el neumático lo más fielmente posible. Para ello se realizan medidas de la altura de la superficie en distintos puntos y posteriormente se realiza una interpolación (Mitschke 1997).

1.2.5. Tiempo Real.

Muchos de los modelos descritos en el apartado anterior son muy útiles en el diseño de vehículos, pero no en aquellas aplicaciones que requieren que el programa proporcione resultados en tiempo real. Sin embargo, en la investigación sobre vehículos inteligentes es esencial disponer de un modelo de la dinámica del vehículo capaz de ejecutarse en tiempo real. La simulación en tiempo real es el punto de encuentro entre el mundo virtual (p.ej. el modelo multibody de un vehículo) y el mundo real (p. ej. una unidad de control electrónico, ECU). Sus aplicaciones típicas son simuladores de conducción (human-in-the-loop), aplicaciones en plataformas de pruebas (model-in-the-loop), o pruebas de unidades de control electrónico (hardware-in-the-loop).

Como se ha visto en las páginas anteriores, los vehículos actuales ayudan al conductor (u operador en el caso de máquinas industriales), mediante un número cada vez mayor de unidades de control electrónico como pueden ser los ABS o ESP, y muchos otros. Para diseñar y verificar los correspondientes subsistemas de control, es necesario realizar numerosas pruebas con el vehículo. Sin embargo, las pruebas con un vehículo real son costosas en tiempo y dinero (requieren disponer de un

prototipo real que es posible que necesite ser modificado repetidamente hasta llegar al diseño final) y, algunas veces, pueden ser peligrosas.

Las aplicaciones en el campo de la seguridad activa y pasiva del automóvil son también muy sugerentes. Es posible emplear un modelo multicuerpo como observador de estados, de tal manera que, junto con información de otros sensores (visión artificial, GPS, sensores inerciales, etc.), se detecten situaciones de riesgo. En ese caso, el sistema se encargaría de alertar al conductor, o bien, asumir el guiado del vehículo para evitar una posible colisión o minimizar los daños si esta fuese inevitable. Asociado a esto, se encuentra la problemática del diseño de trayectorias (Bertolazzi et al. 2007) y de los algoritmos de control para su seguimiento (Antos and Ambrosio 2004 y Hattori et al. 2006). En este caso, es necesario que el sistema resuelva la dinámica del vehículo adelantándose incluso al tiempo real.

Junto a esto, las ECUs determinan cada vez más el comportamiento del vehículo teniendo en cuenta las reacciones del conductor, las señales de los sensores y otros inputs, como puede ser la información del entorno (Arnold et al. 2007). Por este motivo, disponer de un simulador de la dinámica del vehículo en el que probar los sistemas de control y la influencia de las ECUs en el comportamiento del mismo resulta esencial, al suponer un ahorro de tiempo y de dinero.

Con frecuencia, se presenta la situación de que una ECU se encuentra ya fabricada, pero no se ha fabricado el vehículo. Es posible emplear entonces, como prototipo virtual, un modelo multicuerpo que represente el comportamiento mecánico del vehículo y suministre a la ECU la información que necesita. Para verificar el comportamiento del vehículo completo es necesario comunicar el modelo virtual con el hardware de la ECU. De esta forma, nos encontramos en un caso de simulación con hardware-in-the-loop (HIL). La respuesta de la unidad de control se realiza como lo haría en el vehículo real, e influye sobre la dinámica del vehículo. La exigencia de tiempo real implica que las ecuaciones del movimiento deben ser generadas y resueltas numéricamente con suficiente rapidez y ajustadas en el tiempo, de tal manera que la actuación de la unidad de control sea tenida en cuenta en el momento oportuno. Esto quiere decir que, en un ciclo de simulación, deben plantearse al menos una vez las ecuaciones diferenciales de la dinámica y debe realizarse su integración.

Como se ha visto, otra de las aplicaciones que requieren tiempo real son los simuladores de conducción (Heydinger et al. 2002, Reymond et al. 2000, Shiiba and Suda 2007). En ellos se requiere un modelo que tenga en cuenta los inputs del conductor (actuación sobre volante y pedales), y devuelva la reacción del vehículo. En los simuladores más sofisticados, la reacción del vehículo es percibida por el conductor mediante su representación a través de una plataforma que reproduce las aceleraciones que sufre el vehículo en las direcciones de los tres ejes y alrededor de ellos, y la información visual. Por este motivo, durante mucho tiempo se emplearon modelos simplificados de vehículos, como el de la bicicleta que se ha expuesto anteriormente. Hoy por hoy, se suele recurrir a algún software comercial, aunque éstos, en muchas ocasiones, emplean tablas para resolver el carácter no-lineal de algunos elementos, como pueden ser las suspensiones.

En la simulación de la dinámica del vehículo son corrientes pasos de tiempo de simulación entre 0,5 y 2 ms. Si se introducen en la simulación actuadores y sensores, como en el caso de las ECUs, es posible que se necesiten pasos de tiempo aún más pequeños. Actualmente, los integradores empleados en la resolución en tiempo real de sistemas multicuerpo suelen ser integradores explícitos o semi-implícitos de paso fijo. De hecho, es frecuente recurrir a un esquema de integración simple como puede el de Euler explícito. Esto conlleva que la dinámica del modelo multicuerpo debe ser descrita como un conjunto de ecuaciones diferenciales ordinarias no *stiff*. Sin embargo, en ocasiones, p. ej. al modelizar suspensiones, sistemas de dirección y/o *bushings*, estos elementos conducen a ecuaciones diferenciales de tipo *stiff*. Para resolver esta dificultad, los procedimientos más comunes son los siguientes:

- Escribir modelos multicuerpo particulares para cada nuevo prototipo virtual, empleando modelos simplificados de las suspensiones, la dirección y el chasis.
- Emplear tablas para recoger el comportamiento cinemático y elastocinemático de estos elementos.
- Recurrir a formulaciones adaptadas a la topología y estructura de cada prototipo.
- No tener en cuenta en la simulación frecuencias superiores a 2 Hz.

Como consecuencia, hoy por hoy los modelos multicuerpo son todavía muy simplificados y están ligados a un prototipo, sin poseer un carácter generalista. Debido a esto, ligeros cambios de diseño en el prototipo obligan con frecuencia a esfuerzos elevados para su implementación en el modelo virtual. Existe además una compatibilidad muy limitada de códigos con otros ya existentes de otras aplicaciones.

Un campo de estudio en la creación del modelo multicuerpo es la introducción de elementos flexibles y la correcta representación del comportamiento, altamente no lineal, de los bushings. De forma sintética, los campos que permanecen todavía abiertos en el terreno de los modelos multicuerpo son los siguientes:

- La generación automática de modelos que funcionen en tiempo real a partir de modelos multicuerpo más complejos, pero que respeten la topología, la parametrización, el almacenamiento de datos y las subestructuras.
- La representación de elementos altamente no-lineales evitando el uso de tablas.
- Incluir elementos flexibles respetando la resolución en tiempo-real.
- Incluir efectos dinámicos de rango superior a los 25 Hz.

1.3. Objetivos.

A partir de lo anteriormente expuesto, se deduce que la dinámica multicuerpo en tiempo real puede ser de gran ayuda para la simulación y el control de automóviles, tanto más en la medida que se vayan superando los distintos retos a los que se enfrenta dicha disciplina para su aplicación en este campo.

El objetivo de esta tesis es demostrar que se puede utilizar una formulación de dinámica multicuerpo que permita realizar modelos detallados de automóviles, y que sea a la vez suficientemente eficiente como para ser utilizada en aplicaciones de tiempo real (human y/o hardware-in-the-loop), y suficientemente robusta como para abordar la simulación de maniobras exigentes. Dicho objetivo general se concretará en los siguientes objetivos particulares:

- Construcción de un prototipo de automóvil que sea accesible para la instalación de sensores y actuadores.

- Desarrollo de un programa de computador que resuelva, en tiempo real, la dinámica del modelo del prototipo y de todos sus subsistemas con la mayor precisión posible.
- Dotar al programa de un interfaz que permita la conducción del modelo por un conductor real. Esto es equivalente al desarrollo de un simulador de conducción.
- Generación de un módulo de control que permita el diseño de controladores para la conducción automática, total o parcial, del automóvil.
- Desarrollo de unos controladores que permitan al vehículo realizar unas maniobras concretas, verificando la aptitud de la herramienta para el diseño de controladores, y comparando los resultados con los obtenidos por un conductor real al llevar a cabo esas mismas maniobras.
- Evaluación de la eficiencia y robustez de la formulación. Para ello, se simularán las siguientes situaciones:
 1. Un entorno con obstáculos que permita realizar maniobras violentas, que comporten la aparición de fuerzas elevadas para dificultar la integración.
 2. Control parcial, es decir, situaciones de actuación simultánea por parte de conductor real y controlador, bien sea manejando todo el vehículo o algún subsistema.
 3. Un entorno lo más realista posible que permita la comparación en tiempo real entre los vehículos controlados y los conducidos por un conductor. De esta manera, será posible evaluar la eficiencia del modelo: se tratará de ejecutar simultáneamente, en un único PC de características convencionales, tanto los modelos de los vehículos como la representación del entorno y la comunicación con los diversos periféricos.

1.4. Contenido de la Tesis.

A partir de los objetivos citados en el apartado anterior, la presente tesis describe el prototipo real que se ha fabricado en el Laboratorio de Ingeniería Mecánica, su modelización y el empleo del modelo en una herramienta orientada al diseño de algoritmos de control.

El Capítulo 2 recoge los datos físicos del prototipo real, especialmente de los elementos más relevantes de cara a la simulación, como pueden ser el chasis, el grupo motor (puesto que habrá que modelar tanto la fuerza motriz como el comportamiento del cambio de marchas), el sistema de frenos, la suspensión y la dirección.

En el Capítulo 3 se describe la modelización del vehículo, así como las ecuaciones del movimiento para el sistema multicuerpo y su integración mediante una metodología desarrollada en el Laboratorio: una formulación aumentada de Lagrange de índice 3, con proyección de velocidades y aceleraciones. Seguidamente, se expone la modelización de las fuerzas de inercia del chasis y el motor, y la fuerza en las suspensiones. Se ha procurado detallar con la mayor hondura posible la caracterización y la introducción en el modelo de las fuerzas derivadas del contacto neumático-suelo, así como de las de aceleración y de frenado. También se describe el módulo de colisiones con el entorno y con otros vehículos. Finalmente, se explica la introducción de las fuerzas aerodinámicas en el modelo.

A partir del modelo del vehículo se ha desarrollado un simulador de conducción que sirva tanto para la interacción de un conductor real con el modelo virtual, como para el desarrollo y evaluación de controladores. Tras describir la organización del simulador, en el Capítulo 4 se explica el procedimiento para asegurar su funcionamiento en tiempo real. A continuación, se relata la interacción con el conductor: los inputs y outputs del simulador. Se dedica particular atención al entorno de simulación, tanto al procedimiento seguido para la creación de entornos, como al algoritmo de detección del contacto del vehículo con el entorno.

El Capítulo 5 se podría considerar como un preámbulo a la aplicación del modelo del vehículo al diseño de controladores. Se describe brevemente el comportamiento de la Lógica Borrosa (Fuzzy Logic) y se justifica su empleo. A continuación, se analizan las distintas posibilidades de conexión entre el código del modelo y Matlab, puesto que esta aplicación comercial dispone de numerosas herramientas de gran utilidad para el control. Finalmente, el capítulo detalla el método que se ha desarrollado para el empleo de los controladores en aplicaciones que requieren tiempo real, y la posibilidad de realizar control parcial (que no requiera la conducción total del vehículo por parte del controlador, sino únicamente el control de algún subsistema) en el simulador.

En el Capítulo 6 se emplea el modelo como herramienta para el desarrollo de controladores. Se describen dos maniobras que el vehículo debe realizar autónomamente, y la estrategia con que se han diseñado los controladores. Se analiza el comportamiento del vehículo al efectuar las maniobras, y se compara con los resultados obtenidos por un conductor real.

Como se ha visto en el apartado anterior, uno de los objetivos de la tesis es evaluar la eficiencia y robustez de la formulación en el contexto de las aplicaciones de simulación y control de automóviles. Este tema es tratado en el Capítulo 7. En él se describen los entornos que se han diseñado para tal propósito. Uno de ellos, orientado a la evaluación de la robustez, contiene obstáculos en los que el vehículo se enfrenta con la integración de fuerzas elevadas en maniobras exigentes. Para evaluar la eficiencia, se ha elaborado un entorno en el que un vehículo guiado por un conductor real coexiste con otros dos guiados por controladores.

El Capítulo 8 recoge el trabajo en el ámbito experimental que se ha llevado a cabo durante la tesis. Tras describir la instrumentación del prototipo, se detalla el algoritmo desarrollado para la determinación de la posición, velocidad y aceleración del vehículo, y se propone un algoritmo para la actuación sobre el vehículo orientado a la conducción autónoma. En el apartado instrumental se ha tratado de seguir un paralelismo con los inputs y outputs del modelo virtual que se han tenido en cuenta, en el Capítulo 6, para el diseño de los controladores.

En el Capítulo 9 se exponen las conclusiones y aportaciones de la tesis, así como futuras líneas de investigación que ésta deja abiertas.

2.

El Prototipo Real

A la hora de realizar un prototipo real, surge una primera disyuntiva: emplear un vehículo comercial o construir un vehículo nuevo. La principal función del prototipo es facilitar la realización de las pruebas necesarias para la investigación. Desde este punto de vista, los requerimientos del prototipo son un bajo coste y flexibilidad de uso, de forma que pueda instrumentarse con facilidad, pudiendo instalar los sensores oportunos en cada ocasión sin encontrar limitaciones de accesibilidad, espacio, y poder realizar modificaciones estructurales con facilidad. Estas características son más fáciles de reunir en un prototipo nuevo que en un vehículo comercial. Por otro lado, el diseño y desarrollo de un vehículo automóvil constituía un trabajo de elevada carga didáctica en el que poder implicar a los alumnos de Ingeniería Industrial. Fueron ellos, en gran medida, quienes fabricaron el prototipo, diseñando alguna de sus partes, como el chasis, y adaptando el material conveniente de los desguaces. Se puede decir que todo ello constituyó una auténtica labor de artesanía. A continuación se procede a su descripción.

El prototipo se encuentra en el Edificio de Talleres Tecnológicos del Campus de Esteiro (Ferrol) de la Universidad de La Coruña. El chasis del vehículo ha sido fabricado mediante soldadura de tubos de acero, mientras

que los distintos componentes (motor, dirección, suspensiones, ruedas, etc.) se han recogido del desguace y adaptado convenientemente.



Figura 2.1 Prototipo experimental.

Las estructuras y mecanismos que conforman la parte mecánica del prototipo se pueden agrupar por su función:

1. Chasis.
2. Grupo motor.
 - a. Bloque motor.
 - b. Carburador.
 - c. Sistema de refrigeración.
 - d. Caja de cambios automática y puente trasero (par cónico y diferencial).
3. Sistema de frenos: pinzas y discos que giran con cada rueda.
4. Suspensión: neumáticos, resortes y amortiguadores.
5. Dirección.

2.1. Chasis.

El chasis es la estructura principal, y como se aprecia en la Figura 2.1, es una estructura de tubos de acero soldados. Se ha fabricado básicamente

con tubería de acero DIN 2440. Su construcción se ha llevado a cabo en el taller de calderería especializado INTAF S.L., que amablemente se prestó a colaborar en este trabajo.

El bastidor constituye la base del chasis, en el que se sujetan los distintos mecanismos. El chasis está sometido a continuos esfuerzos de flexión durante los procesos de aceleración y deceleración del vehículo. También debe soportar esfuerzos de torsión cuando el vehículo circula por carreteras de firme irregular o en las curvas.

El bastidor está formado por dos largueros en dirección longitudinal, curvados de forma que se acercan en la parte delantera del coche. Entre estos dos largueros se sitúan varios travesaños, realizados con vigas de secciones especiales, y distribuidos para soportar el apoyo de los distintos elementos, como el motor en la parte trasera, o el asiento y el conductor en la parte delantera. En esta parte del coche, y para conferir indeformabilidad al bastidor, se sitúan dos tirantes en cruz.

Dos tubos longitudinales unidos mediante tirantes transversales conforman el habitáculo del prototipo. La parte delantera del mismo, en la que se sitúa el conductor, y la trasera, que aloja al motor, quedan separadas por una cruz formada por dos tirantes que rigidizan todo el conjunto.

Las características de los tubos empleados aparecen reflejadas en la Tabla 2.1.

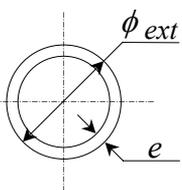
Sección	Diámetro exterior (ϕ_{ext})		Espesor (e)	Peso	
	Pulgadas	mm			
Tubo de acero			mm	Kg/m	
	1	1/2	21.3	2.65	1.22
	2	3/4	26.9	2.65	1.58
	3	1	33.7	3.25	2.44
	4	1 1/4	42.4	3.25	3.14

Tabla 2.1 Secciones principales.

Otras secciones, denominadas especiales, son las recogidas en la Tabla 2.2. Para situar los diversos perfiles en el chasis se aplica el código numérico y de colores indicado en la Figura 2.2.

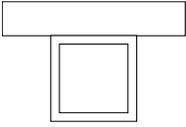
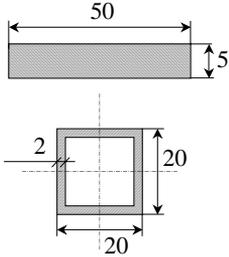
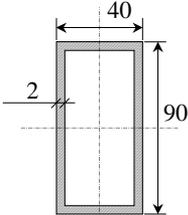
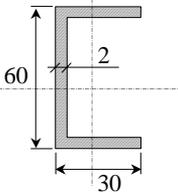
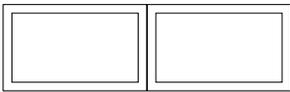
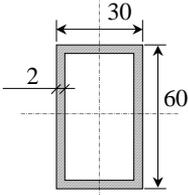
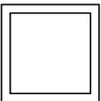
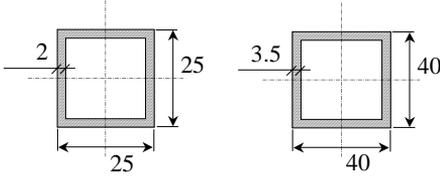
Secciones especiales		Elementos (Dimensiones en mm)	Descripción
5			Pletina de acero soldada a un tubo cuadrado de dimensiones indicadas
6			Tubo rectangular laminado en frío
7			Perfil abierto laminado en frío
8			Sección formada por soldadura de 2 tubos rectangulares de dimensiones indicadas
9, 10			Tubo cuadrado laminado en frío

Tabla 2.2 Secciones especiales.

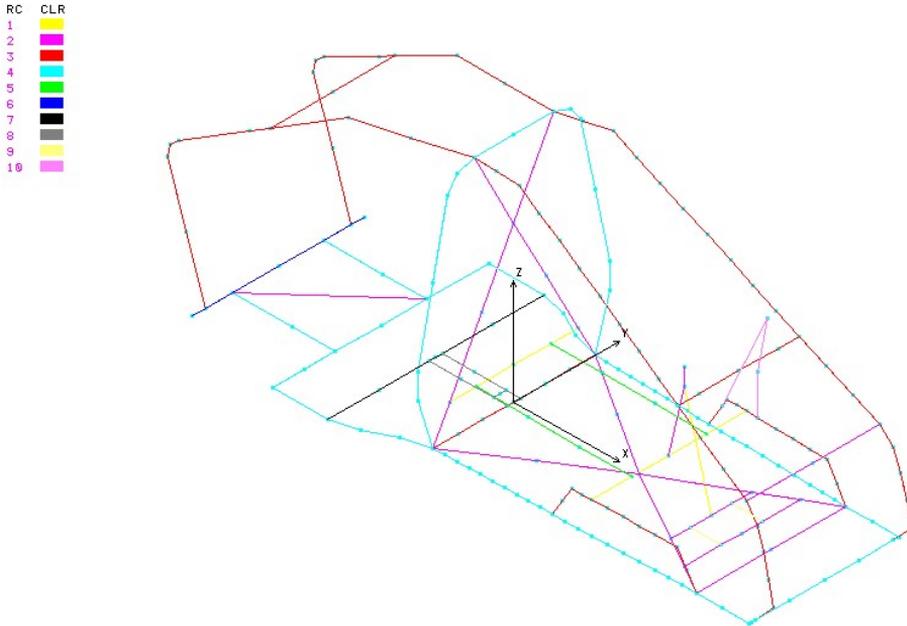


Figura 2.2 Modelo del chasis en elementos finitos.

Las propiedades del material son: densidad, $\rho = 7850 \text{ Kg/m}^3$, módulo de elasticidad, $E = 2.1 \cdot 10^{10} \text{ N/m}^2$, y coeficiente de Poisson, $\nu = 0.30$.

2.2. Grupo Motor.

El motor a montar en el prototipo se seleccionó con la vista puesta en el accionamiento automático. Manejar un motor de cambio automático, en el que incluso es posible detener totalmente el vehículo sin necesidad de alterar la posición de la palanca de cambios, simplifica el sistema de actuadores a instalar en el vehículo. Por este motivo, el motor seleccionado fue el de un Chrysler 150 SX, de cambio automático, con las siguientes características:

Características	Motor Chrysler Referencia: J2-1592
Ciclo	4 tiempos
Orden de encendido	1-3-4-2
Cilindrada (c.c.)	1592
Carrera (mm)	78
Diámetro (mm)	80,6
Nº de cilindros	4

Disposición	En línea
Relación de compresión	9,35:1
Potencia máxima	88 CV a 5400 r.p.m.
Par máximo	14,2 m·kg a 3000 r.p.m.
Combustible	Gasolina súper
Caja de cambios	Automática de 5 velocidades

Tabla 2.3 Características del motor.

El grupo motor está formado básicamente por los siguientes sistemas: bloque motor, carburador y sistema de refrigeración; en nuestro prototipo, también lleva incorporada la caja de cambios automática y el puente trasero (grupo piñón-corona y diferencial), que van acoplados.



Figura 2.1 Motor completo instalado en el vehículo.

El motor se monta en la parte trasera del prototipo y se sujeta al bastidor en los tres puntos de sujeción originales. Estos puntos de sujeción están situados en ambos laterales del motor y en la parte delantera. El motor se ha fijado al chasis mediante tubos de acero y tacos de caucho (silentblocs) que reducen la transmisión de vibraciones. Las escuadras de los tubos laterales están reforzadas mediante nervios de chapa de acero de 4 mm. La barra de sujeción delantera absorbe gran parte del movimiento de cabeceo del motor.

La parte trasera del motor queda libre dejando espacio al colector y al tubo de escape. Además, en la parte trasera se sitúan los palieres que conectan el diferencial con las ruedas motrices.

Sobre este motor se realizaron una serie de modificaciones encaminadas a adaptarlo al prototipo, y a posibilitar y simplificar los ensayos necesarios. Por ejemplo, el carburador original se cambió por otro carburador Weber, modelo DCA-120, por tener un accionamiento más sencillo mediante válvula de mariposa.

El sistema de refrigeración del motor también sufrió varias modificaciones. Se desmontó el sistema de calefacción y el radiador. Debido a que el automóvil no va a circular a velocidades elevadas, el sistema de refrigeración resulta poco eficiente. Para mejorarlo, se optó por trasladar el radiador del sistema a la parte frontal del vehículo e instalarlo junto con un ventilador eléctrico, protegido frente a los golpes. También se eliminó la válvula termostática de apertura y cierre del circuito. Esta válvula tenía como misión mantener cortado el flujo del líquido de refrigeración en el motor para facilitar que éste alcanzase la temperatura de funcionamiento. Una vez alcanzada, la válvula se abría dando paso al refrigerante, con la finalidad de mantener la temperatura en el rango adecuado.

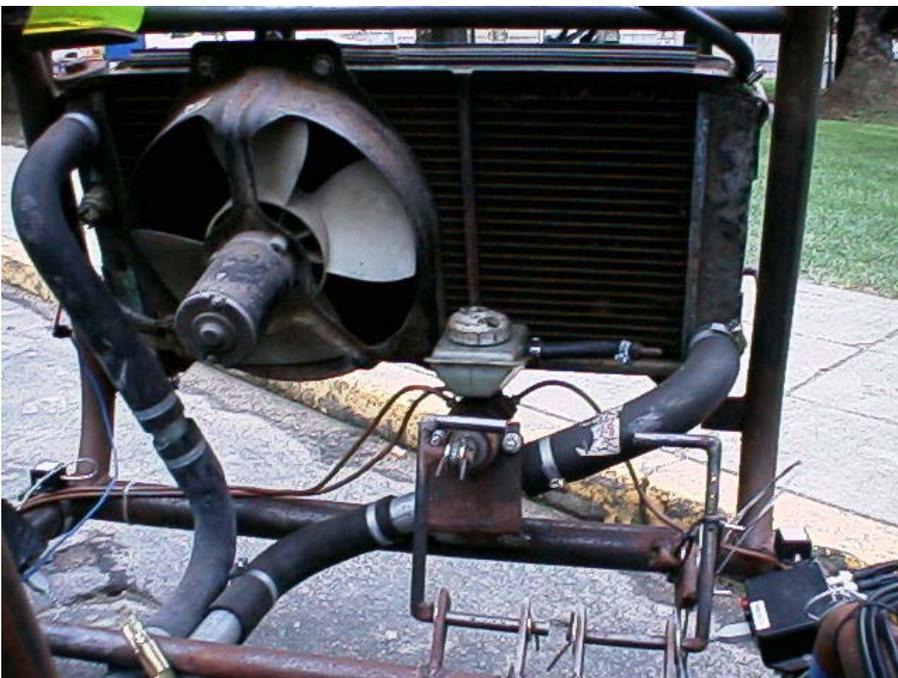


Figura 2.2 Fotografía del radiador, electroventilador y conexiones.

Se realizaron nuevamente todas las canalizaciones de líquido refrigerante. Se efectuó una canalización principal para llevar el líquido refrigerante desde la parte trasera del vehículo, donde está instalado el motor, hasta la parte delantera del mismo, donde está instalado el radiador (Figura 2.2). Se empleó tubo metálico con protección galvánica, de 30 mm de diámetro. Por último, se instaló un vaso de expansión de mayor capacidad para poder satisfacer las necesidades de refrigerante, debido a las mayores variaciones de presión y temperatura en el sistema.

2.3. Masas, Centros de Gravedad e Inercias.

La masa de cada elemento del vehículo que se han empleado en la simulación se muestra en la Tabla 2.4. Todas las masas se obtuvieron por pesando cada uno de los elementos en básculas comerciales. Para obtener el peso del vehículo completo, se situó una báscula bajo cada una de las ruedas. Sin embargo, el peso del motor se dedujo suspendiéndolo de un resorte del que previamente se había calculado su constante elástica colgando de él una pesa de 50 Kg. En el caso del chasis, se realizó una representación en I-deas que permitió hacer una estimación de su peso.

Descripción del elemento	Masa (Kg)
Chasis	107.00
Grupo motor	180.00
Rueda delantera	14.00
Rueda trasera	17.50
Amortiguador delantero	7.00
Amortiguador trasero	6.00
Palier corto	4.00
Palier largo	6.00
Mangueta delantera	8.00
Balancín delantero	2.00
Mangueta trasera	9.50
Balancín trasero	3.00

Tabla 2.4 Masa de cada elemento del vehículo automóvil.

El vehículo incluye dos elementos principales: el chasis y el grupo motor. Estos elementos se representan en la Figura 2.3 y en la Figura 2.4,

respectivamente. En ellas también se indica el sistema de ejes local asociado a cada elemento.

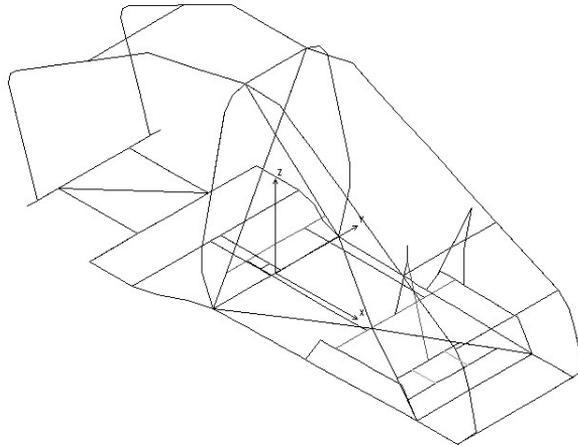


Figura 2.3 Sistema de ejes local del chasis.



Figura 2.4 Sistema de ejes local del grupo motor.

Los datos de coordenadas del centro de gravedad en coordenadas locales e inercias de cada sólido respecto de unos ejes paralelos a los locales con origen en el centro de gravedad se reflejan en la Tabla 2.5.

La posición del centro de gravedad del chasis se obtuvo a partir de la modelización gráfica del vehículo, e igualmente sus momentos y productos de inercia. Sin embargo, para obtener los datos referentes al motor, fue necesario suspenderlo de una grúa, colgándolo en diversas posiciones. En

cada posición de equilibrio, se marcó con una tiza una recta vertical que pasase por el punto de sujeción. El centro de gravedad se localizó en la intersección de los segmentos marcados.

Para obtener los momentos de inercia, una vez calculada la situación del centro de gravedad, se hizo balancear el motor, colgándolo de tal manera que se recogiesen los distintos planos. Midiendo los períodos de oscilación, se pudieron aproximar los valores de los momentos de inercia a partir de la expresión

$$I_o = \frac{Mg\overline{OGT}^2}{4\pi^2} \quad (2.1)$$

Donde \overline{OG} es la longitud desde el punto de giro hasta el centro de gravedad y T el período de las oscilaciones.

Descripción	x_{CDG}	y_{CDG}	z_{CDG}	I_x	I_y	I_z	I_{xy}	I_{xz}	I_{yz}
Unidades	(m)			(kg m ²)					
Chasis	0.164	0.012	0.288	20.56	66.58	61.89	0.985	-2.187	0.079
Grupo motor	-0.620	0.000	0.300	14.80	6.00	12.10	0.00	0.00	0.00

Tabla 2.5 Coordenadas del centro de gravedad e inercias de los elementos principales.

2.4. Sistema de Frenos.

El sistema de frenos consiste en un doble circuito delantero y uno trasero, conectados a una bomba de accionamiento mecánico directo que suministra presión al circuito hidráulico. Los frenos empleados en la cuatro ruedas son de disco. En concreto, el sistema de frenado está compuesto por los siguientes elementos:

- Pinzas de disco y discos delanteros de Seat 127.
- Pinzas de disco y discos traseros de Talbot Hórizon.
- Accionamiento mecánico de Talbot Hórizon.
- Latiguillos delanteros de Talbot Hórizon.
- Latiguillos traseros metálicos fabricados a medida.

El circuito hidráulico está realizado en tubo de cobre normalizado, con diámetro interior de 3 mm y exterior de 5 mm. Debido a la distinta

procedencia de los elementos que lo componen, se han utilizado racores de distinto tipo para poder conectar con los latiguillos originales, y para poder realizar las conexiones entre las pinzas y la bomba hidráulica.

Inicialmente, el sistema de frenos carecía de servofreno. La fuerza a ejercer por el conductor sobre el pedal de freno no resultaba excesiva debido al poco peso del vehículo. Sin embargo, esta fuerza sí resultaba excesiva para ejercerla mediante un motor paso a paso, y por ello se estimó conveniente reducirla. La forma más natural de reducir el esfuerzo sobre el pedal, en tanto que es la solución habitual, es asistirlo mediante un servofreno. El servofreno se acopla al cilindro maestro del sistema de frenos. Aprovecha la depresión que se produce en el colector de admisión durante la carrera de admisión de los cilindros, para disminuir la presión a ejercer en el cilindro maestro para mover el líquido de frenos.

Para facilitar su acoplamiento con el resto de elementos del sistema, se instaló un servofreno de un Talbot Hórizon.

La instalación del servofreno obligó a realizar algunas reformas en el sistema de refrigeración. El conjunto radiador-electroventilador se encontraba muy cercano al cilindro maestro del freno, y no dejaba espacio suficiente para alojar el servofreno. La solución consistió en llevar el radiador más hacia el exterior de la parte frontal del vehículo. Para ello fue necesario aumentar la longitud de sus soportes y modificar las varillas de protección exteriores.

2.5. Suspensión.

El sistema de suspensión es diferente en el tren delantero que en el tren trasero, ya que en nuestro caso el reparto de peso del vehículo es muy desfavorable para el tren delantero. Interesa que el tren delantero sea menos rígido que el tren trasero para que la frecuencia de vibración de ambos sea similar; al tener más masa detrás, el sistema está más equilibrado si el resorte en esta parte es más rígido.

La Figura 2.3 muestra una fotografía de los elementos del sistema de amortiguación delantera del vehículo. Para la suspensión delantera se ha elegido una configuración de doble triángulo, con amortiguador de Seat 127. Para adaptarlos al vehículo, los amortiguadores seleccionados deben situarse inclinados, proporcionando en esta posición menor rigidez, con lo que se logra un buen comportamiento.



Figura 2.3 Sistema de suspensión delantero.



Figura 2.4 Sistema de suspensión trasero.

La Figura 2.4 muestra una fotografía de los elementos del sistema de amortiguación trasera del vehículo. En el tren trasero se ha seleccionado una configuración tipo McPherson, con amortiguador inclinado y con un

entramado de barras soldadas como elemento de sujeción basculante. En este caso, los amortiguadores elegidos son de Seat Ibiza, y se sitúan con inclinación menor que en el tren delantero, presentando un comportamiento más rígido que resulta adecuado.

2.6. Dirección.

El sistema de dirección utilizado es de un Seat 127, y se compone de cremallera, columna de dirección y bieletas. Para poder adaptar dimensionalmente este sistema de dirección a nuestro vehículo, ha sido necesario alargar las bieletas izquierda y derecha en 10 y 15 cm respectivamente, así como la columna de dirección. Asimismo, para cumplir exactamente el principio de Ackerman dentro de los primeros 10° de giro a ambos lados, ha sido necesario situar la cremallera 5 cm hacia atrás, respecto del eje de las ruedas directrices.

También ha sido necesario alargar la columna de dirección para colocarla en una posición descentrada, distinta de la posición original en el vehículo. En la ubicación original, la columna de dirección se sitúa en un plano vertical perpendicular a la cremallera, que contiene la conexión a la cremallera y la sujeción superior del volante. En el prototipo, la pipa de dirección y el volante se encuentran en el centro del vehículo. Así, aparece una excentricidad de 30 cm que es necesario salvar modificando la longitud y los ángulos de la columna de dirección.

3.

El Modelo Virtual del Prototipo

3.1. Modelado: Coordenadas Naturales.

Las coordenadas naturales fueron introducidas en García de Jalón et al. (1981 y 1981 b) y Serna et al. (1982) para casos planos, y en García de Jalón et al. (1986 y 1987) para modelizar sistemas tridimensionales. Las coordenadas naturales describen la posición de cada elemento por medio de coordenadas cartesianas de puntos situados en los pares cinemáticos del sistema, y por medio de componentes cartesianas de vectores unitarios situados también en los pares. Cada elemento precisa de un número mínimo de puntos y/o vectores unidos a él, de manera que dichos puntos y vectores definan completamente el movimiento del elemento. Por lo tanto, si los puntos y vectores de los pares no son suficientes, es preciso añadir otros puntos y/o vectores en los elementos que no queden bien definidos. La modelización en coordenadas naturales suele conducir a tener que plantear ecuaciones de restricción de sólido rígido entre los puntos y vectores empleados en la modelización de un elemento. Las coordenadas de estos puntos y vectores no son, en general, independientes, y para modelizar un único sólido rígido en coordenadas naturales es necesario

introducir ecuaciones de restricción. Otras ecuaciones de restricción serán las encargadas de establecer los pares cinemáticos que unen los distintos elementos del sistema multicuerpo. El resultado suele ser un problema con un gran número de variables, y un gran número de restricciones que ligan estas variables, unas procedentes de las condiciones de sólido rígido de los elementos, y otras procedentes de los pares cinemáticos presentes en el mecanismo.

Las ventajas de las coordenadas naturales son:

1. Su definición es simple y sistemática.
2. Sitúan cada elemento con independencia de los demás.
3. Si se eligen un número suficiente de puntos y vectores para modelizar los elementos, las matrices de masas de los mismos son constantes.
4. Las ecuaciones de restricción son sencillas y fáciles de establecer: suelen ser funciones lineales o a lo sumo cuadráticas de las variables, y no aparecen funciones trigonométricas.
5. El número de coordenadas necesarias para modelizar es reducido, en general intermedio entre las relativas y las coordenadas de punto de referencia, que son las técnicas alternativas de modelización.
6. La matriz de masas global, matriz jacobiana del vector de restricciones y matrices de rigidez y amortiguamiento suelen ser altamente dispersas.
7. La interpretación de los resultados es sencilla, puesto que se trata de coordenadas cartesianas de puntos y vectores unitarios.

En el caso del modelo virtual del prototipo, el conjunto del vehículo se ha implementado en coordenadas naturales mediante 44 puntos, 7 vectores unitarios (una base ortonormal unida al chasis y cada uno de los vectores perpendiculares a los planos de las ruedas), 5 distancias (una para representar el alargamiento de cada uno de los amortiguadores, y otra que recoge el desplazamiento de la barra de dirección) y 5 ángulos (que describen el giro del volante y el giro de las cuatro ruedas). Por simplificar el problema, el giro del volante se ha ligado directamente con el desplazamiento de la dirección mediante una restricción. De esta manera,

se ha evitado la modelización detallada de la cadena del volante. El número total de variables del problema es 163.

En la Figura 3.1 se puede ver un esquema de la modelización que se ha desarrollado. Para mayor claridad sólo se ha recogido la mitad del vehículo.

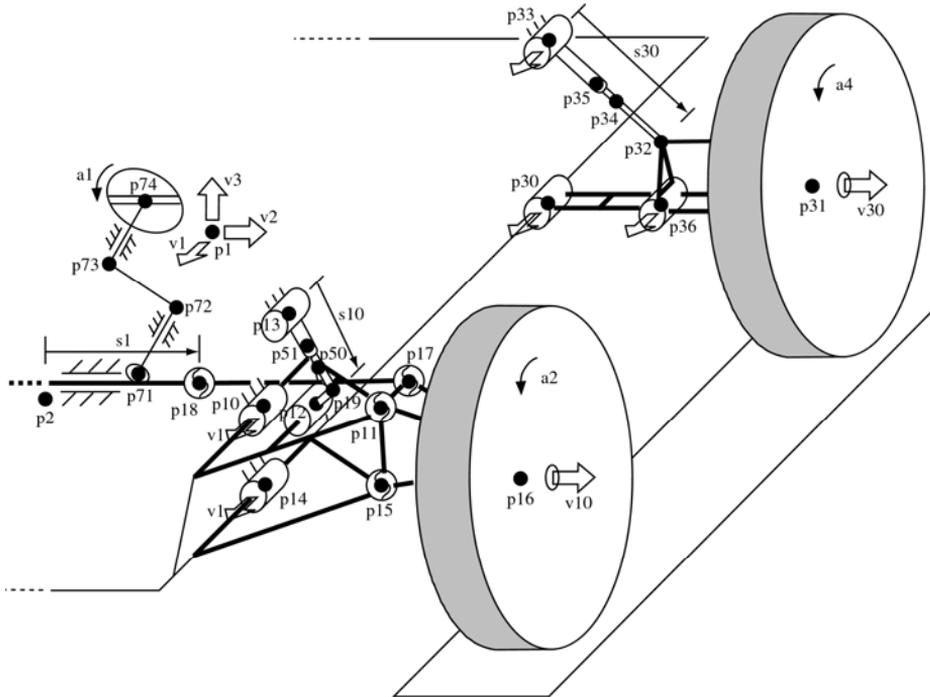


Figura 3.1 Esquema del modelo virtual del prototipo.

Para completar la modelización, se han planteado 161 ecuaciones de restricción y se han contemplado 15 grados de libertad del sistema: la posición y orientación del chasis, el giro del volante, la coordenada vertical del centro de cada una de las ruedas, y el giro de cada una de las ruedas.

3.2. Ecuaciones del Movimiento e Integración Numérica.

La formulación empleada para resolver la dinámica del modelo del prototipo ha sido desarrollada en el Laboratorio de Ingeniería Mecánica de la Universidad de La Coruña. Consiste en una versión mejorada de una formulación aumentada de Lagrange de índice 3 que emplea una modificación de las proyecciones ortogonales propuestas en Bayo y Ledesma (1996).

La forma empleada para las ecuaciones del movimiento es la siguiente:

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_q^T \alpha \mathbf{\Phi} + \mathbf{\Phi}_q^T \lambda^* = \mathbf{Q} \quad (3.1)$$

donde \mathbf{M} corresponde a la matriz de masas del sistema, $\ddot{\mathbf{q}}$ son las aceleraciones, $\mathbf{\Phi}_q$ la matriz jacobiana de las ecuaciones de restricción, α es el factor de penalización, $\mathbf{\Phi}$ el vector que contiene las ecuaciones de restricción, λ^* los multiplicadores de Lagrange, y \mathbf{Q} el vector de fuerzas generalizadas. Los multiplicadores de Lagrange se obtienen mediante el siguiente proceso iterativo:

$$\lambda_{i+1}^* = \lambda_i^* + \alpha \mathbf{\Phi}_{i+1}, \quad i = 0, 1, 2, \dots \quad (3.2)$$

donde el valor de λ_0^* se extrapola de los coeficientes de Lagrange ya calculados en pasos anteriores de tiempo (conviene notar que el subíndice n indica el paso de tiempo, mientras que el subíndice i hace referencia al paso de iteración dentro de un mismo paso de tiempo)

$$\left(\lambda_0^*\right)_{n+1} = 2\lambda_n - \lambda_{n-1} \quad (3.3)$$

Para la integración se ha empleado un integrador implícito de paso simple, la regla trapezoidal. Las ecuaciones correspondientes en velocidades y aceleraciones son (Δt es el paso de tiempo)

$$\dot{\mathbf{q}}_{n+1} = \frac{2}{\Delta t} \mathbf{q}_{n+1} + \hat{\mathbf{q}}_n \quad ; \quad \hat{\mathbf{q}}_n = -\left(\frac{2}{\Delta t} \mathbf{q}_n + \dot{\mathbf{q}}_n\right) \quad (3.4)$$

$$\ddot{\mathbf{q}}_{n+1} = \frac{4}{\Delta t^2} \mathbf{q}_{n+1} + \hat{\mathbf{q}}_n \quad ; \quad \hat{\mathbf{q}}_n = -\left(\frac{4}{\Delta t^2} \mathbf{q}_n + \frac{4}{\Delta t} \dot{\mathbf{q}}_n + \ddot{\mathbf{q}}_n\right) \quad (3.5)$$

Entonces, se pueden escribir las ecuaciones que representan el equilibrio dinámico en el paso $n+1$ introduciendo las ecuaciones (3.4) y (3.5) en las ecuaciones del movimiento (3.1), obteniéndose así la siguiente expresión

$$\frac{4}{\Delta t^2} \mathbf{M} \mathbf{q}_{n+1} + \mathbf{\Phi}_{\mathbf{q}_{n+1}}^T (\alpha \mathbf{\Phi}_{n+1} + \lambda_{n+1}^*) - \mathbf{Q}_{n+1} + \mathbf{M} \hat{\mathbf{q}}_n = 0 \quad (3.6)$$

Sin embargo, debido a cuestiones numéricas (los pasos de tiempo son pequeños y dividir por Δt^2 haría que se desaprovechase la precisión de la

máquina), es aconsejable escalar esta expresión, multiplicándola por el factor $\frac{\Delta t^2}{4}$. De este modo, la ecuación resulta

$$\mathbf{M}\mathbf{q}_{n+1} + \frac{\Delta t^2}{4} \mathbf{\Phi}_{\mathbf{q}_{n+1}}^T (\alpha \mathbf{\Phi}_{n+1} + \lambda_{n+1}) - \frac{\Delta t^2}{4} \mathbf{Q}_{n+1} + \frac{\Delta t^2}{4} \mathbf{M}\hat{\mathbf{q}}_n = 0 \quad (3.7)$$

que, simbólicamente, designaremos como $f(\mathbf{q}_{n+1}) = 0$.

Para obtener la solución de este sistema no-lineal, se aplica el método iterativo de Newton-Raphson

$$\left[\frac{\partial f(\mathbf{q})}{\partial \mathbf{q}} \right]_i \Delta \mathbf{q}_{i+1} = -[f(\mathbf{q})]_i \quad (3.8)$$

donde el residuo vale

$$[f(\mathbf{q})] = \frac{\Delta t^2}{4} (\mathbf{M}\ddot{\mathbf{q}} + \mathbf{\Phi}_q^T \alpha \mathbf{\Phi} + \mathbf{\Phi}_q^T \lambda^* - \mathbf{Q}) \quad (3.9)$$

y la matriz tangente aproximada es

$$\left[\frac{\partial f(\mathbf{q})}{\partial \mathbf{q}} \right] = \mathbf{M} + \frac{\Delta t}{2} \mathbf{C} + \frac{\Delta t^2}{4} (\mathbf{\Phi}_q^T \alpha \mathbf{\Phi}_q + \mathbf{K}) \quad (3.10)$$

en la que \mathbf{C} y \mathbf{K} representan la contribución de las fuerzas elásticas y de amortiguación que puedan estar presentes en el sistema.

Un estudio atento de la matriz tangente revela que pueden aparecer problemas de mal condicionamiento numérico para pasos de tiempo muy pequeños. En efecto, se puede observar en la ecuación (3.10) que la matriz \mathbf{K} y los términos que hacen referencia a las restricciones están multiplicados por Δt^2 , y \mathbf{C} por Δt , mientras que la matriz de masas, \mathbf{M} , no está multiplicada por el paso de tiempo. Como consecuencia, cuando Δt toma unos valores muy bajos, aparecen errores de redondeo de gran magnitud. De hecho, se ha demostrado en Brenan et al (1989) que, para una ecuación diferencial algebraica de índice 3, la matriz tangente tiene un número de condición del orden $1/\Delta t^3$. Consecuentemente, el método presenta errores de redondeo para pasos de tiempo menores que 10^{-5} s, lo que proporciona un rango suficiente para resolver problemas realistas. En

nuestro caso, el paso de tiempo empleado en la simulación del vehículo es de 10 ms, con lo cual se está suficientemente lejos de los problemas debidos al redondeo.

La resolución de las ecuaciones anteriores proporciona un conjunto de valores para el vector \mathbf{q}_{n+1} de posiciones, que satisface la ecuación del movimiento (3.1) junto con las restricciones $\Phi = 0$. Sin embargo, nada hace previsible que los correspondientes conjuntos de velocidades y aceleraciones satisfagan $\dot{\Phi} = 0$ y $\ddot{\Phi} = 0$, puesto que estas condiciones no se han impuesto durante el proceso de resolución. Para superar esta dificultad, se realizan proyecciones en velocidades y aceleraciones. La matriz principal empleada en las proyecciones es la misma matriz tangente de la ecuación (3.10). De esta forma, se evita una nueva triangularización dentro del paso, y las proyecciones en velocidades y aceleraciones requieren únicamente la reducción hacia delante y la posterior sustitución hacia atrás.

Si $\dot{\mathbf{q}}^*$ y $\ddot{\mathbf{q}}^*$ son las velocidades y aceleraciones obtenidas una vez que se ha llegado a la convergencia en el proceso iterativo de Newton-Raphson, se pueden calcular las correspondientes velocidades y aceleraciones “limpias” mediante

$$\begin{aligned} \left[\mathbf{M} + \frac{\Delta t}{2} \mathbf{C} + \frac{\Delta t^2}{4} (\Phi_q^T \alpha \Phi_q + \mathbf{K}) \right] \dot{\mathbf{q}} = \\ \left[\mathbf{M} + \frac{\Delta t}{2} \mathbf{C} + \frac{\Delta t^2}{4} \mathbf{K} \right] \dot{\mathbf{q}}^* - \frac{\Delta t^2}{4} \Phi_q^T \alpha \Phi_t \end{aligned} \quad (3.11)$$

para las velocidades, y

$$\begin{aligned} \left[\mathbf{M} + \frac{\Delta t}{2} \mathbf{C} + \frac{\Delta t^2}{4} (\Phi_q^T \alpha \Phi_q + \mathbf{K}) \right] \ddot{\mathbf{q}} = \\ \left[\mathbf{M} + \frac{\Delta t}{2} \mathbf{C} + \frac{\Delta t^2}{4} \mathbf{K} \right] \ddot{\mathbf{q}}^* - \frac{\Delta t^2}{4} \Phi_q^T \alpha (\dot{\Phi}_q \dot{\mathbf{q}} + \dot{\Phi}_t) \end{aligned} \quad (3.12)$$

para las aceleraciones.

En la resolución de los sistemas lineales indicados en (3.8), (3.11) y (3.12), se utilizan técnicas de matrices dispersas. También se tiene en

cuenta el hecho de que la matriz principal del sistema es simétrica y definido positiva.

3.3. Propiedades Inerciales del Chasis y del Motor.

Las propiedades inerciales del chasis y del motor tienen gran importancia en el comportamiento del vehículo. Resultaba conveniente obtenerlas de la manera más aproximada posible. Sin embargo, esta tarea presentaba su dificultad al carecer de información del fabricante del motor sobre estas propiedades.

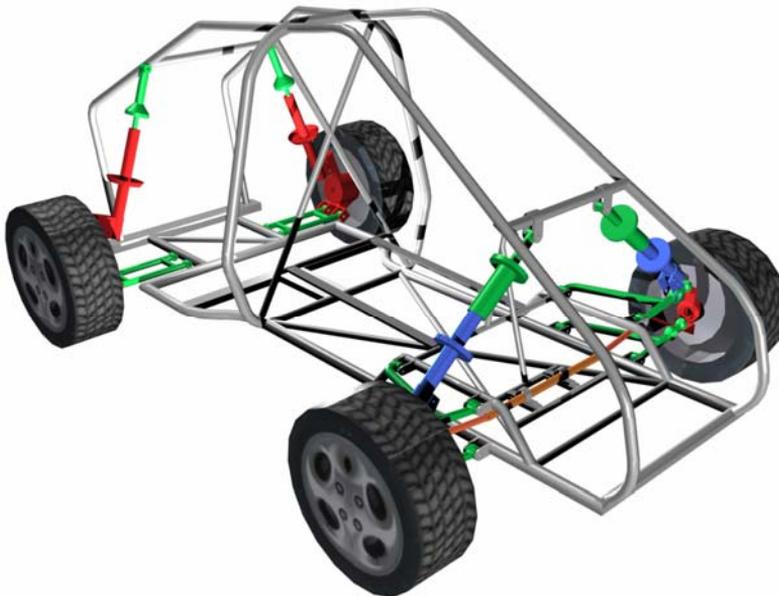


Figura 3.2 Modelo del prototipo desarrollado en I-deas.

Para la obtención de las propiedades del chasis, se realizó un modelo en I-deas que contemplase las secciones de los tubos empleados en la fabricación. Dicho modelo aparece recogido en la Figura 3.2. Las características de las secciones aparecen recogidas en la sección 2.1 de esta tesis. A partir de estos datos, I-deas calcula y proporciona las siguientes propiedades inerciales del chasis:

Más complicado resultaba obtener los datos referidos al motor. No disponíamos datos del fabricante por lo que se decidió realizar una serie de experimentos para determinar el peso y los momentos de inercia. Además, era necesario determinar la posición del centro de gravedad del motor.

Para determinar el peso se decidió suspenderlo de un muelle del que se había determinado antes su rigidez.

Para determinar los momentos de inercia, el motor se suspendió de una grúa y se le hizo balancearse alrededor de distintos ejes, midiendo la inclinación de la que partía y el período de las oscilaciones.

3.4. Fuerzas Transmitidas por la Suspensión.

Como se mencionó al describir el prototipo real del vehículo, el vehículo emplea unos amortiguadores de vehículos comerciales, recuperados de un desguace. Por este motivo, no se conoce la expresión exacta de la fuerza que introducen en el modelo. Para solventar esta dificultad, se ha empleado un modelo lineal de amortiguador y resorte como el que se muestra en la Figura 3.3. Los valores del coeficiente de disipación C del amortiguador y de la rigidez K del muelle que se han empleado se recogen en la siguiente tabla:

	Amortiguador delantero	Amortiguador trasero
K	16000 N/m	10595 N/m
C	10000 N·s/m	6000 N·s/m

Tabla 3.1 Valores empleados en el modelo de los amortiguadores.

Estos valores se aproximaron a partir de los correspondientes a la amortiguación de un vehículo real. Posteriormente, se fueron corrigiendo mediante la realización de maniobras en el simulador, hasta lograr en el vehículo un comportamiento realista.

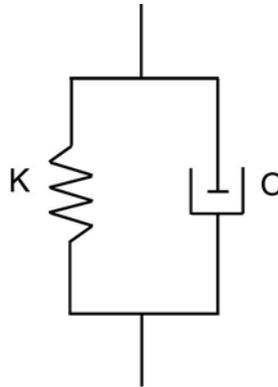


Figura 3.3 Modelo de amortiguador y resorte.

3.5. Caracterización del Contacto Neumático-Suelo.

El neumático es uno de los elementos del vehículo de más difícil caracterización. Es el encargado de transmitir las fuerzas entre el vehículo y el suelo. Esta fuerza depende de múltiples factores, como la composición del neumático, su geometría, la presión de inflado, las características de la otra superficie, la carga vertical que recibe, etc.

Atendiendo a su simulación, las fuerzas y momentos transmitidos por el neumático se pueden dividir en dos grandes categorías, que estudiaremos por separado:

- La fuerza vertical y de compresión del neumático.
- Las fuerzas contenidas en el plano del suelo (lateral y longitudinal), y el momento autoalineante.

3.5.1. Fuerza Vertical y de Compresión.

En este apartado se engloban las fuerzas producidas al comprimir el neumático, bien sea bajo el peso del automóvil o al colisionar contra un obstáculo que aparezca en su trayectoria, como puede ser un escalón, un cambio de pendiente o una pared.

En el simulador, el neumático es el nexo de unión entre el modelo del coche y el entorno virtual en el que éste se mueve. Por este motivo, además del modelo del neumático, la forma de definir y modelizar las

superficies que conforman el entorno es también muy importante. Las superficies en las que se mueve el coche se han representado mediante triángulos. Los triángulos son fáciles de generar y cualquier superficie, más o menos compleja, puede ser reducida a triángulos.

El primer paso para introducir las fuerzas de compresión sobre el neumático es detectar los planos del entorno que están en contacto con el mismo, esto es, que se encuentran a una distancia inferior al radio del neumático. Un polígono que se encuentre a una distancia menor estará comprimiendo la rueda, por lo que, consecuentemente, generará una fuerza de reacción conforme al comportamiento elástico del neumático.

Para hallar la distancia de cada uno de los triángulos al punto central de la rueda, se usará la expresión

$$d = \frac{n_x x + n_y y + n_z z + D}{\sqrt{n_x^2 + n_y^2 + n_z^2}} \quad (3.13)$$

en la que \mathbf{n} es un vector normal al plano que contiene al triángulo, D es una constante que determina la posición del plano dada su orientación, y (x, y, z) son las coordenadas del centro de la rueda. Entre la información relativa a cada uno de los polígonos que componen el entorno, se almacena su vector normal y la constante D . El valor de ésta se despeja de la expresión general del plano, $n_x x + n_y y + n_z z + D = 0$ sustituyendo x, y, y, z por cualquier punto perteneciente al plano. En este caso, se emplea uno cualquiera de los vértices del polígono.

Una vez conocida la distancia, la rueda está en contacto con el suelo si ésta es inferior al radio del neumático. Para ello se comprueba el valor del parámetro s , que y se calcula según la expresión

$$s = \frac{d}{r_0} \quad (3.14)$$

donde r_0 es el radio indeformado del neumático. Cuando la rueda está en contacto con el suelo, s toma valores entre 0 y 1, como se observa en la Figura 3.4.

El paso siguiente consiste en verificar que el punto de contacto está ubicado en el interior del triángulo, es decir, que la rueda se halla realmente en contacto con el triángulo y no solamente con el plano que lo contiene.

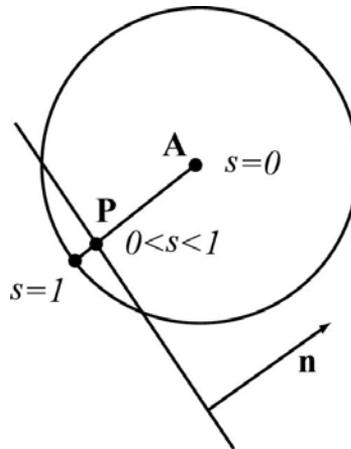


Figura 3.4 Representación paramétrica del segmento de intersección entre eje y suelo.

Para ello, se debe determinar el punto de contacto entre el radio de la rueda y el plano, \mathbf{P} . Dado que el contacto rueda-plano es siempre normal al plano, si denominamos \mathbf{A} al punto central de la rueda, podemos calcular la posición del punto de contacto restándole al centro de la rueda un vector orientado según la normal al plano, \mathbf{n} , que tenga como módulo el radio deformado, r_{def} , como

$$\mathbf{P} = \mathbf{A} - \mathbf{n} \cdot r_{def} \quad (3.15)$$

El radio deformado es conocido puesto que coincide con la distancia d calcula en la ecuación (3.13) cuando el neumático está en contacto con el plano, es decir, cuando $s \leq 1$.

Para determinar si el punto \mathbf{P} se encuentra dentro de los límites del triángulo tomaremos como base dos vectores formados por dos de los lados del triángulo (por ejemplo los indicados en la Figura 3.5 como \mathbf{r} y \mathbf{t}). Cualquier punto perteneciente al plano se puede escribir como una combinación lineal de estos dos vectores.

$$\mathbf{V}(c_1, c_2) = \mathbf{V}_0 + c_1 \mathbf{r} + c_2 \mathbf{t} \quad (3.16)$$

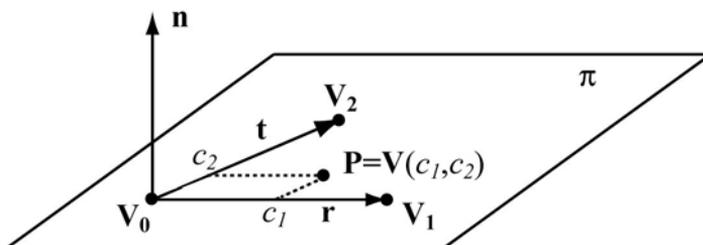


Figura 3.5 Determinación de la intersección entre segmento y triángulo.

Sin embargo, para que un punto se encuentre “dentro” del triángulo, los coeficientes de combinación lineal deben de verificar que

$$\begin{aligned} c_1 &\geq 0 \\ c_2 &\geq 0 \\ c_1 + c_2 &\leq 1 \end{aligned} \quad (3.17)$$

Estas condiciones constituyen la frontera para cada una de las aristas del triángulo.

Si establecemos un sistema de coordenadas con origen en V_0 , y definimos el vector w como aquél que relaciona el punto de intersección entre la rueda y el triángulo, P , con el origen de coordenadas recién establecido,

$$w = P - V_0 \quad (3.18)$$

Por pertenecer al plano, este vector lo podremos escribir como

$$w = c_1 r + c_2 t \quad (3.19)$$

Si el vector se halla encerrado dentro del triángulo, los coeficientes de combinación lineal satisfarán las condiciones impuestas en la ecuación (3.17). Para determinar su valor procedemos de la siguiente manera. Si se multiplica escalarmente ambos miembros de la ecuación (3.19) por $n \times t$, se obtiene

$$w(n \times t) = (c_1 r + c_2 t)(n \times t) = c_1 r(n \times t) + c_2 t(n \times t) = c_1 r(n \times t) \quad (3.20)$$

Expresión que conduce a

$$c_1 = \frac{\mathbf{w}(\mathbf{n} \times \mathbf{t})}{\mathbf{r}(\mathbf{n} \times \mathbf{t})} \quad (3.21)$$

Y, análogamente, al multiplicar por $\mathbf{n} \times \mathbf{u}$, se obtiene

$$c_2 = \frac{\mathbf{w}(\mathbf{n} \times \mathbf{r})}{\mathbf{t}(\mathbf{n} \times \mathbf{r})} \quad (3.22)$$

Una vez se ha comprobado que la rueda está en contacto con dicho triángulo, se usa la distancia (previamente calculada para saber si el polígono podía contactar potencialmente con la rueda) para calcular la fuerza ejercida sobre la rueda y se orienta en la dirección de la normal.

Este procedimiento, sin embargo, presenta un pequeño inconveniente. Como se habrá observado, la distancia desde el centro de la rueda a los planos que forman cada uno de los triángulos se mide en todas las direcciones. Esto equivale a considerar la rueda como una esfera. Obviamente, no es así, por lo que debemos limitar las direcciones en las que tenemos en cuenta el contacto.

Para esto, se ha supuesto que la banda de rodadura corresponde al arco de circunferencia comprendido por un ángulo de 40° , como se puede observar en la Figura 3.6. Este arco, para unos neumáticos como los que monta el prototipo de relación de aspecto 185/70, equivale a un ancho de la banda de 185 mm, que se ajusta al derivado de la relación de aspecto.

Por este motivo, es necesario verificar el ángulo que forma la normal al plano, \mathbf{n} , con el vector \mathbf{u} , perpendicular al plano de la rueda. Puntos exteriores a la banda de rodadura verificarán

$$\mathbf{u} \cdot \mathbf{n} \geq \cos 20^\circ \quad (3.23)$$

En principio, estos puntos no deberían conllevar una situación de contacto del neumático con el suelo, como se puede ver en la Figura 3.6a. Pero es posible que se produzca una situación como la de la Figura 3.6b. Esto se debe a que se detecta la distancia mínima al plano, que correspondería al punto \mathbf{B} , aunque el contacto realmente se produzca en el punto \mathbf{P}_c . Para discriminar esta situación, una vez que se ha detectado un punto como el \mathbf{B} que verifica la ecuación (3.23), se comprueba si el punto \mathbf{P}_c está situado a una distancia del centro de la rueda inferior al radio indeformado y, en el caso de que sea así, si está contenido en el triángulo que da origen al

plano. Si se verifican estas dos condiciones, se aplica la fuerza en ese punto según la dirección de la normal al plano.

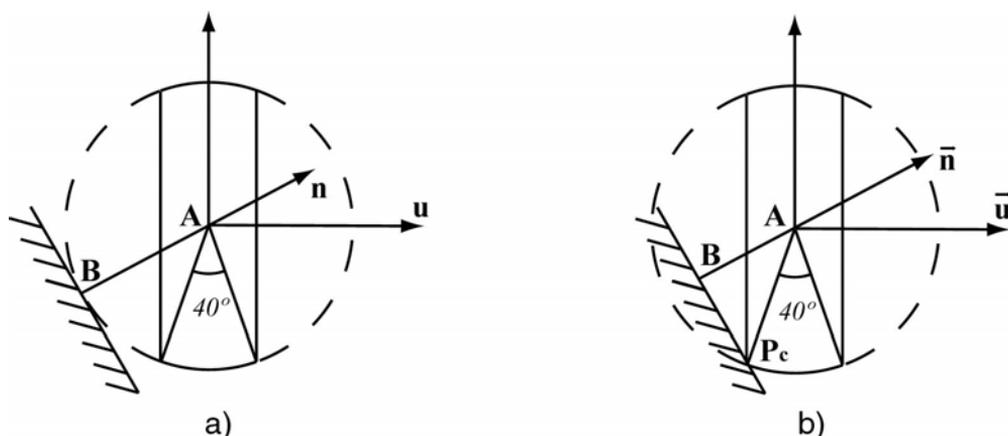


Figura 3.6 Direcciones a considerar en la detección del contacto.

Para el cálculo de la fuerza radial de compresión, tradicionalmente se ha considerado el neumático mediante un modelo de resorte. La rigidez del resorte se denomina en este caso rigidez radial del neumático, k_T . Entonces, la fuerza radial a aplicar en el centro de la rueda será:

$$F_r = k_T \cdot (r_0 - r_{def}) \quad (3.24)$$

La dirección de la fuerza será la de la normal al suelo, e irá orientada hacia el centro de la rueda.

$$\mathbf{F} = F_r \cdot \mathbf{n} \quad (3.25)$$

Al ir orientada hacia el centro de la rueda, la fuerza se puede aplicar directamente en el punto del centro de la rueda. Aquellos casos, como el representado en la Figura 3.6 b, en los que la fuerza no pasa por el centro de la rueda, requerirán un tratamiento específico, que se verá en el apartado 3.5.4.

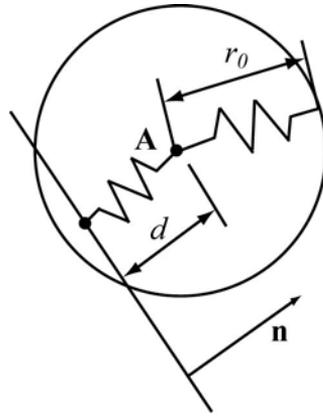


Figura 3.7 Comportamiento lineal elástico de la rueda en las direcciones normales al suelo.

Por último, la consideración del neumático como un muelle nos obliga a tener en cuenta su contribución a la matriz de rigidez del sistema.

De forma general, los muelles actúan entre dos sólidos distintos del sistema, siendo A y B los puntos entre los que actúan y que forman parte de las variables del problema, pudiendo ser uno de ellos fijo, y la fuerza que introducen depende de la distancia que los separa. La expresión general de la fuerza es

$$\mathbf{F}_m = g_m(s) \frac{(\mathbf{r}_B - \mathbf{r}_A)}{s} \quad (3.26)$$

donde $s = |\vec{r}_B - \vec{r}_A|$ y $g_m(s)$ una función cualquiera que puede ser no lineal (Dopico 2004). La contribución de estas fuerzas a la matriz de rigidez supone derivar la expresión de la fuerza, (3.26), con respecto al vector de variables:

$$\mathbf{K}_m = - \left(\frac{\partial \mathbf{F}_m}{\partial s} \frac{\partial s}{\partial \mathbf{q}} + \frac{\partial \mathbf{F}_m}{\partial \mathbf{q}} \right) = \frac{1}{s^2} \left(\frac{\partial g_m(s)}{\partial s} - \frac{g_m(s)}{s} \right) [(\mathbf{r}_B - \mathbf{r}_A)(\mathbf{r}_B - \mathbf{r}_A)^T] [\mathbf{I}_3 - \mathbf{I}_3] + \frac{g_m(s)}{s} [\mathbf{I}_3 - \mathbf{I}_3] \quad (3.27)$$

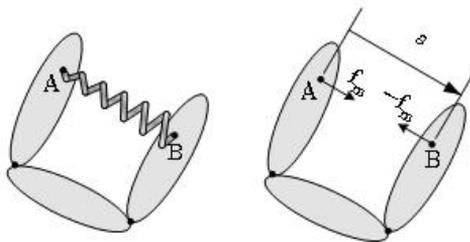


Figura 3.8 Modelización de un muelle actuando entre dos sólidos.

En el caso que nos ocupa, los puntos forman parte de las variables del problema pues uno de los puntos, tomemos por caso el punto B , es el suelo y el otro, el centro de la rueda. La fuerza la podemos escribir como

$$\mathbf{F}_m = -k_T (\mathbf{n}^T (\mathbf{r}_B - \mathbf{r}_A) - r_0) \mathbf{n} \quad (3.28)$$

y la contribución a la matriz de rigidez

$$\mathbf{K}_m = -\frac{\partial \mathbf{F}_m}{\partial \mathbf{r}_A} = k_T (\mathbf{n} \cdot \mathbf{n}^T) \quad (3.29)$$

El proceso se repite, verificando el contacto con todos y cada uno de los triángulos que conforman el escenario. Los entornos que se han empleado en el presente trabajo no son muy complejos y, consiguientemente, no poseen un elevado número de triángulos. En el caso de que fuese así, siempre es posible dividir el entorno en sectores, de tal manera que no sea necesario verificar siempre el contacto con todos los triángulos del entorno.

A simple vista, el método puede parecer computacionalmente muy costoso. Una observación más detallada revelará la existencia de muchos factores constantes e independientes de la posición del vehículo, y que dependen únicamente de la posición del terreno que, a su vez, permanece inalterable. En efecto, para cada triángulo puede precalcularse el siguiente conjunto de variables:

La constante D de la ecuación del plano, con lo que si se tiene en cuenta además que el módulo de la normal es igual a la unidad, la distancia de un punto cualquiera a dicho plano puede calcularse como

$$d = \frac{n_x x + n_y y + n_z z + D}{\sqrt{n_x^2 + n_y^2 + n_z^2}} = n_x x + n_y y + n_z z + D = \mathbf{n} \cdot \mathbf{r} + D \quad (3.30)$$

siendo \mathbf{r} el vector que caracteriza a dicho punto.

También permanecen invariantes las siguientes expresiones durante la evaluación de la pertenencia de un punto a un triángulo:

$$\mathbf{n} \times \mathbf{u} \quad (3.31)$$

$$\mathbf{n} \times \mathbf{v} \quad (3.32)$$

$$\frac{1}{\mathbf{u}(\mathbf{n} \times \mathbf{v})} \quad (3.33)$$

$$\frac{1}{\mathbf{v}(\mathbf{n} \times \mathbf{u})} \quad (3.34)$$

$$n_x^2 + n_y^2 + n_z^2 = 1 \quad (3.35)$$

Conviene reseñar que se almacena la inversa de las expresiones (3.33) y (3.34), ya que, en las computadoras actuales, es habitualmente más rápida la multiplicación que la división por un factor.

La comprobación queda reducida, pues, a 2 productos escalares, 5 multiplicaciones y 2 sumas por cada triángulo que esté muy próximo a la rueda, lo que constituye un coste computacional bastante razonable.

3.5.2. Fuerza Longitudinal.

La fuerza longitudinal es la responsable de la transmisión de fuerzas de aceleración y frenado al vehículo. En general, esta fuerza no es simplemente el producto de la fuerza normal sobre el neumático por el coeficiente de adherencia de la superficie, sino que depende del deslizamiento entre el neumático y el suelo. Este deslizamiento no sólo ocurre cuando los neumáticos patinan completamente sobre la calzada. El comportamiento del neumático es, en ocasiones, una mezcla de situaciones de adherencia y deslizamiento. Como se puede ver en la Figura 3.9, que corresponde a un caso de frenada, la proyección de la velocidad del centro de la rueda en la dirección longitudinal del neumático, \mathbf{v}_x , se puede descomponer en la velocidad de rodadura, \mathbf{v}_r , y la velocidad de deslizamiento, \mathbf{v}_s . La velocidad de rodadura es la que tendría la rueda en el caso de que se produjese una rodadura pura, y su módulo será:

$$v_r = \Omega \cdot r_{def} \tag{3.36}$$

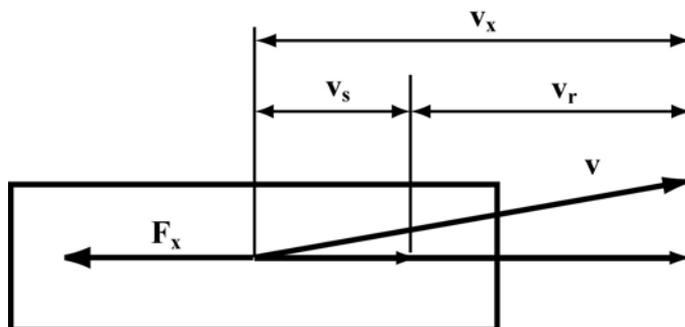


Figura 3.9 Vista superior del neumático mostrando los diagramas de velocidades y de fuerza, en un caso de frenada.

A partir de aquí, se define el deslizamiento o pseudodeslizamiento como (Bakker and Pacejka 1989)

$$K = -\frac{v_s}{v_x} \tag{3.37}$$

En la simulación, el pseudodeslizamiento no se obtiene directamente, sino que es preciso calcularlo a partir de la componente longitudinal de la velocidad del centro de la rueda y la velocidad angular de la rueda como:

$$K = -\frac{v_x - v_{rod}}{v_x} = \frac{\Omega \cdot r_{def}}{v_x} - 1 \tag{3.38}$$

Donde Ω es la velocidad angular de la rueda, y r_{def} es el radio deformado (bajo carga) del neumático.

En función del deslizamiento, la fuerza longitudinal que el neumático es capaz de transmitir tiene el aspecto que se presenta en la Figura 3.10. Como se puede observar, presenta una zona lineal hasta alcanzar el máximo, correspondiente al valor estático del coeficiente de adherencia, μ_x , para valores de deslizamiento entre 0,15 y 0,2 (Luque et al. 2004). Después la fuerza desciende hasta que el coeficiente de adherencia toma el valor dinámico, μ_{xd} .

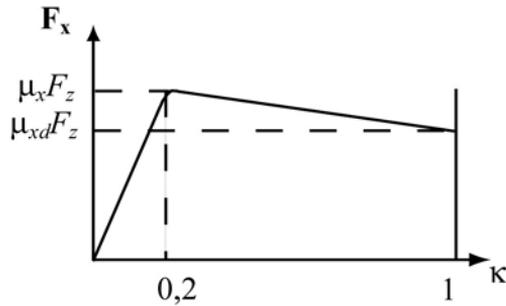


Figura 3.10 Fuerza longitudinal desarrollada por el neumático en función del deslizamiento longitudinal.

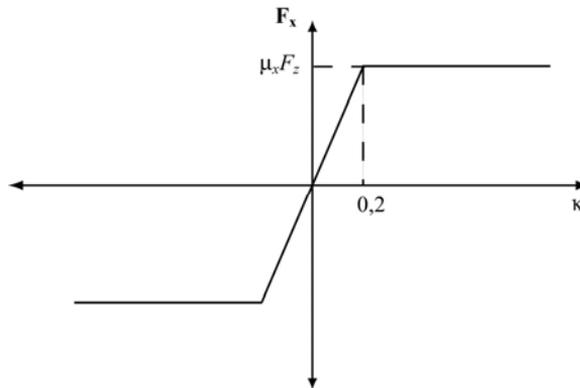


Figura 3.11 Curva correspondiente a la fuerza longitudinal empleada en el modelo.

Es habitual recurrir a un modelo matemático, como la Fórmula Mágica de Pacejka (Bakker and Pacejka 1989) o el modelo de Dugoff (Dugoff et al. 1970), por citar dos de los más empleados, para representar la fuerza longitudinal desarrollada por el neumático a partir del deslizamiento. A partir del modelo del vehículo, la velocidad del centro de la rueda, v , y la velocidad angular de las ruedas, ω , son perfectamente conocidas, y es posible calcular el deslizamiento longitudinal según la ecuación (3.38). Sin embargo, no se poseen datos del fabricante que permitan modelizar el neumático. Por este motivo, se ha supuesto la curva de la Figura 3.11 para el comportamiento longitudinal del neumático. Las ecuaciones que la definen son:

$$F_x = \frac{\mu_x F_z |\kappa|}{0,2} \quad \text{para } |\kappa| \leq 0,2 \quad (3.39)$$

$$F_x = \mu_x F_z \quad \text{para} \quad |\kappa| > 0,2 \quad (3.40)$$

Junto con esto, en la transmisión de las fuerzas, el coeficiente de adherencia entre el neumático y la carretera es otro factor muy importante. En el simulador no se ha caracterizado de ninguna forma el suelo por el que circula el vehículo, y se ha supuesto un coeficiente de adherencia $\mu_x = 0,5$.

Finalmente, para la introducción de la fuerza longitudinal en el modelo, la fuerza que el suelo ejerce sobre el neumático se ha trasladado al centro de la rueda. Como se puede ver en la Figura 3.12 en el que se representa una situación de frenado, la fuerza que actúa en la banda de rodadura se convierte en esa misma fuerza actuando en el centro de la rueda junto con un par. De esta manera, la fuerza se puede aplicar directamente sobre el punto correspondiente al centro de la rueda y el par se aplica a la coordenada del ángulo de giro de la misma.

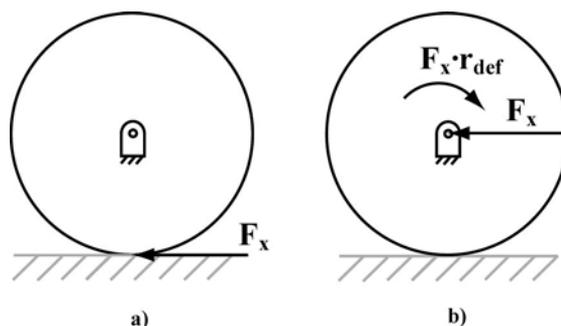


Figura 3.12 Introducción de la fuerza longitudinal en el modelo numérico.

3.5.3. Fuerza Lateral y Momento Autoalineante.

La fuerza lateral que el neumático soporta y el momento autoalineante dependen del ángulo de deslizamiento, α , y del ángulo de caída, γ , de las ruedas. En ocasiones, la trayectoria seguida por el neumático no coincide con la dirección que marcaría un vector contenido en el plano de la rueda y orientado en la dirección de avance del neumático. En la zona de contacto entre el neumático y el suelo, denominada huella, se produce adherencia entre la banda de rodadura y el suelo. Al girar el volante del automóvil, las ruedas giran un ángulo determinado. Sin embargo, debido a la adherencia del neumático y a su capacidad de deformación, es posible que la banda de

rodadura, en contacto con la carretera, no llegue a girar el ángulo previsto. La diferencia entre la intersección de un plano que contiene a la rueda, y la trayectoria del avance realmente seguido por la huella del neumático es el ángulo de deriva, α , que se muestra en la Figura 3.13.

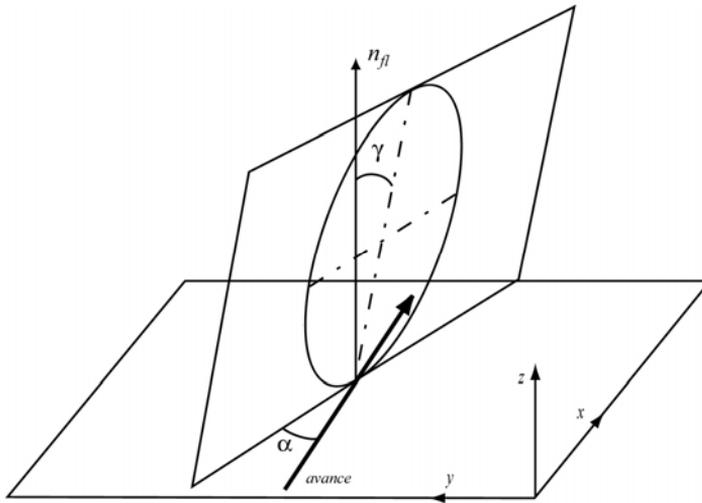


Figura 3.13. Ángulos de deriva, α , y caída, γ .

En esa misma figura se puede apreciar también el ángulo γ que forma el plano que contiene al neumático con la normal al plano del suelo. Este ángulo es denominado ángulo de caída.

Tanto α como γ son fáciles de determinar en el modelo del vehículo. Por tanto, a diferencia de la fuerza longitudinal, sí se pueden modelizar de una manera más realista la fuerza lateral y el momento autoalineante.

Efectivamente, el ángulo α se puede determinar mediante la siguiente ecuación

$$\alpha = \arcsin \left[\mathbf{n}_f (\mathbf{b} \times \mathbf{v}^*) \right] \quad (3.41)$$

donde \mathbf{v}^* es un vector que recoge las componentes de la velocidad del centro de la rueda contenidas en un plano paralelo al del suelo, \mathbf{b} es un vector unitario en el sentido del avance de la rueda, y \mathbf{n}_f es la normal al suelo, como se muestra en la Figura 3.14.

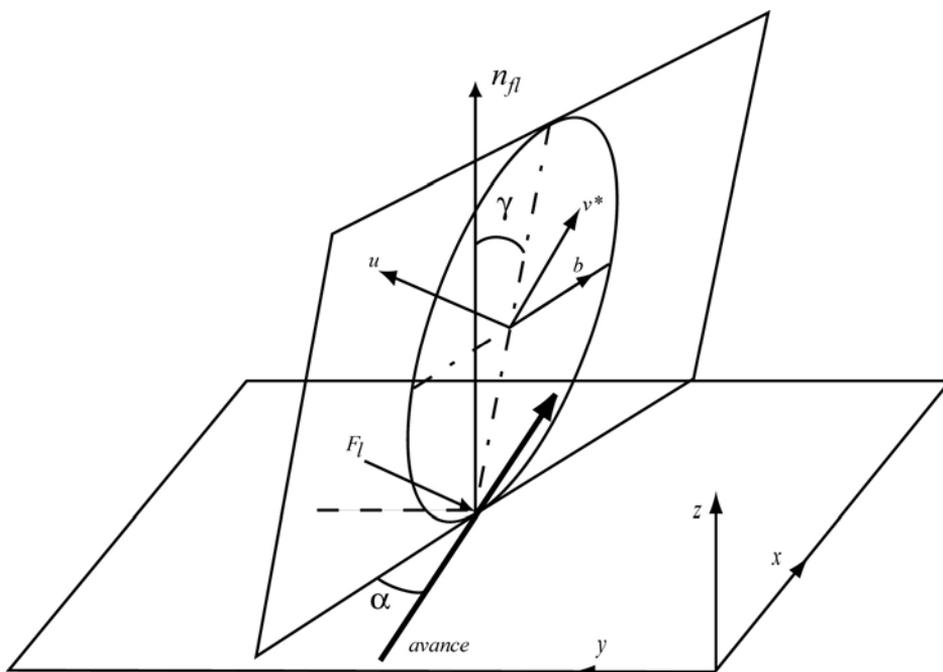


Figura 3.14. Determinación de los ángulos α y γ .

El ángulo de caída se determina apoyándose en un vector unitario, u , perpendicular al plano de la rueda, y la normal al suelo, n_{fl} , como

$$\gamma = \arcsin(\mathbf{u} \cdot \mathbf{n}_{fl}) \tag{3.42}$$

En función del ángulo de deslizamiento y del ángulo de caída, el aspecto que presentan la fuerza lateral y el momento autoalineante se puede observar en la Figura 3.15.

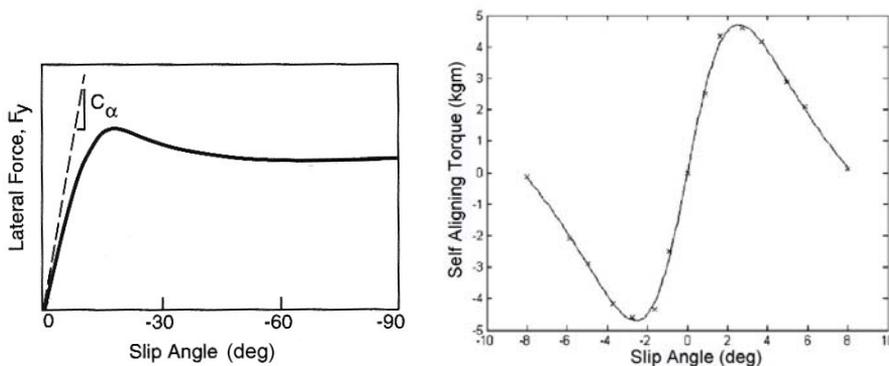


Figura 3.15 Fuerza lateral y momento autoalineante.

Como puede verse, la relación entre la fuerza y el ángulo es altamente no lineal. A lo largo de la historia se han hecho, y se continúan haciendo, grandes esfuerzos por emplear una fórmula que modele el comportamiento dinámico de los neumáticos con la mayor precisión posible, teniendo en cuenta las propiedades físicas que caracterizan el neumático. Actualmente, uno de los más empleados es la Fórmula Mágica de Bakker, Nyborg y Pacejka. Fue presentada por primera vez en el año 1986. Posteriormente, ha sufrido revisiones por parte de sus autores para ajustar cada vez más sus resultados al comportamiento real del neumático. El origen de la fórmula es la búsqueda de expresiones matemáticas con coeficientes que puedan identificarse con los valores característicos de los neumáticos, tales como la rigidez longitudinal, la rigidez transversal o los valores máximos de los esfuerzos que se producen entre el neumático y el suelo.

La fórmula expresa la fuerza lateral F_y y el momento autoalineante M_z como una función del ángulo de deriva α . El ángulo de caída se introduce en la fórmula a través de uno de los coeficientes. La fuerza longitudinal F_x se expresa en función del deslizamiento longitudinal κ . En las fórmulas que se presentan a continuación, la variable de salida y puede representar F_y , M_z o F_x . La variable independiente x toma los valores de α si se trata de representar la fuerza lateral o el momento autoalineante. Si representa la fuerza longitudinal, la variable independiente asume los valores de κ . La forma general de la fórmula, para una carga vertical y un ángulo de caída determinados, es:

$$y(x) = D \operatorname{sen} \left[C \operatorname{arctg} \left\{ Bx - E \left(Bx - \operatorname{arctg} (Bx) \right) \right\} \right] \quad (3.43)$$

donde B , C , D y E son constantes cuyo significado se explicará a continuación. Esta fórmula produce una curva que pasa por el origen de coordenadas, alcanza un máximo y, a continuación, tiende a una asíntota horizontal.

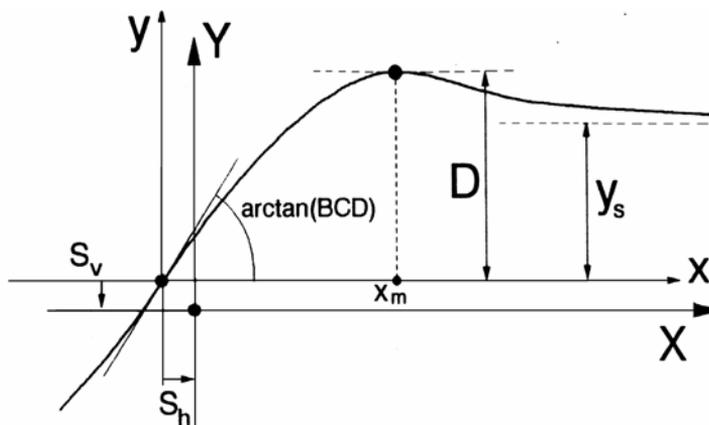


Figura 3.16 Curva característica de un neumático indicando el significado de algunos de los coeficientes de la Fórmula Mágica.

Para ajustarse más al comportamiento real del neumático, que no es antisimétrico, se hace el siguiente cambio de variables:

$$\begin{aligned} Y(X) &= y(x) + S_v \\ x &= X + S_h \end{aligned} \quad (3.44)$$

Los parámetros de traslación S_h y S_v recogen el efecto del ángulo de caída, la conicidad del neumático y del ángulo entre cordones (ángulo del trenzado de las fibras que conforman materialmente el neumático), que son quienes provocan la pérdida del comportamiento antisimétrico. La figura representa la gráfica típica de la fuerza lateral, y en ella se ilustra el significado de algunos de los coeficientes que aparecen en las ecuaciones (3.43) y (3.44): el coeficiente D representa el valor máximo (con respecto a los ejes x) y el producto BCD corresponde a la pendiente en el origen ($x=y=0$).

El factor C modula los límites del rango del argumento de la función seno. Este coeficiente determina la forma de la curva resultante. Los valores típicos para el factor de forma C son: 1.3 para la fuerza lateral, 2.4 para el momento autoalineante y 1.65 para la fuerza de frenado (deslizamiento longitudinal). El factor B es llamado factor de rigidez pues tiene como misión ajustar la pendiente en el origen. Finalmente, el factor E tiene como misión modelar la curva en el entorno de su máximo al mismo tiempo que controla el punto x_m en el que aparece ese máximo. Se puede determinar mediante la siguiente expresión:

$$E = \frac{Bx_m - \operatorname{tg}\left(\frac{\pi}{2C}\right)}{Bx_m - \operatorname{arctg}(Bx_m)} \quad (3.45)$$

El valor de la asíntota a la que tiende y para valores elevados del deslizamiento es

$$y_s = D \cdot \operatorname{sen}\left(\frac{\pi}{2}C\right) \quad (3.46)$$

Es conveniente tener en cuenta que los valores de C suelen ser fijos para un determinado tipo de curva. Utilizando estas expresiones, se hallan los valores de fuerza lateral y momento autoalineante en cada instante de tiempo de la simulación, en función del ángulo de deslizamiento, del ángulo de deriva, y de la reacción normal del suelo.

Para la aplicación práctica de la Fórmula Mágica, las expresiones que se han empleado son las siguientes (F_z representa en todos los casos la fuerza vertical que soporta el neumático):

Fuerza lateral:

$$\begin{aligned} Y_y &= F_y \\ X_y &= \alpha \\ D_y &= \mu_y F_z \\ \mu_y &= (a_1 F_z + a_2)(1 - a_{15} \gamma^2) \\ BCD_y &= a_3 \sin(2 \arctan(F_z/a_4))(1 - a_5 |\gamma|) \\ C_y &= a_0 \\ E_y &= (a_6 F_z + a_7)(1 - (a_{16} \gamma + a_{17}) \operatorname{sgn}(\alpha + S_{hy})) \\ B_y &= BCD_y / C_y D_y \\ S_{hy} &= a_8 F_z + a_9 + a_{10} \gamma \\ S_{vy} &= a_{11} F_z + a_{12} + (a_{13} F_z^2 + a_{14} F_z) \gamma \end{aligned} \quad (3.47)$$

Momento autoalineante:

$$\begin{aligned}
 Y_z &= M_z \\
 X_z &= \alpha \\
 D_z &= (c_1 F_z^2 + c_2 F_z) (1 - c_{18} \gamma^2) \\
 BCD_z &= (c_3 F_z^2 + c_4 F_z) (1 - c_6 |\gamma|) \exp(-c_5 F_z) \\
 C_z &= c_0 \\
 E_z &= (c_7 F_z^2 + c_8 F_z + c_9) (1 - (c_{19} \gamma + c_{20}) * \text{sgn}(\alpha + S_{hz})) / (1 - c_{10} |\gamma|) \\
 B_z &= BCD_z / C_z D_z \\
 S_{hz} &= c_{11} F_z + c_{12} + c_{13} \gamma \\
 S_{vz} &= c_{14} F_z + c_{15} + (c_{16} F_z^2 + a_{17} F_z) \gamma
 \end{aligned} \tag{3.48}$$

Estas expresiones corresponden a la versión de la Fórmula Mágica presentada por Pacejka y Bakker en 1991. Los coeficientes empleados en la formulación aparecen recogidos en la Tabla 3.2. Han sido proporcionados por el fabricante Pirelli, y se corresponden con los neumáticos que monta el coche, de dimensiones 185/70 (155/80 13), sometidos a una presión de 2 kg/cm².

3.5.4. Peculiaridades de la Fuerza Lateral.

Como se ha podido observar, la fuerza lateral y el momento autoalineante dependen fundamentalmente del ángulo de deslizamiento α . Esto hace que se produzcan algunas situaciones en las que el comportamiento del neumático no es satisfactorio. Cuando el vehículo se mueve a baja velocidad, las fuerzas que introduce el neumático van a ser las mismas que para altas velocidades. De esta manera, aparecen fuerzas cuya magnitud es desproporcionada a la maniobra que se está realizando. Por este motivo, se ha moderado el valor de la fuerza mediante un factor, f , de expresión

$$f = 2 \frac{v}{v_0} - \left(\frac{v}{v_0} \right)^2 \tag{3.49}$$

en el que v_0 corresponde con el valor de una velocidad umbral, que, en este caso, toma el valor de 4 m/s. Este factor disminuye de forma cuadrática al hacerlo la velocidad el valor que se había obtenido al evaluar la fuerza lateral del neumático en función del ángulo de deriva.

Fuerza lateral		Momento autoalineante	
a_0	1.30	c_0	2.40
a_1 [1/N]	1.1334E-05	c_1 [m/N]	3.7470E-06
a_2	-8.1131E-01	c_2 [m]	9.9556E-05
a_3 [N/rad]	3.5904E+04	c_3 [m/(N*rad)]	1.8502E-04
a_4 [N]	4.0953E+03	c_4 [m/rad]	1.0535E-01
a_5 [1/rad]	-5.6655E-03	c_5 [1/N]	1.8691E-04
a_6 [1/N]	-2.3009E-04	c_6 [1/rad]	1.7481E+00
a_7	-1.2548E+00	c_7 [1/(N ²)]	-3.2837E-07
a_8 [rad/N]	-1.5503E-06	c_8 [1/N]	1.4776E-03
a_9	-2.5091E-03	c_9	-3.5911E+00
a_{10} [rad/rad]	3.2613E-03	c_{10} [1/rad]	8.4912E-04
a_{11}	-4.4437E-02	c_{11} [rad/N]	2.6390E-07
a_{12} [N]	-3.1931E+01	c_{12} [rad]	-1.0448E-02
a_{13} [1/(N*rad)]	-2.2765E-05	c_{13} [rad/rad]	-6.2098E-02
a_{14} [1/rad]	9.1590E-01	c_{14} [m]	3.5182E-05
a_{15} [1/(rad) ²]	-2.8731E-01	c_{15} [N*m]	-2.7552E+00
a_{16} [1/rad]	6.0520E-01	c_{16} [m/(N*rad)]	-1.8291E-06
a_{17}	3.5269E-02	c_{17} [m/rad]	3.0691E-02
		c_{18} [1/(rad) ²]	1.0696E+01
		c_{19} [1/rad]	-2.1257E-02
		c_{20}	-9.2067E-03

Tabla 3.2 Coeficientes empleados en la Fórmula Mágica.

Este modo de proceder tiene también su contrapartida. Cuando el vehículo se detiene en una superficie con caída lateral, la velocidad es nula, y los neumáticos no aportarán la fuerza necesaria para contrarrestar la gravedad y mantener el vehículo en la posición de equilibrio. Esta situación es fácil de detectar, puesto que la velocidad es muy baja –prácticamente nula-, el ángulo de deslizamiento muy grande y el vector normal al suelo no

es del tipo $\mathbf{n}_{fl} = (0 \ 0 \ 1)$. En esos casos, se ha recurrido a introducir unas fuerzas adicionales en el neumático. Estas fuerzas se comportan como un conjunto formado por un resorte y un amortiguador que, unidos por un lado a la rueda y por otro a una pared, impiden el deslizamiento del vehículo por la pendiente. Esa pared estaría situada en la primera posición de velocidad nula, x_0 , a partir de la cual empieza a deslizarse el neumático. Esta forma de proceder se aplica en la modelización de la fricción, para caracterizar la evolución de la fuerza en las zonas de tránsito desde el valor estático al dinámico del coeficiente de fricción de Coulomb, y es conocida como *bristle method* (Haessig and Friedland 1991 y Ma 1995).

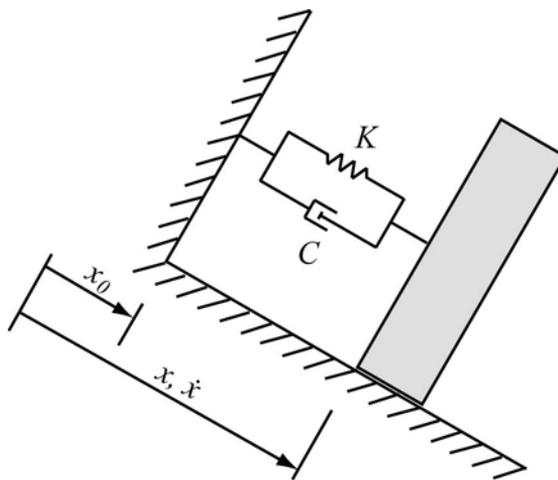


Figura 3.17 Fuerzas laterales introducidas con el vehículo parado en rampa lateral.

Los valores que se han impuesto a la rigidez del muelle y a la constante del amortiguador son $K=31.000 \text{ N/m}$ y $C=2.125 \text{ N}\cdot\text{s/m}$. Los valores de umbral para esta situación son $|\mathbf{v}| < 0,5 \text{ m/s}$ y $\alpha \geq 0,1 \text{ rad}$. Es necesario imponer una pendiente mínima a partir de la cual se emplean estas fuerzas para lograr el equilibrio del vehículo, puesto que, de lo contrario, podrían actuar en casos indeseados (por ejemplo, cuando el coche hace un trompo), alterando el comportamiento del vehículo. Se ha impuesto que las rampas tengan una inclinación superior a 5° .

3.5.5. Introducción de las Fuerzas.

Puesto que el modelo está definido con coordenadas naturales, las fuerzas se introducen a través del vector de fuerzas generalizadas. Este vector posee tantos elementos como coordenadas naturales se utilizan para el modelo. Las fuerzas generalizadas representan la “contribución” de una fuerza sobre cada punto o vector unitario. Cuando una fuerza se aplica directamente sobre un punto, no hay ningún problema, puesto que la fuerza se introduce en el vector de fuerzas generalizadas en la posición correspondiente a ese punto. Lo mismo ocurre cuando un par actúa sobre un ángulo ya definido. Sin embargo, cuando una fuerza actúa en un punto que no se corresponde con uno de los empleados en el modelo, es necesario representarla a través de un conjunto de fuerzas generalizadas actuando sobre el resto de los puntos.

Se comienza con el caso de una fuerza aplicada en un punto **P**, genérico, de un elemento, según se muestra en la Figura 3.18.

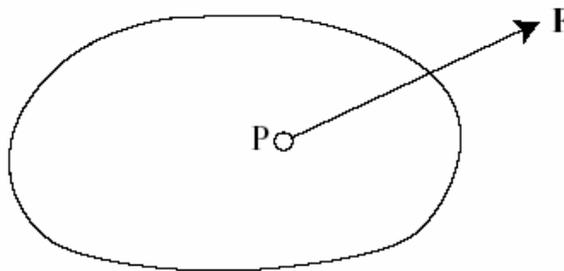


Figura 3.18 Elemento sometido a la acción de una fuerza.

Supongamos que el elemento se ha definido con un punto y tres vectores no coplanarios, según se observa en la Figura 3.19. La posición de un punto cualquiera del elemento puede escribirse de la siguiente forma,

$$\mathbf{r} = \mathbf{r}_1 + c_1\mathbf{u} + c_2\mathbf{v} + c_3\mathbf{w} = \mathbf{r}_1 + [\mathbf{u} \quad \mathbf{v} \quad \mathbf{w}]\mathbf{c} = \mathbf{r}_1 + \mathbf{Xc} \quad (3.50)$$

donde \mathbf{X} es una matriz 3x3 cuyas columnas son las componentes de los vectores \mathbf{u} , \mathbf{v} y \mathbf{w} , respectivamente,. La misma relación se verifica en coordenadas locales del elemento,

$$\bar{\mathbf{r}} = \bar{\mathbf{r}}_1 + c_1 \bar{\mathbf{u}} + c_2 \bar{\mathbf{v}} + c_3 \bar{\mathbf{w}} = \bar{\mathbf{r}}_1 + [\bar{\mathbf{u}} \quad \bar{\mathbf{v}} \quad \bar{\mathbf{w}}] \mathbf{c} = \bar{\mathbf{r}}_1 + \bar{\mathbf{X}} \mathbf{c} \quad (3.51)$$

de tal forma que el vector de coeficientes \mathbf{c} puede escribirse como,

$$\mathbf{c} = \bar{\mathbf{X}}^{-1} (\bar{\mathbf{r}} - \bar{\mathbf{r}}_1) \quad (3.52)$$

Estos coeficientes son constantes, a pesar del movimiento del elemento puesto que se han obtenido en coordenadas locales que, siendo el elemento indeformable, no variarán.

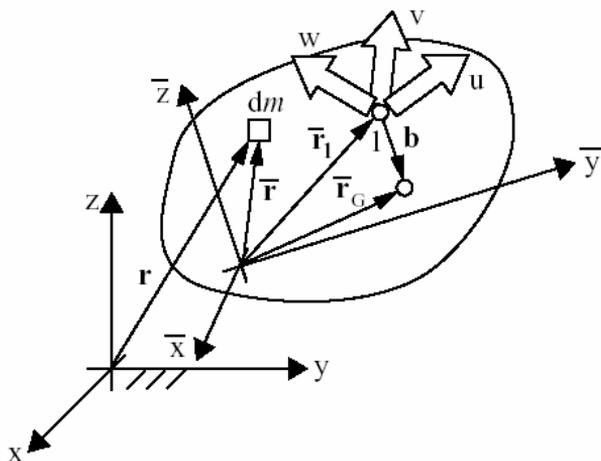


Figura 3.19 Elemento genérico tridimensional.

A partir de aquí, la posición de un punto cualquiera del elemento se puede expresar como

$$\mathbf{r} = [\mathbf{I} \quad c_1 \mathbf{I} \quad c_2 \mathbf{I} \quad c_3 \mathbf{I}] \begin{Bmatrix} \mathbf{r}_1 \\ \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \end{Bmatrix} = \mathbf{C} \mathbf{q} \quad (3.53)$$

donde \mathbf{I} es la matriz identidad 3x3. Entonces, al ser la matriz \mathbf{C} constante, la velocidad del punto genérico \mathbf{P} sobre el que se aplica la fuerza, puede

escribirse en función de las derivadas de las variables que definen el elemento como

$$\dot{\mathbf{r}}_p = \mathbf{C}_p \dot{\mathbf{q}} \quad (3.54)$$

siendo \mathbf{C}_p es una matriz que contiene los coeficientes que expresan la posición del punto \mathbf{P} en la base del elemento.

Para conocer el valor del vector de fuerzas generalizadas a que da lugar la fuerza puntual, se recurre a calcular la potencia virtual de la fuerza. Dos sistemas de fuerzas que produzcan idéntica potencia virtual son equivalentes desde el punto de vista de su efecto dinámico.

$$\dot{W} = \mathbf{F}' \dot{\mathbf{r}}_p = \mathbf{F}' \mathbf{C}_p \dot{\mathbf{q}} = \mathbf{Q}' \dot{\mathbf{q}} \quad (3.55)$$

identificando términos se tiene que,

$$\mathbf{Q} = \mathbf{C}_p' \mathbf{F} \quad (3.56)$$

Supongamos ahora que el elemento no se ha modelizado con un punto y tres vectores, sino con dos puntos, \mathbf{r}_1 y \mathbf{r}_2 , y dos vectores unitarios, \mathbf{u} y \mathbf{v} . Se puede escribir la siguiente relación entre las variables que se han utilizado en el desarrollo del vector de fuerzas, y las variables que realmente modelizan el elemento:

$$\begin{Bmatrix} \mathbf{r} \\ \mathbf{u} \\ \mathbf{v} \\ \mathbf{w} \end{Bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{Bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{u} \\ \mathbf{v} \end{Bmatrix} \quad (3.57)$$

que, de forma compacta, escribiremos como,

$$\mathbf{q}_0 = \mathbf{T} \mathbf{q} \quad (3.58)$$

donde \mathbf{q}_0 son las variables auxiliares, empleadas para el desarrollo, y \mathbf{q} son las que, de verdad, modelizan el elemento. Además, derivando,

$$\dot{\mathbf{q}}_0 = \mathbf{T} \dot{\mathbf{q}} \quad (3.59)$$

puesto que la matriz T es constante.

Ahora, para obtener el vector de fuerzas generalizadas, recurrimos nuevamente a la potencia virtual,

$$\dot{W} = \mathbf{Q}_0^t \dot{\mathbf{q}}_0 = \mathbf{Q}_0^t \mathbf{T} \dot{\mathbf{q}} = \mathbf{Q}^t \dot{\mathbf{q}} \quad (3.60)$$

donde \mathbf{Q}_0 es el vector de fuerzas obtenido cuando se utilizaba un punto y tres vectores, y \mathbf{Q} es el vector de fuerzas sobre la variables que realmente definen el elemento. Por tanto, identificando términos en la expresión anterior se llega a,

$$\mathbf{Q} = \mathbf{T}^t \mathbf{Q}_0 \quad (3.61)$$

que proporciona el vector de fuerzas generalizadas sobre el elemento en las coordenadas naturales que lo definen.

En el modelo del automóvil, se deben introducir las fuerzas en las ruedas delanteras y traseras, distinguiendo entre cada uno de estos dos casos, pues están definidos de diferente forma.

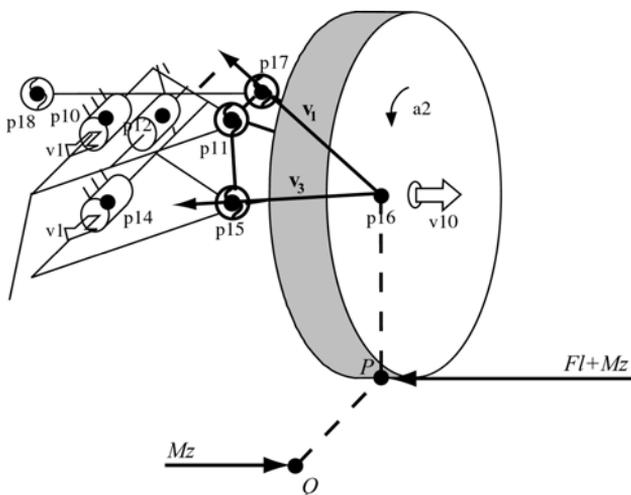


Figura 3.20 Base empleada para la introducción de las fuerzas del neumático en la rueda delantera izquierda.

En el caso de las ruedas delanteras se han empleado tres puntos y un vector. En la rueda izquierda se han utilizado los puntos 15, 16 y 17 y el

vector 10, como se indica en la Figura 3.20. De forma similar, para la rueda derecha se han considerado los puntos 25, 26 y 27 y el vector 20.

La fuerza lateral se introduce toda ella en el punto **P**. Para la introducción del momento, éste se descompone en un par de fuerzas. Una de ellas se aplica al punto **P** y la otra, que tiene como módulo únicamente el momento autoalineante, está aplicada en un punto auxiliar **Q**, a la distancia de un vector unitario en la dirección de avance de la rueda, tal como se representa en la Figura 3.20.

Para las ruedas traseras, tenemos dos puntos y dos vectores –los puntos 36 y 31 y los vectores 30 y 1 para la rueda izquierda, Figura 3.21, y los puntos 46 y 41 y los vectores 40 y 1, para la rueda derecha.

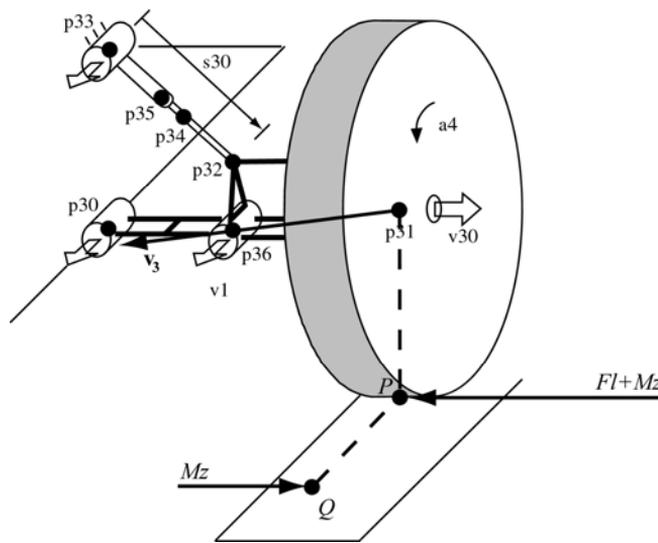


Figura 3.21. Base empleada para la introducción de las fuerzas del neumático en la rueda trasera izquierda.

Siguiendo la metodología expuesta anteriormente, una vez definido el sistema de referencia mediante un punto y tres vectores, se resuelve la ecuación (3.53), aplicada al punto **P**, o al punto **Q**

$$\mathbf{r}_p = \mathbf{C}_p \mathbf{q} \quad (3.62)$$

obteniéndose los coeficientes c_1 , c_2 y c_3 .

La matriz \mathbf{T} empleada en el caso de las ruedas delanteras es la siguiente:

$$\begin{bmatrix} \mathbf{p}_{16} \\ \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{p}_{17} \\ \mathbf{p}_{16} \\ \mathbf{v}_{10} \\ \mathbf{p}_{15} \end{bmatrix} \quad (3.63)$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & -\mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I} & \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (3.64)$$

y, para las ruedas traseras:

$$\begin{bmatrix} \mathbf{p}_{31} \\ \mathbf{v}_1 \\ \mathbf{v}_2 \\ \mathbf{v}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{p}_{31} \\ \mathbf{v}_1 \\ \mathbf{v}_{30} \\ \mathbf{p}_{36} \end{bmatrix} \quad (3.65)$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (3.66)$$

A continuación, se puede obtener ya el vector \mathbf{Q} de fuerzas generalizadas como

$$\mathbf{Q} = \mathbf{T}^T \mathbf{C}_p^T \mathbf{F} \quad (3.67)$$

La forma que toma el producto $\mathbf{T}^T \mathbf{C}^T$ en el caso de las ruedas delanteras es:

$$\mathbf{T}^T \mathbf{C}^T = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & -\mathbf{I} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ c_1 \mathbf{I} \\ c_2 \mathbf{I} \\ c_3 \mathbf{I} \end{bmatrix} = \begin{bmatrix} c_1 \mathbf{I} \\ (1 - c_1 - c_3) \mathbf{I} \\ c_2 \mathbf{I} \\ c_3 \mathbf{I} \end{bmatrix} \quad (3.68)$$

que nos proporciona, como expresión final para el vector de fuerzas:

$$\mathbf{Q} = \mathbf{T}^T \mathbf{C}^T \mathbf{F} = \begin{bmatrix} c_1 \mathbf{I} \\ (1-c_1-c_3) \mathbf{I} \\ c_2 \mathbf{I} \\ c_3 \mathbf{I} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \quad (3.69)$$

$$\mathbf{Q}^T = \left[c_1 \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \quad (1-c_1-c_3) \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \quad c_2 \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \quad c_3 \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \right] \quad (3.70)$$

En cuanto a las ruedas traseras:

$$\mathbf{T}^T \mathbf{C}^T = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & -\mathbf{I} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ c_1 \mathbf{I} \\ c_2 \mathbf{I} \\ c_3 \mathbf{I} \end{bmatrix} = \begin{bmatrix} (1-c_3) \mathbf{I} \\ c_1 \mathbf{I} \\ c_2 \mathbf{I} \\ c_3 \mathbf{I} \end{bmatrix} \quad (3.71)$$

que conduce finalmente a la expresión de la fuerza generalizada:

$$\mathbf{Q} = \mathbf{T}^T \mathbf{C}^T \mathbf{F} = \begin{bmatrix} (1-c_3) \mathbf{I} \\ c_1 \mathbf{I} \\ c_2 \mathbf{I} \\ c_3 \mathbf{I} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \quad (3.72)$$

$$\mathbf{Q}^T = \left[(1-c_3) \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \quad c_1 \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \quad c_2 \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \quad c_3 \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \right] \quad (3.73)$$

Estos vectores se ensamblarán en el vector de fuerzas total de todo el vehículo.

3.5.6. Resistencia a la Rodadura.

El contacto entre el neumático y el suelo no es puntual, sino que se produce sobre toda una zona de la banda de rodadura, denominada huella del neumático. La forma de la huella depende del neumático y de las

características del suelo sobre el que se produce la rodadura, puesto que ambos pueden deformarse.

Cuando el suelo no es deformable, la resistencia a la rodadura se debe a que, debido a una combinación de diversos fenómenos, la fuerza normal que el suelo ejerce sobre la rueda se encuentra adelantada con respecto al centro de la rueda, Figura 3.22. De esta forma, crea un par que se opone al movimiento.

Sin embargo, cuando el suelo es deformable, la normal entre el suelo y la rueda sí pasa por el centro de la rueda, y se encuentra inclinada en la dirección contraria al movimiento. Entonces, a pesar de que la normal no produce un par contrario al avance del vehículo, una componente de la normal se opone directamente a él.

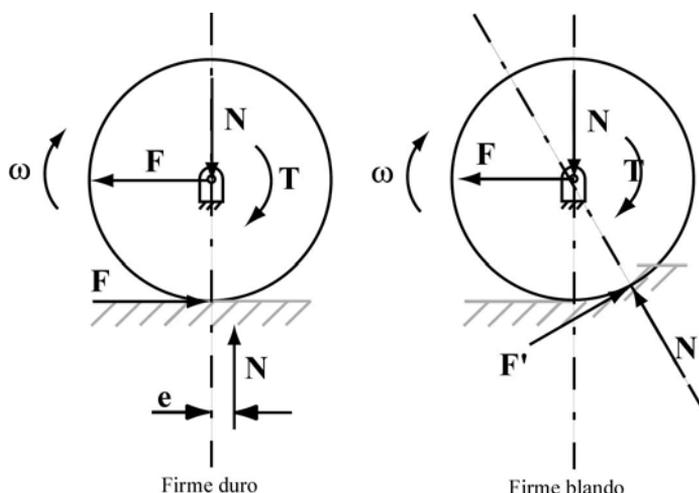


Figura 3.22 Resistencia a la rodadura en firme duro y blando.

Aunque el fenómeno de resistencia al avance en ambas ocasiones no se produzca de igual manera, en la práctica, para caracterizar este fenómeno se emplea el coeficiente de resistencia a la rodadura, f_r . Este coeficiente tiene su origen en el estudio del caso de firme blando, y viene dado por la relación:

$$f_r = \frac{e}{r_{def}} \tag{3.74}$$

siendo r_{def} el radio efectivo del neumático y e... En el modelo virtual del vehículo se considera la resistencia a la rodadura a través de este

coeficiente, y se introduce en la rueda un par contrario al movimiento, P_{rod} , de valor

$$P_{rod} = r_{def} f_r F_z \tag{3.75}$$

siendo F_z la fuerza de compresión del neumático [que el vehículo ejerce sobre la rueda]. En el simulador, se han tomado los valores de f_r indicados por Gillespie (1992) que se recogen en la Tabla 3.3.

Tipo de vehículo	Superficie		
	Cemento	Dureza media	Arena
Turismo	0,015	0,08	0,30
Camiones pesados	0,012	0,06	0,25
Tractores	0,02	0,04	0,20

Tabla 3.3 Valores típicos del coeficiente de rodadura.

3.6. Fuerzas de Aceleración y Frenado.

3.6.1. Curvas de Par del Motor.

Simular la fuerza de aceleración del vehículo supone representar lo más fielmente posible el comportamiento del motor que, como se recordará, pertenecía a un Chrysler 150 SX automático. En el presente caso se disponía de los datos mínimos para poder reconstruir las curvas de par-revoluciones del motor. Estos datos, proporcionados por el fabricante, son los siguientes:

- Potencia: 88 CV DIN a 5400 rpm.
- Par motor máximo: 14.0 m kg a 3000 rpm.
- Cambio de marchas automático.
- Numero de marchas de avance: tres.
- Relaciones internas de la caja de velocidades:

Marcha	Relación de velocidades, ρ
1ª	2,475:1
2ª	1,475:1
3ª	1:1

- Tipo del grupo: recto helicoidal, doble. Reducción: 3,673:1
- Desarrollo final de la transmisión: 29,4 km/h en 3ª teóricos sin resbalamiento.

A partir de estos datos se ha ajustado un polinomio de tercer grado que represente la curva de par frente a las revoluciones del motor. Para ello, se imponen las siguientes condiciones que permiten determinar los coeficientes de la ecuación:

- Par de 96 N·m a 1000 rpm.
- Par máximo de 137 N·m a 3000 rpm.
- Y par nulo a 6000 rpm, que correspondería en la tercera marcha a una velocidad de 200 km/h.

La ecuación resultante del par, donde n representa las revoluciones del motor, es

$$T = -9.9444 \cdot 10^{-10} n^3 - 3.2888 \cdot 10^{-6} n^2 + 0.046583 \cdot n + 53.7 \quad (3.76)$$

Esta curva de par se corresponde al par de máxima admisión, es decir, suponiendo que el acelerador está completamente pisado. Sin embargo, en el caso contrario, cuando el motor no recibe admisión de combustible, se opone al giro, presentando un par que se denomina par de contrapresión. En nuestro caso, la ecuación que describe este par de contrapresión es:

$$T_c = -15 \cdot 10^{-3} n \quad (3.77)$$

En la Figura 3.23 se representan los pares de máxima admisión y de contrapresión en función de las revoluciones del motor.

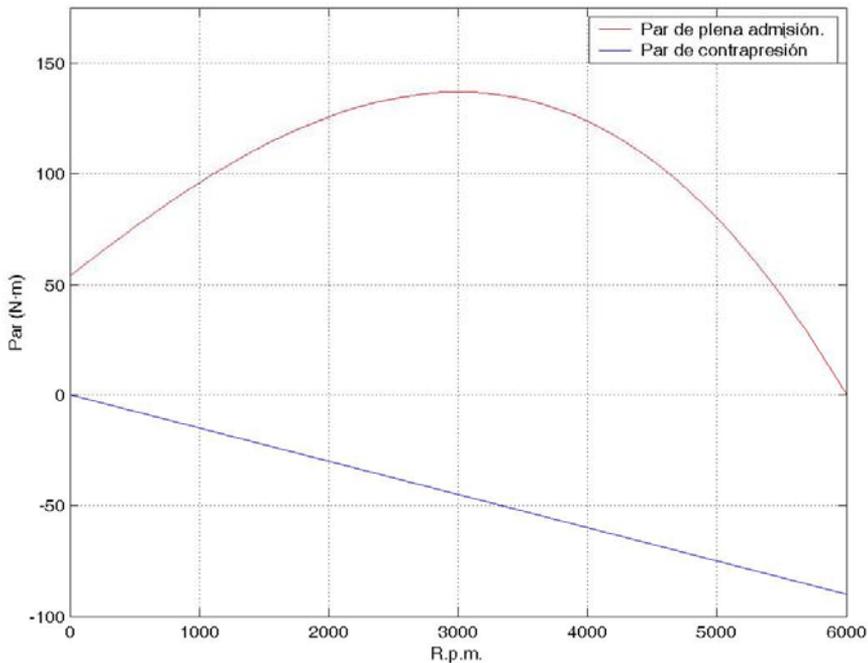


Figura 3.23 Curva par-revoluciones del motor.

En el modelo del vehículo se introduce directamente el par motor en las ruedas motrices que, en este caso, son las traseras. Este par, en un instante dado, depende de los siguientes parámetros: el par para máxima admisión, el par de contrapresión, el grado de admisión de combustible en el motor (que corresponde al desplazamiento del pedal acelerador), las revoluciones a las que gira el motor, y la marcha en la que se encuentra el vehículo.

En primer lugar, es necesario determinar el par que proporciona el motor y que es función del par motor, del par de contrapresión, y del grado de desplazamiento del pedal, f . El valor de f corresponde a 1 cuando el pedal está totalmente desplazado (en ese caso el par se corresponde totalmente con el par de plena admisión), y a 0 cuando no se acelera (entonces el motor opone la resistencia del par de contrapresión). La expresión del par entregado por el motor se puede escribir como:

$$T_{motor} = T \cdot f + T_c \cdot (1 - f) \quad (3.78)$$

A continuación, hay que tener en cuenta la transmisión y la relación de velocidades entre el giro del motor y el de las ruedas. Si despreciamos las pérdidas en la transmisión, y asumimos que la potencia en el eje del motor y el eje tractor es la misma, obtenemos:

$$T_{motor} \omega_{motor} = T_{rueda} \omega_{rueda} \quad (3.79)$$

El valor de la reducción entre el motor y la rueda varía con la marcha, y se define como:

$$\frac{\omega_{motor}}{\omega_{rueda}} = \frac{T_{rueda}}{T_{motor}} = \frac{1}{\rho} \quad (3.80)$$

donde ρ corresponde a la relación de velocidades que establece la caja de cambios.

Para determinar el par que el grupo motriz introduce en cada instante en la rueda, es necesario conocer las revoluciones de giro del motor a partir de la velocidad que, en ese instante, tiene la rueda. A partir de las ecuaciones anteriores se puede obtener ya la relación entre las revoluciones por minuto del motor, n , y la velocidad en el centro de las ruedas, v_{rueda} ,

$$n = \frac{60 \cdot \omega_{motor}}{2\pi} = \frac{60 \cdot \omega_{rueda}}{2\pi \cdot \rho} = \frac{60 \cdot v_{rueda}}{2 \cdot \pi \cdot r_{rueda} \cdot \rho} \quad (3.81)$$

Una vez que se conocen las revoluciones del motor, se calcula directamente, a partir de la curva de par y la ecuación (3.78), el par que proporciona T_{motor} . A partir de él, el par que se aplica al ángulo con el que se ha modelizado la rueda será:

$$T_{rueda} = \frac{T_{motor}}{\rho} \quad (3.82)$$

De forma orientativa, la Figura 3.24 representa las fuerzas que llegan a las ruedas para cada una de las marchas, dando los correspondientes valores a la relación de velocidades, ρ .

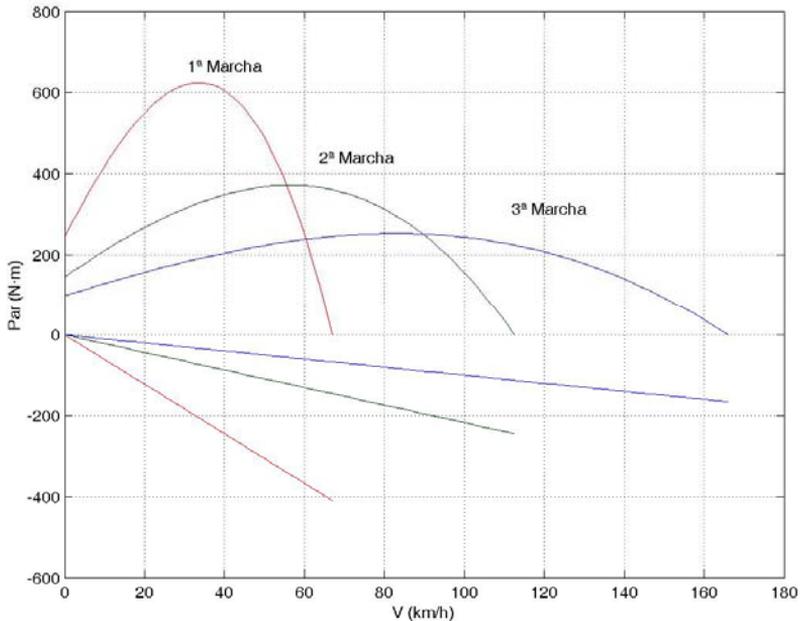


Figura 3.24 Relación fuerza-velocidad para cada una de las marchas.

3.6.2. Cambio Automático de Velocidades.

El motor que monta el vehículo real y que se ha modelizado, posee un cambio de marchas automático. En el apartado anterior se ha indicado cómo determinar las curvas de par correspondientes a las distintas velocidades. Como es sabido, la mayoría de los cambios automáticos realizan los cambios de marcha al llegar a un número determinado de revoluciones del motor. Por este motivo, el cambio se ha modelizado estableciendo las velocidades concretas en las que se produce cambio de marcha, tanto para pasar a una marcha más larga como para reducir la velocidad. La elección de los puntos en que se produce el cambio de velocidades en el modelo, se ha efectuado llevando a cabo pruebas con el prototipo real. En la Tabla 3.4 se recogen los rangos de velocidades establecidos para cada marcha.

En el modelo no se han tenido en cuenta los efectos dinámicos propios del grupo turbina-compresor que constituye la caja de cambios. Tampoco se han simulado algunos efectos típicos de las cajas automáticas, como puede ser la reducción al pisar a fondo el acelerador, porque no se ha visto necesario.

Marcha	Velocidad mínima (m/s)	Velocidad máxima (m/s)
1^a	0	9
2^a	9	16
3^a	16	-

Tabla 3.4 Rango de velocidades establecido para cada marcha.

Sí se ha recogido el efecto propio de los coches dotados de cambio automático, consistente en que es posible detener completamente el vehículo con una marcha puesta. Para ello, lo que se ha hecho ha sido establecer un par mínimo para el motor. A velocidades inferiores a 3 m/s el motor ejerce un par de 10 N·m. Para mantener el coche totalmente detenido es necesario pisar el pedal del freno. De esta forma, también se consigue que el vehículo arranque automáticamente al soltar el pedal de freno.

También es posible cambiar manualmente de marcha. Para ello, se han aprovechado los botones de que dispone el volante que se emplea en la interacción con el usuario en el modo simulador. Pulsando el botón de la derecha, la marcha se incrementa y, pulsando el botón izquierdo, disminuye. Este es un procedimiento típico empleado en el mundo de los videojuegos para el cambio de velocidades.

La marcha atrás, igual que en un vehículo real, sólo se puede introducir manualmente mediante los botones. El par motor que se ha establecido para ella, a falta de datos, ha sido el de la primera velocidad de avance.

3.6.3. Fuerzas de Frenado.

Como se ha indicado en el capítulo anterior, el vehículo está dotado de frenos de disco en las cuatro ruedas. Para simular su comportamiento, se ha calculado el par máximo que proporcionan los frenos a partir de las características geométricas de los discos.

El par proporcionado por un freno de disco, en el supuesto de presión uniforme, viene dado por la siguiente ecuación (Shigley and Mischke 2001):

$$T = \mu p \phi \frac{D^3 - d^3}{12} \quad (3.83)$$

donde:

- μ es el coeficiente de fricción. Se ha empleado un valor $\mu = 0,35$ que está dentro de los valores típicos para el contacto entre asbesto y hierro fundido.
- p es la presión máxima que se ejerce sobre la zapata. El valor que se ha supuesto es $p = 1000 \text{ kPa}$.
- ϕ es el ángulo abrazado por la zapata. En el presente caso, su valor es $\phi = 50^\circ$.
- D es el diámetro exterior del disco, en este caso, $D = 0,23 \text{ m}$.
- d es el diámetro interior, que toma un valor de $0,15 \text{ m}$.

Con esos valores, el resultado del par máximo es $T = 225 \text{ N}\cdot\text{m}$.

Como se recordará, para la modelización de la rueda se empleó el punto central y un ángulo. El par de frenado se aplica directamente sobre el ángulo y con sentido contrario al de giro de la rueda.

Sin embargo, al actuar el freno, no siempre lo hace con el máximo par posible, sino que el conductor regula la fuerza de frenado actuando sobre el pedal del freno. Este efecto se tiene en cuenta multiplicando la fuerza de frenado por un valor entre 0 y 1, que se corresponde con el desplazamiento del pedal: 0 corresponde a un desplazamiento nulo del pedal, y 1 al pedal desplazado a fondo.

3.6.4. Introducción de las Fuerzas.

El par de aceleración se introduce solamente en las ruedas traseras, sumándolo directamente a la fuerza generalizada correspondiente al ángulo de cada una de las ruedas. Los ángulos que modelizan la rueda están definidos de tal manera que el sentido positivo del ángulo coincide con el par correspondiente al avance del vehículo, y el negativo al de retroceso.

Si estos pares fuesen exteriores al sistema, bastaría con haberlos introducido directamente sobre los ángulos de las ruedas. Sin embargo, al tratarse de pares generados por elementos internos, se deben introducir también sobre el chasis del vehículo, puesto que es el encargado de absorber la reacción de estos pares. Sin embargo, en la modelización del chasis no se dispone de un ángulo sobre el que introducir directamente el valor del par, y es necesario modelizarlo de otra manera.

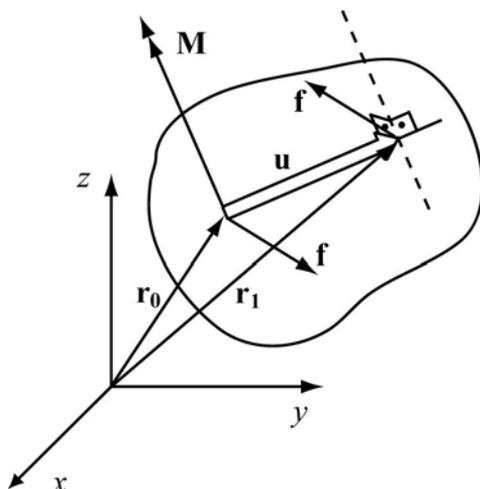


Figura 3.25. Momento actuante sobre un elemento, e introducción de un momento mediante una fuerza actuando sobre un vector.

La forma más directa para introducir un par de fuerzas es aplicándolo sobre una variable de ángulo. En el caso de que no se disponga de un ángulo como variable del problema, un par es equivalente a la introducción de dos fuerzas, \mathbf{f} , opuestas, de módulo igual al módulo del par, separadas entre sí por un vector unitario, \mathbf{u} , perpendicular al vector del par, Figura 3.25.

Para la introducción en el vector de fuerzas, el par se puede aplicar sobre el vector, \mathbf{u} , si este es una variable del problema. El vector \mathbf{u} , se puede escribir como:

$$\mathbf{u} = \mathbf{r}_1 - \mathbf{r}_0 \quad (3.84)$$

La potencia virtual de las fuerzas aplicadas sobre el vector se calcula como el producto de las fuerzas por las velocidades de sus puntos:

$$\dot{W} = \mathbf{f}^T \dot{\mathbf{r}}_1 - \mathbf{f}^T \dot{\mathbf{r}}_0 = \mathbf{f}^T (\dot{\mathbf{r}}_1 - \dot{\mathbf{r}}_0) = \mathbf{f}^T \dot{\mathbf{u}} \quad (3.85)$$

Como consecuencia, a la hora de introducir la fuerza en el vector de fuerzas generalizadas, se puede aplicar a las componentes del vector unitario puesto que el efecto de la fuerza \mathbf{f} aplicada a las componentes del vector \mathbf{u} es equivalente al momento \mathbf{n} actuando sobre el elemento, siempre que $\|\mathbf{f}\| = \|\mathbf{n}\|$.

Nos serviremos de esta propiedad, para introducir el momento como una fuerza aplicada a las componentes del vector \mathbf{v}_3 de la base unida al chasis del vehículo.

En el caso del par de frenado se procede de igual manera. El sentido del par de frenado es el contrario al movimiento de las ruedas. Como en el caso del par de aceleración, también se introducen los pares de frenado en el chasis como fuerzas aplicadas a las componentes del vector \mathbf{v}_3 .

El par de frenado presenta otra peculiaridad. A bajas velocidades, próximas al estado de reposo del vehículo, la introducción del par de frenado con el valor que le correspondería directamente, al calcularlo tal como se ha expuesto anteriormente, da problemas de convergencia numérica, puesto que es demasiado elevado. Por este motivo, se ha moderado mediante la introducción de un factor, f , del tipo

$$f = 2 \frac{v}{v_0} - \left(\frac{v}{v_0} \right)^2 \quad (3.86)$$

al igual que se había hecho en el caso de las fuerzas laterales. Se recordará que v_0 corresponde al valor de una velocidad umbral, que, en este caso, toma el valor de 0,1 m/s. Este factor hace que el par desarrollado por los frenos disminuya de forma cuadrática con la velocidad.

Este comportamiento de la fuerza de frenado, por otro lado, se asemeja a la forma real en que se comportan frenos. La fuerza que desarrollan depende de la fricción entre las pastillas y el disco, y ésta fricción no se produce si no hay una velocidad relativa entre los dos sólidos en contacto. Asimismo, se evita que se produzca, alrededor de una velocidad muy próxima al cero, una oscilación de la fuerza entre un sentido y otro, como podría ocurrir con valores de fuerza muy elevados.

3.7. Colisiones.

La interacción entre el vehículo y el entorno no se produce únicamente entre los neumáticos y el suelo. También está previsto que el coche pueda colisionar con el entorno y chocar con otros vehículos. Para ello, se ha realizado un sencillo módulo que detecta las colisiones e introduce las fuerzas pertinentes.

El módulo de colisiones consiste en ocho esferas situadas según se muestra en la Figura 3.26: tres en la parte delantera, tres en la trasera y otras dos más en la parte central de los laterales del coche, de tal forma que cubran prácticamente todo el perímetro lateral del vehículo. Las esferas tienen un radio de 0,45 m.

El algoritmo empleado para detectar la interacción entre el neumático y el suelo, descrito en el apartado 3.5.1, se emplea tal cual para detectar la interacción entre las esferas y los planos del entorno. Por simplificar el problema, sólo se han tenido en cuenta las colisiones de estas ocho esferas, aunque, como se puede observar en la Figura 3.26, no cubren toda la superficie lateral. Esto conlleva que un obstáculo pueda penetrar más en algunas partes del vehículo que en otras. A pesar de ello, el efecto logrado es realista.

Para introducir las fuerzas, las esferas se han considerado como resortes de constante $K = 100\ 000\ N/m$. Todos los resortes se han caracterizado con la misma rigidez, aunque se ha tenido en cuenta que las rigideces en las distintas bolas pueden ser diferentes. El módulo de la fuerza será el producto de la rigidez, K , por la deformación de la esfera, y la dirección es la del vector normal al plano que contacta con la esfera.

En el caso de las colisiones entre vehículos, el algoritmo es un poco distinto. En cada esfera de un vehículo se debe chequear la distancia a todas y cada una de las esferas de los demás vehículos. Este cálculo es más sencillo que en el caso de las distancias a los planos, puesto que basta con evaluar las distancias entre centros de las esferas.

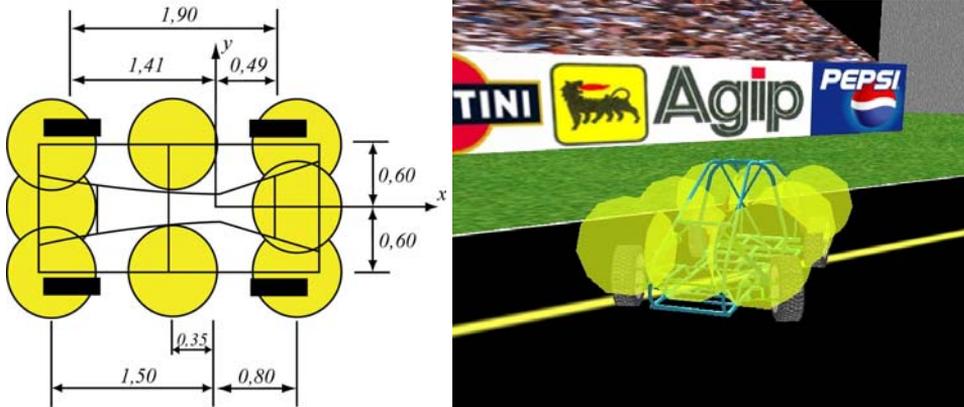


Figura 3.26 Situación de las esferas para la detección de colisiones.

Al colisionar las dos esferas, se ha supuesto que la dirección de la fuerza que actúa entre ellas es la del vector que une los dos centros. Como se ha tenido en cuenta la posibilidad de que la rigidez de las esferas no sea la misma, la fuerza resultante al chocar una esfera i con otra j se ha calculado como:

$$\mathbf{F} = \left(\frac{K_i \cdot K_j}{K_i + K_j} \right) \left(1 - \frac{d_{ij}}{\delta_{ij}} \right) \mathbf{u} \quad (3.87)$$

donde K_i y K_j son las rigideces, d_{ij} es la distancia entre centros a partir de la cual se produce penetración, δ_{ij} es la distancia real entre los centros, y \mathbf{u} es un vector unitario en la dirección que une los centros de las esferas.

Puesto que las esferas están situadas en el chasis, pero sus centros no están sobre ninguno de los puntos empleados para la modelización del vehículo, para introducir las fuerzas se debe seguir la sistemática expuesta en el apartado 0. Como se recordará, Figura 3.1, en la modelización del chasis se emplea un punto $-p1-$, y tres vectores unitarios $-v1$, $v2$ y $v3-$, que son los que se emplean para aplicar las fuerzas, junto con los valores de combinación lineal de los centros de las esferas expresados en ese sistema de referencia.

Suponer que las esferas se comportan como resortes implica que, en el momento en que se produce una colisión, se deba tener en cuenta la contribución de los resortes de las esferas implicadas a la matriz de rigidez del sistema. Nuevamente, la introducción de estos valores en la matriz de rigidez se hace siguiendo las ecuaciones (3.27) y (3.29). El hecho de que

los puntos en que están situados los centros de las esferas no pertenezcan a la modelización del chasis, hace que se deban tener en cuenta los coeficientes de combinación lineal de los puntos en la base de referencia, por lo que la ecuación (3.29), para el caso de una esfera situada en el punto A , se expresa ahora de la siguiente manera

$$\mathbf{K}_m = -\frac{\partial \mathbf{F}_m}{\partial \mathbf{r}_A} = \mathbf{C}_A^T K (\mathbf{n} \cdot \mathbf{n}^T) \quad (3.88)$$

donde \mathbf{C}_A es la matriz que contiene los coeficientes de combinación lineal del punto A en la base de referencia empleada en la modelización.

3.8. Fuerzas Aerodinámicas.

La incorporación de fuerzas debidas al viento, o simplemente al propio avance del vehículo, aporta un mayor grado de realismo a la simulación. Por este motivo, se ha pretendido dotar al simulador de la posibilidad de representar este tipo de fenómenos. En lugar de la geometría propia del chasis de tubo del prototipo real cuyo comportamiento aerodinámico se ignora, se ha tenido en cuenta la geometría de un vehículo comercial, para simular estos efectos que dependen, en gran medida, de la superficie que el chasis presenta a la corriente de aire. Dentro de este apartado de fuerzas aerodinámicas se ha considerado únicamente la resistencia aerodinámica. El efecto de la incidencia del viento sobre cualquiera de las superficies no se ha incluido finalmente, aunque se ha desarrollado todo lo relativo a la determinación de la superficie expuesta al viento. En este aspecto, queda pendiente incluir un generador de viento, e introducir las fuerzas originadas por el empuje del mismo.

La resistencia aerodinámica es la oposición que presenta aire al avance del vehículo. Debido a esto, su dirección será perfectamente conocida en todo momento, dado que es la dirección opuesta a la de la velocidad instantánea del vehículo. El módulo de la fuerza de resistencia aerodinámica sobre un vehículo cualquiera responde a la siguiente ecuación (Luque et al, 2004):

$$F_x = \frac{1}{2} \rho C_x S v^2 \quad (3.89)$$

En esta expresión, S representa la superficie sobre la que incide el aire, ρ es la densidad del aire, v la velocidad instantánea del vehículo, y C_x corresponde al coeficiente aerodinámico. Un valor de C_x igual a 1 es el que opone una superficie plana enfrentada contra el aire. El coeficiente aerodinámico, C_x , depende de la geometría del chasis estudiado. Pequeños detalles, como el enrasado de las ventanillas, las formas de los vierteaguas, etc., afectan al valor de este coeficiente. El valor del coeficiente aerodinámico se determina mediante ensayo en un túnel de viento y suele tomar valores alrededor de $C_x = 0,35$ en turismos normales. Para el simulador se ha empleado este valor, que corresponde con el del vehículo que se ha considerado como modelo. El área frontal del vehículo que se ha utilizado, proporcionada por el fabricante, es $S = 1,92 \text{ m}^2$.

La International Standard Atmosphere (ISA) es un modelo atmosférico terrestre creado por la Organización de Aviación Civil Internacional, que proporciona un valor para la densidad del aire, ρ , al nivel del mar de $1,225 \frac{\text{kg}}{\text{m}^3}$. Este es el valor que se ha implementado en la simulación para la densidad del aire.

La fuerza procedente de evaluar la ecuación (3.89) se introduce directamente sobre el punto 1 del chasis, con dirección opuesta a la velocidad del vehículo.

4.

Simulador de Conducción con Control Humano

4.1. Organización del Simulador.

El modelo que se ha detallado en el capítulo anterior se ha empleado para crear un simulador de conducción. Empleando únicamente el modelo computacional, es posible imponer al vehículo una maniobra determinada y obtener unas gráficas con los datos que se deseen. Sin embargo, una vez que se dispone de un modelo matemático del vehículo, la posibilidad de conducirlo a través de un escenario, del mismo modo que un vehículo real, y poder ver su comportamiento, resulta de mucha utilidad. En primer lugar, para valorar el comportamiento de la simulación del vehículo observando su respuesta ante una maniobra determinada. Comportamientos anómalos son detectados inmediatamente y pueden ser corregidos con facilidad. También facilita la búsqueda de errores de programación ejecutando el programa en modo de depuración.

En el presente trabajo, se propone abordar el diseño de controladores para que el vehículo realice autónomamente algunas maniobras. El simulador facilita el diseño de estas maniobras. Una vez que se ha determinado la maniobra, es posible que un conductor la lleve a cabo,

identificando posibles dificultades, y evitando así que se le pida al controlador algo que es totalmente imposible para un conductor real. O, si se le pide, que sea conociendo la dificultad que entraña.

En este mismo sentido, al disponer del simulador, es posible comparar los resultados obtenidos por el controlador y por un conductor. De esta manera, se evidencia la eficacia del controlador y las mejoras que introduce. Además, durante el proceso de diseño de los controladores, es posible tener en cuenta el modo de proceder del conductor.

Otra facilidad que proporciona el simulador es que hace posible diseñar maniobras en las que el conductor y un controlador interactúen simultáneamente. De esta forma, se puede observar la reacción del conductor ante el comportamiento automatizado del vehículo, probar controles para algunos sistemas del vehículo, como el sistema de frenos, los sistemas de control de tracción, etc.

Desde el punto de vista del análisis de la formulación multicuerpo utilizada, el simulador permite evaluar sus prestaciones en el contexto de una aplicación concreta, de tiempo real, que va más allá de la mera ejecución del modelo computacional en un ordenador, ya que implica también una salida gráfica realista, entradas por periféricos, ...

En general, un simulador de conducción es un dispositivo capaz de reproducir el comportamiento total o parcial del vehículo, empleando uno o varios ordenadores. Su aplicación puede ser variada, desde el diseño de automóviles hasta el entrenamiento y reciclaje de conductores.

Considerados como dispositivos de entrenamiento, los simuladores de conducción se pueden clasificar en tres grandes sistemas (Brazález, 2002): los dispositivos de entrenamiento mediante ordenador, más conocidos como CBT (Computer Based Training), los simuladores de tareas parciales o PTS (Part Task Simulator), y los simuladores completos de entrenamientos o FTS (Full Training Simulator).

Un CBT suele estar compuesto por uno o varios PCs que reproducen los instrumentos de la cabina, el puesto de conducción o los esquemas de los subsistemas a estudiar. No pretenden una sensación de inmersión completa en el entorno y, por este motivo, la interacción con el ordenador no es muy realista. Su uso puede centrarse en proporcionar un conocimiento detallado de los subsistemas que conforman el vehículo o un chequeo previo a la conducción real.

Los PTS incorporan los controles de la cabina del vehículo pero suelen ser de carácter genérico, sin representar un vehículo concreto. También se caracterizan por poseer un campo limitado de visión, sin lograr una inmersión total del conductor. Se suelen emplear para las etapas iniciales de entrenamiento.

Finalmente, los FTS suelen hacer una réplica completa del modelo a representar (Freeman et al. 1995). En ocasiones, llegan a incorporar la totalidad del vehículo (Heydinger et al. 2002 y Reymond et al. 2000). Disponen de sistemas de movimiento, generalmente plataformas dotadas de 6 grados de libertad, que replican la sensación de movimiento mediante algoritmos denominados *washout*. Poseen sistemas de visión completa. Reproducen la mayor parte de las tareas que se pueden llevar a cabo en el vehículo real. Estos simuladores deben cumplir forzosamente dos requisitos: ejecutarse en tiempo inferior al real, y que reproduzcan el vehículo real con una fidelidad suficiente como para convencer al conductor de estar conduciendo en el mundo real.



Figura 4.1 Distintos tipos de simuladores de conducción.

El simulador desarrollado podría encuadrarse dentro de los CBTs, aunque posee características de cada uno de los tres grupos: funciona en tiempo real, emplea un único ordenador, posee los controles del puesto de conducción de una manera genérica, el campo de visión es limitado, y el grado de inmersión no es muy grande puesto que no se simula el movimiento del puesto de conducción.

El modelo recibe los inputs provenientes del conductor a través de un volante y unos pedales, del tipo de los empleados en los juegos de ordenador. Para la representación gráfica, se han desarrollado varios entornos con diversas finalidades:

- Un entorno totalmente plano, sin obstáculos, con la única representación de la trayectoria a seguir, que se ha empleado en el diseño de controladores.

- Un entorno con obstáculos en el que el vehículo pueda realizar maniobras violentas, con la finalidad de verificar la robustez del simulador.
- Una representación del Circuito de Cataluña. Este entorno permite realizar la simulación simultánea de tres vehículos, uno conducido por un conductor real y los dos restantes guiados por controladores, y constituye una buena herramienta para evaluar la eficiencia del simulador.
- Una recreación de un barrio de Ferrol, para disponer de un entorno realista. Este entorno incluye los mapas de alturas tanto de las calles como de los edificios, texturas, etc.

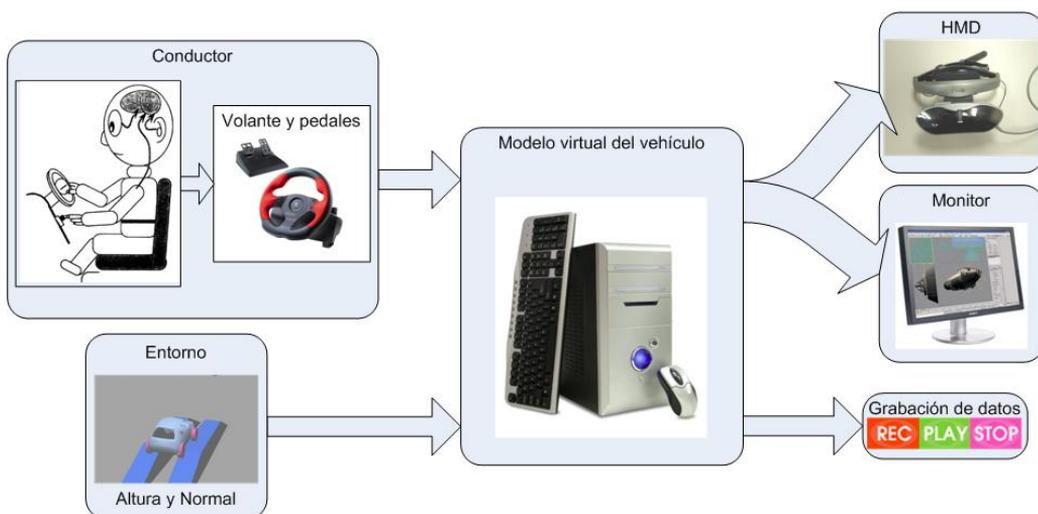


Figura 4.2 Esquema del simulador

La organización del software del simulador está reflejada en la Figura 4.3. Las subrutinas INI y CALCULO corresponden al modelo virtual del prototipo que se ha detallado en el capítulo anterior. Como se ha mencionado, estas subrutinas se han programado en Fortran. La función principal que aglutina todas las subrutinas del esquema está programada en C. Algunas de las subrutinas, como la que gobierna la conexión del simulador con el sensor que detecta la orientación de la cabeza empleado por las gafas de realidad virtual, y las subrutinas para cálculos relativos al entorno gráfico, están programadas en C++. El lenguaje C++ ha sido particularmente ventajoso para los cálculos relativos al entorno, puesto que permite definir clases y operaciones de los elementos de las clases, simplificando el manejo de la gran cantidad de información de puntos y vectores relativa al entorno. Por

este motivo, en el esquema de flujo se han resaltado aquellas subrutinas, como las referentes a las colisiones o las fuerzas sobre los neumáticos, en las que Fortran y C++ se entrelazan.

Como compiladores se han empleado Compaq Visual Fortran 6.5 y Microsoft Visual Studio 6.0. Estos compiladores comparten el entorno del Microsoft Developer Studio, y facilitan la programación en lenguajes mixtos. Este último factor, junto con el interfaz de programación sencillo, ha tenido gran relevancia a la hora de la elección de estos compiladores.

En las siguientes secciones, se describen con mayor detalle el esquema de funcionamiento del programa y los dispositivos empleados para la interacción entre el usuario y el simulador.

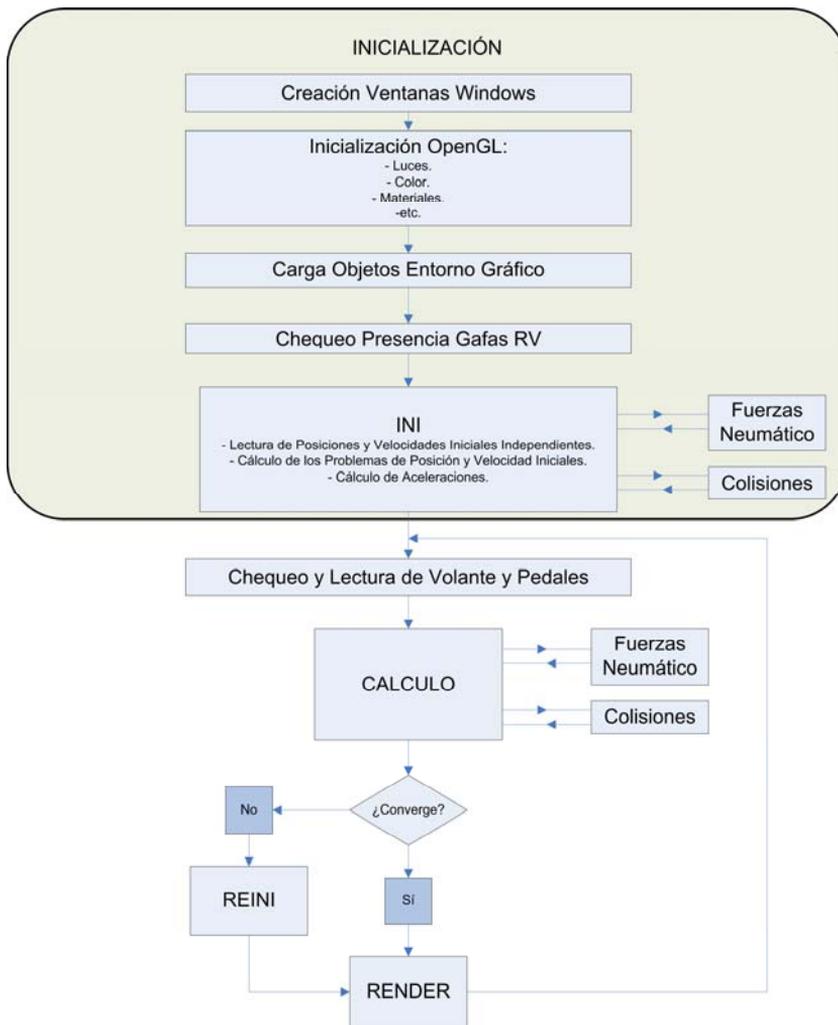


Figura 4.3 Organización del Software del Simulador.

4.1.1. Desarrollo del Programa.

En el esquema de funcionamiento del simulador se han distinguido las partes correspondientes a INI y CALCULO porque responden, además, a situaciones distintas del simulador. En el primer bloque, que corresponde a la inicialización, además de verificar los dispositivos que en ese momento se encuentran conectados (volante y pedales, HMD) se realiza la lectura de las posiciones y velocidades iniciales independientes. A partir de ellas, se realiza el cálculo de los problemas de posición y velocidad inicial, y el cálculo de aceleraciones iniciales. Las condiciones que se han impuesto son de reposo y, para el problema de posición inicial, se ha exigido una tolerancia de 10^{-14} m, entendida como la diferencia entre las posiciones obtenidas en dos iteraciones sucesivas durante la resolución del correspondiente sistema no lineal (Newton-Raphson).

A continuación, el programa pasa a realizar los cálculos del vehículo con un paso de tiempo de 10 ms. Como se puede ver, recibe las fuerzas del entorno a través de las ruedas y de las posibles colisiones con otros vehículos, con el entorno, etc. En el problema dinámico, se abandona la iteración para la resolución del sistema cuando la variación entre la posición resultado y la de la iteración anterior es igual o inferior a 10^{-5} m. Sin embargo, en ocasiones, debido a la exigencia de una maniobra o a la magnitud de las fuerzas que pueden sobrevenir en un instante determinado, es posible que el programa no alcance la convergencia en un número razonable de iteraciones (se ha tomado un valor de 10 iteraciones de manera que se asegure el tiempo real en la simulación). En ese caso, a partir de las posiciones y velocidades independientes, se vuelven a resolver los problemas de posición y velocidad, como se hizo en el instante inicial, y se realiza el cálculo de aceleraciones. Se da por válida una variación en las posiciones respecto a la iteración anterior de 10^{-10} m. Este paso es realizado en la subrutina denominada REINI.

4.1.2. Sincronización Temporal de la Simulación.

Un aspecto de vital importancia en el simulador es mantener la ejecución del programa en tiempo real. Además, tan importante como mantener el tiempo de simulación es que la representación gráfica no esté retrasada respecto del cálculo, que las acciones del conductor sobre el volante y los pedales sean leídas en el instante adecuado, y que su representación

gráfica se haga sin retrasos (por ejemplo, que al girar el volante giren las ruedas, sin que haya desfase de tiempo).

Para este fin, se ha optado por primar la correcta sincronización de la simulación con el tiempo real sobre todos los demás aspectos de ésta. Así, en cada paso del bucle principal del programa simulador, se comprueba el desfase temporal entre el cálculo de la simulación y el tiempo real. Si el cálculo está retrasado con respecto al tiempo real y el desfase es superior a un paso de tiempo, dt , no se realiza la representación gráfica en ese instante de tiempo, y se continúa calculando hasta que la diferencia con el tiempo real sea inferior a un paso de integración. En ese instante, se representa gráficamente el estado del sistema. Si el cálculo se realiza en mucho menos tiempo del establecido para un paso de tiempo de integración, no se permite que el programa *adelante* al tiempo real. Mientras la diferencia entre ambos tiempos sea inferior a un paso de tiempo de integración, el programa representa el estado del sistema tantas veces como sea posible.

De este procedimiento se derivan dos consecuencias, dependiendo de si el computador empleado es lo suficientemente rápido para mantener la simulación en tiempo real o no:

1. Si el ordenador es más rápido de lo necesario, no se calculará ningún paso nuevo de integración, aunque sí se representará el estado del entorno, que resulta ser idéntico al del *frame* anterior.
2. Si el ordenador no es capaz de ejecutar el cálculo en el tiempo especificado, ejecutará tantos pasos de integración como sean necesarios para alcanzar el instante real actual y, una vez conseguido esto, representará el estado del entorno. En casos extremos, el intervalo de tiempo entre dos actualizaciones de la pantalla puede ser excesivo, pero se garantiza que la simulación se ejecuta en tiempo real.

El diagrama de flujo que ilustra el procedimiento es el que se muestra en la Figura 4.4:

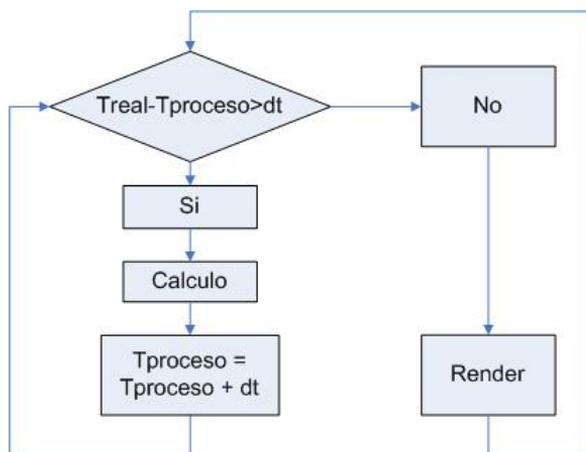


Figura 4.4 Proceso de Control de Tiempo de Simulación.

El Tiempo Real es obtenido mediante la función `LeerRelojPrecision`, que nos proporciona el tiempo del sistema. Por este motivo, debe ser corregido restando el tiempo en el que comenzó la simulación. Puesto que el Tiempo de Proceso es conocido, a continuación se ejecutan tantos pasos de integración como sean necesarios hasta que el tiempo de desfase se haya reducido a una magnitud inferior a la de un paso de integración, dt .

4.2. Input: Volante y Pedales, Datos Topográficos.

Como se acaba de ver, los inputs que recibe el sistema son de dos tipos: los que provienen del usuario y los relativos a la interacción con el entorno. Las entradas que el modelo debe recibir del usuario son las correspondientes a la posición del volante, los pedales del acelerador y el freno, y la posición de la palanca de cambios. Además, es muy conveniente poder cambiar el punto de vista desde el que se observan el coche y el escenario. Para ello, se ha habilitado el control de las cámaras, o puntos de vista a desde los que se observa la simulación, mediante una entrada externa que puede ser manipulada por el conductor.

Se han empleado un volante y unos pedales marca Logitech, de los que habitualmente se utilizan en los juegos de ordenador para realizar estas funciones. Estos dispositivos disponen de un volante dotado de varios botones y dos pedales, uno para el acelerador y otro para el freno.

Los botones del volante se han aprovechado para realizar distintas funciones. Los dos botones situados en la parte anterior izquierda se

emplean para acercar y alejar el punto de vista de la cámara. Los dos situados a la derecha cambian la situación de las cámaras. Como se verá más adelante, se han programado seis posibles posiciones para las cámaras. Finalmente, dos botones situados en la parte posterior del volante sirven para manejar la palanca de cambios. Pulsando el botón de la derecha se incrementa la marcha seleccionada, mientras que el botón de la izquierda disminuye la marcha. Esta distribución de las funciones de los botones está muy extendida en los juegos de ordenador.



Figura 4.5 Volante empleado en el simulador de conducción.

Para la lectura de datos del volante desde el PC, se ha empleado la librería DirectInput de DirectX. En esta librería se dispone de los comandos necesarios para gestionar cualquier dispositivo de entrada y salida al PC, y en nuestro caso hemos utilizados los comandos propios de un volante y unos pedales.

En cuanto a la interacción con el entorno virtual, el modelo debe recibir la altura y la normal del suelo en el punto de contacto. En este sentido, la forma de modelizar el entorno juega un papel muy importante. Debe adaptarse a formas complejas y permitir representar superficies muy irregulares. Respetar las formas geométricas que aparecen en una superficie topográfica real puede suponer una división del espacio demasiado minuciosa, que se traduciría en tiempos de cálculo muy elevados. Además, la representación gráfica debe permitir que a un mismo punto del plano le correspondan dos o más posibles alturas, como sucede en el caso de edificios, túneles, intersección de viales, etc.

La solución que se ha adoptado es sencilla, y cumple con los anteriores requisitos. Consiste en la aplicación de mallas tridimensionales compuestas por triángulos, que pueden adquirir cualquier orientación en el espacio. De esta forma, la malla poligonal puede describir cualquier tipo de superficie,

puesto que cualquier forma es susceptible de ser aproximada mediante la triangularización de su contorno.

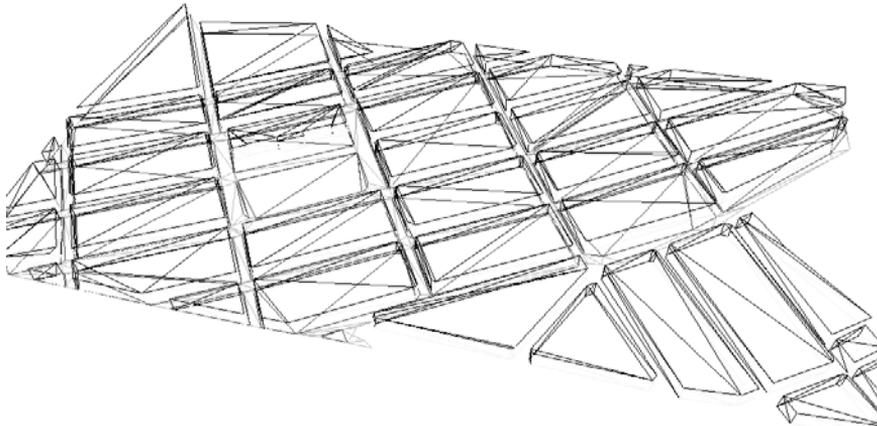


Figura 4.6 Imagen de detalle de la triangularización del entorno gráfico.

La triangularización no sigue ningún patrón rígido: el tamaño y la forma de los triángulos puede ser cambiante. Esto permite que se pueda representar cualquier detalle con la precisión que éste merezca en una zona determinada, sin añadir complejidad al resto de las zonas.

Al no existir imposiciones previas para la localización o el tamaño de los triángulos, éstos pueden ser situados en cualquier punto del espacio, incluyendo aquellos que están situados por encima o por debajo de otros polígonos.

Sin embargo, una desventaja de esta modelización frente a otras (por ejemplo, discretizar el entorno mediante una cuadrícula) es que la búsqueda de los polígonos que pueden contactar con el vehículo ya no es sencilla, debido a que la triangularización no sigue ningún patrón para adaptarse con más fidelidad al terreno. A cambio, esta flexibilidad de la división del espacio mediante triángulos, hace que sea posible representar grandes áreas mediante muy pocos triángulos cuando se trata de superficies muy lisas. Esto compensa de alguna manera el incremento de dificultad en el algoritmo para encontrar el punto de contacto entre el vehículo y el terreno. Además, siempre es posible dividir el entorno, cuando sea muy extenso, en sectores limitando el universo de la búsqueda del punto de contacto. El algoritmo de búsqueda del punto de contacto ha sido descrito en la sección 3.5.1 al detallar la introducción de las fuerzas en el neumático.

4.3. Output: Entorno Gráfico, Gafas de Realidad Virtual, Grabación de Maniobras.

Se ha dotado al simulador de tres posibles salidas de información. En primer lugar, se dispone de un entorno gráfico de representación que es posible visualizar de dos maneras: en la pantalla del ordenador o mediante unas gafas de realidad virtual (Head Mounted Display o HMD). Mediante estas dos opciones es posible visualizar el comportamiento instantáneo del vehículo. Pero también es necesario poder almacenar datos y extraer gráficas de las maniobras realizadas. Para ello, el simulador posee la capacidad de grabar y reproducir maniobras, almacenando los datos en un fichero. A partir de los datos del fichero, es posible trazar gráficos o bien ejecutar nuevamente la maniobra, visualizándola en el propio entorno del simulador tantas veces como se considere oportuno.

No hay que perder de vista que lo que se pretendía era desarrollar un entorno para la simulación, en el que se pudiera probar y visualizar el comportamiento del vehículo. En este sentido, el entorno más sencillo es un plano sobre el que se mueva el vehículo. Pero, sin embargo, con tanta sencillez se pierden muchos datos (por ejemplo, se carece por completo de sensación de velocidad), aunque ese entorno, como se verá es suficiente para varias aplicaciones. Pero para poder observar el comportamiento del modelo de vehículo, su maniobrabilidad, y la robustez del método, es conveniente que el entorno disponga de rampas, obstáculos, etc.

Para llevar a cabo la representación gráfica de un entorno tridimensional como el que se emplea en el simulador existen dos interfaces de programación de aplicaciones –API’s- en la actualidad: OpenGL (Schreiner et al. 2003) y DirectX (Snook 2003). Vamos a examinar y comparar ambas API’s.

4.3.1. OpenGL frente a DirectX.

La primera diferencia es que DirectX no sólo posee un componente de representación gráfica. DirectX incluye también aplicaciones para sonido, entrada de datos, música, trabajo en red y otras aplicaciones multimedia. Sin embargo, OpenGL es estrictamente una interfaz gráfica.

Ambos, tanto OpenGL como DirectX, utilizan lo que se denomina sistema tradicional de *Rendering Pipeline*, y que indica un orden en la serie de

pasos que se llevan a cabo para el renderizado –proceso mediante el cual se discretiza un volumen mediante un mallado y se proyecta en un plano para representarlo en pantalla-, una serie de plataformas de proceso y este *Pipeline*. Y este sistema es el mismo que se viene utilizando en los gráficos 3D desde los comienzos de los gráficos por ordenador, que aunque han sido modificados con los avances del hardware, han mantenido el esquema básico.

En lo que se refiere al modo de definir los objetos en el espacio, ambos sistemas describen los vértices de los elementos como un conjunto de datos, que definen tanto la localización del vértice como otros datos relacionados con sus características. La caracterización de lo que se denominan *primitivas gráficas* –puntos, líneas, triángulos, polígonos– se realiza del mismo modo, es decir definiendo un conjunto ordenado de vértices. La diferencia en este apartado entre las API's estriba en cómo son manejados los vértices para crear las primitivas.

Hay un gran número de diferencias y similitudes técnicas entre los dos sistemas pero, en conclusión, con respecto a esta disyuntiva, sobre la cual ya se ha debatido mucho en foros especializados, quizás no se pueda establecer una valoración para inclinarse por una u otra opción en cuanto a prestaciones técnicas, es decir, tiempo de ejecución, uso de recursos de memoria o efectos gráficos.

En definitiva, lo que ha inclinado la balanza hacia la plataforma OpenGL se resume en dos características no técnicas que posee OpenGL y no DirectX, que son la portabilidad y la sencillez formal:

- OpenGL es una API que es portable a otros sistemas operativos como Unix, Macintosh, etc., es decir, las mismas llamadas a funciones seguirían siendo válidas en estos otros sistemas, mientras que DirectX posee unas librerías que solo están preparadas para el entorno Windows.
- OpenGL dispone de un entorno de trabajo mucho más claro e intuitivo, frente a la complicación formal de las llamadas a las funciones de DirectX.

OpenGL (Open Graphics Library) es una librería de funciones para la generación de gráficos 2D y 3D. OpenGL no genera imágenes altamente realistas, pero consigue muy buenas imágenes a un bajo coste computacional, siendo apropiado cuando se requiere una velocidad de refresco elevada, que es lo que se busca en este caso. OpenGL se usa

para una amplia variedad de fines, desde las aplicaciones CAD de ingeniería y arquitectura, o la creación de videojuegos de última generación, a la realización de efectos especiales en películas.

OpenGL soporta hardware 3D, y es altamente recomendable poseer este tipo de hardware gráfico. Si no se dispone de él, las rutinas de representación correrán por software; esto quiere decir que, si existe una tarjeta gráfica aceleradora instalada en el ordenador, la gestión de las operaciones encaminadas a la representación gráfica se realizarán en el procesador de la propia tarjeta, liberando de trabajo al microprocesador principal. De otro modo, el microprocesador llevaría a cabo estas operaciones, disminuyendo en gran medida la velocidad de las rutinas de representación y, al mismo tiempo, cualquier otro proceso de cálculo del que se estuviese encargando, como es el caso de los cálculos de integración numérica de las ecuaciones del movimiento del modelo que se realizan paralelamente a la representación gráfica.

4.3.2. Generación de Entornos.

Una vez escogido OpenGL para la creación de los entornos, todo el proceso para su generación es un tanto complejo. Lo podemos dividir en los siguientes tres apartados:

- Diseño de los entornos.
- Almacenamiento de los datos del entorno.
- Gestión y representación del entorno en el simulador.

Para el diseño de los entornos se ha buscado un procedimiento que permita partir de planos cartográficos y diseñar otros entornos, menos complejos, de una manera sencilla y rápida. Para el desarrollo de entornos 3D, existen programas específicos como puede ser Multigen y 3D Studio. Estos programas permiten crear entornos muy complejos y detallados.

Muchas veces, los planos de ciudades de los que disponen los Ayuntamientos han sido realizados con AutoCAD. En concreto, se disponía de unos planos en este formato de la ciudad de Ferrol facilitados por el Ayuntamiento. Estos planos se han empleado para generar un entorno realista, correspondiente a parte de un barrio de la ciudad, con texturas y la topografía real del barrio. Además, para generar otros entornos sencillos, con formas geométricas simples como cubos, pirámides, etc., AutoCAD es una herramienta fácil de manejar. Por este motivo, ha sido la herramienta

empleada para el diseño de entornos, aunque no sea específica para estos fines. Sin embargo, permitía respetar los dos fines planteados: emplear los planos proporcionados por el Ayuntamiento, y desarrollar los otros entornos sin tener que hacer grandes inversiones de tiempo y dinero.

Los planos del entorno están divididos en capas con datos para las distintas partes a representar. En los planos del barrio, la división entre capas está hecha con un criterio funcional, distinguiendo la nomenclatura de las calles, la delimitación de las zonas verdes, la ubicación de los postes de transporte eléctrico, buzones, luminarias, arquetas, etc. Esta división en capas permite seleccionar aquellas de interés para la representación gráfica, y el tipo de tratamiento que se le dará a cada capa. Se han tenido en cuenta capas para fachadas, aceras, calzada y, en los entornos sencillos, rampas.

La capa correspondiente a la calzada se representa tal cual. Sin embargo, las aceras, en el caso de que existan en ese entorno, se crean mediante una extrusión. Como veremos más tarde, la capa de aceras contiene la información correspondiente a los contornos de las aceras. Este contorno es extruido una altura dada sobre la superficie correspondiente a la calzada. Los edificios se generan igualmente mediante extrusión a partir de su contorno, que está almacenado en la capa correspondiente. La capa denominada rampas se emplea para la colocación de obstáculos sobre la calzada. Todas las superficies en cada una de las capas deben estar formadas por contornos perfectamente cerrados.

Como hemos visto en el apartado 4.2, es necesario que las superficies estén representadas mediante triángulos para detectar la interacción entre el vehículo y el entorno. Por este motivo, es necesario reducir a triángulos aquellas superficies que aparezcan en los planos representadas de otra manera. El proceso para representar mediante triángulos una superficie, llamado *teselación*, puede ser llevado a cabo manualmente o bien mediante un algoritmo. En nuestro caso, para procesar las superficies del entorno del barrio, se ha empleado un algoritmo consistente en crear segmentos que vayan desde un único punto hacia los restantes vértices del contorno. Éste es un método muy adecuado para los contornos de aceras y edificios. En el caso de los entornos nuevos generados en AutoCAD, todas las superficies se han dibujado mediante triángulos, sin hacer ningún proceso posterior.

Una vez reducidas a triángulos las cuatro capas de datos, se guarda el fichero en formato *DXF*, y se anotan los nombres por los que se ha de

identificar a las cuatro capas recién creadas. No es necesario borrar el resto de la información original que puedan contener los planos. El fichero ya está listo para ser procesado automáticamente por el simulador.

El formato *DXF* (*Drawing Exchange Format*) es un formato de fichero diseñado especialmente para favorecer el intercambio de datos de geometría tridimensional entre diversos programas. Es un formato de texto sencillo. Distingue los datos que contiene en dos tipos básicos: objetos y entidades. Los objetos no poseen una representación gráfica, mientras que las entidades sí. Los datos están organizados en parejas compuestas por un dígito de código y sus valores asociados. Los códigos, llamados “códigos de grupo”, indican el tipo de valor que lo sucede: *string*, *entero de 16 bits*, etc. El fichero *DXF* se organiza, mediante el uso de estas parejas de códigos de grupo y sus valores correspondientes, en *secciones*, que a su vez contienen varios *registros*, que, de nuevo, no son más que una serie de parejas de código de grupo y su valor asociado correspondiente.

Las secciones tienen un código de grupo “0”, al que le sigue su valor asociado “SECTION”. A continuación, se emplaza siempre la siguiente pareja de valores, cuyo código de grupo será “2”, y como valor asociado una cadena con el nombre del tipo de la sección. Los nombres de tipos posibles de sección son HEADER, CLASSES, TABLES, BLOCKS, ENTITIES y OBJECTS. Una sección termina con un código de grupo “0” y el valor asociado “ENDSEC”.

Una de las grandes ventajas del formato *DXF* es que posee una gran modularidad: esta articulado en secciones independientes, y éstas a su vez en diferentes subgrupos, que contienen diferentes tipos de parámetros. Esto permite que el programa lector del fichero no esté obligado a procesar toda la información que se halla en el mismo. Únicamente es necesario buscar la sección deseada, para después obtener los datos requeridos en ella.

Para la representación de los entornos sólo son relevantes, de entre los datos contenidos en los ficheros *DXF*, los segmentos de línea que determinan los contornos que delimitan las aceras y líneas de fachada de del tipo *ENTITIES*, que, de forma general, guardan la información correspondiente las calles, y los segmentos correspondientes a los triángulos que componen la calzada y las rampas y obstáculos que pueda haber. El módulo de lectura busca las secciones que albergan la información correspondiente a las coordenadas de estos segmentos. Estas

secciones son a las diferentes entidades que constituyen los dibujos, como líneas, arcos, curvas, etc., aunque, en este caso, únicamente se tienen en cuenta las líneas.

Una vez localizada la sección *ENTITIES*, la rutina encargada de la lectura del fichero *DXF* busca, de entre los grupos siguientes, aquellos que correspondan al identificador *LINE*. Dentro de este grupo se hallan todos los parámetros que definen al segmento de línea correspondiente. El primero que se busca es el que tiene el código “8”, correspondiente a la capa en la que se encuentra el segmento (*LAYER*). Basándose en esta información, el programa puede descartar todos aquellos segmentos que no pertenezcan a la capa que esté procesándose.

El siguiente paso consiste en obtener las coordenadas del inicio y final del segmento, una vez se ha garantizado que pertenece a la capa de datos deseada. Las coordenadas *x*, *y*, *z* del inicio se encuentran tras los códigos de grupo “10”, “20” y “30” respectivamente. De igual manera, las del final del segmento se encuentran tras los grupos “11”, “21” y “31”. Los puntos se van almacenando en una lista. Para cada capa se genera una lista con los puntos correspondientes.

Se ha programado un módulo encargado de leer la información de los ficheros *DXF*, que está formado por una serie de rutinas sencillas que extraen las coordenadas de los segmentos de línea pertenecientes a la capa deseada, y generan las listas de puntos.

A partir de las listas de puntos de cada capa es necesario identificar los polígonos que conforman las calles, las aceras, etc. Para esto, no hay que perder de vista que un segmento puede pertenecer a uno o a dos triángulos. Por este motivo, cada segmento cuenta con un contador que inicialmente toma el valor de “2”. Cuando se haya determinado un triángulo al que pertenece el polígono el valor del contador disminuye. Todavía hay que tener en cuenta ese segmento en la búsqueda de triángulos a menos que su contador valga “0”.

En las listas de segmentos, los datos están organizados de seis en seis. Es decir, las tres coordenadas del punto de inicio del segmento y, a continuación, las tres coordenadas del punto final. El proceso para la detección de los triángulos se muestra en la Figura 4.7, y es explicado a continuación.

Partimos de un segmento, por ejemplo el A. Se busca otro en el que alguno de sus extremos coincida con uno de los extremos del segmento inicial. En el segmento B, el extremo B_1 coincide con el extremo A_2 del segmento A. A partir de aquí, la búsqueda continúa, teniendo como base el segmento B. Ahora buscamos otro segmento que cierre el triángulo y que, por tanto, sus extremos coincidan con el B_2 y el A_1 . Este segmento será el C.

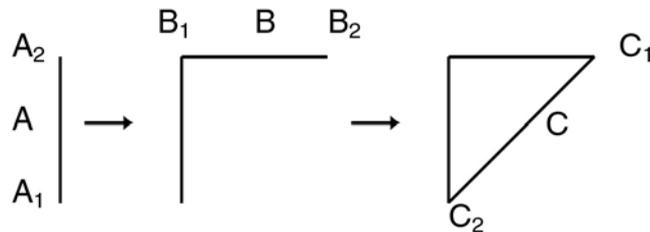


Figura 4.7 Algoritmo de búsqueda de los triángulos que componen el entorno.

Ahora bien, ¿qué ocurriría si el segmento C, que comparte un extremo con el B, no cierra el triángulo? En el caso de que no haya ningún segmento que cierre el triángulo, Figura 4.8, se desecha el segmento B y se busca otro segmento que tenga en común un extremo con el A. Con este nuevo segmento, D, se comprueba que comparte extremo con otro que efectivamente cierre el triángulo.

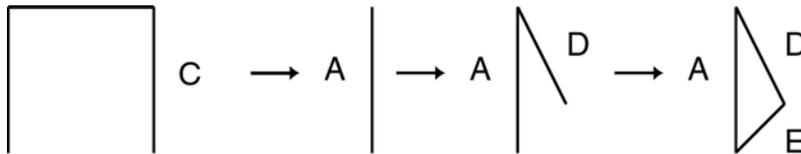


Figura 4.8 Algoritmo en el caso de que el tercer segmento no cierre el triángulo.

Por último, es necesario que el tercer segmento, que posee un punto en común con cada uno de los otros dos, no tenga el mismo punto en común con los dos anteriores. Es decir, que no se presente una configuración en racimo como la representada en la Figura 4.9. Una subrutina es la encargada de realizar esta verificación.



Figura 4.9 Tres segmentos configurados en racimo.

En este punto, ya se puede afirmar que los tres segmentos delimitan una cara, que se añade a una nueva lista que almacena las caras. A continuación, se reduce el valor del contador de “polígonos adyacentes” de cada línea como se había indicado anteriormente. En caso de que alguno de ellos fuese ahora nulo, el programa deja de tenerlo en cuenta para la búsqueda.

Este proceso de búsqueda de triángulos se realiza en las listas de segmentos pertenecientes a cada una de las capas. En el caso de la capa de la calzada o de la capa de rampas, el proceso de generación del entorno se termina aquí. Sin embargo, para los conjuntos de las aceras y los edificios se debe dar un paso más, puesto que, en estos casos, los polígonos delimitan la superficie sobre la que irá una acera o un edificio. Estos volúmenes correspondientes a aceras y edificios, se generan mediante una extrusión de las secciones representadas por las superficies anteriormente halladas. Dicho procedimiento se describirá a continuación para el caso de la generación de los volúmenes que describen a los bloques de edificios. El correspondiente a las aceras de las calles es exactamente el mismo.

Para generar los bloques y las aceras se parte de la lista de segmentos previamente procesada para la capa en cuestión, y de la lista de superficies triangulares hallada en el proceso anterior. El primer paso es determinar los contornos que delimitan las secciones en planta de los bloques de edificios. Esto se realiza concatenando los segmentos que tienen un extremo en común y sólo pertenecen a un triángulo. Los segmentos que forman parte del contorno son aquellos que tienen su contador de caras adyacentes a “1”, puesto que solamente forman parte de un triángulo.

Para obtener los contornos, se parte de un segmento y se marca como “usado”. A continuación se recorre la lista en busca de otro segmento con el que comparta un extremo y no se haya empleado anteriormente. El proceso se repetirá hasta que se encuentre un segmento que, formando parte del contorno, conecte con el primer segmento de éste.

Una vez determinado el contorno del edificio, se guarda toda la información necesaria para su representación en un vector llamado *edificio*, que está formado por: una lista con el contorno, la lista de triángulos que forman las caras, y el valor de la altura de extrusión.

Las caras laterales de los edificios se construyen a partir del contorno. Éste se extruye la altura indicada. Cada uno de los segmentos que forman el contorno del edificio pasa a ser un paralelepípedo que, a continuación, es dividido en dos triángulos. Para construir las caras laterales de las aceras se sigue un procedimiento similar.

Por último, para la representación del entorno correspondiente a un barrio de Ferrol, se han empleado también texturas. Las texturas dotan a la representación de una mejora visual muy significativa, puesto que proporcionan a los elementos geométricos representados un detalle que sólo una malla poligonal extremadamente densa podría superar. Por otro lado, la aplicación de texturas tiene un coste computacional muy bajo. Actualmente, son las tarjetas gráficas quienes se encargan del mapeado de texturas. Esto es debido a que existen algoritmos muy simples para la aplicación de las texturas, que apenas suponen una carga de cálculo con respecto a las demás operaciones gráficas. Sin embargo, la sencillez del algoritmo es inversamente proporcional a la calidad del mapeado de textura, algo que se debe tener en cuenta a la hora de sopesar qué método usar. En este trabajo se ha empleado el algoritmo que OpenGL tiene como predeterminado.

Para confeccionar un entorno tan variado como sea posible, se han recopilado imágenes de las fachadas de muchos edificios del barrio de La Magdalena de Ferrol. Sin embargo, hay que tener en cuenta las limitaciones técnicas de espacio de memoria y tiempo de proceso. Aunque se incidió antes en el escaso coste computacional que implicaba el texturizado en sí, el trasiego de datos desde y hacia la memoria de vídeo provocado por el cambio intensivo de texturas, puede provocar una severa pérdida de rendimiento. Por este motivo, no se han representado exactamente todos los edificios que constituyen una manzana, tal y como son en la realidad, sino que se ha buscado una simplificación próxima a la realidad, que al mismo tiempo sea demostrativa de la capacidad de representación de esta técnica.

Con las imágenes de los edificios se elabora una “tira” continua de fachada, correspondiente a una manzana de casas, que será aplicada posteriormente a los polígonos que conformen los edificios. La proyección de la imagen (mapeado) sobre el objeto tridimensional se llevará a cabo rodeando al edificio con la imagen que se desea proyectar. Posteriormente, se determinará qué coordenadas del mapa bidimensional de textura corresponderán a cada vértice de los que componen el objeto. Se han

elaborado varias “tiras” que se repiten, pero evitando que dos manzanas consecutivas sean iguales.

Se ha empleado un mapeado cilíndrico para rodear y proyectar la imagen sobre el edificio. Este procedimiento consiste en deformar conceptualmente la imagen de textura para que englobe al objeto, tal y como se muestra en la Figura 4.10.

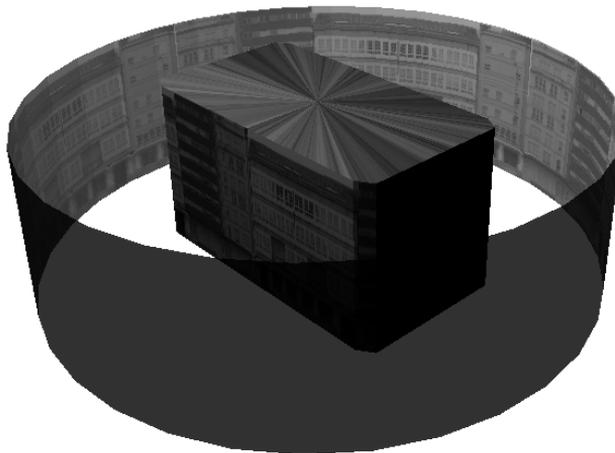


Figura 4.10 Proyección cilíndrica de la textura sobre el objeto que representa al edificio.

Para llevar a cabo este procedimiento, el primer paso consiste en especificar una orientación que defina la colocación del mapa de texturas con respecto al edificio. Se toma una dirección perpendicular a una fachada de edificio, de tal manera que la zona central de la textura represente el frontal del bloque. Los puntos extremos de la textura irán situados en las posiciones que forman 180° y -180° . Los puntos intermedios reciben su posición de una interpolación, tal como se representa en la Figura 4.11.

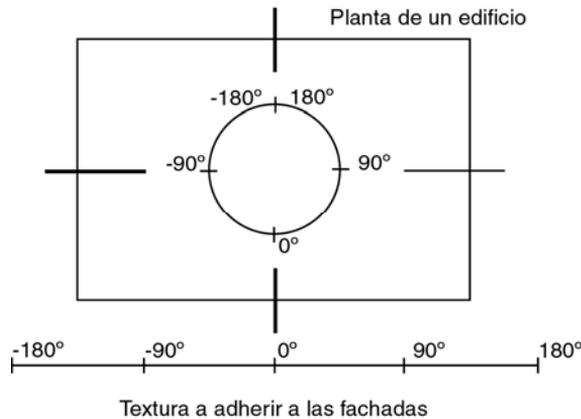


Figura 4.11 Relación entre los puntos de la fachada y de la textura.

4.3.3. Representación del Vehículo y Otros Objetos.

Para la representación del vehículo se han desarrollado dos modelos: una representación del prototipo, y una representación de un vehículo comercial. La representación de los elementos del prototipo se realizó en Ideas, y posteriormente se exportó en un formato adecuado para su introducción en el simulador. Aunque esta representación es realmente vistosa y, sobre todo, permite observar perfectamente el comportamiento de cada uno de los elementos del vehículo, tiene el inconveniente de que es computacionalmente muy costosa, debido a que contienen un número muy elevado de triángulos lo que en ordenadores un tanto antiguos, ralentiza la simulación. Por este motivo, se ha considerado utilizar también el chasis de un vehículo utilitario como modelo.

Para la representación de este último, se parte de un plano del vehículo que tenga las vistas necesarias y englobe todos los detalles a representar. A continuación, se muestran los planos usados en la construcción del modelo 3D del utilitario.

A partir de esos planos, se debe modelar, en cualquier programa 3D, la carrocería, teniendo como referencia dichas imágenes. Para el modelado del vehículo se ha optado por el programa *Blender* (www.blender.org). Las ventajas de utilización de *Blender* con respecto a cualquier otro editor 3D son numerosas.

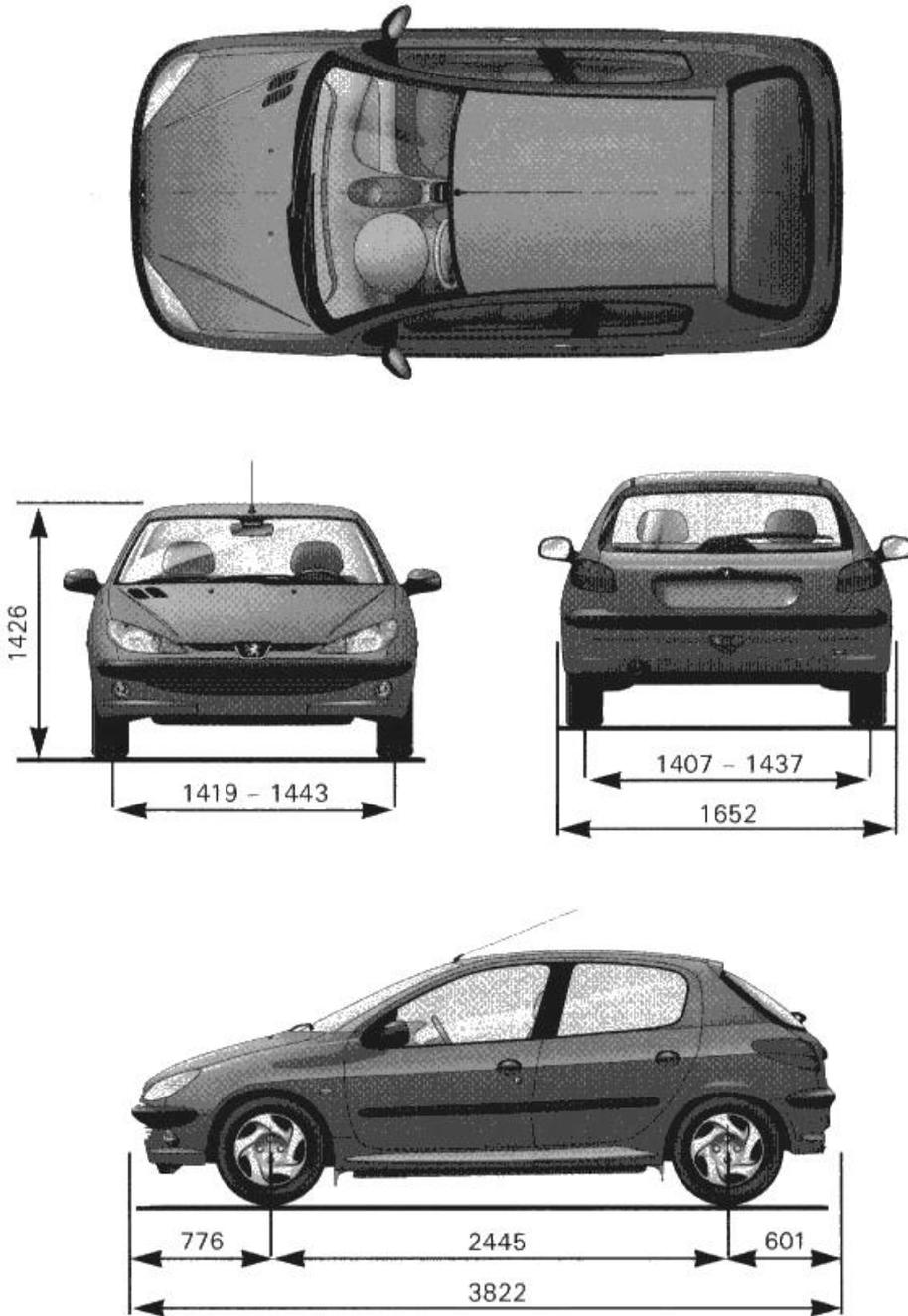


Figura 4.12 Vehículo comercial empleado en el simulador.

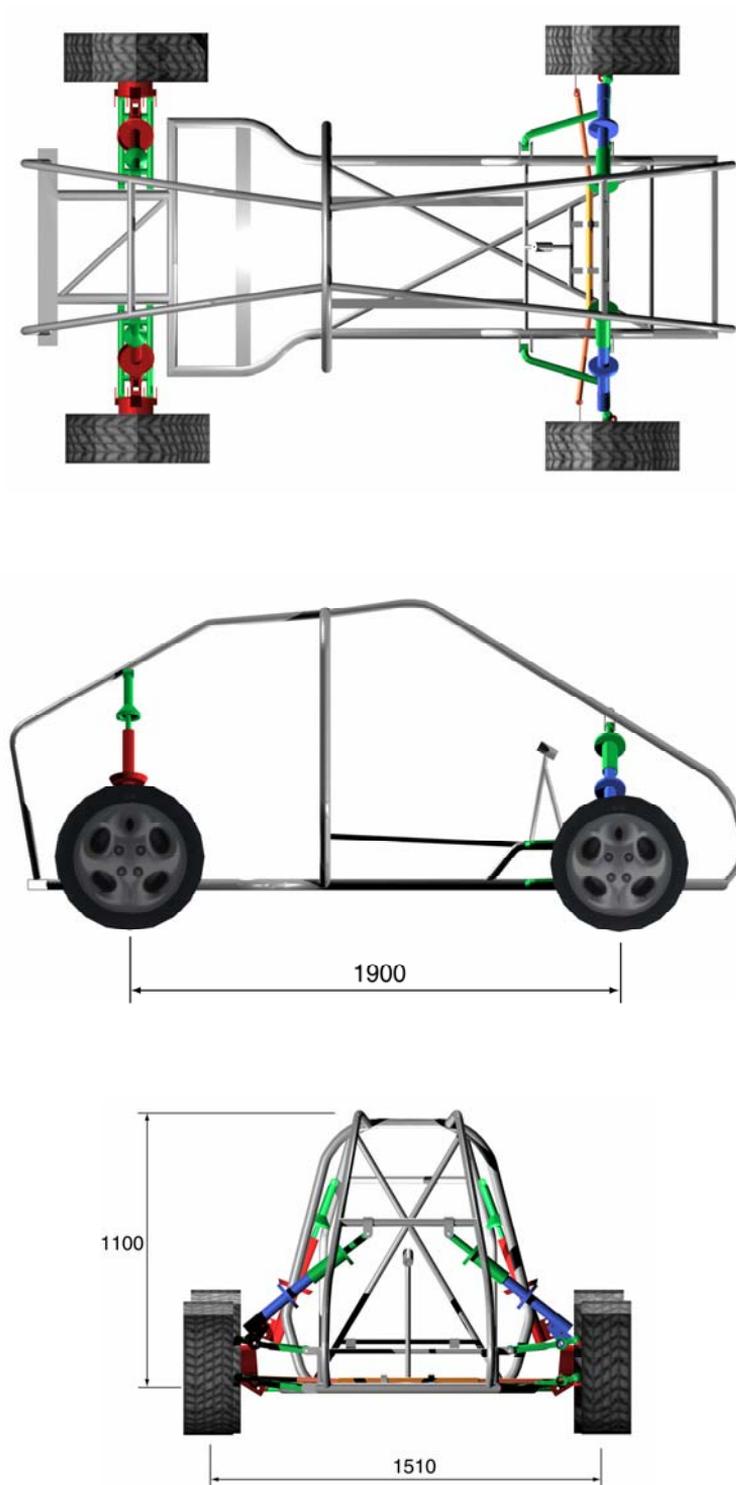


Figura 4.13 Representación del prototipo.

En primer lugar, *Blender* es gratuito y *Open Source*, con lo que no solamente no hay que adquirir licencias de uso de ningún tipo, sino que además el código fuente está disponible, tanto para aclarar dudas sobre su funcionamiento como para corregir posibles errores. Se trata además de un programa muy ligero en su segmento, ya que apenas ocupa 10Mb. Además, está disponible en la gran mayoría de las plataformas (todas las de PC), con lo que permite que el desarrollo pueda extenderse a diferentes entornos y sistemas operativos. Sin embargo, una de las características más potentes y que más útil ha resultado durante el desarrollo del proyecto, ha sido la posibilidad de expansión del programa mediante la escritura de *scripts* en un lenguaje orientado a objetos muy intuitivo y sencillo, *Python*. Esto ha posibilitado la escritura de un módulo cuya función es almacenar la información del modelo en un fichero de texto que, como se podrá ver, es muy sencillo de procesar.

El proceso para obtener la malla poligonal de la carrocería (y por extensión de cualquier otro objeto empleado) se describe a continuación, sin entrar en los detalles de uso del programa.

El primer paso consiste en colocar las imágenes de los planos del vehículo sobre los planos principales del espacio, de tal manera que sirvan de guía durante el modelado.

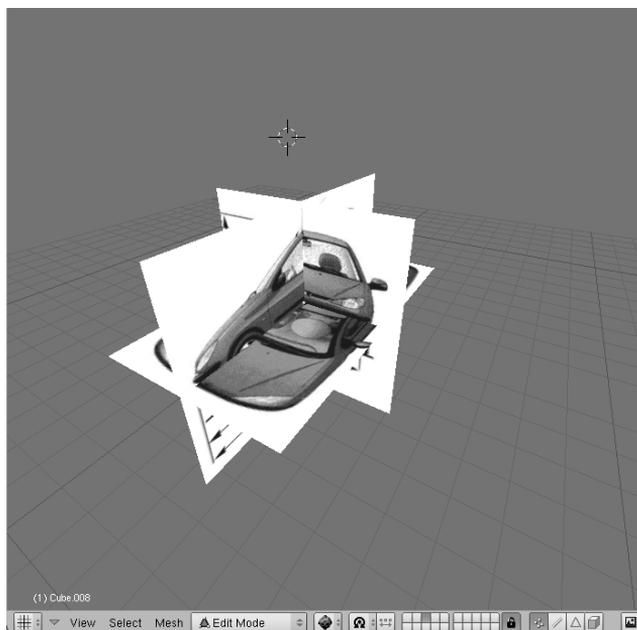


Figura 4.14 Establecimiento de los tres planos de referencia para el modelado.

Posteriormente, se añade un cubo que se modifica para que tenga un volumen que contenga las vistas que hemos representado anteriormente. A continuación, se subdivide, eliminando partes de material y se le añaden nuevas aristas, que se ajusten a los contornos que en cada vista definen los planos.

Únicamente es necesario modelar un lado de la carrocería. Al igual que otros muchos programas de modelado 3D, *Blender* puede actualizar automáticamente el otro lado. De esta manera, se garantiza la simetría entre los lados izquierdo y derecho, y se agiliza mucho el trabajo. Como ocurría con el entorno, para que pueda ser representado dentro del programa del simulador, hay que realizar una triangularización de las superficies de la carrocería para representarlas mediante los comandos de OpenGL.

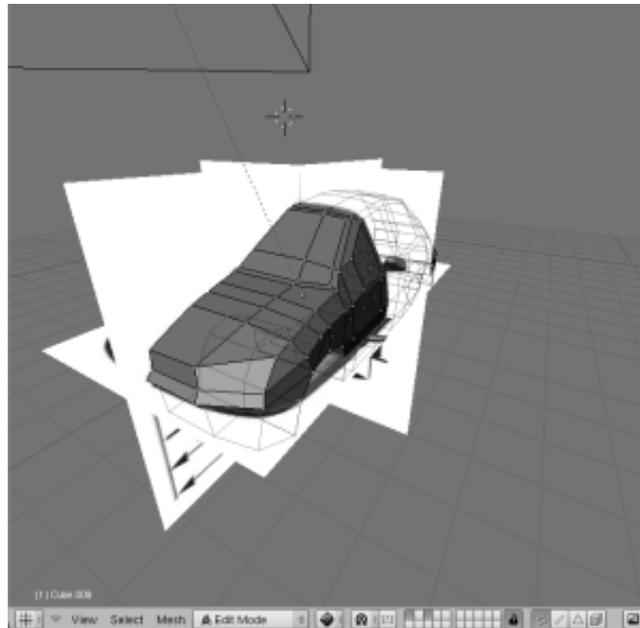


Figura 4.15 Vista del proceso de modelado en base a las imágenes de referencia.

Una vez diseñada la representación gráfica de la carrocería es necesario almacenar los datos de los vértices de los triángulos en un fichero. Se ha creado una pequeña aplicación o *script*, para ejecutarla dentro del programa Blender. Esta aplicación accede a los datos internos del programa para obtener las coordenadas de los vértices y de las normales del objeto deseado y escribe estos datos en un fichero del disco. A los ficheros generados por este procedimiento se les ha dado arbitrariamente

la extensión *.pnt*. Además, escribe otra lista con el orden en que se debe recorrer la lista de vértices para formar las caras del modelo.

Esta forma de proceder (almacenar los vértices una sola vez, e indicar después el orden en el que se han de recorrer mediante una lista de índices), conduce a una mayor eficiencia de procesamiento durante el uso del modelo, ya que es menor la cantidad de datos a transferir entre la CPU y la tarjeta gráfica. Además, los cálculos de transformación (traslación y rotación) de cada uno de los vértices sólo se llevan a cabo una vez, ya que aparecen listados de igual manera sólo una vez, aunque estén compartidos por varias caras.

Una vez dentro del programa del simulador, la lectura de datos es muy sencilla, debido a que el propio fichero de datos indica el número de vértices, normales e índices que hay contenidos en éste. De esta manera, se puede asignar de antemano la memoria necesaria para albergar a los datos. Los pasos a seguir para la lectura son:

- Abrir el fichero.
- Extraer el tamaño de los arrays de caras, normales e índices.
- Reservar la memoria necesaria para albergarlos.
- Leer línea por línea, almacenando los datos en la memoria.

Esta disposición de los datos hace posible enviárselos directamente a la tarjeta mediante una única llamada a una función de OpenGL, sin necesidad de iterar a lo largo de las listas.

Este mismo procedimiento se puede aplicar a cualquier otro objeto que se desee introducir en la simulación. En concreto, se ha empleado para la representación del volante, los cristales del vehículo, el salpicadero y las agujas y esferas de los relojes del salpicadero.

4.3.4. Gafas de Realidad Virtual (HMD).

Buscando una mayor inmersión en el entorno, se ha dotado al simulador de unas gafas de realidad virtual o HMD (Head Mounted Display). Este dispositivo presenta dos características principales:

- Genera una imagen tridimensional del entorno.
- Integra los movimientos de la cabeza en el punto de vista de la imagen que percibe el usuario.

De esta forma, el entorno que el usuario ve pasa a ser únicamente el generado por el ordenador. Para que la sustitución del entorno sea efectiva, los movimientos de la cabeza deben ser recogidos y enviados al ordenador, de modo que se presente, a través de las gafas, la vista correspondiente. Para ello, existe una amplia gama de detectores de movimiento en el mercado, los cuales, conectados al ordenador, son capaces de transmitir varias veces por segundo la posición instantánea del sensor o sensores.



Figura 4.16 Equipo completo de sensor, unidad central y generador de campo.

Se ha considerado como el conjunto más adecuado para adaptar estas técnicas al simulador, el formado por las gafas de Realidad Virtual y el sensor de movimiento. Existen otros simuladores en los que el usuario se introduce en una cabina, con varias pantallas fijas a las paredes del habitáculo. A su vez, este habitáculo es móvil, de manera que transmite las aceleraciones que se producirían durante la conducción real. Las gafas de Realidad Virtual, gracias a su sistema binocular, son capaces de representar ante el usuario un punto de vista diferente para cada uno de sus ojos, de tal manera que se genera una ilusión de tridimensionalidad, o efecto estereoscópico. Además, como están fijas a su cabeza, garantizan que la imagen generada por el ordenador siempre esté enfrente de ésta. El sensor de posición y movimiento tiene encomendada una tarea de igual o mayor importancia, ya que contribuye a sostener la credibilidad de las imágenes generadas, comunicando al ordenador la posición de la cabeza, con lo que el simulador puede construir la imagen que se observaría desde

esa posición. Esto hace que el usuario tenga la sensación de que puede moverse realmente dentro del entorno generado por el simulador. Otra ventaja del uso de estos aparatos es que también pueden ser instalados en una cabina como la antes descrita, de tal manera que se sumen el efecto óptico de las gafas y el físico de la plataforma.

El sistema que sensoriza el movimiento se compone principalmente de tres partes:

La unidad central, a la que se conectan todos los demás dispositivos, incluido el ordenador y la alimentación independiente del aparato. Dispone del interruptor de encendido/apagado y de otros, únicamente usados cuando se conecta el aparato mediante el puerto de serie del ordenador.



Figura 4.17 La unidad central.

El generador de campo magnético. Se trata de una pequeña caja que genera a su alrededor un campo magnético orientado según los ejes definidos por sus aristas. Este generador sirve para proporcionar el campo de referencia y, por tanto, debe estar unido al sólido con respecto al cual se va a medir el movimiento. Debido a que el campo magnético presenta simetría, las medidas sólo se podrán efectuar en la zona definida por $0^\circ < \varphi < 180^\circ$ y $0^\circ < \theta < 180^\circ$, según las coordenadas indicadas en la Figura 4.18.

En caso de que el sensor saliese de dicha zona, el aparato comenzaría a transmitir las coordenadas reflejadas con respecto al plano de simetría. En el presente trabajo no se ha contemplado un rango de movimientos tan amplio. Asimismo, es conveniente depositar el generador en algún sitio aislado, donde no se pueda perturbar el campo magnético. Es aconsejable usar madera para este fin.

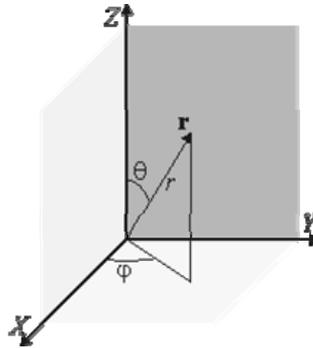


Figura 4.18 Representación de las coordenadas esféricas empleadas.

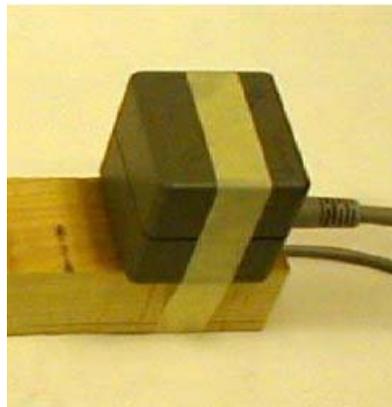


Figura 4.19 Generador de campo magnético.

El sensor capta, en el punto en que se encuentre, el campo magnético producido por el generador. El valor del flujo magnético que llegue al sensor es comunicado a la unidad central, que calcula, a partir de estos datos, la posición y orientación del sensor con respecto al generador. Cabe destacar que la intensidad del campo magnético no es muy elevada y, por tanto, no es muy conveniente que el generador esté más alejado de 1 m del sensor, puesto que se empezaría a recibir una señal de ruido muy elevada. Esto tampoco ha presentado ningún problema, puesto que el movimiento normal de la cabeza del conductor es bastante más reducido.

Para controlar el dispositivo, existe una librería, codificada en lenguaje C++ y proporcionada por el fabricante, que facilita el acceso a todos los parámetros que configuran el detector. Desde el programa principal del simulador, al comienzo, es necesario inicializar el dispositivo, mediante una llamada a la librería que garantice que el aparato está correctamente conectado y en funcionamiento.



Figura 4.20 Detalle del sensor.

Entre las opciones que se pueden configurar en el sensor están el formato de los datos y las unidades en las que se reciben. El software permite escoger entre unidades del Sistema Internacional y el anglosajón para las mediciones de posición. Para describir la orientación del sistema, se puede escoger entre la matriz de rotación, los ángulos de Euler o los cuaterniones. Se ha optado por obtener la matriz de cosenos directores, por la sencillez de uso posterior. Los datos de posición se registran en unidades del Sistema Internacional.

A partir de los datos anteriores se forma la matriz de transformación correspondiente a la posición y orientación de la cabeza del conductor. Esta matriz de transformación se aplica a una de las cámaras, que recoge el punto de vista que se ha diseñado para el conductor. Al comenzar el programa se inicializa el sistema de coordenadas del sensor. El programa considera que esta posición inicial constituye el origen de coordenadas con respecto al que se recogen los movimientos de la cabeza, y que la orientación inicial del sensor está completamente alineada con respecto a los ejes dibujados en la caja del generador. Esta forma de proceder permite que no sea relevante la forma en que el usuario se adhiera el sensor, sino cuánto se mueva durante la simulación con respecto a la posición inicial.

La recogida de datos se puede realizar de diversas maneras: consulta puntual, el método de la cola de mensajes y el de medición continua.

El primero consiste sencillamente en llamar a una rutina para obtener la posición instantánea del sensor cada vez que se necesite. La gran desventaja de este procedimiento consiste en que el mecanismo de

petición-obtención de datos es muy lento, y además, el programa se detiene hasta que llegan esos datos.

El segundo método tiene como objetivo utilizar la cola de mensajes habitual en los programas Windows, de tal forma que la llegada de los datos se comporte como un mensaje del sistema operativo más. El programa lee la cola de mensajes y, cuando llega alguno perteneciente a la adquisición de datos, lo envía a una rutina encargada de actualizar la información sobre posición y orientación. Seguir este método conlleva que el programa tenga una gran dependencia del sistema Windows, y también de la programación orientada a eventos. El simulador no es un programa orientado a eventos, sino que, por el contrario, trata de efectuar todos sus cálculos en tiempo real. Por estos motivos, se ha desechado también este método.

El último método es más sencillo y transparente. Tras una orden, mediante una llamada a la librería, el aparato comienza a realizar las mediciones por su cuenta, almacenándolas en la memoria del ordenador. Esto se realiza mientras el programa principal está ocupado en otros cálculos. Cuando es necesario redibujar la pantalla, el dispositivo proporciona la dirección de memoria donde se encuentran almacenados los datos, de tal manera que el programa no tiene que esperar a la llegada de los datos, simplemente los lee de la memoria. Allí están almacenadas todas las mediciones, desde la más reciente a la más antigua. Generalmente, sólo se usará la más reciente, puesto que podría ser considerada “instantánea”. Sin embargo, si por alguna razón se necesitase recoger los datos de instantes anteriores (como, por ejemplo, para promediar los últimos datos obtenidos con una *media móvil*, obteniendo así una respuesta suavizada), también se podrían hallar en la memoria. La única limitación es que el *buffer*, o memoria que usa el dispositivo para almacenar los datos, es limitado, aunque en la inicialización se podría especificar dicho límite si fuera necesario.

Un último detalle, en lo que se refiere al sensor, es que la matriz de orientación del mismo con respecto a la base no es la que se necesita para representar la vista subjetiva del usuario, sino precisamente su inversa. Afortunadamente, las matrices de rotación tienen la propiedad de ser ortogonales, por lo que será suficiente con transponer la matriz proporcionada por el sensor antes de componer la matriz de transformación.

En lo referente a las gafas de Realidad Virtual, éstas se comportan a todos los efectos como un monitor para ordenador cualquiera y, por lo tanto, no es necesario establecer ninguna configuración adicional en el programa para poder hacer uso de ellas. La única diferencia en el comportamiento del programa es que, para obtener visión estereoscópica, es necesario generar dos imágenes –cuyos puntos de vista corresponden a cada uno de los ojos–, en vez de una, en cada paso de tiempo.

Es preciso establecer de alguna manera qué imagen de las dos será generada en cada una de las dos pantallas de las gafas, y cómo hacerle llegar a éstas esa información. Existen dos métodos: el primero requiere que el ordenador disponga de una tarjeta de vídeo con dos salidas, a las que se conectarían las dos entradas de las gafas. Cada imagen sería transmitida independientemente. La desventaja de este método consiste precisamente en la necesidad de disponer de una tarjeta de doble salida, que hasta el momento están instaladas en muy pocos ordenadores (para trabajos muy específicos), y que restringiría enormemente el número de usuarios que pudiesen utilizar el programa.



Figura 4.21 Gafas de realidad virtual empleadas.

El segundo método sólo requiere que la tarjeta de vídeo posea una salida, por lo que es el que se ha empleado en el presente trabajo. Consiste en enviar a las gafas una única imagen *entrelazada*, que será decodificada posteriormente por éstas para generar las dos vistas. El *entrelazamiento* se reduce a mezclar las dos imágenes en el mismo espacio que ocuparía una sola de ellas. Esto se logra dibujando las líneas horizontales de número impar de una de ellas, y las líneas horizontales de número par de la otra.

Dado que el número final de líneas de la imagen entrelazada debe ser el mismo que el de cada una de las imágenes a partir de las que se forma, se observa que sólo se podrá enviar la mitad de las líneas de cada imagen. Una consecuencia de este modo de proceder es una disminución en la calidad de la imagen: cada ojo pierde la mitad de la información que le correspondería, y la resolución vertical de cada imagen disminuye a la mitad.

En la Figura 4.22 se observa un esquema del entrelazamiento. Los datos de entrada son dos imágenes diferentes, la del cuadrado y la del triángulo, y el patrón o máscara de mezcla está compuesta por un emparrillado horizontal. Gracias a ésta última, se desecha la mitad de los datos proporcionados por cada imagen de manera complementaria, es decir, de una se eliminan las líneas pares y, de la otra, las impares. Así se cumple el requisito de que, tras la operación, ambas imágenes pueden formar una nueva completa a partir de las dos.

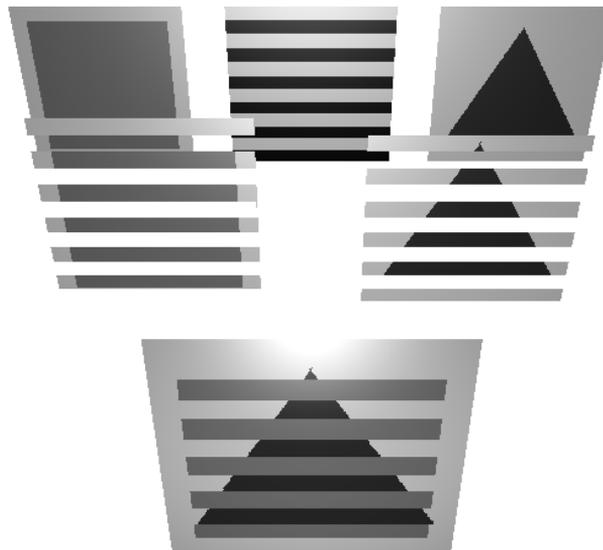


Figura 4.22 Proceso de entrelazado de imágenes.

Al generar la imagen entrelazada, surge un problema de rendimiento. Manipular directamente las dos imágenes para obtener una tercera, exige copiar la memoria de vídeo donde se hallan ambas a la memoria convencional, generar una tercera y, finalmente, enviarla de nuevo a la memoria de vídeo. Este proceso es necesario porque no es posible modificar directamente los contenidos almacenados en la memoria de vídeo.

Afortunadamente, existen formas de realizar este trabajo aprovechando las capacidades de la tarjeta gráfica. Antes de presentar la imagen en la pantalla, OpenGL realiza una serie de tests o comprobaciones. En ellos se verifica el cumplimiento de alguna propiedad. Por ejemplo, el test de profundidad comprueba que un objeto no se encuentre oculto tras otro, más próximo al punto de vista del observador. De esta manera, se evita la representación de absolutamente todos los objetos, representando solamente los que están a la vista del observador.

El test que se ha empleado para entrelazar las imágenes es el *stencil test* o test de estarcido. OpenGL dispone de memorias auxiliares o *buffers*. Uno de ellos es el *stencil buffer* o *buffer de estarcido*. En esta memoria se puede almacenar un patrón o máscara y, a continuación, verificar si los puntos que se van a representar quedan ocultos por la máscara o no, mediante el test de estarcido. Como se puede observar, es una forma de descartar un fragmento antes de que vaya a ser representado. Existen tests específicos para definir áreas de recorte en la pantalla, como el *scissor test* o test “de tijera”, pero, desgraciadamente, estos test sólo contemplan áreas de recorte en forma de rectángulo. El test de estarcido, en cambio, permite aceptar o descartar la representación de un píxel mediante un patrón de comparación arbitrario previamente suministrado.

Concretamente, en el *stencil buffer* se dibuja primero la máscara mediante los comandos usuales de OpenGL, activando previamente la escritura en dicho *buffer*. Para ello, hay que tener en cuenta primero que hay que desactivar la característica por defecto de OpenGL, que es dibujar directamente en la pantalla. Esto evitará que el dibujo de la máscara se represente en la imagen final. Una vez creada la máscara, se vuelve a permitir que OpenGL dibuje sobre la pantalla, y se representa una de las imágenes. Posteriormente, se limpia el buffer de profundidad (*z-buffer*), ya que dicha información es diferente para cada imagen y, de no hacerse así, la segunda se vería desvirtuada. A continuación, se activa la función de comparación del *stencil buffer*, de tal manera que sólo se pueda dibujar a través de los “huecos” definidos por la máscara creada anteriormente. De esta manera, la máscara protegerá las líneas pares de la primera imagen (si así se ha definido), mientras que las impares resultarán sobrescritas por la segunda.

En este caso, la máscara que se ha empleado es muy sencilla. Como se sabe, la imagen que se representa en la pantalla está formada por líneas. Según la resolución de la pantalla se tendrán más o menos líneas

horizontales y verticales. La máscara empleada está formada por líneas alternas, de manera que la imagen que va a recibir un ojo, en concreto el derecho, está compuesta por las líneas pares, mientras que el ojo izquierdo recibirá la imagen correspondiente a las impares. Como resultado se obtendrá la imagen combinada de ambas representaciones.

Al recibir las gafas estereoscópicas esta imagen, envían la señal procedente de las filas pares al ojo derecho, y las de las impares al izquierdo, con lo que logran desempaquetar ambas y crear la ilusión de tridimensionalidad.

Por último, se indican algunos datos que se han tenido en cuenta para programar el punto de vista del conductor. Se ha supuesto que la separación entre los ojos es de 6 cm. Otro dato importante es el punto de focalización. Inicialmente, se supuso que ambos ojos miraban hacia el frente. En ese supuesto, el punto de focalización es el infinito. El efecto que se obtenía daba la sensación de tener un recinto tridimensional -que se correspondería con el interior del coche- en un entorno plano. Para evitar este efecto, se ha puesto el foco a 10 m, de tal manera que los haces de vista de los dos ojos se crucen en ese punto. La percepción estereoscópica del entorno aumentó sensiblemente mediante este procedimiento. Por otro lado, las gafas obligan a que se utilice una resolución de 800x600 píxeles. Debido al entrelazado de la imagen, la resolución para cada ojo es de 800x300 píxeles. Por este motivo, los colores originales resultan un poco desvaídos en la representación final.

Resta hacer un comentario sobre el rendimiento debido a esta nueva característica. En un principio, el esfuerzo de representación se ha duplicado, ya que como se ha descrito, la visión estereoscópica requiere dibujar dos pantallas desde puntos de vista distintos para un mismo instante dado. En la práctica, hay dos motivos por los que este incremento en la carga de trabajo de la tarjeta gráfica no tiene una influencia directa en el rendimiento del programa, como se puede observar experimentalmente:

En primer lugar, el tiempo destinado a la representación es claramente poco significativo en comparación con el tiempo empleado en el cálculo dinámico. Esto se puede comprobar cuando se aplica un estudio de tiempos a las funciones del programa (*profiling* en la jerga del compilador). Se ha realizado un estudio de tiempos durante una simulación de 90 s de duración. Se ha escogido este tiempo, puesto que las funciones relativas a la inicialización del programa tienen una duración de 9,8 s, y se trataba de

evitar que tuviesen una relevancia excesiva dentro del conjunto. De esta forma, en la simulación, las funciones relativas al cálculo consumen 78,15 s, el 78,6 % del tiempo de la simulación, mientras que las funciones de representación gráfica emplean 3,96 s de tiempo, lo que supone el 4 %.

En segundo lugar, las tarjetas de aceleración gráfica no esperan a que la CPU principal del sistema llegue a determinado punto del programa, sino que empiezan a hacer sus cálculos en paralelo con la CPU, una vez se han enviado todos los comandos a la tarjeta. En el presente caso, tras la llamada a *SwapBuffers()*, se envían las órdenes de representación a la tarjeta, pero no se espera a que éstas se hayan completado; la tarjeta gráfica estará, en efecto, representando un instante, mientras la CPU se dedica a calcular el instante siguiente en paralelo. Por esta razón, mientras el tiempo consumido durante la representación no sea mayor que el empleado durante el cálculo, un incremento en la carga del proceso gráfico no debería aumentar el tiempo de proceso global del programa, es decir, la representación no será el factor crítico en el rendimiento.

4.3.5. Grabación de Maniobras.

Una de las grandes ventajas del simulador, frente a un programa que sólo calcule datos dada una entrada conocida de antemano, es la posibilidad de interactuar que posee el usuario. Ciertos tipos de comportamiento en la conducción no pueden ser parametrizados completamente, o por lo menos de forma absolutamente satisfactoria. Es el caso del tiempo de respuesta del conductor ante cualquier señal o imprevisto, la precisión a la hora de seguir una trayectoria prefijada, la influencia de la visibilidad en la conducción, etc.

En cambio, la interactividad del simulador puede resultar desventajosa. Puede ocurrir que, al analista o al usuario del sistema, le interese estudiar en detalle algún aspecto de una maniobra realizada previamente y que, por tanto, ésta deba ser repetida con exactitud. Otra posibilidad es que se intente analizar la respuesta del sistema bajo diferentes configuraciones o parámetros. En este caso, para poder efectuar la comparación, es imprescindible que las entradas sea completamente idénticas, de tal forma que únicamente se observe la variación de la respuesta con respecto al parámetro estudiado. Incluso, durante el desarrollo del propio simulador, es muy útil poder reproducir una sesión cuando, por alguna circunstancia, aparecen fallos que sólo se observan bajo ciertas condiciones. Repetir dos

o más veces en el simulador la misma maniobra de forma manual es, al igual que en la vida real, imposible. En definitiva, parece muy conveniente que el simulador disponga de un sistema de captación de movimientos.

Para poder reproducir fielmente una maniobra cualquiera, será necesario disponer de las lecturas de los grados de libertad del automóvil en cada instante. El entorno no supondría un problema, ya que la simulación siempre comienza en el mismo punto de éste. Se garantiza de este modo que, ante idénticas entradas en los controles del vehículo, se obtendrán idénticas respuestas. En el caso de este trabajo, las entradas son las siguientes: giro del volante, desplazamiento de los pedales de freno y acelerador, y estado del cambio de marchas. Al simularse un automóvil con cambio de marchas automático, la posición de éste sólo es importante en ciertos instantes, como cuando el usuario pasa de “punto muerto” a “avance”, o a “marcha atrás”. En el resto de los casos (la mayor parte del tiempo), la marcha vendrá dada en función de la velocidad.

El módulo de grabación se puede activar en cualquier momento por el conductor, tanto mediante una pulsación en el teclado, como pinchando en el icono correspondiente de la pantalla, “REC”. En ese instante, los iconos de grabación y reproducción se sombrearán, y sólo funcionará el icono de detención de la operación, “STOP”. Internamente, el programa asigna un espacio de memoria en el que poder almacenar la información proporcionada por dichos grados de libertad. La subrutina encargada de leer los pedales y el volante, justo después de la lectura, registra estos datos, escribiéndolos en la memoria. Si la sesión se extiende más allá del límite de la porción de memoria reservada, el módulo de grabación comenzará a sobrescribir los datos más recientes sobre los más antiguos, de tal forma que siempre permanezcan almacenados los últimos instantes (segundos, minutos, dependiendo del tamaño de esa porción de memoria) de la maniobra. Cuando se ordena al programa la finalización de la grabación, éste vuelca toda la información en un fichero del disco duro.

El módulo de reproducción funciona de una manera muy similar al anterior, sólo que de manera inversa. Al dar la orden de reproducción, “PLAY”, el módulo carga el fichero del disco duro a la memoria. A partir de ese momento, el programa deja de leer los pedales y el volante, para suministrar al módulo de cálculo únicamente los datos que provienen del fichero. De esta manera, se está reproduciendo fielmente la acción en el tiempo. Cuando se llega hasta el final de la porción de memoria donde se

almacenan los datos, el módulo finaliza automáticamente la reproducción, y cede de nuevo el control al volante y los pedales.

El almacenamiento de maniobras no es la única utilidad posible de este módulo. De hecho, puede utilizarse para registrar cualquier tipo de resultados generados por la simulación. Se puede, por ejemplo, almacenar la posición, velocidad o aceleración de un punto cualquiera de los empleados en la modelización del vehículo. Este almacenamiento de la información en un fichero es también muy útil para su estudio posterior, representación de gráficas, etc.

5.

Módulo para el Diseño de Controladores

5.1. Organización del Módulo.

Actualmente, los requerimientos de la industria en la simulación de sistemas multicuerpo crecen continuamente. El diseño de sistemas activos como el ESP o las suspensiones activas de los automóviles o, en el mundo del ferrocarril, la basculación activa (control de la inclinación de los vagones), o el control de tracción, requiere una gran flexibilidad, que permita integrar en la simulación del sistema mecánico la simulación del control. De esta manera, los métodos de sistemas multicuerpo se emplean integrados en el denominado CACE (computer aided control engineering).

Otro campo en que simulación y control interactúan son las llamadas simulaciones con “hardware-in-the-loop” (HIL), en las que alguno de los componentes del sistema simulado es un dispositivo físico real. Esta es la estrategia que se sigue para verificar, por ejemplo, el comportamiento de las unidades de control electrónico (ECU) que manejan una parte o todo un sistema mecánico. Antes de probar el equipo en el sistema real, se chequea mediante una simulación en la que un modelo virtual sustituye al sistema mecánico controlado. Estos modelos deben trabajar

necesariamente en tiempo real, y representar lo más fiablemente posible el comportamiento del sistema.

En una fase previa, para el diseño y ajuste de los controladores, se llevan a cabo simulaciones en las que un algoritmo recrea el comportamiento del sistema de control que interactúa con el sistema multicuerpo. En este caso, la técnica empleada recibe el nombre de “software-in-the-loop” (SIL). Así, en el caso de las ECU, la unidad de control es sustituida por un modelo virtual. Al contrario que las simulaciones con HIL, las simulaciones con SIL no necesitan correr en tiempo real. Sin embargo, la velocidad y eficiencia de la simulación se traducirá en una reducción de los tiempos de diseño, por lo que la velocidad de ejecución no deja de ser un parámetro relevante.

Una etapa importante en la interacción entre el sistema multicuerpo y el control, es la comunicación entre el modelo mecánico y el software para el desarrollo de controladores. La forma de abordar esta etapa a dado lugar a diversas estrategias, entre las que cabe citar la co-simulación, y el intercambio de ecuaciones como código generado por uno de los programas, el programa de simulación del sistema multicuerpo o el de control, y que es incorporado por el otro.

En el presente trabajo, se ha desarrollado un “conductor virtual” para el modelo de vehículo. Dicho conductor se encarga del control total del vehículo, y para ello se han diseñado dos controladores: uno decide la actuación sobre el volante, y el otro maneja los pedales de freno y acelerador.

Según se ha visto, el proceso de diseño requiere la conexión del modelo con un software para el desarrollo de controladores. En este caso se ha empleado Matlab. Como técnica de control se ha escogido la Lógica Borrosa. Durante la fase de diseño, el modelo interactúa con una sesión de Matlab para el diseño y ajuste de los controladores. Observando el comportamiento de los controladores, el ajuste de sus parámetros se realiza manualmente hasta conseguir el comportamiento adecuado. Posteriormente, los controladores se preparan para ser empleados en tiempo real. Para ello, se almacenan en matrices de tal manera que su aplicación sea más rápida.

Si nos fijamos en el proceso completo, la totalidad del programa puede ser considerada como una herramienta para el diseño de controladores. Ésta permite emplear el modelo virtual del vehículo para el diseño, ajuste e implementación de controladores. En el caso presente, estos controladores

afectan al vehículo completo, pero podrían corresponder a alguno de sus subsistemas.

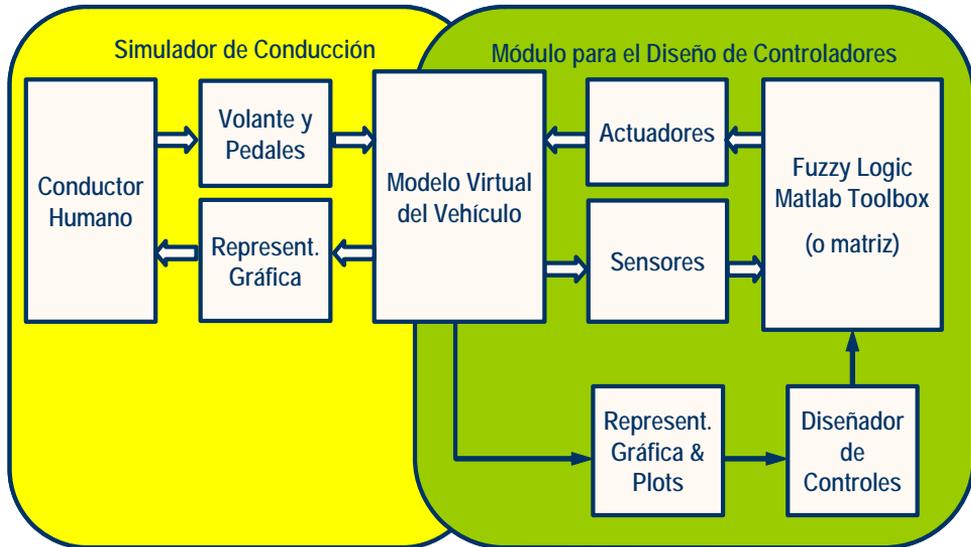


Figura 5.1 Esquema del Módulo y Conexión con el Simulador.

La Figura 5.1 representa un esquema del módulo y su conexión con el simulador de conducción. Como se puede observar, los controles reciben información de lo que podríamos denominar unos “sensores virtuales”, definidos en el modelo virtual, y opera sobre unos “actuadores virtuales” también definidos en el modelo virtual. Para determinar qué sensores (qué información se recibe) y qué actuadores (qué información se devuelve) se introducen en el modulo, el criterio ha sido replicar en lo posible la información que se obtendría del vehículo real, y la forma de actuar sobre él. En el caso de los actuadores, los controles actúan sobre el volante y los pedales. No se actúa sobre la palanca de velocidades puesto que, al ser la caja de cambios del motor automática, no es necesaria su manipulación en las maniobras para las que se han diseñado los controles. En lo que respecta a los sensores, como se verá más tarde, todas las magnitudes requeridas como inputs por los controles se pueden obtener a partir de la información sobre posición, velocidad y aceleración del vehículo. En el caso del modelo computacional, esta información se obtiene directamente. En el prototipo real, en cambio, se necesita un módulo intermedio para calcular los datos necesarios a partir de la información de los sensores.

Otra consideración a tener en cuenta es que, en el presente trabajo, el diseño y ajuste de los controles es realizado por el diseñador mediante un proceso iterativo. El comportamiento del vehículo al ser guiado mediante el

control diseñado, es visualizado a través de la representación gráfica que proporciona el simulador, siendo además posible obtener gráficas de los resultados que el diseñador necesite para el análisis y ajuste del controlador. A partir de aquí, surge una nueva aplicación que consistiría en la sustitución del diseñador humano por algún algoritmo de optimización que llevara a cabo el ajuste del control de manera automática.

En los apartados siguientes, se van a analizar con detalle cada uno de los componentes de esta herramienta: la técnica de control, la conexión con el software para el diseño del controlador (Matlab), y la implementación de los controladores en aplicaciones de tiempo real.

5.2. Técnica de Control Elegida.

5.2.1. Justificación del Empleo de Fuzzy Logic.

En la actualidad, las técnicas de control disponibles son muy numerosas. Junto a aquellas técnicas más tradicionales como el control PID, se han desarrollado muchas otras como las redes neuronales, los sistemas borrosos o los algoritmos genéticos. Incluso estas técnicas, a su vez, admiten ser combinadas entre sí, por ejemplo empleando una red neuronal para ajustar un control borroso.

A su vez, para modelizar el comportamiento de un conductor, se han ensayado diferentes técnicas de control pero, en general, no son suficientemente realistas. El principal problema estriba en el hecho de que el comportamiento del conductor depende de factores físicos y psicológicos, así como de la situación en que se encuentre el tráfico. Estos problemas se han tratado de superar empleando tanto la teoría clásica de control como la borrosa.

En la teoría clásica, el conductor se modeliza como un controlador PID continuo (Kiencke and Nielsen 2000). El controlador PID reduce los errores laterales de una forma eficaz, pero no se comporta como un conductor humano. Para una simulación vehículo-conductor en lazo cerrado, a menudo es suficiente con esta aproximación.

La aproximación del controlador PID consiste en dos subcontroladores casi independientes, uno para la dinámica lateral y otro para la longitudinal. El acoplamiento entre ambos subcontroles tiene lugar en situaciones de

conducción críticas, en las que el control longitudinal disminuye la actuación sobre el pedal del acelerador.

Sin embargo, como se ha dicho ya, el comportamiento de un conductor se ve afectado por numerosos factores muy difíciles de reflejar mediante las técnicas clásicas. En general, el diseño de un controlador clásico sigue un proceso que consiste en modelizar el sistema aplicando técnicas de linealización, desarrollar un controlador y evaluar su comportamiento (Ellis 1991). Sin embargo, un automóvil se comporta como un sistema altamente no lineal. No es una tarea fácil desarrollar un modelo del vehículo que represente el comportamiento completo de la planta y refleje, al mismo tiempo, las características no lineales del sistema. Debido a su capacidad para reflejar la no-linealidad de un sistema, frente al planteamiento clásico, la Lógica Borrosa es una buena elección para llevar a cabo el control tanto de un vehículo completo (Kageyama and Pacejka 1991) como de algunos de los subsistemas que lo componen (Zhang 2003, Chen and Wang 1996 y Yuan et al. 1999).

Entre las características de las aplicaciones de la Lógica Borrosa al control de vehículos, uno de los rasgos más destacados es que se emplean modelos basados en las expresiones comunes del lenguaje para describir el control de un proceso (Zadeh 1996), y que puede aplicarse a sistemas formados por mecanismos similares. Esto hace posible diseñar un controlador borroso genérico para sistemas análogos.

Una práctica común al abordar el problema del seguimiento automático de una trayectoria con un vehículo automóvil (path tracking problem), es el empleo, para el diseño del controlador, de un modelo simplificado del vehículo, como puede ser el modelo de la bicicleta. Al emplear Lógica Borrosa no es necesario recurrir a este modelo simplificado.

El control borroso es una técnica de control basada en el conocimiento, que imita las estrategias del hombre para controlar sistemas complejos, (Hopgood 2001). A la hora de imitar el comportamiento de un conductor, el modelo debe tener en cuenta sus capacidades físicas, tales como la anticipación, la adaptación al cambio de las características dinámicas del vehículo controlado, el aprendizaje, y la capacidad de planificar sus tareas. De alguna manera, un conductor tiene en mente lo que se podría llamar un “modelo interno”, intuitivo, del vehículo. Esto le permite relacionar su actuación sobre el vehículo con un comportamiento esperado del mismo, y reaccionar ante las posibles discrepancias (Macadam 2003). Las fronteras

de este “modelo interno” no son muy exactas, y este fenómeno encaja muy bien, como se verá, con el concepto de la Lógica Borrosa de pertenencia a un conjunto borroso.

Una última razón, de índole práctica, es que Matlab dispone de una aplicación para el desarrollo de controladores mediante Lógica Borrosa. Este hecho simplifica enormemente la programación y el ajuste de los controladores.

5.2.2. Características Principales de la Lógica Borrosa.

La Lógica Borrosa o “Fuzzy Logic” se puede definir como un modelo de razonamiento lógico que se basa en el concepto de la verdad parcial, es decir, un valor de verdad entre lo absolutamente verdadero y lo absolutamente falso. En este sentido, la Lógica Borrosa se puede interpretar como un superconjunto de la Lógica Clásica, la cual sólo contempla la verdad y falsedad absolutas.

El modelo de la Lógica Borrosa fue presentado por primera vez por el profesor Lofti A. Zadeh en los años sesenta con el objetivo de modelar la incertidumbre del lenguaje humano (Zadeh 1965). En este sentido, la Lógica Borrosa surgió como una respuesta a las insuficiencias que presentaban los lenguajes de programación convencionales ante procesos complejos, difíciles de modelar matemáticamente, y en los que la simplificación de los cálculos no conducía a buenos resultados. De esta manera, la Lógica Borrosa ofrece a los sistemas de control la capacidad de emplear reglas de carácter intuitivo propias del lenguaje humano, lo cual hace posible abordar con relativa sencillez la programación de tareas complejas relacionadas con el comportamiento y razonamiento de las personas.

La inferencia borrosa es el proceso de asociar unos datos de entrada a unos de salida a partir de un conjunto de reglas, más bien genéricas, que orientan acerca de qué decisiones adoptar o qué patrones seguir. Los métodos de inferencia borrosa más empleados son el Mamdani (Mamdani and Asslian 1975) y el Sugeno, (Sugeno and Kang 1988). En este trabajo se ha empleado el método Mamdani, que se describe en detalle a continuación.

En el proceso de inferencia difusa intervienen tres elementos principales: las funciones miembro, los operadores lógicos borrosos, y las reglas si-entonces (if-then).

Una función miembro es una curva que asocia a los valores de una variable de entrada unos valores de pertenencia comprendidos entre 0 y 1. La Fuzzy Logic Toolbox de Matlab incluye varios tipos de funciones miembro, desde simples funciones lineales a distribuciones gaussianas o curvas polinómicas. El rango de los valores de entrada se denomina universo del discurso.

Los operadores lógicos difusos tienen su equivalencia con los operadores lógicos clásicos AND, OR y NOT. Considerando dos variables A y B cuyos valores están limitados a un rango entre 0 y 1, la operación lógica A AND B equivale a la función $\min(A,B)$, que toma el valor mínimo de A y B. Del mismo modo, la operación A OR B se corresponde con el máximo de ambos valores, es decir, con la función $\max(A,B)$. Por último, la operación NOT A es equivalente a $1-A$, es decir, todo los valores del universo de discurso excepto los de la variable A. En resumen, los correspondientes operadores difusos para los operadores clásicos intersección (AND), unión (OR) y complemento (NOT) son, respectivamente, los operadores mínimo, máximo y complementario.

Las reglas si-entonces (if-then) son declaraciones condicionales de la forma “si X es A entonces Y es B”, donde A y B son valores lingüísticos definidos por conjuntos difusos pertenecientes al universo del discurso de X e Y. La parte “X es a A” se denomina antecedente, mientras que “Y es a B” se denomina consecuente. La interpretación de la regla si-entonces implica la evaluación del antecedente, y la aplicación de ese resultado al consecuente.

A partir de estos tres elementos, el proceso de evaluación de un sistema borroso se puede dividir en tres partes, como se puede ver en la Figura 5.2.



Figura 5.2 Etapas de proceso en un sistema borroso.

Los inputs del proceso son valores numéricos exactos, pertenecientes a una variable, que suelen ser fruto de la medida de sensores. Estos inputs son transformados en variables lingüísticas con grados de pertenencia entre 0 y 1 a las funciones miembro. A este proceso se le denomina “fuzzificación”. Los inputs de carácter lingüístico son evaluados mediante las reglas borrosas si-entonces y dan lugar, a su vez, a outputs de tipo

lingüístico. Outputs numéricos y exactos se obtienen a continuación mediante el proceso de “defuzzificación”.

El proceso de defuzzificación se puede llevar a cabo mediante diferentes métodos. El más empleado, que es el que se ha aplicado en este trabajo, es el método del Centro de Gravedad o del Centroide. Como se ha visto, cada una de las reglas de inferencia asocia, a un conjunto de inputs de carácter lingüístico, un grado de pertenencia a los conjuntos borrosos de las salidas. El resultado concreto depende conjuntamente de todas las reglas. El método del centroide determina la posición del centro de gravedad del área que representa el grado de pertenencia a los conjuntos borrosos de salida.

La Figura 5.3 ilustra un ejemplo de inferencia mediante Lógica Borrosa. Los inputs son dos variables, T y P . La variable T tiene su universo dividido en tres conjuntos (bajo, medio y alto), mientras que la variable P se divide en dos conjuntos (bajo y alto). A su vez, como output se tiene la variable y y cuyo universo está dividido en tres conjuntos (pequeño, medio y grande).

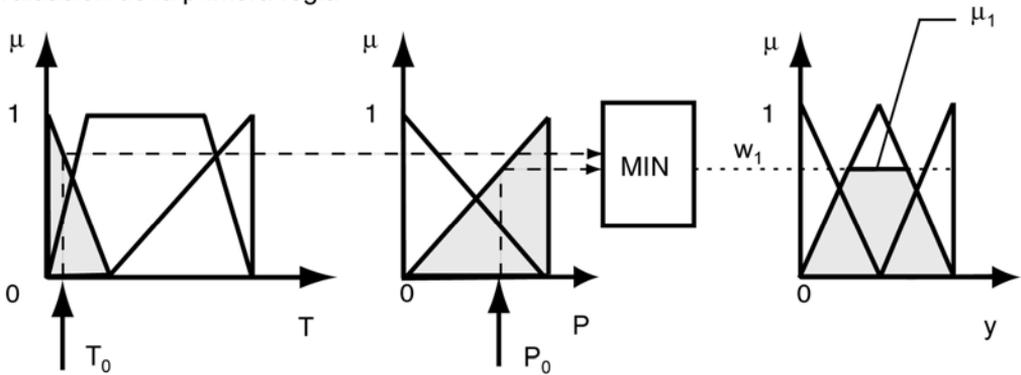
El conjunto de reglas está formado por las siguientes dos reglas:

SI	T =bajo	Y	P =alto	ENTONCES	y =medio
SI	T =medio	Y	P =alto	ENTONCES	y =pequeño

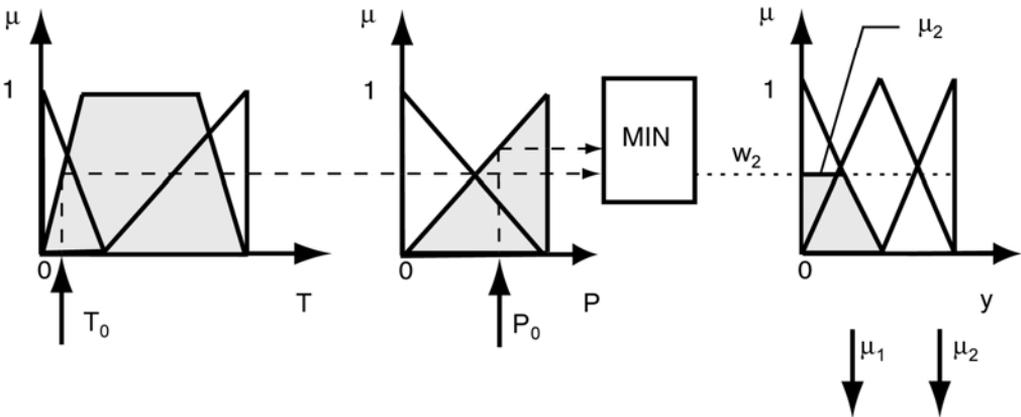
A continuación se evalúan unos valores concretos de las variables de entrada, T_0 y P_0 . Como esquema de inferencia se escoge el mínimo. Al evaluar la primera regla comprobamos el grado de pertenencia de T_0 al conjunto bajo y el grado de pertenencia de P_0 al conjunto alto. El mínimo de ambos es el grado de pertenencia de P_0 al conjunto alto. A este valor lo denominamos w_1 y determina, a su vez, el grado de pertenencia al conjunto medio de la variable de salida, μ_1 . Si el conjunto de reglas estuviese formado sólo por esta regla, el valor de la variable de salida y estaría determinado por la posición del centro de gravedad del área sombreada.

Una vez evaluada la primera regla, se procede de igual manera con la segunda. Se comprueba el grado de pertenencia de T_0 al conjunto medio y el de P_0 al conjunto alto. Cabe resaltar cómo un mismo valor de T_0 pertenece, con distintos grados de pertenencia, a dos conjuntos. Lo mismo ocurre con el valor de P_0 , pero las reglas no exigen evaluar el grado de pertenencia al conjunto bajo. Ahora, w_2 determina el grado de pertenencia de la variable de salida al conjunto pequeño, μ_2 .

evaluación de la primera regla



evaluación de la segunda regla



agregación

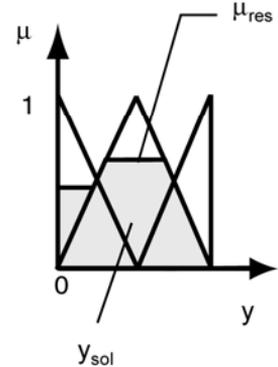


Figura 5.3 Ejemplo de evaluación de un sistema de reglas.

Por último, para determinar el valor final de las variables de salida, tiene lugar el proceso de agregación. En este paso, se suman las áreas determinadas al evaluar las reglas, y se determina el centro de gravedad

del área resultante. La posición del centro de gravedad determina el valor concreto de la variable de salida, y_{sol} .

5.3. Diseño de los Controladores: Conexión con Matlab.

Para muchas aplicaciones relacionadas con el control de vehículos la eficiencia es de vital importancia, tanto para acortar el proceso de diseño, como para su ejecución en tiempo real. En este trabajo se contemplan ambas posibilidades. Por un lado, durante el proceso de diseño, una simulación rápida del vehículo gobernado por el controlador es muy deseable, puesto que permite al diseñador la realización de más iteraciones del proceso de diseño en menos tiempo. Mediante este procedimiento se alcanza antes el ajuste del controlador (este requerimiento es todavía más importante si el diseñador es sustituido por un algoritmo de optimización para el ajuste del controlador). De todas formas, si el objetivo perseguido no es el control total del vehículo sino sólo el control parcial, como puede ocurrir en el caso de aplicaciones dirigidas a la asistencia al conductor, la simulación también necesita la participación de una persona dentro del lazo de simulación (human-in-the-loop). Esto conlleva el funcionamiento en tiempo real de la simulación, incluido el control. Por otro lado, una vez que el controlador está ajustado, es posible que se utilice en aplicaciones que trabajen en tiempo real. Para ello, es necesario asegurar que su implementación es suficientemente rápida para asegurar ese propósito.

Como se ha indicado anteriormente, para el desarrollo de controladores con Lógica Borrosa se va a emplear la Fuzzy Logic Toolbox de Matlab. Desde el punto de vista de la eficiencia, los puntos críticos son la conexión entre Fortran y Matlab y el elevado consumo de tiempo que conlleva la evaluación de los controladores en Matlab, especialmente cuando éstos están formados por un elevado número de reglas. Por este motivo, se han estudiado dos alternativas disponibles para conectar ambos lenguajes de programación: Matlab Engine y el empleo de ficheros MEX. La metodología a seguir en ambos casos se detalla en los siguientes apartados. Como alternativa más eficiente para el caso de empleo del controlador una vez ajustado, se plantea el almacenamiento de los controladores diseñados en matrices. Este procedimiento se detalla en el apartado 5.4.2.

La Fuzzy Logic Toolbox proporciona un interfaz bastante intuitivo para el diseño de controladores. Se accede a él tecleando el comando *fuzzy* en el

espacio de trabajo de Matlab. Los controladores diseñados son almacenados en ficheros de extensión *.fis*. Para trabajar con ellos es preciso cargarlos en el espacio de trabajo. Esto se lleva a cabo mediante el comando *readfis* cuyo argumento es el fichero a cargar. Para evaluar un vector de inputs del controlador se emplea la función *evalfis*. Los argumentos de esta función son el vector de inputs y el controlador a emplear, y devuelve un vector con los outputs correspondientes.

Finalmente, es necesario tener en cuenta una última consideración antes de abordar las distintas formas de conexión con Matlab. Matlab es un programa orientado al trabajo con vectores y matrices. En este sentido, el programa considera los escalares como un caso particular de matrices con dimensión 1. Además, trabaja siempre en doble precisión, es decir, guardando cada dato en 8 bytes, con 15 cifras decimales exactas. Por este motivo, los distintos tipos de variables empleados en Fortran deben ser transformados a doble precisión.

5.3.1. Matlab Engine.

Matlab Engine es una librería que contiene una serie de subrutinas que permiten comunicarse con Matlab desde otros programas. De esta manera, se puede emplear Matlab como motor de cálculo. Los programas que pueden recurrir a esta utilidad han de estar escritos en Fortran o C, y se comunican con Matlab a través de un canal de comunicación. La librería de rutinas de Fortran o C permite iniciar y terminar una sesión de Matlab, enviar y recibir datos, y enviar comandos para ser procesados.

Para emplear Matlab Engine es necesario compilar el programa incluyendo las librerías correspondientes. Matlab proporciona las librerías para los compiladores más habituales. Existen también dos posibilidades para compilar y montar. Una de ellas es crear un fichero *make* en Matlab, y ejecutarlo desde la ventana de comandos. Para proceder de esta manera, se indica que se va a crear un fichero MEX, y se introducen las opciones de compilación correspondientes a Engine, que fundamentalmente sirven para indicar la situación de las librerías. Por ejemplo, el siguiente código

```
mex -f C:\MATLAB6p1\bin\win32\mexopts\df60engmatopts.bat ...
```

```
Coche.f
```

se encarga de compilar y linkar el programa Coche.exe a partir del fichero Fortran, *Coche.f*. El parámetro *-f* indica que, para compilar y linkar, se

emplea *df60engmatopts.bat*. Este fichero contiene las opciones de compilación con las librerías necesarias para el compilador Digital Fortran 6.0.

Esta opción es eficaz, pero se pierde una de las grandes ventajas de emplear Matlab Engine, que consiste en poder ejecutar el programa en modo depuración empleando Matlab. Para ello, algunos compiladores permiten incluir las librerías correspondientes a Matlab Engine. En la Figura 5.4, se representa su inclusión en el entorno Microsoft Visual Studio, que es el que se ha empleado en este trabajo. Como se puede ver, las librerías correspondientes son *libeng.lib*, específica para Matlab Engine, y *libmx.lib*, que contiene las funciones para manipulación e intercambio de variables entre los dos códigos (esta librería se emplea también cuando se utilizan los ficheros MEX). La Tabla 5.1 recoge las funciones empleadas de ambas librerías.

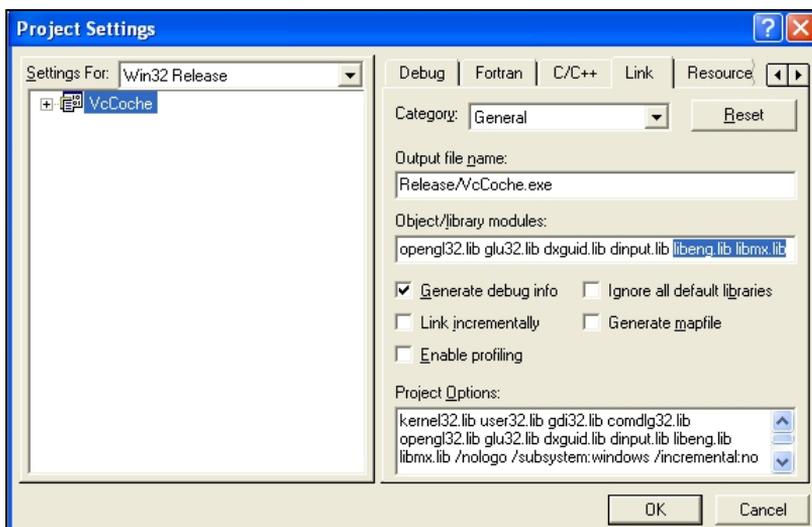


Figura 5.4 Introducción de las librerías de Matlab Engine.

Una vez hecho esto, la Figura 5.5 detalla el procedimiento que se ha seguido en el programa. El primer paso consiste en abrir una sesión de Matlab, nada más inicializar el programa, mediante la función *engOpen*. Esta sesión permanecerá abierta durante la ejecución del programa. En este punto hay también dos posibilidades, puesto que se puede abrir una sesión nueva o bien conectarse a una sesión ya abierta de Matlab. Un inconveniente para conectarse a una sesión ya abierta es que es necesario compilar desde Matlab, y no se puede hacer incluyendo las librerías en el compilador. Por otro lado, si se inicia una nueva sesión, llevará más tiempo

la ejecución del programa puesto que incluye también el tiempo de inicio del mismo.

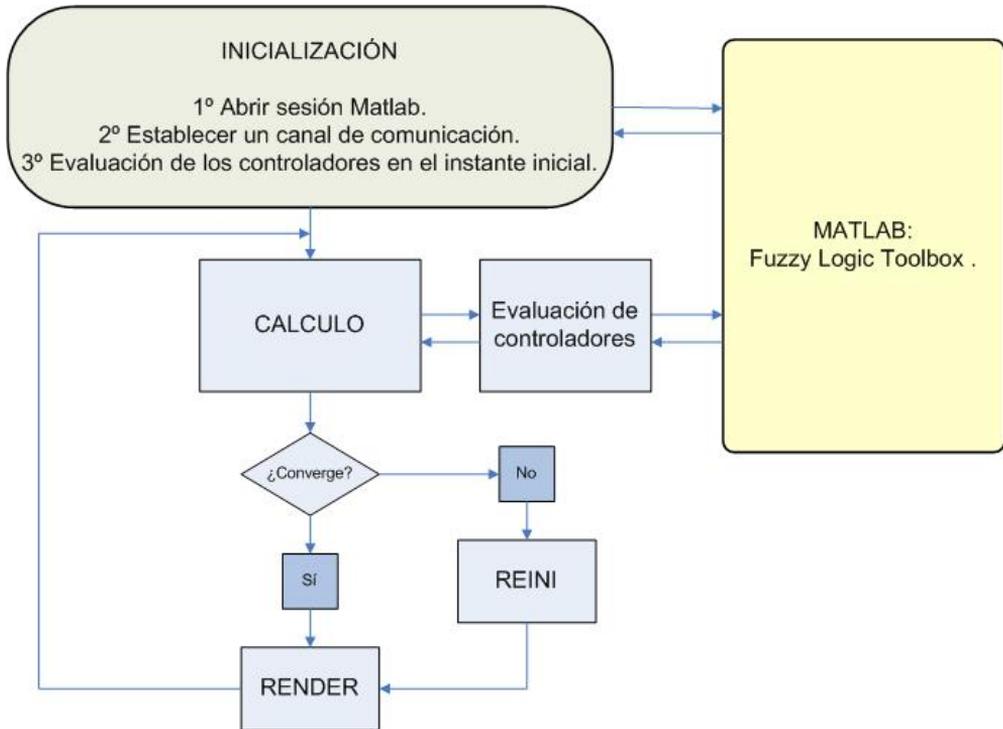


Figura 5.5 Esquema del empleo de Matlab Engine.

La función *engOpen*, tras iniciar la sesión de Matlab, establece un canal de comunicación entre ambos códigos. Este canal es el que se utilizará para la transmisión de datos durante la ejecución del programa, y se cierra al finalizar éste. También conviene tener en cuenta que, para poder ejecutar comandos de Matlab, es necesario escribir a través del canal de comunicación el comando correspondiente. Este procedimiento ralentiza la ejecución del programa. Por este motivo, conviene que los nombres de las variables que se transmiten y se manipulan sean cortos.

Una vez inicializada la sesión de Matlab, el siguiente paso consiste en la carga de los controladores borrosos, que están almacenados en el disco duro, en la sesión activa. De esta forma, los controladores quedan disponibles para su utilización. Para ello, se emplea la función *engEvalString*, que se encarga de “escribir” una cadena de caracteres a través del canal de comunicación. En la Fuzzy Logic Toolbox, la función *readfis* es la encargada de cargar un controlador almacenado en el disco.

Funciones de la librería ENGINE	
Función	Propósito
engOpen	Abre el canal de comunicación con Matlab
engClose	Cierra el canal de comunicación con Matlab
engEvalString	Ejecuta un comando de Matlab
engGetArray	Obtiene un array de Matlab a través del canal de comunicación
engPutArray	Envía un array de Matlab a través del canal de comunicación
Funciones de la librería MX	
Función	Propósito
mxSetName	Especifica el nombre de un array <i>mx</i>
mxCreateFull	Crea un array <i>mx</i> tridimensional
mxGetPr	Obtiene los datos de un array <i>mx</i>
mxCopyReal8ToPtr	Copia un array de reales de 8 bits a un array <i>mx</i>
mxCopyPtrToReal8	Copia un array <i>mx</i> a un array de reales de 8 bits

Tabla 5.1 Funciones empleadas en la conexión Fortran-Matlab.

A continuación, se procede a la evaluación en el instante inicial de los controladores. Como se ha indicado anteriormente, la evaluación de los controladores proporciona la posición del volante y el desplazamiento de los pedales de freno y acelerador. Para evaluar los controladores es necesario pasar el vector de inputs desde Fortran a Matlab. Como se ha mencionado ya, los tipos de datos que emplean los códigos son distintos, por lo que el proceso es un tanto complejo y consiste en:

- Crear un array de Matlab mediante la función *mxCreateFull*.
- Empleando la función *mxSetName*, dar nombre al vector.
- Introducir en el array de Matlab los datos del vector de inputs. La función *mxCopyReal8ToPtr* es la encargada de esta tarea.
- Trasladar con el comando *engPutArray* los datos a Matlab.
- Evaluar el controlador y tomar el vector de outputs mediante *engGetArray*.
- Trasladar los datos del array *mx* a un vector de reales en Fortran con la función *mxCopyPtrToReal8*.

Una vez superado el instante inicial, durante el proceso iterativo necesario para obtener la solución en el próximo instante, el canal de comunicación

permanece abierto y la sesión de Matlab continúa activa. En cada paso de tiempo el sistema vuelve a evaluar los controladores borrosos, recibiendo los nuevos inputs para el estado del vehículo.

Finalmente, las ventajas de Matlab Engine se podrían resumir en las siguientes:

- Proporciona una forma sencilla y eficaz de emplear Matlab como motor de cálculo.
- Hace posible ejecutar el programa en modo depuración, pudiendo acceder prácticamente a todas las partes del programa, tanto al modelo virtual del vehículo como al módulo de control.
- El tiempo de ejecución es suficientemente rápido.
- Facilita la tarea de obtención de gráficas, etc. que son de gran ayuda para el análisis y el ajuste de los controladores.

Como inconvenientes se podrían citar los siguientes:

- La conexión con Matlab requiere la inicialización de una sesión de Matlab. Esto ralentiza considerablemente el tiempo de ejecución. Es posible conectar con una sesión preexistente, pero eso requiere compilar perdiendo todas las opciones de depuración.
- El modo de ejecutar los comando en Matlab es escribiéndolos a través del canal de comunicación.

5.3.2. Ficheros MEX.

Este procedimiento se podría considerar como el contrario al empleo de Matlab Engine, puesto que, en este caso, se trata de llamar desde Matlab a las funciones y subrutinas programadas en Fortran o C. De esta manera, el código Fortran se puede emplear como si se tratase de un fichero propio de Matlab. Este procedimiento se suele emplear para llamar desde Matlab a programas extensos programados en Fortran o C, sin tener que reescribirlos en código de Matlab, o bien para aprovechar la rapidez de estos lenguajes y escribir en ellos los cuellos de botella (generalmente bucles), haciendo así más rápida la ejecución del programa en Matlab.

Al emplear este procedimiento, los programas escritos en Fortran se compilan y linkan formando librerías que pueden ejecutarse dentro del entorno de Matlab. El fichero resultante posee una extensión que depende de la plataforma en la que se ha generado. En Windows, las librerías

tendrán como extensión *.dll*. La librería *libmx.lib* que se ha empleado en Matlab Engine, se puede emplear también con los ficheros MEX para realizar el intercambio de datos entre los ficheros en Fortran y Matlab, y para la ejecución de funciones.

Para compilar un fichero MEX en Fortran, además del código, se debe añadir otro fichero de entrada. Este fichero actúa como interfaz entre el código y Matlab, y requiere cuatro parámetros: *prhs*, *nrhs*, *plhs* y *nlhs*. *prhs* es un vector que contiene los inputs que va a recibir el código; *nrhs* especifica el número de inputs a recibir; *plhs* es otro vector que contiene los outputs; y *nlhs* indica el número de outputs. Este fichero se encargará de llamar al programa en Fortran como una subrutina.

Nuevamente, todo el código se compila desde Matlab. Ahora, el aspecto que tendría el *make* del mismo ejemplo que se empleó al describir Matlab Engine es el siguiente:

```
mex Coche.f Cocheg.f
```

Como se ve, ahora aparece un nuevo fichero *Cocheg.f*, que corresponde al archivo de enlace. En este caso, no se han especificado, como en el caso de Matlab Engine, opciones para el compilador, aunque podría haberse hecho.

La Figura 5.6 muestra el esquema seguido por el programa al emplear ficheros MEX. En este caso, se parte de una sesión de Matlab iniciada por el usuario. Una subrutina de Matlab se encarga de cargar los controladores que se van a emplear, de tal manera que estén presentes en el espacio de trabajo y puedan ser llamados desde Fortran. A continuación, es necesario llamar al código Fortran. Para ello, basta con escribir el nombre de la subrutina inicial en el espacio de trabajo como si se llamase a un fichero o a una función de Matlab.

En este caso, se ha compilado todo el programa del simulador para ejecutarlo como fichero MEX. Es decir, desde Matlab se llama directamente a *main.f*. La inicialización y el cálculo se hacen exactamente igual que antes. Sin embargo ahora, en el momento de evaluar los controles, se accede desde la subrutina de Fortran al espacio de trabajo de Matlab, obteniendo la información necesaria sobre el volante y los pedales.

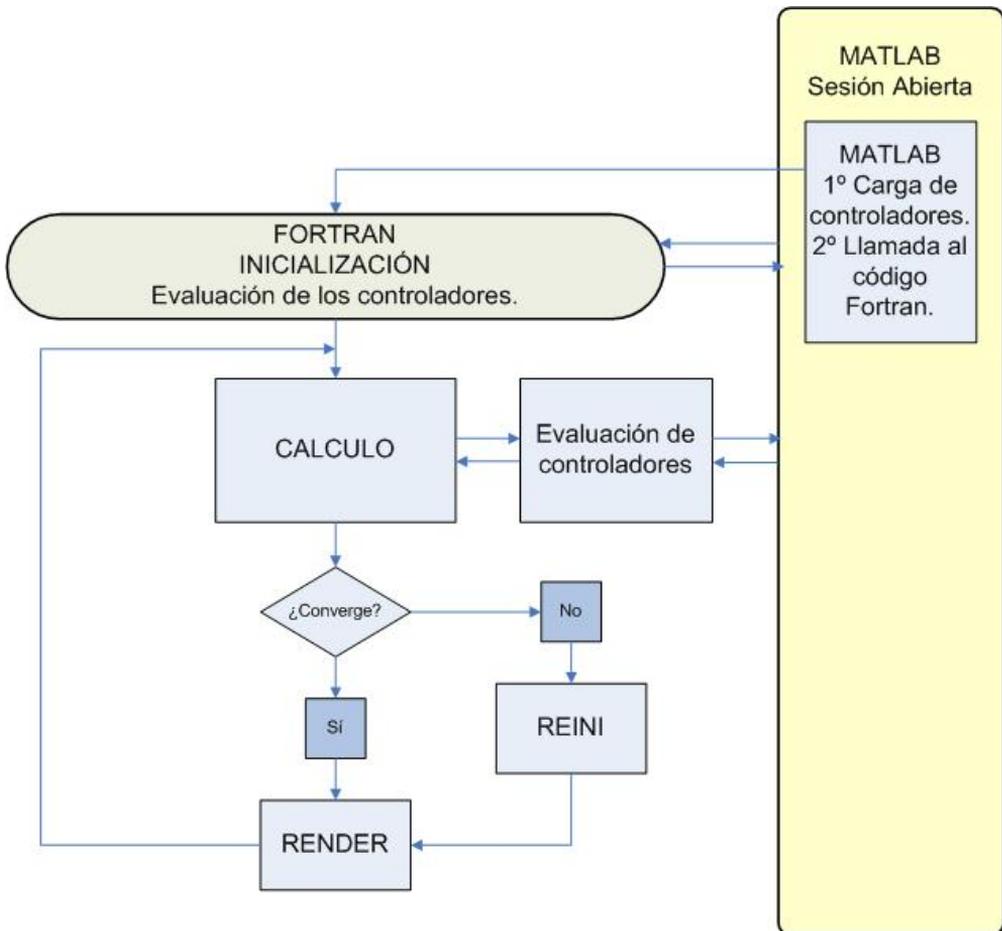


Figura 5.6 Esquema del empleo de los archivos MEX.

Al igual que con Matlab Engine, el proceso de intercambio de datos requiere una manipulación previa. Ahora, el intercambio de datos se hace siguiendo los siguientes pasos:

- Crear un array de Matlab mediante la función *mxCreateFull*.
- Dar nombre al vector empleando la función *mxSetName*.
- Introducir en el array de Matlab los datos del vector de inputs. La función *mxCopyReal8ToPtr* es la encargada de esta tarea.
- Emplear la función *mexCallMATLAB* para ejecutar una función de Matlab que evalúe los controladores para el vector de inputs, y devuelva el vector de outputs. Este paso se podría haber hecho siguiendo un esquema similar al empleado con Matlab Engine, pero ahora las funciones a emplear serían *mexPutArray*, *mexEvalString* y

mexGetArray, en lugar de sus homólogas *engPutArray*, *engEvalString* y *engGetArray*.

- Trasladar los datos del array *mx* de outputs a un vector de reales en Fortran con la función *mxCopyPtrToReal8*.

Las ventajas más reseñables de este procedimiento son:

- Es ligeramente más rápido que el empleo de Matlab Engine, como se verá en las maniobras que se han programado y resuelto mediante los dos procedimientos. De todas formas, los tiempos empleados son prácticamente similares cuando Matlab Engine se conecta a una sesión ya abierta.
- Al igual que en Matlab Engine, la realización de gráficas es sencilla puesto que es posible el intercambio de datos entre Fortran y Matlab en cualquier parte del programa.

Los inconvenientes son los siguientes:

- El gran inconveniente del empleo de los ficheros MEX estriba en que no es posible ejecutar el programa en modo depuración introduciéndose en la parte relativa al código Fortran. Por supuesto, sí es posible hacerlo en la parte relativa al cálculo en Matlab.
- Es necesario agregar un fichero de entrelazamiento entre Matlab y el código en Fortran/C.
- Al igual que en Matlab Engine, los comandos se ejecutan mediante escritura. Por lo que también se ralentiza la ejecución del programa sin permitir alcanzar, muchas veces, el tiempo real.

Funciones de la librería MEX	
Función	Propósito
mexCallMATLAB	Evalúan una función en el espacio de trabajo de Matlab.
Funciones de la librería MX	
Función	Propósito
mxSetName	Especifica el nombre de un array <i>mx</i>
mxCreateFull	Crea un array <i>mx</i> tridimensional
mxGetPr	Obtiene los datos de un array <i>mx</i>
mxCopyReal8ToPtr	Copia un array de reales de 8 bits a un array <i>mx</i>
mxCopyPtrToReal8	Copia un array <i>mx</i> a un array de reales de 8 bits

Tabla 5.2 Funciones empleadas en la conexión ficheros MEX-Matlab.

Tras el estudio de los dos procedimientos, se ha visto que Matlab Engine resulta más conveniente para la fase de ajuste de los controladores. Aunque el empleo de ficheros MEX es un poco más eficiente, no lo es lo suficiente como para que aventaje claramente a la otra opción. Además, la posibilidad que ofrece Matlab Engine de ejecutar el programa en modo depuración, recorriendo tanto el código en Fortran o C como el relativo a los controladores, hace mucho más versátil y ventajosa esta forma de proceder.

Otra opción posible que se ha desechado sin llegar a probarla es la de almacenar las funciones de Matlab, compilándolas como una librería en C. Sin embargo, este procedimiento no parece muy eficiente puesto que supone el encapsulamiento del controlador que se está ajustando y, además, aunque Matlab dispone de una herramienta para compilar el código desarrollado dentro del programa, sólo es posible compilar en librerías de lenguaje C (Vaculin et al. 2004).

5.4. Aplicación de los Controladores al Simulador de Conducción.

Con los métodos descritos en los apartados anteriores no es posible asegurar que el programa, incluyendo los controladores, funcione en tiempo real. El retraso en el tiempo se debe a diversos factores. Por un lado, se encuentra el establecimiento de una sesión de Matlab y la escritura de comandos a través del canal de comunicación. Junto a esto, la evaluación de un control borroso se hace tanto más pesada cuanto más elevado es el número de reglas que contiene. Por este motivo, si los controladores se van a emplear en aplicaciones que requieran el funcionamiento del programa en tiempo real, es necesario buscar una alternativa.

Para afrontar el problema conviene tener en cuenta que la Lógica Borrosa, en cierto sentido, genera una hipersuperficie que relaciona, de forma biunívoca, un conjunto de variables de entrada (error, velocidad, etc.) con las variables de salida (posición del volante y los pedales). Esta hipersuperficie, por lo general, no será posible aproximarla mediante una expresión matemática. Además, si fuese así, no tendría sentido el empleo de la Lógica Borrosa. Pero sí es posible almacenar en tablas el conjunto de outputs correspondiente a un conjunto representativo de inputs. Es posible evaluar una matriz que contenga las posibles combinaciones de inputs del controlador, obteniendo la correspondiente matriz de outputs. Este es el

procedimiento al que se ha recurrido para mejorar la eficiencia, en aquellas aplicaciones que así lo requieren.

5.4.1. Organización del Simulador.

Al proceder de esta manera, la organización general del simulador no se ve afectada. Únicamente, desaparece la conexión entre el programa del modelo virtual del vehículo y el software para el diseño de controladores. En su lugar, el modelo virtual interactúa con los valores almacenados en una matriz. Para ello, al principio del programa tiene lugar la carga de las matrices del controlador. Esto equivale a leer un fichero de texto donde se almacenan los datos y guardarlo en un variable. Cada uno de los controladores ha sido almacenado en un fichero distinto.

Como se puede observar en la Figura 5.7, el esquema se simplifica, al eliminar la conexión con otro programa.

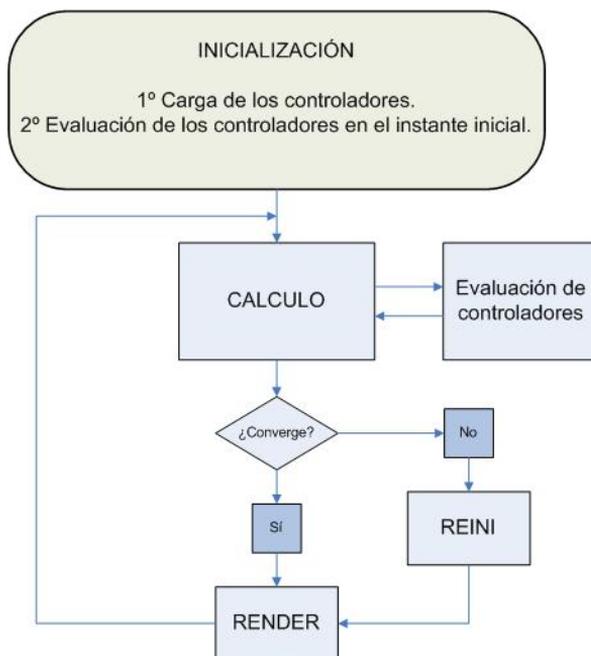


Figura 5.7 Esquema del empleo de los controladores almacenados en matrices.

Tras la carga de los controladores en variables del sistema, los controladores se evalúan, tanto en el instante inicial como en cada uno de los pasos de la iteración requerida por el cálculo de la solución en un nuevo instante.

5.4.2. Matrices del Controlador.

A la hora de elaborar las matrices, puesto que se trata de discretizar un conjunto continuo, es necesario establecer un conjunto de inputs que sea representativo del universo de discurso. La malla no puede ser demasiado fina puesto que daría lugar a matrices demasiado grandes. Al ser la malla más gruesa, se hace necesario recurrir a la interpolación. En el presente trabajo se ha buscado que la discretización sea tan gruesa como sea posible sin llegar a tener que recurrir a la interpolación. Como se ha visto, la Lógica Borrosa establece grados de pertenencia a conjuntos borrosos. La discretización debe ser tal que la pertenencia de un input a un conjunto borroso no se vea alterada. El grado de pertenencia varía entre 0 y 1 a lo largo de un intervalo. Para salvaguardar el grado de pertenencia, la forma de las funciones de pertenencia no es irrelevante. En el presente trabajo se han empleado funciones de pertenencia trapezoidales y triangulares, como las representadas en la Figura 5.8. Como se puede observar en la figura, los intervalos de la variable x en los que el grado de pertenencia al conjunto es inferior a 1 no son siempre de igual ancho. En este caso, los intervalos serían a y b . El paso de discretización, s , que se ha empleado es la décima parte del intervalo de variación de pertenencia más pequeño, el b en la Figura 5.8. Este valor no es arbitrario sino que se ha observado que, con ese valor, el comportamiento del controlador es similar a cuando se emplea Matlab Engine y se evalúan los inputs como variables continuas.

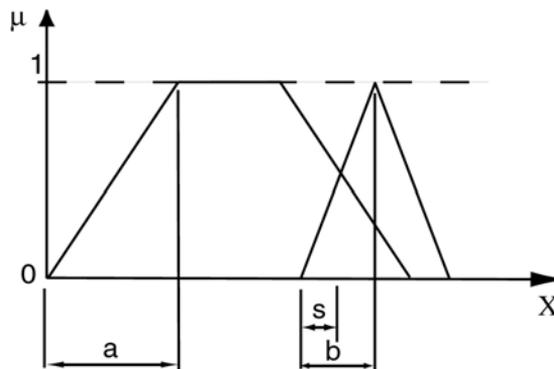


Figura 5.8 Conjuntos borrosos y paso de discretización.

Una vez seleccionado el paso a emplear en la discretización, se forma una matriz de inputs que recoge las distintas combinaciones posibles de los inputs del controlador. Cada fila de la matriz representa una combinación

de inputs. Por consiguiente, la matriz tendrá tantas columnas como inputs tenga el controlador.

Es posible evaluar esta matriz en Matlab mediante la función *evalfis* de la Matlab Toolbox. Esta función admite que sus inputs sean también matrices y no sólo valores escalares, con la única condición de que cada fila de la matriz esté formada por un vector de inputs. El resultado será una nueva matriz con los outputs correspondientes.

A la hora de ejecutar el programa, los inputs para los controladores son reducidos al valor más próximo de la matriz de inputs. Entonces, es posible determinar su posición en la matriz de inputs. El valor de los outputs se encontrará buscando la fila correspondiente en la matriz de outputs.

5.4.3. Control Parcial.

Los controladores desarrollados responden a lo que se podría denominar un “conductor virtual”. Sin embargo, se puede emplear el módulo de diseño de controladores desarrollado para diseñar e implementar controladores que afecten únicamente a una parte del sistema. Por ejemplo, como se ha indicado con anterioridad, se han diseñado dos controladores, uno para gobernar el volante y otro para actuar sobre los pedales. Es muy sencillo emplear solamente uno de los dos controladores, dejando al conductor real manipular el otro.

Se ha desarrollado un ejemplo, un tanto más complejo, de control parcial, que consiste en que tres vehículos (el número es arbitrario) forman parte de la simulación. Uno de ellos es guiado por un conductor real y los otros dos por los controladores diseñados (Figura 5.9).

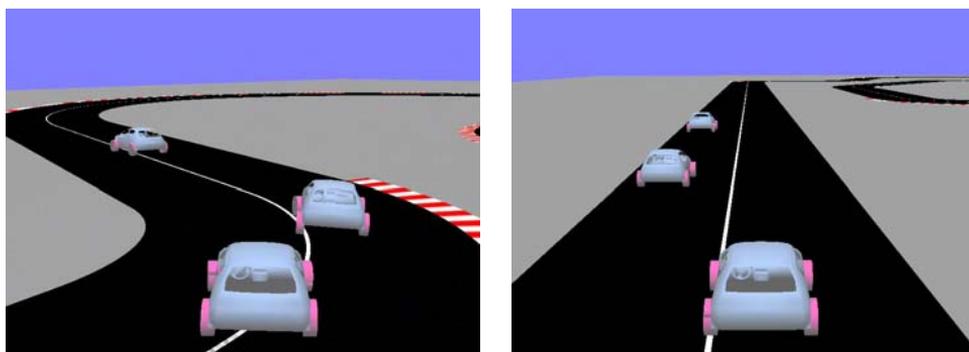


Figura 5.9 Dos situaciones en la simulación de tres vehículos.

Como se puede suponer, esta posibilidad facilita el desarrollo de controladores para subsistemas como suspensiones activas, sistemas de frenado, control de estabilidad, técnicas de control del ángulo de guiñada (yaw control), etc.

6.

Diseño de Controladores: Ejemplos

6.1. Elección de las Maniobras.

Para ilustrar las capacidades de uso de la herramienta de diseño de controladores, se han escogido dos maniobras sencillas, y se han desarrollado los controladores que permitan que el vehículo las realice automáticamente. Conviene no perder de vista que el propósito de las maniobras no es buscar los mejores controladores posibles, sino observar que, merced al uso de la herramienta propuesta, se pueden obtener unos resultados más que aceptables con rapidez.

Para decidir las maniobras a realizar, el simulador juega un papel muy importante. Diseñar una maniobra “en abstracto” puede conducir a situaciones sin solución, tanto para un controlador como para un conductor humano, o a que los requerimientos impuestos al controlador sean excesivamente exigentes. Por este motivo, la maniobra se lleva a cabo en primer lugar en el simulador, verificando la posibilidad de realizar la maniobra y observando el comportamiento del conductor.

Para el diseño de los controladores, el procedimiento comienza con el conductor realizando la maniobra. A partir de la información obtenida y de la percepción del desempeño de la maniobra (velocidad máxima alcanzada, forma de acelerar y frenar, etc), se diseña un controlador inicial. Para ajustar este controlador se sigue un proceso iterativo. Se ejecuta la maniobra con el controlador detectando las carencias y corrigiéndolas hasta alcanzar el comportamiento deseado.

Asimismo, para evaluar el desempeño que los controladores llevan a cabo de las maniobras, se ha optado por compararlo con la misma maniobra realizada por un conductor real en el simulador. Este procedimiento podría haberse desarrollado más mediante un estudio estadístico, empleando varios conductores y hallando los valores característicos del grupo en la resolución de la maniobra. Tampoco se ha tenido en cuenta la influencia del simulador y de sus características específicas (volante y pedales tipo videojuego, representación gráfica en la pantalla, etc.) en los conductores. Sin embargo, parece que el estudio estadístico y la influencia de los factores propios del simulador excedían los objetivos del presente trabajo, puesto que simplemente se trataba de disponer de una información realista para evaluar, en una primera aproximación, el comportamiento de los controladores.

6.2. Primera Maniobra: Trayectoria Recta.

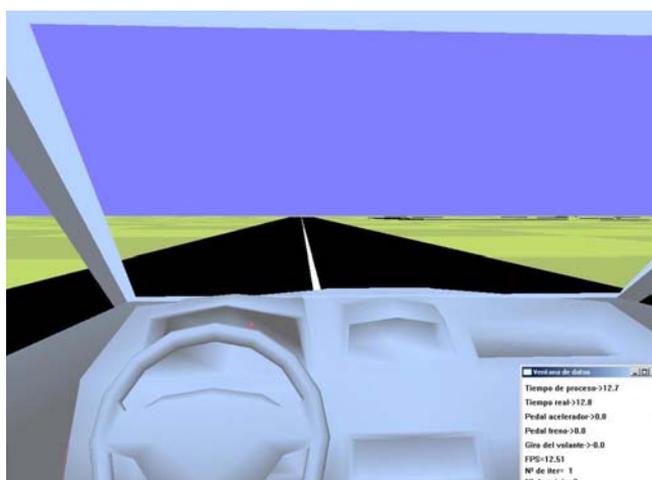


Figura 6.1 Primera maniobra: trayectoria recta.

La primera maniobra, representada en la Figura 6.1, es muy sencilla. El vehículo parte del reposo, recorre un camino recto y frena, deteniéndose a

20 m de la posición de partida. El propósito de esta maniobra es detener el vehículo lo más cerca posible de la distancia objetivo. Para esta maniobra, solamente se necesita el control de los pedales de freno y acelerador. Como se ha indicado en el apartado anterior, la maniobra es llevada a cabo inicialmente por un conductor. A partir de los resultados obtenidos por éste, se ha permitido una velocidad máxima de 5 m/s, y una aceleración límite de 3 m/s², tratando de evitar que la aceleración y el frenado sean muy bruscos. El tiempo para realizar la maniobra no está limitado.

El esquema de actuación que se ha desarrollado para la actuación automática sobre los pedales se representa en la Figura 6.2. Como se puede observar, las variables de entrada son la posición, la velocidad y la aceleración del coche. En el vehículo real, el valor de estas variables se obtendría mediante sensores. El controlador, actuará sobre los pedales del acelerador y el freno. En el vehículo real esto se podría realizar accionando unos motores. En el caso de una maniobra real, se debería controlar también el volante para corregir desviaciones de la trayectoria debidas a irregularidades del terreno, defectos en la alineación inicial, etc. Sin embargo, en la maniobra teórica, ejecutada sobre un suelo totalmente plano, este control resulta irrelevante y, consecuentemente, no se ha tenido en cuenta.

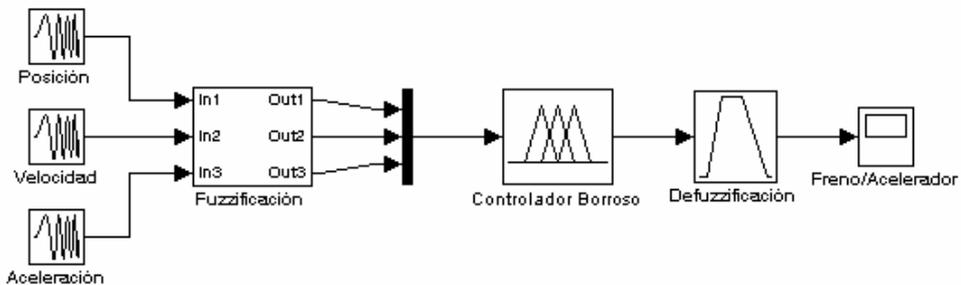


Figura 6.2 Esquema del control de la actuación sobre acelerador y freno.

La elección de los conjuntos borrosos se ha realizado de una forma intuitiva. Se ha pretendido reflejar la descripción simplificada del movimiento del coche durante la maniobra que se haría en términos lingüísticos, así como las decisiones tomadas por el conductor. Si la descripción de los conjuntos relativos a la posición se hiciese en términos de control, en lugar de la posición tendríamos que hablar más bien del error. En el fondo, el error no es más que la distancia a recorrer menos la posición del vehículo, y se puede considerar como la misma variable. Sin

embargo, para el conductor parece más intuitivo hablar de arranque, llegada, freno, etc.

El primer input o variable que se considera es la posición del coche. La distancia total a recorrer se ha dividido en cinco intervalos: el arranque (desde 0 a 4,5 m), la marcha (que cubre el tramo más largo, entre 2,5 y 17,5 m), la frenada (de 16,5 a 19,5 m), la parada (de 18,75 a 20), y el rebase (a partir de 20 m).

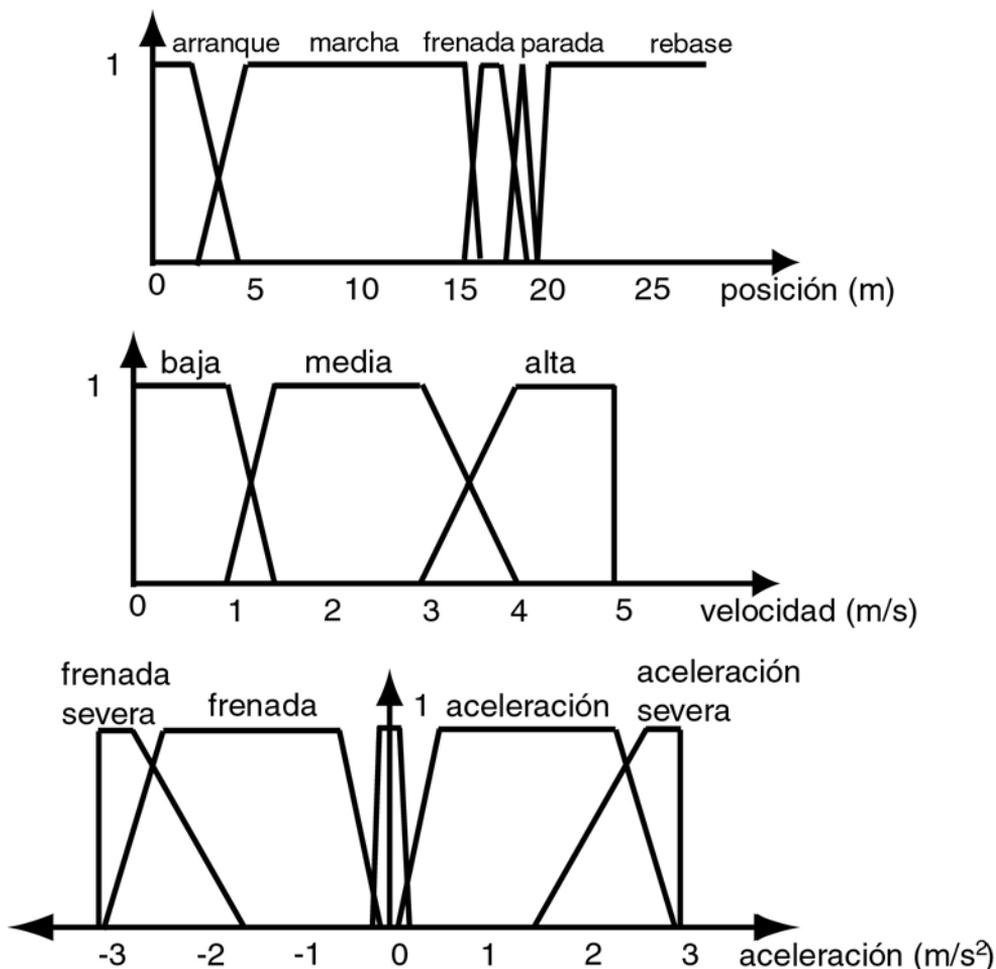


Figura 6.3 Conjuntos borrosos para los inputs de posición, velocidad y aceleración.

El segundo y el tercer input son la velocidad y la aceleración del vehículo, respectivamente. Para evitar un número excesivo de posibles combinaciones a la hora de formular las reglas, sólo se han empleado tres conjuntos borrosos para cada uno de ellos: valores de las variables bajo, medio y alto, como se puede observar en la Figura 6.3. No se contemplan

valores negativos para la velocidad y, en el caso de la aceleración, los valores de los conjuntos que se han supuesto se traducen en frenada severa, frenada, aceleración nula, aceleración y aceleración severa. Tampoco se ha contemplado el caso de velocidad nula. Dicho estado solamente puede presentarse en la situación de arranque o de parada y, en ambas situaciones, es posible enunciar las reglas de actuación sin tener en cuenta la velocidad (v. gr. “si la posición es parada, el pedal está en freno a fondo”). Asimismo, cuando alguno de los inputs rebasa el máximo del universo de discurso, se le da el valor máximo.

Se han empleado conjuntos trapezoidales para las variables que recogen los inputs del sistema. En el diseño de los conjuntos se ha procurado que se produzca un solapamiento entre los mismos, de tal manera que la intersección entre dos conjuntos se produzca a mitad del tramo inclinado del trapecio. De esta forma, se asegura una transición suave para los inputs que toman valores en los intervalos entre zonas de pertenencia total a un conjunto.

En lo que respecta a los outputs, la actuación sobre el acelerador y el freno se ha reducido a una única variable, que tiene como dominio el intervalo $[-1, 1]$. Los valores extremos corresponden respectivamente a la posición de freno pisado a fondo (-1), y a la misma posición del acelerador (1). Los valores que posteriormente se hacen llegar a ambos actuadores deben estar contenidos entre el 0 y el 1, que se corresponden a ningún desplazamiento o al máximo de los respectivos dispositivos. Mediante este procedimiento se evita que ambos pedales puedan ser accionados simultáneamente. Otra ventaja derivada del empleo de una única variable es que, una vez diseñado y ajustado el controlador, a la hora de almacenarlo en matrices, el espacio de memoria necesario será menor.

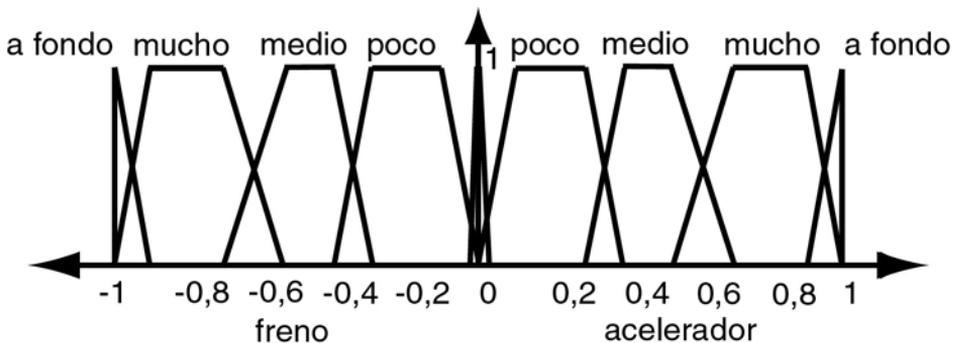


Figura 6.4 Conjuntos borrosos para la actuación sobre el freno y el acelerador.

Para los conjuntos borrosos que describen los desplazamientos de los pedales también se han empleado conjuntos trapezoidales. En esta ocasión, como se puede observar en la Figura 6.4, también se han empleado conjuntos triangulares para representar las situaciones de pedal pisado a fondo o no pisado.

En total, se han establecido 21 reglas para relacionar la variable de salida con las de entrada. Se ha dado más peso a posición y aceleración que a velocidad, especialmente en los intervalos de arranque (en esa posición la velocidad siempre será baja y habrá que acelerar), parada (es un intervalo muy pequeño en el que, independientemente de la velocidad del vehículo, es necesario frenar para detener totalmente el coche) y rebase. Este último conjunto se ha tenido en cuenta para evitar que, en el caso de que el vehículo sobrepasase la distancia objetivo sin detenerse completamente, se inhibiese la actuación sobre los pedales y, por tanto, no se lograse detener completamente el coche. Por este motivo, en el caso de alcanzar esas posiciones, la actuación consistirá en frenar a fondo, con independencia de la velocidad.

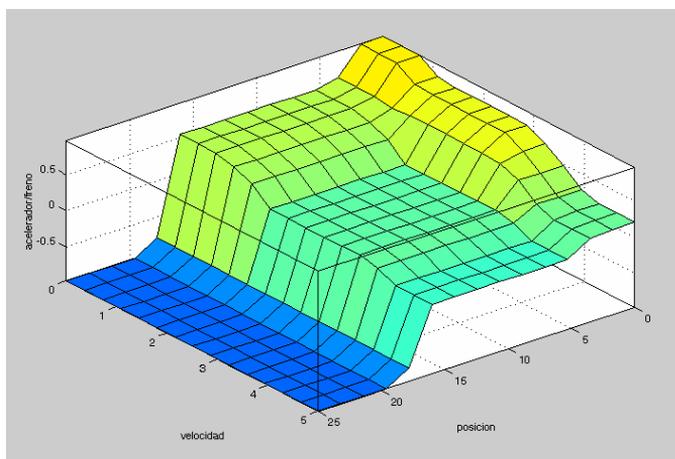


Figura 6.5 Superficie de control generada por las reglas de actuación sobre acelerador y freno.

Los valores de los límites de los conjuntos trapezoidales se han determinado tras un proceso iterativo de ajuste. Este proceso es manual y consiste en ajustar los valores y observar el comportamiento del vehículo realizando una nueva simulación. Como se puede observar en las distintas figuras, algunos conjuntos (actuación nula, aceleración nula y parada) son triangulares, y apenas se solapan con los otros conjuntos. Esto es debido a que recogen situaciones muy singulares y que, generalmente, tienen que

ser logradas completamente. Una actuación nula sobre un pedal quiere decir que no debe existir un desplazamiento del mismo. La parada del vehículo debe producirse en un entorno del objetivo lo más reducido posible (al mismo tiempo, hay que tener en cuenta la inercia del vehículo y, por este motivo, el conjunto no se puede reducir a la posición objetivo).

En la Figura 6.5 se muestra la superficie que relaciona dos de los inputs (posición y velocidad), y el output (actuación sobre freno y acelerador). Como se puede observar, el carácter no lineal aparece fuertemente marcado.

6.2.1. Realización de la Maniobra.

Para evaluar el comportamiento del controlador a la hora de llevar a cabo la maniobra, se ha comparado su actuación con la desarrollada por un conductor real empleando el simulador. La Figura 6.6 muestra la comparación entre la trayectoria del vehículo cuando es conducido por el conductor real y cuando lo es por el controlador, así como los gráficos correspondientes a la actuación sobre el freno y el acelerador a lo largo del tiempo desarrollada por ambos conductores.

Como se puede observar, la ejecución de la maniobra por parte del controlador se podría decir que es más “limpia”. El controlador no necesita realizar sucesivas aproximaciones. El conductor humano realiza dos pequeñas paradas para tomar referencia de la posición frente al objetivo y adecuar la velocidad, antes de alcanzar definitivamente la meta. Asimismo, la actuación sobre los pedales de freno y acelerador es más eficiente cuando el vehículo es controlado automáticamente, y la maniobra se realiza en menos tiempo.

La Tabla 6.1 recoge los valores numéricos de las magnitudes características de la maniobra. En ella se verifica que el controlador emplea 4,5 s menos de tiempo, a pesar de que tanto las velocidades como las aceleraciones se mantienen en valores inferiores a los desarrollados por el conductor real. Esto se debe a que la actuación del controlador sobre los pedales es más constante, evitando la brusquedad. De todas formas, el valor máximo de la frenada supera el valor de 3 m/s^2 . Esto es debido a que, como se recordará, cuando una variable excede el máximo de su universo de discurso se toma el valor máximo como valor de la variable.

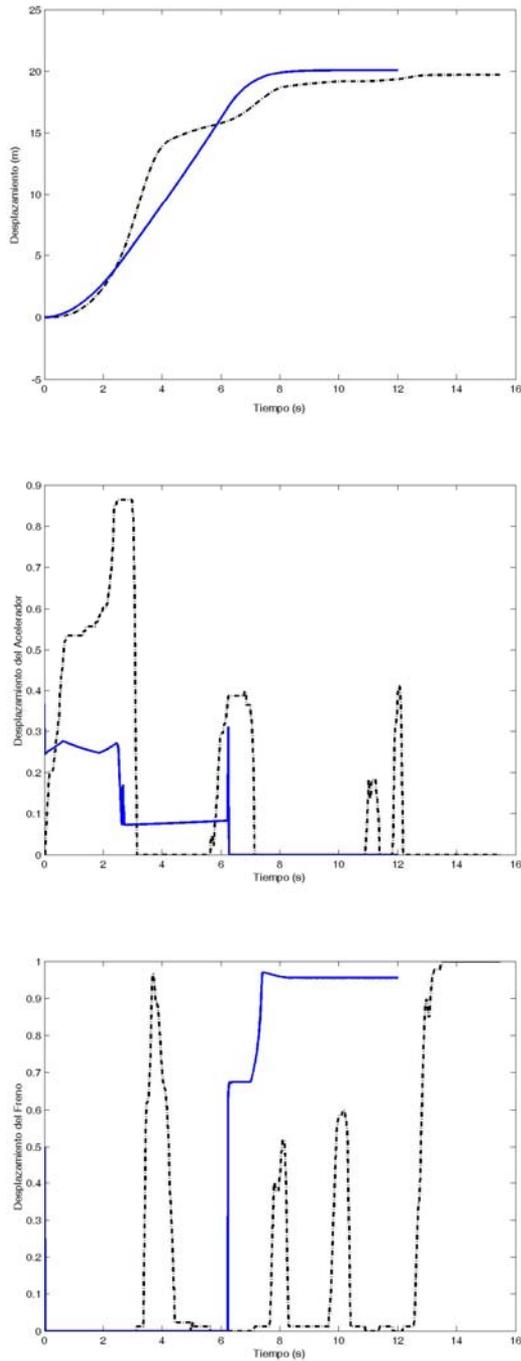


Figura 6.6 Controlador frente a conductor real (línea discontinua): a) trayectoria; b) actuación sobre el acelerador; c) actuación sobre el freno.

Por otro lado, el error (distancia final a la posición objetivo) obtenido por el controlador es de un orden de magnitud inferior al cometido por el conductor. El valor negativo del error cometido por éste último significa que el vehículo se detiene antes de alcanzar la posición objetivo.

	Tiempo maniobra (s)	Error (m)	Velocidad Máxima (m/s)	Aceleración máxima (m/s ²)	Frenada máxima (m/s ²)
Conductor real	13.1	-0.314	8.11	5.8	-9.6
Controlador	8.6	+0.081	3.7	1.7	-4.1

Tabla 6.1 Valores característicos obtenidos en la realización de la primera maniobra.

6.3. Segunda Maniobra: Elusión de un Obstáculo.

El objetivo de la segunda maniobra es representar el efecto de evitar un obstáculo que pueda presentarse ante el vehículo. En este caso, se requiere tanto el control de los pedales como el del volante. El vehículo parte del reposo, recorre un tramo recto de 20 m, sigue un período completo (de pico a pico) de una senoide de amplitud 1,75 m y, finalmente, retorna a un tramo recto. Nuevamente, la maniobra ha sido desarrollada previamente por un conductor real para comprobar su viabilidad y determinar los valores característicos de la maniobra. De esta forma, se ha establecido una región de velocidades permisibles, que permite diseñar los conjuntos borrosos dentro de esa región. Como resultado, la velocidad se ha mantenido por debajo de los 8 m/s, y la aceleración longitudinal se ha restringido al intervalo de $\pm 3 \text{ m/s}^2$, de forma que se eviten maniobras bruscas.

6.3.1. Descripción de la Trayectoria a Seguir: Curvas de Bézier.

La trayectoria que debe seguir el vehículo se representa mediante una secuencia de segmentos de curva conectados. De esta manera, es posible

representar cualquier trayectoria con la precisión que sea necesaria, únicamente añadiendo más segmentos de curva. Estas curvas son curvas de Bézier.

Mediante curvas paramétricas se puede representar cualquier tipo de trayectoria, en el plano o en el espacio. En el simulador, dada la naturaleza del control del vehículo, se ha optado por usar trayectorias bidimensionales. Esto no significa que el vehículo no pueda seguir trayectorias en el espacio (subidas, bajadas, etc) sino que la trayectoria está almacenada mediante su proyección en el plano. El uso de curvas de Bézier 3D únicamente implicaría usar puntos 3D en lugar de 2D, sin ningún cambio adicional en el algoritmo; no obstante, sólo se ha empleado la segunda opción.

Las curvas de Bézier son curvas paramétricas caracterizadas por cuatro puntos P_1 , P_2 , P_3 y P_4 , llamados puntos de control (Foley et al. 1997). Dichos puntos determinan una zona llamada el “cascarón convexo” (*convex hull*). Está garantizado que una curva de Bézier no tiene nunca puntos contenidos fuera de este cascarón. Esto es de utilidad a la hora de hacer comprobaciones rápidas, es decir, sencillas y de bajo coste computacional, de la proximidad de un punto cualquiera a la curva Bézier.

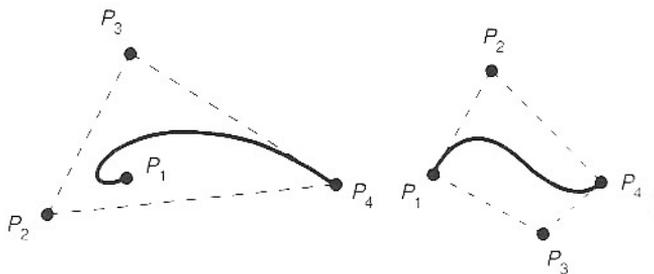


Figura 6.7 Curvas de Bézier y puntos de control.

Los puntos P_1 y P_4 determinan los extremos de la curva, por lo que esta pasa necesariamente por ellos. Mediante los puntos P_2 y P_3 se determinan las tangentes a la curva en sus extremos. Para ello, el vector formado por un punto extremo y su contiguo, p. ej. P_1 y P_2 , tiene la dirección de la tangente a la curva en ese punto extremo.

La expresión paramétrica de la curva es función de los puntos de control antes mencionados y tiene el siguiente aspecto:

$$Q(t) = (1-t)^3 P_1 + 3t(1-t)^2 P_2 + 3t^2(1-t) P_3 + t^3 P_4 \quad (6.1)$$

En la expresión anterior, los coeficientes de cada uno de los puntos corresponden a los llamados *polinomios de Bernstein*. Dichos polinomios tienen como particularidad que su suma, para cualquier valor del parámetro t , es igual a la unidad, lo que los hace ideales para la ponderación de cada uno de los puntos de control conforme se recorre la curva. Los valores $t=0$ y $t=1$ representan los puntos extremos de la curva, esto es, P_1 y P_4 respectivamente, aunque las ecuaciones siguen siendo válidas para cualquier t externo a ese rango.

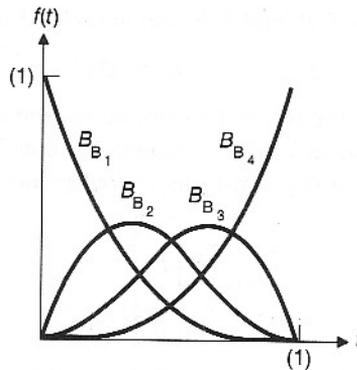


Figura 6.8 Polinomios de Bernstein.

Las curvas presentan continuidad C_0 , cuando el punto P_4 de un segmento coincide con el P_1 del segmento siguiente. Además, gracias a la posibilidad de imponer directamente el valor de las tangentes en los extremos de cada segmento, es posible alcanzar el grado de continuidad C_1 entre segmentos cuando la tangente definida por los puntos P_4 - P_3 de la primera curva es idéntica a la definida por los puntos P_2 - P_1 de la siguiente.

La forma de la expresión paramétrica de la curva (6.1) es polinomial, lo que garantiza la continuidad y derivabilidad de la misma para cualquier valor del parámetro t . Esto facilita en gran medida la obtención de ciertas características de la curva, necesarias para el algoritmo del controlador. Como ejemplo de dichas características está la obtención de la tangente para cualquier punto de la curva. Mediante la derivación con respecto al parámetro t de la expresión polinomial antes mostrada, se obtiene también

la curvatura gracias a la expresión general
$$\kappa = \frac{\dot{x}\ddot{y} - \ddot{x}\dot{y}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}.$$

La parametrización permite desacoplar la descripción de la trayectoria y el código, dado que aquella se carga en tiempo de ejecución desde un fichero

de datos. La trayectoria puede ser, por lo tanto, creada desde un programa externo.

La propia naturaleza de la trayectoria formada por las curvas de Bézier, al estar formada por múltiples segmentos de curva, facilita la tarea de determinar en qué punto se encuentra el vehículo. Es decir, se impone una segmentación implícita de la trayectoria que puede ser muy útil en casos particulares, como en zonas de cruce de la misma, donde la información será suficiente para poder determinar qué ruta se ha de tomar. Esto es posible dada la secuencialidad de los segmentos, que impone el sentido de recorrido de la ruta además de su localización.

La densidad en cuanto a número de segmentos de curva por unidad de área no es fija, con lo que es posible dedicar más segmentos a las zonas más complejas y menos a las más sencillas (rectas prolongadas).

La naturaleza paramétrica de las curvas permite evitar los casos de singularidades que se encontrarían con representaciones cartesianas ($\text{tg}90^\circ$, etc.). Además, permite que un único segmento pueda cortarse consigo mismo (lazo).

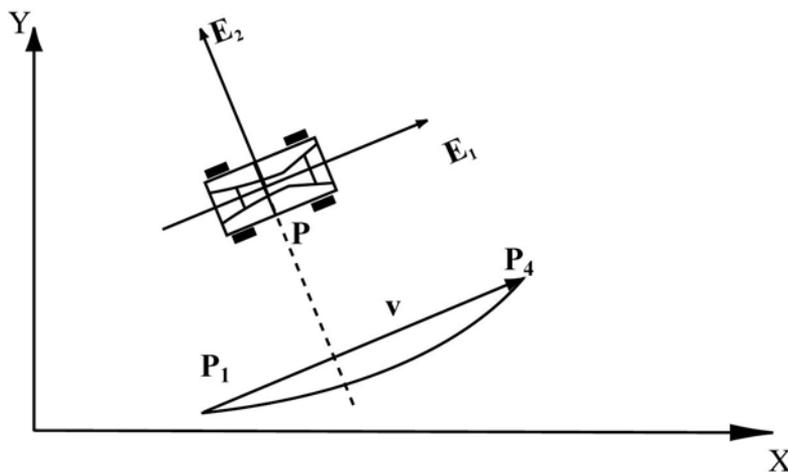


Figura 6.9 Intersección de la normal al vehículo con la curva objetivo.

Una vez descrita la trayectoria a seguir por el vehículo, el procedimiento para detectar la intersección entre la normal al éste y la curva, Figura 6.9, es sencillo. Este procedimiento se necesita para determinar los errores de posición del vehículo y el error angular (discrepancia entre la orientación del vehículo y la tangente a la trayectoria).

El primer paso consiste en detectar con qué curva se produce la intersección. Para ello, se calcula la intersección entre una recta que pasa por el centro del coche (punto \mathbf{P}) y está orientada según el vector normal, \mathbf{n} , y otra que pasa por el punto \mathbf{P}_1 y está orientada según el vector \mathbf{v} , correspondiente a la cuerda de la curva entre \mathbf{P}_1 y \mathbf{P}_4 . Si se expresan estas rectas en paramétricas se trata de resolver el sistema:

$$\begin{aligned} \mathbf{r}_1 &= \mathbf{P} + \lambda \mathbf{n} \\ \mathbf{r}_2 &= \mathbf{P}_1 + \mu \mathbf{v} \end{aligned} \quad (6.2)$$

Como sólo interesa saber si ambas rectas se cortan, basta con despejar el parámetro μ y verificar que su valor está entre 0 y 1. Si no, se pasa a comprobar la intersección con otra curva.

Una vez que se ha determinado con cual de las curvas se produce la intersección, se calcula el punto de corte con la misma. Para ello, la expresión de la curva (6.1) se expresa como un polinomio en función del parámetro t , que adopta la siguiente expresión:

$$\mathbf{Q}(t) = (-\mathbf{P}_1 + 3\mathbf{P}_2 - 3\mathbf{P}_3 + \mathbf{P}_4)t^3 + 3(\mathbf{P}_1 - 2\mathbf{P}_2 + \mathbf{P}_3)t^2 + 3(\mathbf{P}_2 - \mathbf{P}_1)t + \mathbf{P}_1 \quad (6.3)$$

El polinomio se puede escribir como:

$$\mathbf{Q}(t) = \alpha t^3 + \beta t^2 + \gamma t + \delta \quad (6.4)$$

Nuevamente, los valores del parámetro t correspondientes a la parte de la curva situada entre los puntos de control, son aquellos comprendidos entre 0 y 1. De hecho, se verifica que $\mathbf{Q}(0) = \mathbf{P}_1$ y $\mathbf{Q}(1) = \mathbf{P}_4$.

A continuación, se plantea la intersección de la curva definida por el polinomio (6.4) con la recta \mathbf{r}_1 . Esto implica igualar las distintas componentes. Al igualar la componente x , resulta la siguiente ecuación:

$$\alpha_x t^3 + \beta_x t^2 + \gamma_x t + \delta_x = P_x + \lambda n_x \quad (6.5)$$

De ella se puede despejar el parámetro λ en función de t como:

$$\lambda = \frac{1}{n_x} (\alpha_x t^3 + \beta_x t^2 + \gamma_x t + \delta_x - P_x) \quad (6.6)$$

Este valor se introduce en la ecuación correspondiente a igualar las componentes en y , con lo que resulta la siguiente ecuación cúbica en t :

$$\begin{aligned} & (n_x \alpha_y - n_y \alpha_x) t^3 + (n_x \beta_y - n_y \beta_x) t^2 + (n_x \gamma_y - n_y \gamma_x) t \\ & + n_x (\delta_y - P_y) - n_y (\delta_x - P_x) = 0 \end{aligned} \quad (6.7)$$

Para la obtención de las raíces del polinomio se sigue el método propuesto en Press et al. (1992). Finalmente, una vez obtenidas las tres raíces, nos quedaremos con aquella raíz real que tome valores entre 0 y 1.

6.3.2. Diseño de los Controladores.

En este caso, se han diseñado controladores tanto para el volante como para los pedales de acelerador y freno. Estos controladores se han diseñado de forma independiente. Sin embargo, como se verá más adelante, se ha procurado que, cuando el vehículo no sigue la trayectoria deseada, el error cometido en la dirección transversal se traduzca en una reducción de la velocidad del vehículo como ocurriría con un conductor real. Para ello, el control de los pedales sigue la misma estrategia que se ha explicado en el primer ejemplo, con la excepción de que la posición, que se empleaba como primero de los inputs, ha sido sustituida por el error en el seguimiento de la trayectoria. Este error se calcula como la media de los dos errores empleados para el control del volante, que se describirán más adelante, y toma valores entre 0 y 1.

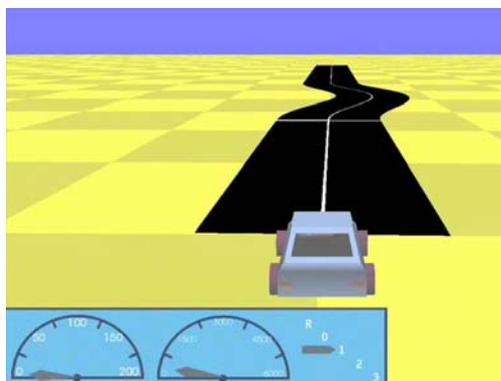


Figura 6.10 Segunda maniobra: elusión de un obstáculo.

La Figura 6.11 ilustra el esquema de control del volante. Se han definido dos tipos de errores para la maniobra. El primero es el error de posición del vehículo en el instante considerado, ε_{pos} . La definición de este error se muestra en la Figura 6.12, y es calculado como la distancia del coche a la trayectoria deseada, medida sobre el eje transversal del vehículo. Este error toma valores entre -5 y 5 m. Los errores superiores a 5 m se han

reducido a este valor. Valores del error superiores a 1 m han sido muy penalizados en el diseño del controlador.

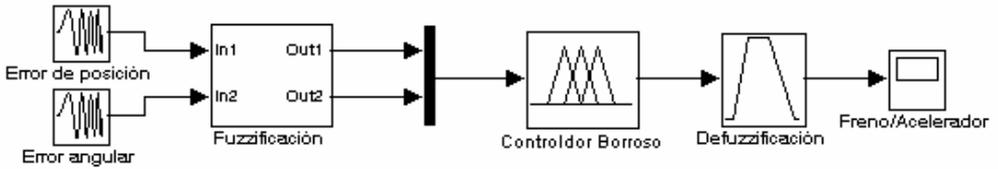


Figura 6.11 Diagrama de control para la actuación sobre el volante.

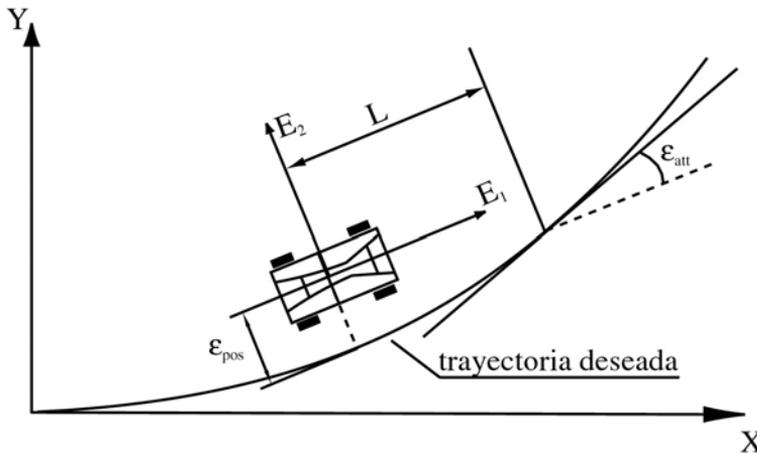


Figura 6.12 Funciones de error para el control de la dirección.

El segundo error empleado pretende tener en cuenta una parte fundamental del comportamiento del conductor, que consiste en tratar de anticiparse a los obstáculos que se encuentran más adelante (Sharp et al. 2000). Para ello, se ha empleado el error angular, ϵ_{att} , que considera la orientación del vehículo en los instantes siguientes. Para definir este error se compara la tangente a la trayectoria real del vehículo en el instante actual, con la tangente a la trayectoria a seguir en un punto más avanzado. El conductor “observa” el camino a seguir proyectando hacia delante una distancia L . Esta distancia se corresponde con el hecho de que el conductor es capaz de anticiparse una cantidad de tiempo, T_p , a la maniobra. Entonces, empleando este procedimiento, la distancia resulta ser una función de la velocidad longitudinal del vehículo:

$$L = v_x \cdot T_p \tag{6.8}$$

Experimentalmente, se ha hallado un valor para T_p de 0,425 s, que resulta apropiado para la maniobra. Se ha considerado para la desviación un valor máximo de 50° , y se asigna este valor de 50° a los valores de la variable que lo sobrepasen.

Por supuesto, es posible tener en cuenta otras definiciones para los errores. Por ejemplo, podría ser conveniente considerar más de un punto de anticipación para el error angular (Sharp et al. 2000). Sin embargo, el presente trabajo se ha centrado en el desarrollo de una herramienta para el diseño de controladores más que en la obtención de un controlador óptimo para un caso determinado.

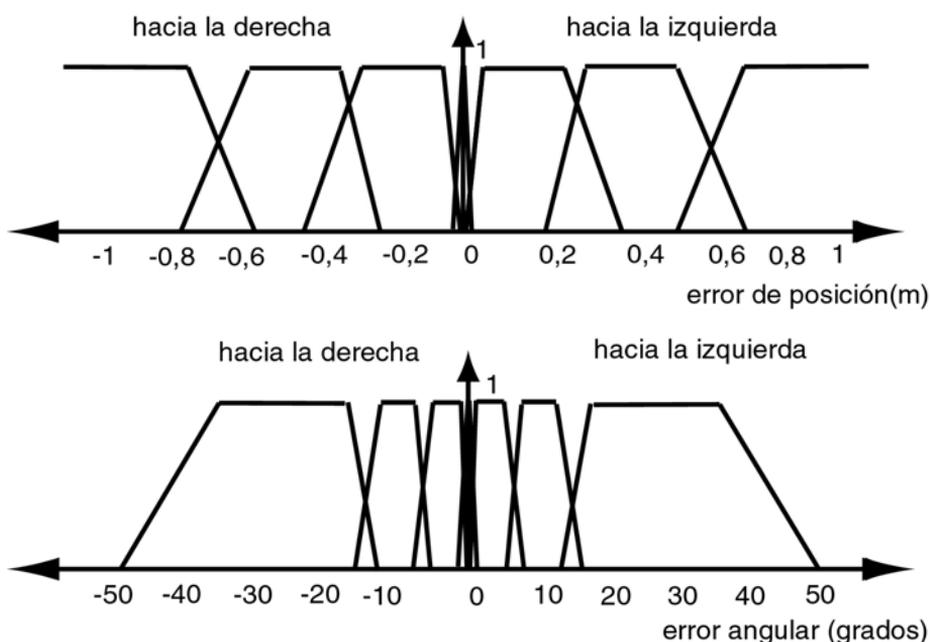


Figura 6.13 Conjuntos borrosos para los inputs del controlador.

Los conjuntos borrosos que describen los errores se han representado en la Figura 6.13. Para el error angular, los conjuntos son más estrechos en el entorno del error nulo con el fin de que una ligera desviación de la trayectoria deseada pueda ser rápidamente corregida. En el caso del error de posición, la figura representa únicamente la parte central del universo de discurso $[-5, 5]$. Nuevamente, los conjuntos no son tan anchos en las proximidades del error nulo con el fin de reaccionar rápidamente en el caso de una ligera desviación.

Los conjuntos de salida se representan en la Figura 6.14. La variable de salida representa la posición absoluta del volante, con unas posiciones

extremas de 450° tanto hacia la derecha como hacia la izquierda. Se ha permitido un giro máximo de 15° entre instantes de integración consecutivos. Este valor se ha escogido teniendo presente una posible implementación futura del controlador en el prototipo. En ese contexto, si el controlador se maneja mediante un motor paso a paso, con un paso del motor de $1,8^\circ$ y un paso de integración de 10 ms, ese desplazamiento correspondería a una frecuencia de excitación de 833 Hz.

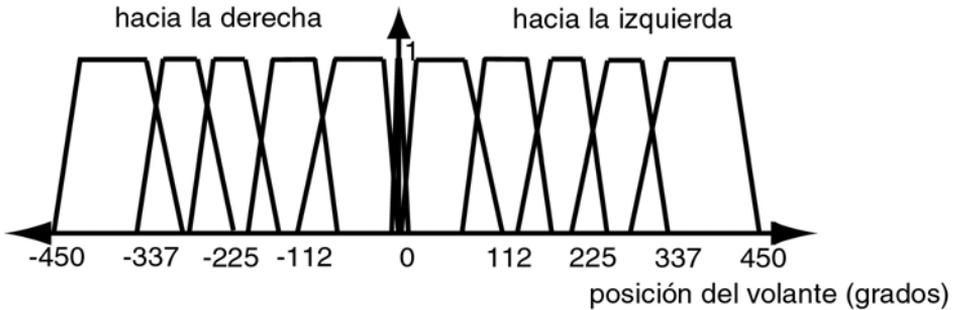


Figura 6.14 Conjuntos de salida para la posición del volante.

Se han establecido 49 reglas para el control del volante. La superficie resultante se puede observar en la Figura 6.15. Dicha superficie proporciona la posición angular absoluta del volante para cualquier par de valores de los errores.

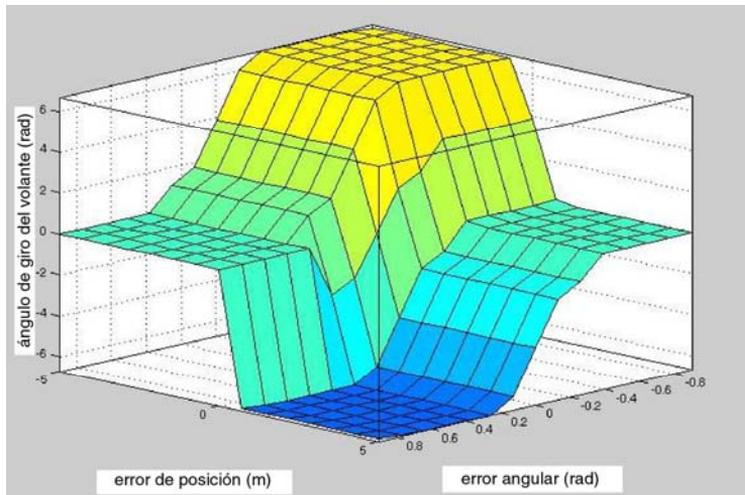


Figura 6.15 Superficie de control generada por las reglas de control del volante.

Volviendo al control de los pedales en esta segunda maniobra, los conjuntos borrosos para la velocidad y la aceleración son similares a los

descritos para la primera maniobra. Únicamente se ha variado el rango de la velocidad, que se ha elevado de 5 m/s a 12 m/s, y los conjuntos se han ampliado proporcionalmente. Sin embargo, como se ha indicado al principio de esta sección, el error de posición que se contemplaba en la primera maniobra se ha sustituido por el error en el seguimiento de la trayectoria. Este error se ha calculado como la media del error de posición y el error angular empleados para el control del volante. Los conjuntos borrosos definidos para este nuevo error se muestran en la Figura 6.16.

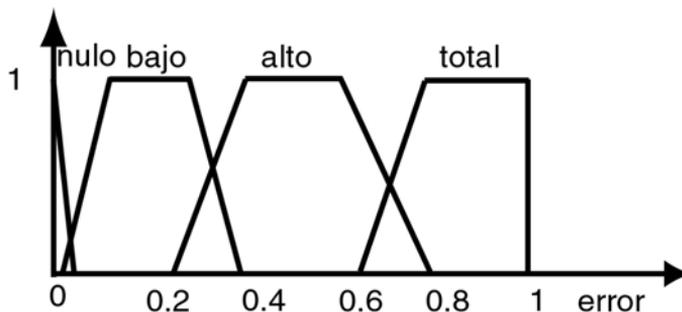


Figura 6.16 Conjuntos borrosos para el error en el seguimiento de la trayectoria.

6.3.3. Realización de la Maniobra.

Al igual que para la primera maniobra, la ejecución desarrollada por el controlador obtenido se ha evaluado comparándola con la de un conductor real. La Figura 6.17 muestra la comparación de las trayectorias seguidas, y la velocidad y el desplazamiento del volante en función del tiempo.

Se puede observar que el conductor real tiende a suavizar la trayectoria en las curvas. Debido a la anticipación que se ha introducido en el diseño del controlador, esta tendencia es menos acusada en el comportamiento del vehículo guiado por el controlador. De hecho, el controlador se podría ajustar para que siga mejor la trayectoria deseada, reduciendo el máximo error angular permitido, y considerando el error angular sin anticipación. Es decir, considerando los errores de posición y angular para la posición del vehículo en ese instante de tiempo. Sin embargo, un controlador ajustado de esa manera es demasiado sensible a pequeñas excitaciones y, consecuentemente, bastante inestable. Por este motivo, es necesario buscar un compromiso entre ambos factores (de hecho, se puede observar una cierta inestabilidad de la actuación del volante al final de la maniobra).

Llama la atención que el conductor real no “anticipe” el último giro. Esto se debe, probablemente, a una velocidad excesiva.

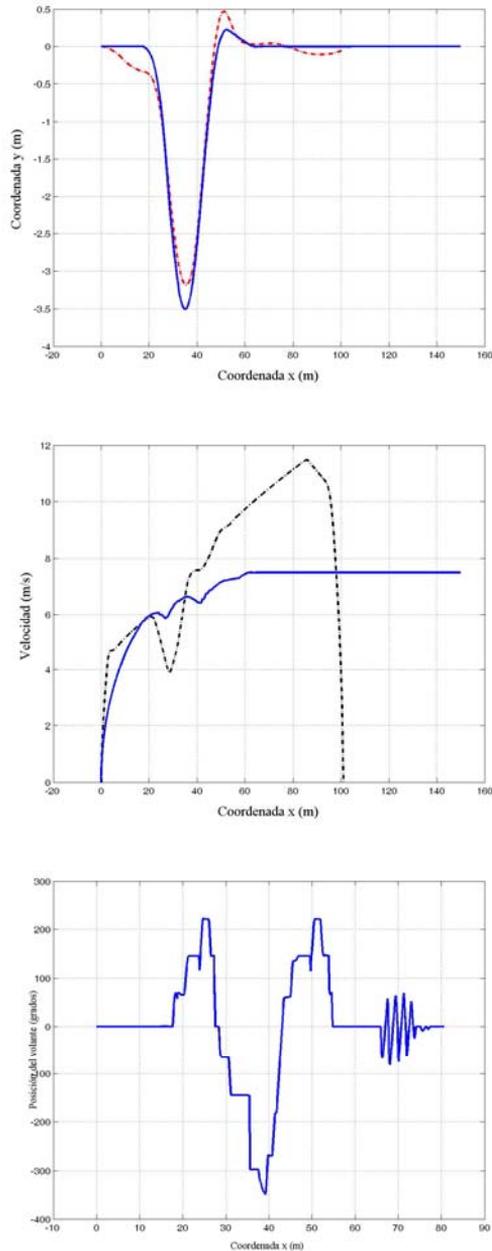


Figura 6.17 Controlador frente a conductor humano: a) trayectoria; b) velocidad; c) posición del volante.

Conviene notar que, durante el tramo curvo, el error cometido por el controlador es prácticamente nulo, y la reducción de velocidad es menos

acusada que cuando la maniobra la realiza el conductor humano. El máximo error cometido por el controlador y el conductor humano tiene lugar en el momento de retornar al tramo recto final. Si el controlador se diseña imponiéndole velocidades más bajas, el error cometido se puede reducir prácticamente a cero. El controlador diseñado realiza la maniobra a una velocidad casi crítica desde el punto de vista de la estabilidad del vehículo.

	Tiempo Maniobra (s)	Máximo Error Posición (m)	Velocidad Máxima (m/s)	Error Posición Medio (m)
Conductor Real	14.05	0.455	7.55	0.0047
Controlador	14.91	0.205	6.55	0.000833

Tabla 6.2 Valores característicos obtenidos en la realización de la segunda maniobra.

La Tabla 6.2 proporciona los valores numéricos de las magnitudes características de la maniobra. El valor máximo de la velocidad corresponde solamente al tramo curvo de la maniobra. Como se puede observar, el conductor real guía el vehículo a una velocidad más elevada. Sin embargo, al cotejarla con las gráficas, se observa que esta velocidad se produce en el instante anterior a tomar la curva, e induce al vehículo a cometer un mayor error al trazar el último giro. El controlador es ligeramente más lento, pero comete un error menor en el seguimiento de la trayectoria. Además, a la luz del comportamiento del vehículo con ambos conductores, se puede deducir que la velocidad a la que se realiza la maniobra está próxima al máximo valor con que el vehículo puede seguir la trayectoria sin salirse de la carretera.

La media del error se ha definido como la media algebraica del error de posición a lo largo de la maniobra,

$$\bar{\varepsilon}_{pos} = \frac{\sum_{i=1}^n |\varepsilon_{pos}(i)|}{n} \quad (6.9)$$

donde n es el número de pasos de tiempo de la integración numérica requeridos para la realización completa de la maniobra. Este valor puede

considerarse como una evaluación de la precisión en el seguimiento de la trayectoria. Como se puede observar en la Tabla 6.2, el error cometido por el controlador es unas cinco veces más pequeño. Junto con esto, conviene tener en cuenta que, como se puede observar en la Figura 6.17, el error cometido por el conductor real es mayor en los tramos en los que la trayectoria cambia de recta a curva o viceversa. En esos casos, el vehículo controlado se ajusta mucho mejor al trazado. El error máximo que comete es la mitad que el del conductor real, y se produce cuando retorna al tramo recto del final de la maniobra, mientras que, en el resto de la misma, el error es muy inferior al máximo.

7.

Evaluación de Eficiencia y Robustez

Una vez comprobada la validez de la herramienta para el desarrollo de controladores, se trata de evaluar de alguna manera la eficiencia y la robustez del simulador desarrollado. Una simulación se considera estable cuando proporciona una solución a lo largo del tiempo, independientemente de que esta sea correcta o no. El programa es además robusto cuando sigue siendo estable ante problemas *stiff*, aceleraciones elevadas, grandes fuerzas que aparezcan súbitamente, simulaciones prolongadas, etc. Se entiende por eficiencia la velocidad de cálculo del programa.

En el caso de un simulador de conducción, el programa será suficientemente eficiente si es capaz de mantener la simulación en tiempo real en las condiciones requeridas por la simulación. Las exigencias, en el caso de la robustez, son también elevadas, puesto que un vehículo real es capaz de realizar maniobras violentas, soportar colisiones con otros vehículos o con el entorno, etc.

Para evaluar las capacidades del simulador, se han desarrollado dos entornos en los que verificar las características de eficiencia y robustez. El primero consiste en un entorno con obstáculos para comprobar la reacción del vehículo ante maniobras bruscas. El segundo es un entorno en el que

se ha cuidado más el realismo de la representación. Consiste en una reproducción a escala de un circuito de carreras en el que, además, circulan otros vehículos. De esta manera, es posible comprobar la dificultad de las maniobras que realmente se pueden abordar con el simulador, y tener una idea de hasta qué punto es éste capaz de mantener el tiempo real de la simulación cuando resuelve la dinámica de varios vehículos y representa un entorno con un cierto nivel de detalle.

7.1. Entorno con Obstáculos.

Como se ha dicho, la finalidad del entorno con obstáculos es observar la capacidad del simulador para abordar maniobras bruscas o violentas. Por este motivo, el entorno que se ha desarrollado consta de los siguientes elementos:

- Rampas. En el entorno se dispone de varias rampas. Una de ellas presenta las dimensiones que se muestran en la Figura 7.1. En ella se comprueba el comportamiento del vehículo al saltar al suelo desde una altura de 1 m. Otras rampas se han empleado para verificar el comportamiento lateral, tanto dinámico como estático.

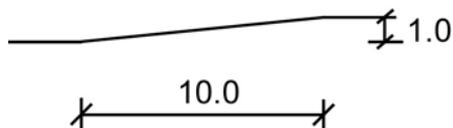


Figura 7.1 Dimensiones de la rampa.

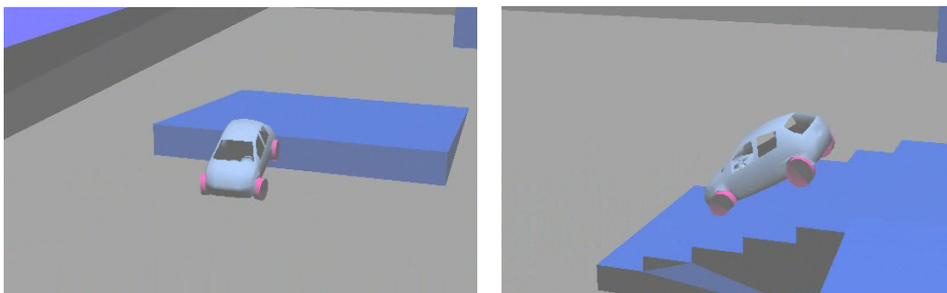


Figura 7.2 El vehículo realizando algunas maniobras.

Al realizar una maniobra consistente simplemente en saltar con una velocidad de 11 m/s (la mínima necesaria para subir la rampa y saltar sin que haya contacto entre el chasis y el borde de la rampa), se obtiene una fuerza máxima de 15.675 N en cada uno de los

amortiguadores delanteros y 10.880 N en los traseros, como se puede observar en la Figura 7.3. En la figura también se puede observar el cabeceo que produce el choque del tren trasero con el suelo, y sus posteriores rebotes.

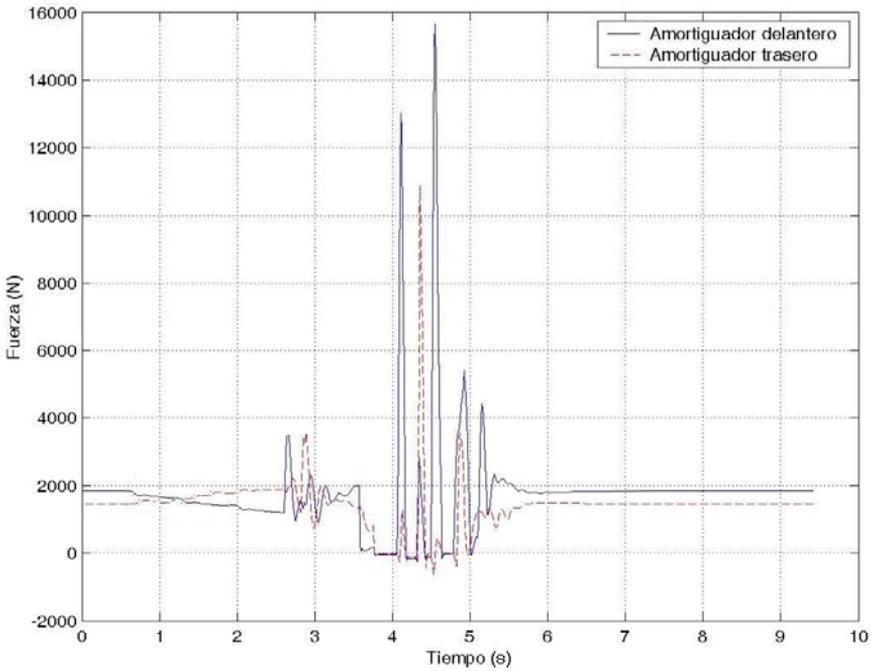


Figura 7.3 Fuerzas obtenidas en los amortiguadores en la maniobra de la rampa.

- Escaleras. Una maniobra todavía más brusca y difícil de integrar es el descenso de escaleras. En el entorno se han introducido unas escaleras con las dimensiones indicadas en la Figura 7.4. Como se puede observar, constan de cinco escalones de 1 m de avance y 0,3 m de descenso, que culminan en un escalón de 0,5 m. de altura.

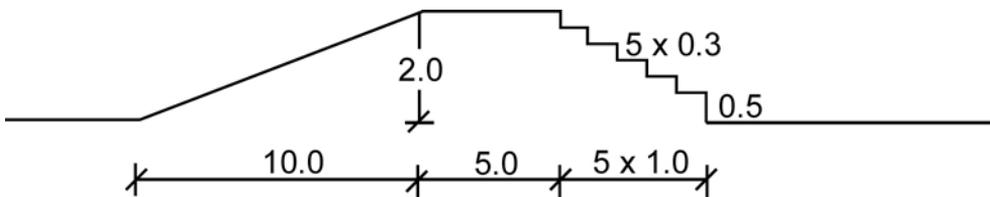


Figura 7.4 Dimensiones de la rampa y escaleras.

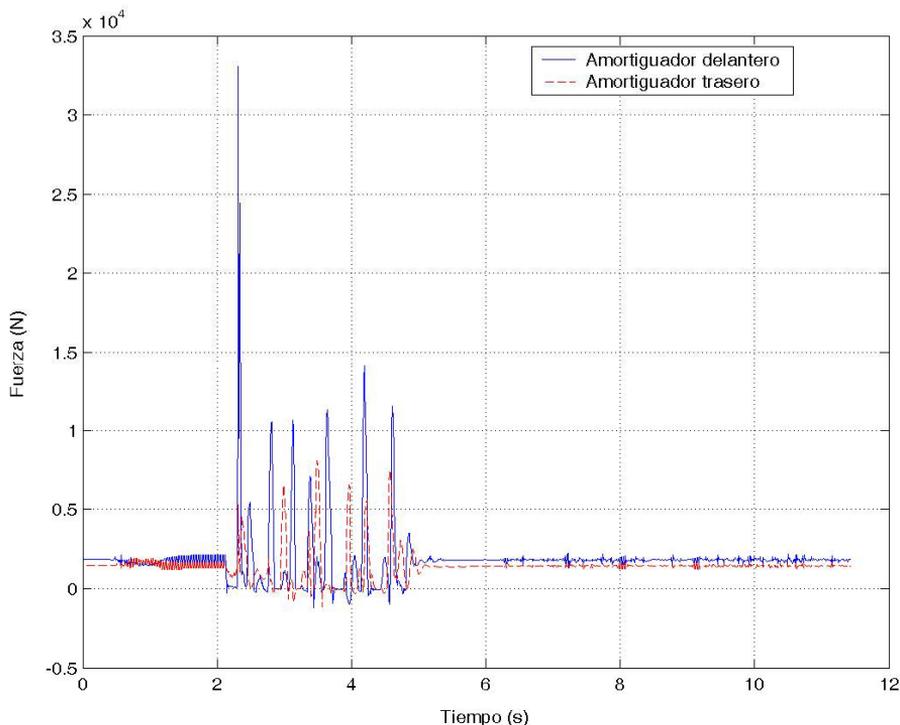


Figura 7.5 Fuerzas obtenidas en los amortiguadores al descender por las escaleras.

El descenso es muy violento. El coche se lanza con una velocidad de 1 m/s, es decir, la velocidad mínima para que caiga por las escaleras. La Figura 7.5 muestra las fuerzas que se obtienen en uno de los amortiguadores delanteros y en uno de los traseros. Como se puede observar, en el amortiguador delantero se produce un pico de fuerza de 33.075 N. Los posteriores rebotes toman unos valores entre un máximo de 14.070 N y un mínimo de 10.500 N para el amortiguador delantero, y un máximo de 8.080 N y un mínimo de 4.580 N para el trasero.

- Cambios de adherencia en la superficie. En el circuito es posible situar zonas con distinta adherencia. Por el momento sólo se ha representado un pequeño rectángulo en el que el coeficiente de adherencia desciende de 0,9 a 0,2.
- Hilera de obstáculos. Se han distribuido 4 columnas separadas una distancia de 13 m para verificar la maniobrabilidad del vehículo.

- Túnel. Desde el punto de vista de la eficiencia o la robustez, la existencia de un túnel no aporta nada, pero sirve para verificar que el algoritmo desarrollado para la interacción con el entorno permite manejar terrenos en los que, para unas coordenadas x , y y determinadas, sea posible la existencia de varias alturas.
- Pista quebrada. Se ha simulado la presencia de una zona de baches mediante una pista con pequeñas rampas de subida y bajada consecutivas, que provoca una excitación en la suspensión del vehículo.

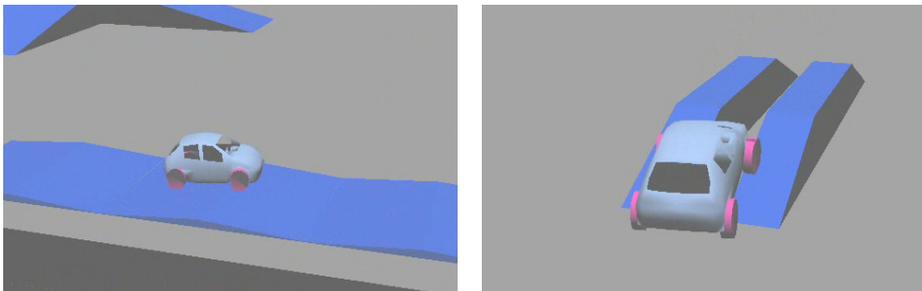


Figura 7.6 Pistas quebrada y elevada.

- Para comprobar la precisión en la determinación del contacto entre el neumático y el suelo, se ha incluido la pista elevada con zanja central que se muestra en la Figura 7.6 derecha.

7.2. Simulación de Varios Vehículos.

Para evaluar la eficiencia se ha desarrollado un entorno más realista, con un mayor detalle en los gráficos. Asimismo, se realiza la simulación simultánea de varios vehículos. Otra utilidad de este ejemplo es la evaluación de un controlador de conducción. Para la elaboración del entorno se ha tomado como modelo el Circuito de Cataluña.

El proceso de diseño de este nuevo escenario se ha dividido en dos partes principales. Por un lado se ha abordado el diseño del trazado de la pista y su representación visual y, por otro, se han añadido diversos objetos y texturas que mejoran sensiblemente el grado de inmersión del conductor en la simulación. Además, se ha creado una nueva ventana gráfica donde el conductor puede comprobar con rapidez la posición y velocidad del vehículo.

7.2.1. El Circuito.

Para la representación gráfica, se ha elegido como modelo un circuito de competición real, el Circuito de Cataluña, por las características de su trazado y por la facilidad de encontrar información sobre el mismo. La característica más importante del Circuito de Cataluña es su trazado irregular, con tramos rectos de distintas longitudes alternándose con diversos tipos de curvas, lo que exige al conductor un buen dominio del vehículo.

Los datos sobre la geometría del circuito se han obtenido de la información suministrada en varias páginas web. La página web oficial del Circuito de Cataluña (<http://www.circuitcat.com>), proporciona abundante información sobre el circuito. Para realizar mediciones de algunas partes del circuito se ha empleado también la aplicación informática SIGPAC del Ministerio de Agricultura, Pesca y Alimentación (<http://sigpac.mapa.es/fega/visor/>).



Figura 7.7 Plano del Circuito de Cataluña.

Para la representación del circuito se ha determinado la posición de los puntos que marcan el paso de los tramos rectos a los tramos curvos. Las curvas del circuito real se han aproximado mediante arcos de circunferencia. En aquellas curvas con un radio claramente variable se han empleado varios arcos de circunferencia en su aproximación.

La Figura 7.8 muestra los extremos de los tramos diseñados, siendo el punto 1 el que se corresponde con la posición de salida de los vehículos. La Tabla 7.1 muestra las coordenadas de los puntos anteriores referidos al

sistema de referencia indicado en la Figura 7.8, con origen en el punto de salida, el punto 1, y con el eje x coincidiendo con el sentido de la marcha en la dirección de la recta principal del circuito (recta de tribunas).

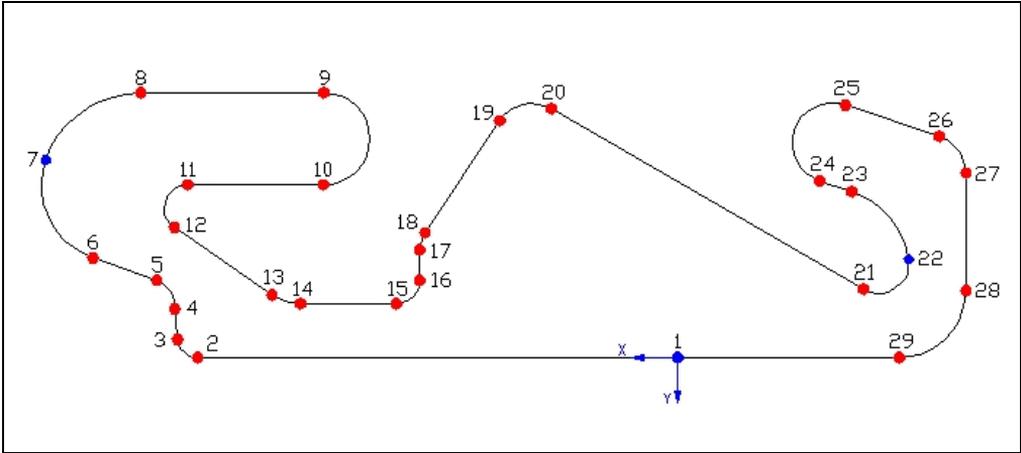


Figura 7.8 Puntos del circuito.

Punto	x (m)	y (m)	Punto	x (m)	y (m)
1	0	0	16	385,40	-115,50
2	717,89	0,00	17	385,40	-161,92
3	747,78	-27,39	18	377,84	-186,89
4	752,53	-81,74	19	265,69	-355,06
5	779,07	-115,97	20	188,33	-373,04
6	874,38	-149,64	21	-278,49	-102,11
7	944,87	-296,23	22	-345,53	-147,12
8	803,22	-396,90	23	-260,97	-248,62
9	529,30	-396,90	24	-212,90	-264,60
10	529,30	-258,90	25	-250,81	-378,56
11	732,99	-258,90	26	-391,84	-331,55
12	752,92	-195,13	27	-431,50	-276,53
13	606,66	-93,84	28	-431,50	-100,00
14	563,96	-80,50	29	-331,50	0,00
15	420,40	-80,50			

Tabla 7.1 Coordenadas de los puntos del circuito.

En los programas de dibujo asistido por ordenador existen dos planteamientos para la representación de curvas. Uno es desarrollar un algoritmo de generación de curvas, con el cual se crea una curva real. El segundo método consisten en aproximar la curva mediante un determinado

número de pequeños segmentos rectos, y tiene un coste computacional menor que el primer planteamiento. Por ello, se ha seleccionado este método para realizar la representación del circuito. De esta manera, cuantos más puntos se usen en la representación de la curva, mayor precisión se obtendrá en la representación de la misma. En este sentido, para que todas las curvas del circuito muestren un aspecto similar, se ha hecho que el número de puntos que definan cada curva sea proporcional a la longitud de su arco.

Se han desarrollado las subrutinas necesarias para la representación gráfica del circuito empleando las primitivas geométricas de OpenGL. Se han elegido variables el ancho de la carretera y la escala de representación del circuito. Se ha incluido esta variable porque el tamaño original del circuito es excesivamente grande para el vehículo. El circuito real está pensado para la celebración de actividades deportivas en las que participan vehículos con unas características muy diferentes a las del vehículo simulado. Por ejemplo, en las competiciones de Fórmula 1, los vehículos pueden alcanzar una velocidad de 310 km/h en algunos tramos del Circuito de Cataluña. Sin embargo, la velocidad máxima del vehículo utilizado en esta tesis supera ligeramente los 100 km/h en las simulaciones realizadas. De esta manera, el tiempo necesario para que el vehículo simulado completara una vuelta del circuito a escala real sería excesivo, y mucho más si se compara con el empleado por un vehículo de Fórmula 1, que tarda alrededor de 80 segundos en completar una vuelta. Con el propósito de obtener unos tiempos similares, se ha decidido representar el circuito a escala 1:4, con lo que se consiguen tiempos que rondan los 120 segundos por vuelta. Aunque dicho tiempo sigue siendo superior al de un vehículo de competición, se considera como un valor razonable. También se debe tener en consideración que, a escala reducida, la dificultad de las curvas es mayor y, por lo tanto, una situación más exigente tanto para el controlador como para el conductor real.

Para dibujar la pista y las líneas y bordes de la misma, se emplea la primitiva `GL_QUAD_STRIP`, que dibuja una banda de cuadriláteros conectados, a partir de las coordenadas de los puntos almacenados. La Figura 7.9 muestra el aspecto de la representación que se obtiene.

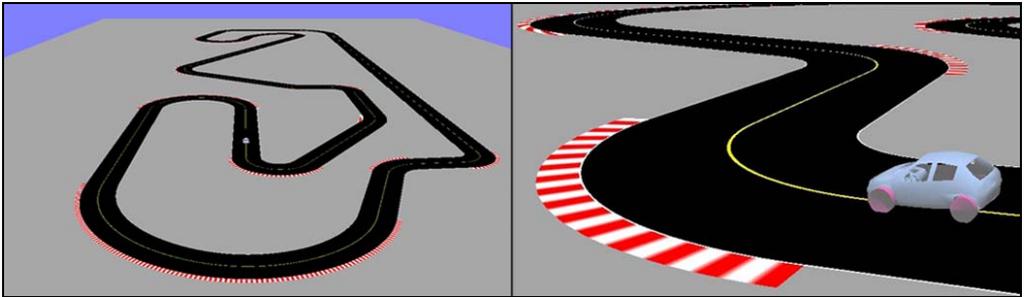


Figura 7.9. Trazado del circuito.

7.2.2. Realismo e Inmersión en el Entorno.

Una vez que se ha representado gráficamente la pista, se ha procurado mejorar el grado de inmersión del conductor en la simulación mediante el aumento del grado de realismo del entorno. Para ello, se emplean las texturas correspondientes a los sólidos utilizados y se representan elementos del circuito tales como gradas y vallas publicitarias. Por último, para aumentar en el conductor la sensación de profundidad, se añade el efecto de niebla.

Sobre el suelo gris se ha introducido una textura de césped, que mejora sensiblemente la vistosidad y el realismo del entorno.

El diseño de las gradas se ha basado en algunas fotografías del circuito. Si la geometría de éstas fuera relativamente sencilla, la implementación gráfica de las mismas podría realizarse especificando directamente las coordenadas de los vértices de cada elemento, así como las normales de sus caras y sus coordenadas de textura. Sin embargo, las gradas reales presentan una cierta complejidad, como se puede observar en la Figura 7.10, que desaconseja este tipo de implementación.



Figura 7.10 Gradas del Circuito de Cataluña.

El proceso que se ha seguido para la representación ha consistido en modelar las gradas de una forma aproximada, mediante el programa de modelado tridimensional Blender, y ejecutar un *script* para extraer los datos necesarios de los objetos modelados. A continuación, estos datos son leídos y representados desde el programa de simulación. Como se recordará, se trata del mismo procedimiento descrito anteriormente, en el apartado 4.3.3.

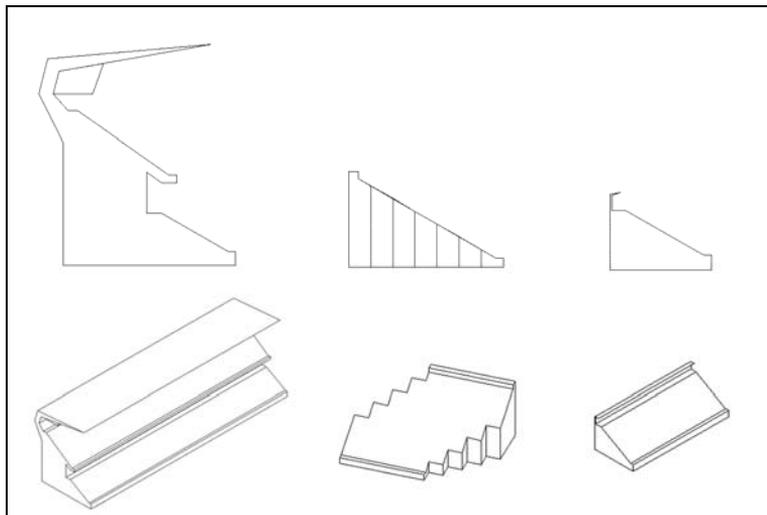


Figura 7.11 Perfiles y geometría de las gradas.

Para el modelado, se dibuja el perfil de la grada como una cara bidimensional, y se extruye en su dirección normal. La Figura 7.11 muestra

los tres perfiles usados, y los objetos generados tras la extrusión de dichos perfiles.

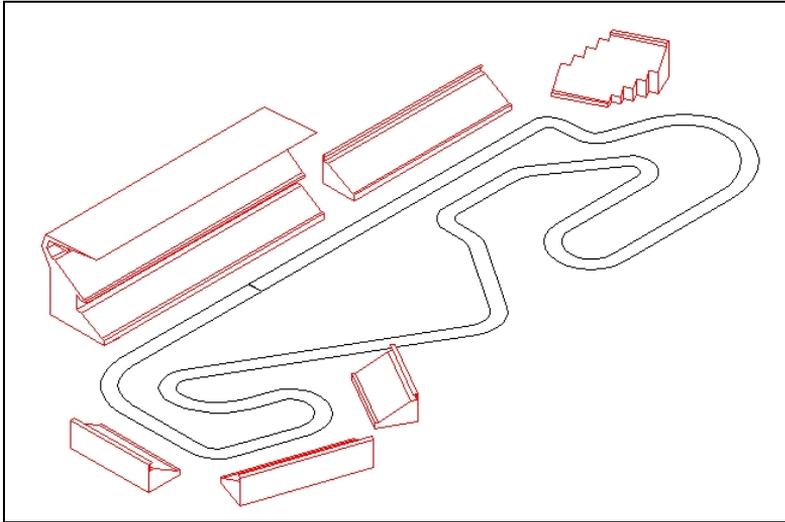


Figura 7.12 Ubicación de las gradas.

La grada de perfil más sencillo (representada a la derecha en la Figura 7.11) se ha empleado en cuatro ocasiones, con lo que se tiene un total de seis gradas que se distribuyen en el circuito según la disposición que muestra la Figura 7.12.

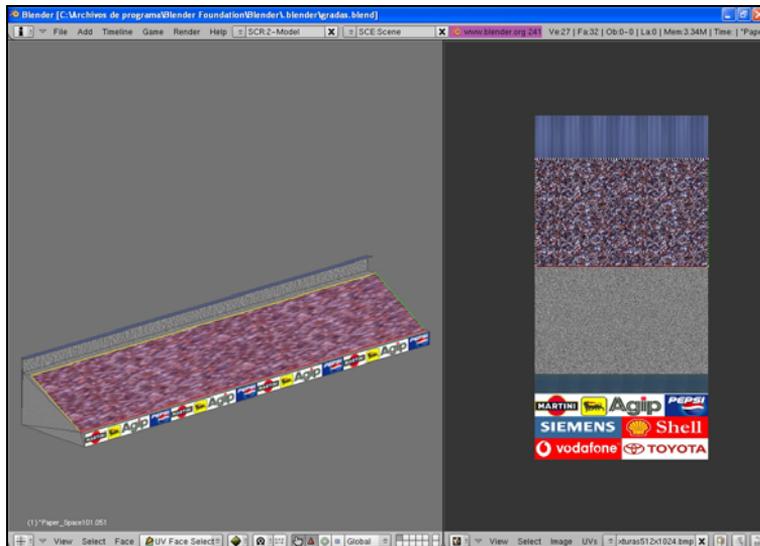


Figura 7.13 Aplicación de texturas a las gradas.

Para añadir la textura, se ha creado una imagen en formato mapa de bits de un tamaño de 512 x 1024 píxeles que incluye todas las texturas que se

emplearán en el texturizado de las gradas: publicidad, público, materiales de la grada, etc. A continuación, a cada cara de la gradas se le asigna el fragmento del mosaico anterior que le corresponda.

Una vez dibujadas y texturizadas todas las gradas, se unen en un único elemento y se ejecuta con un *script* que almacena la información del modelo en un archivo de texto. La información de este archivo de texto contiene el número de vértices de los objetos modelados, las coordenadas y normales de estos vértices, el número de caras de los objetos, los índices de las caras y las coordenadas de textura de los vértices de cada cara. De esta forma, se dispone de toda la información para que las gradas sean representadas en pantalla.

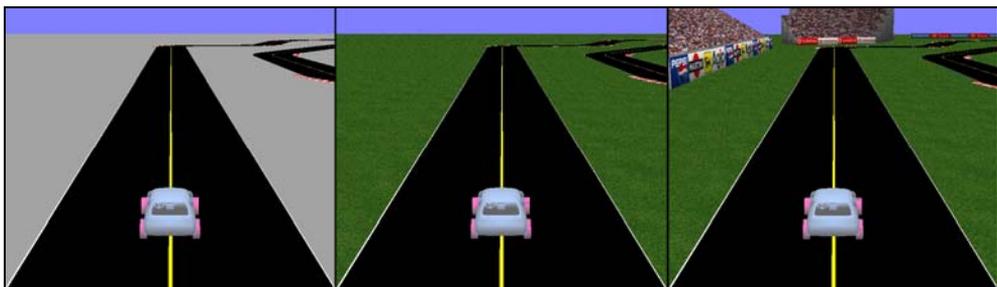


Figura 7.14 Mejora de la inmersión del conductor en el entorno.

Además de las gradas, para cerrar el contorno del circuito se han incluido cinco vallas publicitarias. La geometría de estas vallas consiste en un sencillo poliedro de cinco caras, resultado de extruir un perfil triangular en su dirección normal. El proceso seguido para la representación gráfica de las vallas publicitarias es exactamente igual al seguido para el caso de las gradas.

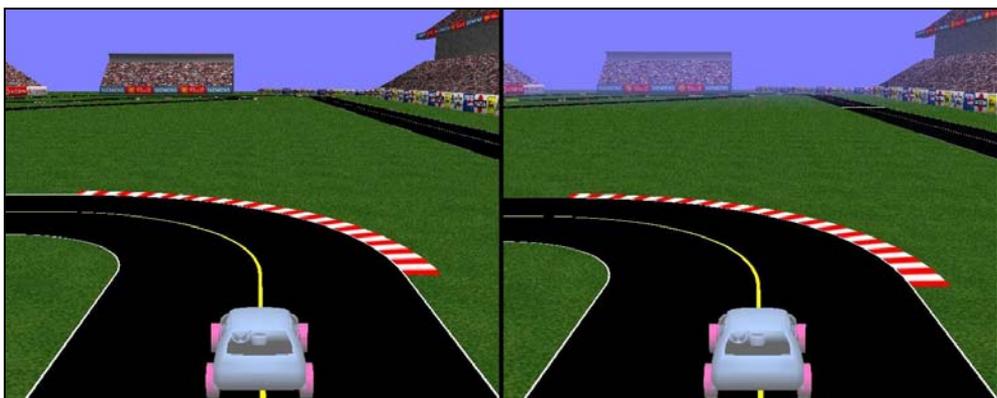


Figura 7.15 Efecto producido por la niebla.

Para terminar, se ha incluido el efecto niebla. La niebla es un efecto especial que admite OpenGL, de fácil uso. Una vez activado este efecto, OpenGL fusiona el color de niebla que se especifique en los elementos geométricos del entorno, una vez efectuados el resto de cálculos de color. La cantidad de color de niebla mezclada con los elementos geométricos varía con la distancia entre dichos elementos y el origen de la cámara.

En este caso, con una ligera cantidad de niebla se busca producir un efecto nebuloso en los objetos distantes que proporcione una sensación realista de profundidad.

7.2.3. Panel de Datos.

Con el objetivo de que el conductor obtenga, de una manera sencilla y rápida, la posición y la velocidad del vehículo a lo largo de la simulación, se ha creado un panel informativo con estos datos en una ventana que se ha situado en la parte superior derecha de la pantalla. Para representar la posición del vehículo se ha decidido dibujar el trazado del circuito y, sobre éste, un cuadrado que indique dicha posición. El resto de los datos se muestran en formato numérico por debajo de la figura del trazado.

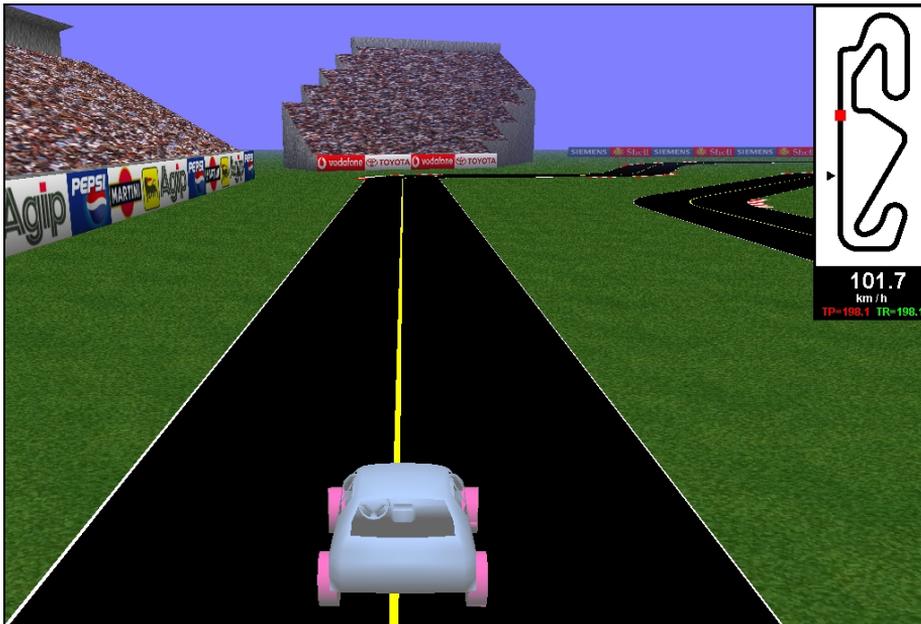


Figura 7.16 Aspecto final del entorno de simulación.

7.2.4. Introducción de Varios Vehículos.

En la aplicación desarrollada se ha contemplado la simulación simultánea de tres vehículos. Uno es guiado por un conductor a través del volante y los pedales del simulador. Los otros dos son guiados mediante sendos controladores.

El controlador empleado no es exactamente uno de los desarrollados en el capítulo 6, sino una versión simplificada. El control del volante es el descrito en apartado 6.3, pero para el control de los pedales no se ha tenido en cuenta la aceleración del vehículo sino únicamente el error y la velocidad.

Asimismo, el universo empleado para la velocidad se ha normalizado tomando valores entre 0 y 1. De esta manera, se puede imponer en cada tramo una velocidad máxima a la que el vehículo puede recorrerlo. Se ha desarrollado un algoritmo para que el conductor virtual decida la velocidad máxima que puede alcanzar en un tramo. La variable con la que se entra en el controlador, será la velocidad instantánea del vehículo dividida por la velocidad máxima establecida para el tramo que está recorriendo.

Siguiendo la técnica para la simulación de colisiones indicada en el apartado 3.7, se ha implementado la posibilidad de que los vehículos colisionen entre sí.

En la resolución con varios vehículos se ha mantenido la misma estructura de variables que en el simulador con un único vehículo. Sin embargo, las variables relativas al vehículo, que se almacenaban en vectores, pasan a estar almacenadas en matrices en las que cada columna guarda los datos de uno de los vehículos. Para cada paso de tiempo se resuelve la dinámica de cada uno de los coches, uno detrás de otro. Se ha contemplado la posibilidad de que alguno de los coches no converja y necesite ser reinicializado, mientras que los demás sigan su proceso normalmente.

7.2.5. Algoritmo de Decisión de la Velocidad Máxima en un Tramo.

Como se ha indicado con anterioridad, la finalidad del algoritmo es que el vehículo decida autónomamente cuál es la velocidad máxima que puede alcanzar en un tramo del circuito sin salirse del mismo. Se ha establecido una velocidad máxima “mínima” de 10 m/s. Si se impone esta velocidad

como máxima, el vehículo no se sale nunca del circuito pero, la velocidad en muchos tramos, especialmente los rectos, es excesivamente reducida. A partir de aquí, se trata de que el vehículo pueda decidir una velocidad mayor en los tramos en que esto sea factible. De entrada, el algoritmo distingue dos situaciones: vehículo situado en tramo recto o en tramo curvo.

Como criterio para decidir la velocidad máxima en un tramo recto se considera la distancia necesaria para frenar el vehículo hasta la velocidad mínima de 10 m/s. Se han realizado pruebas para determinar la distancia que emplea el vehículo en frenar hasta los 10 m/s, partiendo de distintas velocidades iniciales y teniendo en cuenta distintos valores del coeficiente de adherencia de la calzada.

El primer paso del algoritmo es discernir si el vehículo se encuentra ante una recta o una curva. El criterio que se ha empleado es que el error (media entre el de posición y el angular) del vehículo en el seguimiento de la trayectoria sea menor que un umbral determinado. Se comienza evaluando el error en un punto por delante de la posición del vehículo. Si está por debajo del error admisible, se supone que el vehículo se encuentra en una recta, y se comprueba el error en un punto más lejano. Si el error es superior al mínimo admisible, se almacena el último valor de la distancia a la cual el error era inferior al del umbral. Se considera que, si se produce una variación en el trazado del circuito (curva), se producirá un aumento en el error que estará por encima del admisible. De esta manera, se tiene una estimación del tramo recto por delante del vehículo y, por consiguiente, de la distancia que dispone para frenar hasta la velocidad mínima. Si el vehículo circula por un tramo curvo, ya en el primer punto en que se evalúe el error éste se hallará por encima del admisible.

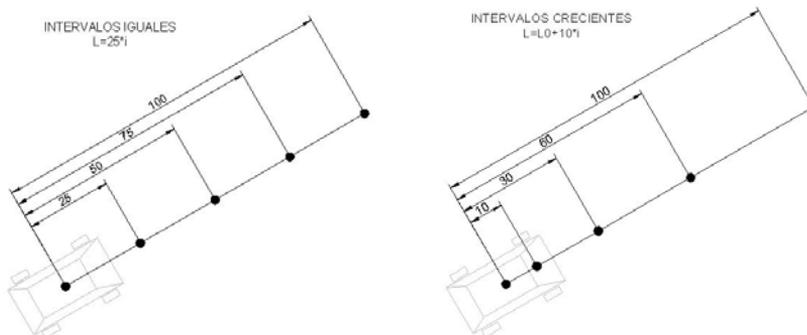


Figura 7.17 Distribución de los puntos de evaluación del error.

Se evalúan los errores hasta una distancia máxima de 100 m por delante del vehículo. Se han estudiado dos alternativas para escoger la situación de los puntos en los que se estudia el error. La primera posibilidad consiste en escoger los puntos de manera que la distancia entre ellos sea constante. La otra alternativa es tomar los puntos de forma que la distancia de separación vaya creciendo a medida que se alejan del vehículo. Estas dos posibilidades son esquematizadas en la Figura 7.17.

Para determinar el valor máximo del error admisible y cuál de los dos métodos resulta más efectivo, se han realizado pruebas con distintos valores para el error, teniendo como criterios que el vehículo no abandone el circuito y que lo recorra en un tiempo mínimo. Esto ha conducido a un valor de 0,5 para el error admisible, y a seleccionar los puntos de evaluación del error a distancia creciente, por proporcionar mejores resultados.

Como se ha indicado anteriormente, un tramo curvo se detecta rápidamente, puesto que el error en el primer punto evaluado es superior al admisible. En este caso, se ha relacionado la velocidad máxima admisible con la curvatura del tramo. Puesto que un conductor real evalúa de alguna manera la curvatura, se ha considerado esta variable como dato conocido. Entonces, se ha ejecutado el programa de simulación imponiendo distintos valores a la velocidad del vehículo en curvas de distinto radio, y observando la máxima a la que la pueden trazar sin salirse de la carretera. La ecuación (7.1) proporciona la relación entre la velocidad máxima y el radio de curvatura de la carretera, ρ .

$$v_{m\acute{a}x} = 1,822 \left(\frac{1}{\rho} \right)^{0,8237} \quad (7.1)$$

7.2.6. Eficiencia del Simulador.

Para evaluar la eficiencia del simulador, conviene tener en cuenta que el programa se ha ajustado de tal manera que resulte prioritario mantener la simulación en tiempo real. De esta manera, en caso de que el cálculo requiera más tiempo, se dejan de representar las imágenes correspondientes hasta que el tiempo de proceso se iguala con el tiempo real. Por este motivo, se puede considerar como medida de la eficiencia el número de imágenes por segundo que se muestran en pantalla.

Para la comparativa, se ha realizado la simulación en el circuito con tres vehículos simultáneos. Uno es guiado mediante el volante y los pedales y los otros dos son guiados mediante control automático. Para evaluar la carga que supone la adición de texturas, se ha probado un entorno con texturas y sin ellas. El algoritmo de decisión de velocidad supone también una carga para el procesador. El programa se ha ejecutado, además, en dos ordenadores distintos. Uno de aproximadamente tres años de antigüedad, con el que se realizaron las maniobras de los controladores, y otro más reciente. Los resultados de imágenes por segundo obtenidas (f.p.s., frames per second) se muestran en la Tabla 7.2.

Como se puede observar, en el ordenador más moderno, el simulador llega a representar en torno a 21 imágenes por segundo en el caso más desfavorable. De esta manera, la percepción visual es bastante buena, proporcionando continuidad en la representación. Es posible mejorar este valor, puesto que en la representación se están evaluando todos los triángulos que forman parte del entorno (incluidos los pertenecientes a gradas y vallas) para comprobar la colisión con ellos. Una opción que alivia el cálculo es dividir el entorno en varias zonas o sectores, de tal manera que, una vez decidida la zona en la que se encuentra el vehículo, sólo hay que evaluar los triángulos pertenecientes a esa zona. En ausencia de texturas, la representación es muy holgada.

El ordenador más antiguo no es capaz de mantener la continuidad en la representación para el caso del entorno con texturas. Sin embargo, sin las texturas proporciona una sensación de continuidad bastante buena aunque, en aquellas ocasiones en que el cálculo requiere un mayor número de iteraciones, pierde la continuidad.

Procesador	Tarjeta Gráfica	Sin Texturas (f.p.s.)	Con Texturas (f.p.s.)
Pentium IV, 2 GHz. 1 GByte RAM	NVidia GeForce2 MX	18,22	2,70
AMD Athlon 64 3000+, 1,81 GHz. 1 GByte RAM	ATI RADEON X300	51,17	21,27

Tabla 7.2 Comparativa de la eficiencia del simulador.

En ambos casos se trata de PC's convencionales sin ninguna característica especial. Estos datos evidencian que la eficiencia del simulador es considerable. El grado de detalle en la simulación mecánica es muy elevado, y se emplea un único PC para resolver tanto el modelo del vehículo como la representación gráfica, proporcionando una simulación en tiempo real y con una buena calidad visual.

8.

Instrumentación del Prototipo

8.1. Motivación.

Como último apartado, se contempla la posibilidad de aplicar al prototipo real los algoritmos de control desarrollados mediante el simulador. Esto tiene varias utilidades, como son:

- Comprobar si los controles desarrollados se pueden aplicar directamente en el vehículo real, o si es necesario efectuar alguna mejora posterior.
- Verificar la precisión del modelo de vehículo que se ha implementado en el simulador para el desarrollo de controladores.
- Evaluar la necesidad de simular los efectos debidos a los actuadores y a los sensores de cara al diseño de los controladores.

Para todo ello se ha instrumentado el prototipo real, de tal manera que se disponga de la misma información que la empleada como inputs en el desarrollo de los controladores con el modelo virtual. En el modelo virtual, posición, velocidad y aceleración son conocidas en cualquier instante. Sin embargo, en el prototipo real es necesario desarrollar un módulo para la determinación del movimiento a partir de las medidas de los sensores. De esta manera, tanto el modelo virtual como el real proporcionan la misma

información para los controladores. En las aplicaciones mostradas, el trayecto que debe seguir el vehículo es conocido y, por tanto, es posible calcular variables como el error en posición o el error angular empleados como inputs de los controladores.

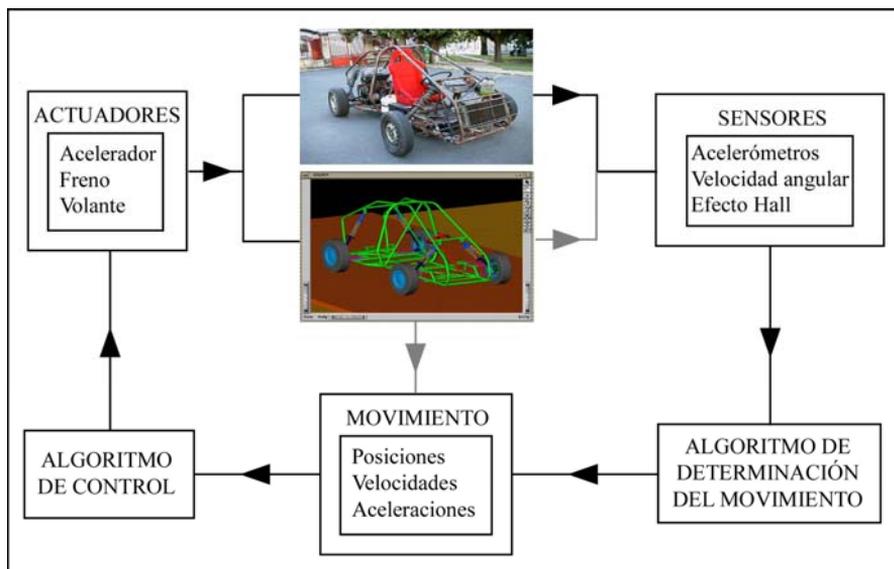


Figura 8.1 Planteamiento para la actuación sobre el vehículo real.

De igual manera, la actuación en el vehículo se hará sobre el volante y los pedales de acelerador y freno. Como se recordará del apartado correspondiente a los controladores, las variables relativas al desplazamiento de los pedales tomaban valores entre 0 y 1, que ahora se reescalan, imponiendo que el valor de 1 corresponda con el desplazamiento máximo del pedal.

8.1.1. Adquisición y Control de Datos.

A la hora de instrumentar el vehículo, es necesario instalar una tarjeta de adquisición y control de datos, que será la encargada de la lectura de la información proporcionada por los sensores, y de emitir las señales necesarias para el manejo de los actuadores.

La tarjeta de adquisición empleada es una Microstar 4200a con las siguientes características:

- CPU
 - Procesador: Intel 80486DX4

- Velocidad de reloj de la CPU: 96 MHz.
- DRAM: 16 MBytes.
- Interface de conexión al PC tipo PCI, con tasa de transferencia máxima de 3.2 muestras/s.
- Entradas analógicas:
 - 16 canales de entrada en modo común.
 - Convertidor analógico digital de 14 bits mediante aproximaciones sucesivas.
 - Máxima frecuencia de muestreo: 769 Kmuestras/s.
 - Rango de entrada: 0 a 5 V; -2,5 V a 2,5 V; -5 V a 5 V; -10 V a 10 V. La configuración empleada ha sido con entradas entre -5 V y 5 V.
- Salidas analógicas:
 - 2 canales de salida con masa común.
 - Convertidor digital analógico de 12 bits.
 - Frecuencia de refresco máxima: 833 Kmuestras/s.
 - Rango de salidas: 0 a 5 V; -2,5 V a 2,5 V; -5 V a 5 V; -10 V a 10 V. Nuevamente, la configuración empleada ha sido con salidas entre -5 V y 5 V.
 - Impedancia de salida: 0,05 Ω .
 - Corriente máxima: ± 1 mA.
- Entradas/salidas digitales:
 - 16 E/S digitales tipo FCT TTL.

Como se puede observar, la tarjeta dispone de un procesador interno que permite la manipulación de datos.

La tarjeta está instalada en un ordenador portátil TOSHIBA Tecra 8000 con procesador Mobile Pentium II @ 300 MHz y 64 MBytes de memoria RAM. Para la instalación de la tarjeta se ha empleado una Deskstation V Plus que proporciona las ranuras PCI de conexión.

Para la alimentación eléctrica se ha instalado un SAI Merlin Gerin Pulsar 8 V. De esta manera, el PC dispone de una autonomía de una hora, aproximadamente.

Para la comunicación con la tarjeta, Microstar Laboratories proporciona un software y, además, los *drivers* necesarios para comunicar con distintos programas comerciales como Matlab, LabView, etc. También es posible crear programas para la tarjeta en C, o en C++. En el presente trabajo, la tarjeta de adquisición de datos se controla desde Matlab. Esto permite la manipulación de los datos, la obtención sencilla de gráficas, y la conexión con otras toolboxes de Matlab como la de Fuzzy Logic que, como se ha indicado anteriormente, se ha empleado en el desarrollo de los controladores.

Al conectar con la tarjeta de adquisición desde Matlab, el primer paso es abrir un canal de comunicación con la tarjeta mediante el comando INIT, proporcionado por el fabricante. A continuación, a través del canal de comunicación, se envía un archivo de comandos para la configuración de la tarjeta.

En el archivo de comandos se especifican los canales que se va a emplear y, acto seguido las distintas operaciones o procedimientos que debe realizar la tarjeta.

El primer bloque de comandos está constituido por las operaciones de toma de datos. En él hay que especificar los canales por los que la tarjeta va a recibir los datos, y el tiempo en microsegundos que transcurre entre la lectura de canales consecutivos mediante el comando TIME. De esta forma, si se van a leer cuatro canales, por ejemplo, y se desea que la frecuencia de muestreo en cada canal sea de un milisegundo, habrá que especificar un tiempo de muestreo de 250 microsegundos.

A continuación, se suceden los siguientes bloques de comandos. Con ellos se pueden manipular los datos, enviarlos al PC, ordenar la salida de datos, etc. En las aplicaciones desarrolladas, lo más común es enviarlos al PC mediante el comando BMERGE.

En el fichero de configuración se detallan los procedimientos que se van a realizar. Estos procedimientos pueden ser iniciados y terminados tantas veces como se estime conveniente. Para ello, se envía desde el PC, a través del canal de comunicación, el comando START seguido del nombre de los procedimientos que deben iniciarse. Análogamente, el comando

STOP seguido del nombre de algún procedimiento termina la ejecución del mismo. Si no se especifica ningún nombre, el comando STOP finaliza todos los procedimientos que se encuentren activos en ese instante.

8.2. Determinación del Movimiento.

Durante los últimos años se ha experimentado un gran desarrollo de la investigación en este campo, motivada por la incorporación de sistemas de posicionamiento en automóviles comerciales. Anteriormente, los sistemas de posicionamiento se empleaban fundamentalmente en navegación marítima y aérea. La navegación se puede automatizar mediante los sistemas de radio Loran y Omega, y el Sistema de Posicionamiento Global (GPS) que emplea satélites (Lawrence 1998). En determinadas ocasiones, estos sistemas pueden no estar disponibles, siendo entonces necesario emplear un sistema sin referencias exteriores, denominado navegación inercial.

La navegación inercial emplea giróscopos y acelerómetros (sensores inerciales) para determinar el movimiento del vehículo a partir de los cambios provocados por las aceleraciones. Este sistema se comenzó a emplear durante la Segunda Guerra Mundial en armas estratégicas guiadas, donde el coste no tenía relevancia. Sólo 20 o 30 años más tarde, el desarrollo de esta tecnología hizo que llegase a ser lo suficientemente barata como para utilizarla comercialmente.

La revolución electrónica ha afectado al campo de la navegación inercial en dos sentidos: el desarrollo del GPS ha inclinado la balanza a favor de los sistemas asistidos, y la microelectrónica ha modificado la tecnología de los sistemas inerciales.

Los sistemas de navegación para vehículos automóviles constituyen un producto emergente en el mercado de Estados Unidos, Japón y, con un cierto retraso, Europa. Tal como se han desarrollado, estos sistemas incluyen, para el posicionamiento, un receptor GPS, un sensor de velocidad, un sensor de velocidad angular y un mapa-base de datos sobre el cual los algoritmos de navegación trazan el plan de viaje y siguen la ruta seleccionada (Ibrahim et al. 2005). El GPS y sus aplicaciones en el Intelligent Transportation System constituyen partes vitales de la sociedad de la información. Hoy por hoy, los sistemas de localización de vehículos (Automatic Vehicle Location Systems, AVLS) que incorporan un GPS, son

más robustos y económicamente más competitivos que los que simplemente se basan en sistemas inerciales.

Los sistemas DGPS (Sistema de Posicionamiento Global Diferencial), que emplean dos antenas para comparar el desplazamiento del vehículo con respecto a una de ellas que permanece fija, llegan a determinar la posición de un vehículo en movimiento, en condiciones normales, con una precisión en torno a los 10 cm (Neumann et al. 1996). El rango de frecuencia de muestreo de la información proporcionada por estos sistemas se encuentra entre 1 y 10 Hz (Bevly 2004). Sin embargo, estos sistemas requieren tener visibilidad de, al menos, cuatro satélites. La visibilidad de un satélite no es problemática en carretera, pero sí lo es en túneles y condiciones urbanas. Las pérdidas de visibilidad se suelen solventar empleando sistemas de navegación inercial. Los datos proporcionados por los sensores se emplean junto con un modelo del movimiento del vehículo para calcular la posición. El estudio de la integración de la información de los sensores inerciales con la proporcionada por el GPS ha sido fuente de un gran número de trabajos como los de Alexander and Donath (1999), Bevly et al. (2002), Ryu and Gerdes (2004), Ibrahim et al. (2005), Huang and Tan (2006), etc. Un enfoque frecuente consiste en emplear filtros de Kalman (Kalman 1960), que modelizan matemáticamente del error cometido por los sensores inerciales, así como el error cometido por el GPS, para estimar una medida mejor de la posición. El punto crítico de la determinación del movimiento de un vehículo automóvil estriba en la necesidad de estimar con precisión el ángulo de deriva del vehículo o, en su lugar, la velocidad del vehículo en la dirección transversal.

Como se puede observar, la frecuencia del GPS es relativamente baja, lo que plantea algunos posibles problemas. Por otro lado, mediante sensores inerciales es posible recuperar el movimiento con una frecuencia más elevada y no es necesario tener en cuenta la problemática derivada de la visibilidad de los satélites. Por este motivo, se ha diseñado un algoritmo para la determinación del movimiento, que emplea datos proporcionados por sensores inerciales. Mediante este algoritmo se obtiene la información de posición, velocidad y aceleración que constituye la entrada de los controladores empleados. Como se verá más adelante, el algoritmo se basa en un método odométrico que, pese a su sencillez, proporciona unos errores bajos y admisibles en las pruebas realizadas.

8.2.1. Sensores Empleados.

Para la determinación del movimiento, se ha optado por instalar en el vehículo un sensor de velocidad angular, un acelerómetro y un sensor de efecto Hall.

El sensor de efecto Hall detecta la presencia ante el sensor de un elemento magnético, y tiene como propósito poder determinar el giro que ha experimentado la rueda. Se ha instalado en el disco de la rueda delantera izquierda. En éste se han efectuado 40 agujeros de 10 mm de diámetro, situados de forma equidistante a lo largo de un diámetro de 205 mm. De esta manera, es posible contar los agujeros que pasan ante el sensor, de la misma manera en que lo hace el sensor del ABS de los vehículos comerciales. El sensor de efecto Hall proporciona un valor de salida de 5 V cuando se encuentra ante un agujero, y de 0 V al situarse frente al hierro del disco. Como se verá en el siguiente apartado, la señal proveniente del sensor de efecto Hall se emplea para el desplazamiento del vehículo y su velocidad.

Para medir la aceleración se ha empleado una de las componentes de un acelerómetro triaxial Longbow. Como se recordará, en los controladores solamente se tiene en cuenta la componente longitudinal de la aceleración. El sensor se ha instalado en la parte superior del chasis del vehículo, Figura 8.2 derecha, #1, en una posición centrada tanto longitudinal como transversalmente. Se ha estudiado la posibilidad de emplear este sensor para integrar su señal obteniendo la velocidad, pero el ruido obtenido de la señal es demasiado elevado y fuente de un error excesivo en la integración.



Figura 8.2 Colocación de los sensores en el vehículo.

El sensor de velocidad angular se emplea para determinar el ángulo girado por el vehículo. Se ha situado tras el asiento del piloto, en un punto próximo al centro del automóvil, Figura 8.2 derecha, #2. De esta manera, se puede determinar la orientación del chasis en un instante dado, despreciando los efectos debidos a deformaciones, y considerando que el movimiento que se realiza es plano. Estos supuestos encajan bastante bien para el caso de las maniobras que se han programado con los controladores.

8.2.2. Algoritmo.

El algoritmo encargado de la determinación del movimiento sigue un método odométrico. En concreto, el sensor de efecto Hall se emplea, como veremos más adelante, como un *encoder* que permite obtener el desplazamiento y la velocidad. El sensor de velocidad angular se emplea para obtener, mediante integración el ángulo girado por el vehículo. Del acelerómetro se obtiene directamente la aceleración longitudinal.

El algoritmo de determinación del movimiento se ha esquematizado en la Figura 8.3. Como se puede observar, la primera etapa consiste en un acondicionamiento de la señal que es peculiar para cada uno de los sensores.

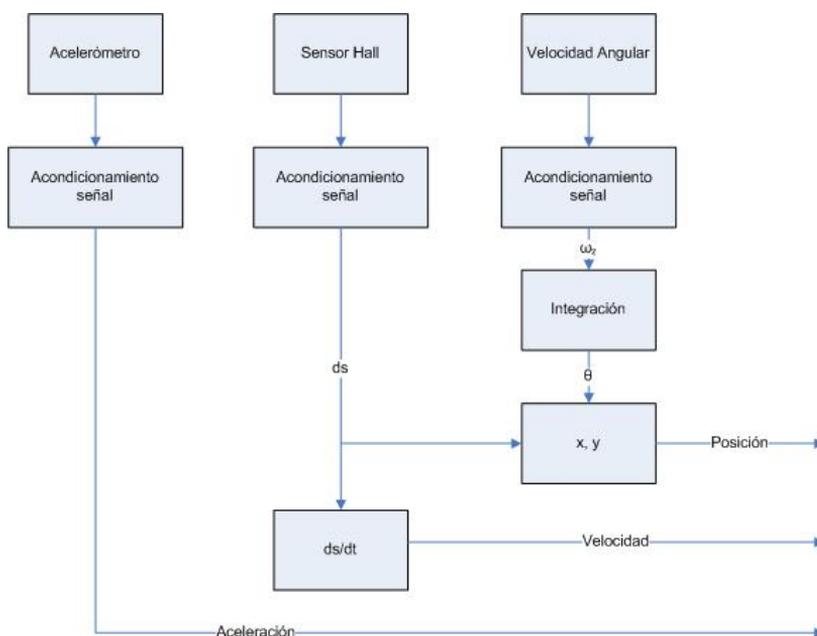


Figura 8.3 Esquema de la determinación del movimiento.

Para la aceleración se emplea la información procedente del eje longitudinal del acelerómetro. El acelerómetro tiene un rango de ± 2 g y un factor de escala de 0,981 V/g. Se ha tomado un valor para la gravedad $g = 9,81 \text{ m/s}^2$. El fondo de escala de las lecturas es 5 V, transformado por el convertidor analógico digital a un valor entero de 32768 en la tarjeta. El acelerómetro, según datos del fabricante devuelve 2,545 V cuando está sometido a una aceleración de 0 g. Por este motivo, para obtener la aceleración se realizan las siguientes operaciones

$$\text{Señal Voltios} = \frac{\text{Lectura Tarjeta}}{32768} \cdot 5 \quad (8.1)$$

$$a = \frac{(\text{Señal Voltios} - 2,545)}{0,981 \text{ V/g}} \cdot 9,81 \text{ m/s}^2 \quad (8.2)$$

La señal que proporciona el sensor de efecto Hall, tras hacer el acondicionamiento indicado por la ecuación (8.3), está formada por escalones entre 0 y 5 V. Los escalones corresponden al paso de los agujeros del disco de freno ante el sensor.

$$\text{Señal Voltios} = \frac{\text{Lectura Tarjeta}}{32768} \cdot 5 \quad (8.3)$$

El sensor de velocidad angular tiene un rango de ± 100 °/s y un factor de escala de 20 mV/°/s. Por este motivo, para obtener la velocidad angular, la señal que se recibe de la tarjeta debe ser reescalada de la siguiente manera, hasta obtener la velocidad angular en rad/s:

$$\text{Señal Voltios} = \frac{\text{Lectura Tarjeta}}{32768} \cdot 5 \quad (8.4)$$

$$\omega_z = \frac{\text{Señal Voltios}}{0,02 \text{ V/°/s}} \cdot \frac{\pi}{180} \text{ rad/s} \quad (8.5)$$

Para la obtención de la posición del vehículo, se emplea simultáneamente la información del sensor de efecto Hall y la del sensor de velocidad angular. Por un lado, se tienen en cuenta los flancos de bajada que se producen en la señal del sensor de efecto Hall. El registro de un flanco de bajada se corresponde, suponiendo que hay una rodadura pura entre el

neumático y el suelo, con un avance del vehículo, s , determinado según la siguiente expresión:

$$s = \frac{2\pi r_{def}}{40} = \frac{2\pi 0,26}{40} = 0,04084 \text{ m} \quad (8.6)$$

r_{def} es el radio efectivo de la rueda, es decir, el radio que realmente tiene la rueda bajo carga. En el caso de la rueda instrumentada, se ha tomado una medida de 0,26 m.

Simultáneamente, la información sobre velocidad angular, ω_z , se integra empleando la regla trapezoidal, para obtener el ángulo de orientación del vehículo en el instante actual, θ .

$$\theta_i = \theta_{i-1} + \frac{dt}{2} (\omega_{z_i} + \omega_{z_{i-1}}) \quad (8.7)$$

donde los valores con subíndice i corresponden a los valores en el instante de tiempo actual, y el subíndice $i-1$ hace referencia al instante anterior.

Una vez que se dispone del avance que se ha producido y del ángulo girado en ese instante de tiempo, es posible proyectar el avance según un sistema de referencia fijo:

$$s_x = s \cdot \cos \theta \quad (8.8)$$

$$s_y = s \cdot \sin \theta \quad (8.9)$$

Para la estimación de la velocidad se emplea también la señal procedente del sensor de efecto Hall. Para obtener una estimación de la velocidad con una frecuencia de 100 Hz (que es la frecuencia del simulador y con la que se han diseñado los controladores), puesto que la información del sensor de efecto Hall se obtiene con una frecuencia de 1000 Hz, se acumulan los incrementos de distancia que se producen durante 10 pasos de tiempo y se divide por el periodo de tiempo transcurrido.

$$v = \frac{\sum_{i=1}^{10} s_i}{10 * dt} \quad (8.10)$$

Este procedimiento impone una serie de limitaciones. Por un lado, para tener una suficiente precisión, es necesario que el número de agujeros por vuelta sea elevado. El número que se ha tomado de 40 agujeros

proporciona una resolución de unos 4 cm, que parece suficiente para las maniobras que realiza el vehículo.

Por otro lado, la frecuencia de muestreo debe ser lo suficientemente elevada como para que no se deje de detectar el paso de un agujero frente al sensor. En dos pasos de tiempo se debe detectar un flanco de subida y uno de bajada. Es decir, en el caso en que la rueda gire más rápido, como mucho debe pasar un agujero por delante del sensor en dos pasos de tiempo. Se ha empleado una frecuencia de muestreo de 1000 Hz. Esto limita la velocidad a 73,5 km/h según se indica en (8.11). Aunque la tarjeta permite ir a frecuencias de muestreo más elevadas esta velocidad es más que suficiente.

$$v_{\text{máx}} = \frac{2\pi r_{\text{def}}}{2 \cdot 0,001} = 20,42 \text{ m/s} = 73,5 \text{ km/h} \quad (8.11)$$

Otros errores del método que afectan especialmente a la determinación de la posición son los siguientes:

- Como hemos visto, se ha empleado un valor aproximado y constante para el radio de la rueda. Sin embargo, el radio puede variar aunque, sea mínimamente, debido a irregularidades del terreno o a pequeñas variaciones en el reparto de carga.
- Se ha supuesto que la rueda se encuentra alineada con la dirección de avance del vehículo.
- Se han despreciado los deslizamientos que se producen en el contacto neumático-suelo durante el movimiento.

Posteriormente, en las pruebas realizadas, se podrá comprobar numéricamente la incidencia de estos errores. Sin embargo, se puede adelantar que, para las maniobras a realizar, la precisión del algoritmo de determinación del movimiento resulta suficiente.

8.3. Actuación.

Para la conducción autónoma del vehículo se ha decidido actuar sobre el volante y los pedales de acelerador y freno. Se ha evitado tener que manipular el cambio de marchas por la complejidad de automatización del embrague y de la caja de cambios. Por este motivo, se seleccionó en su día un motor de cambio automático el vehículo. La actuación sobre la palanca

de cambios en un vehículo automático resulta trivial. Es posible detener completamente el vehículo, sin necesidad de actuar sobre la palanca de cambios, cuando ésta se encuentra en la posición de conducción. Además, en el caso de que se deseara manipular la palanca de cambios, bastaría con instalar un actuador lineal que llevara la palanca a las distintas posiciones (paro, marcha atrás o una marcha concreta), sin desplazamientos en dos direcciones, y sin tener que actuar simultáneamente sobre el embrague, como sería necesario en una caja de accionamiento manual. Como, además, las maniobras para las que se han diseñado los controladores no requieren en ningún momento tener que desplazar el vehículo marcha atrás, no se ha instalado ningún actuador para el accionamiento de la palanca de cambios.

8.3.1. Motores.

A la hora de seleccionar los actuadores, los mayores esfuerzos a vencer se presentan en el volante y el pedal de freno. La dirección es de cremallera y, debido a su estado, presenta bastantes pérdidas. Además, dentro del vehículo no se dispone de una fuente de energía eléctrica constante.

Se desechó la posibilidad de emplear motores de corriente alterna porque no se disponía de una fuente de alimentación con potencia suficiente. Las opciones disponibles eran un motor de corriente continua o un motor paso a paso. Los motores de corriente continua necesitan, además, un *encoder* para el posicionamiento del motor y, consecuentemente, una aplicación de control para gobernar el desplazamiento del motor. Los motores paso a paso son gobernados mediante pulsos eléctricos, cada uno de los cuales conlleva un desplazamiento concreto. De esta manera, tienen la ventaja de que, conociendo los pulsos de control que se han enviado, se conoce el giro que ha realizado el motor. Como contrapartida, si el par resistente es muy elevado se pueden producir deslizamientos entre el eje del motor y los polos del estator. Esta dificultad puede ser superada por el mero hecho de emplear Lógica Borrosa, puesto que, si el giro no ha sido exactamente el requerido, el controlador borroso ordenará un mayor giro del actuador. Por este motivo se seleccionaron como actuadores motores paso a paso.

Para la actuación sobre el volante, se ha empleado un motor que proporciona 7,8 N·m. Se acopla a la dirección como sustituto de la barra de transmisión del volante, y actúa directamente sobre la cremallera.

Realizando pruebas, se puso de manifiesto la necesidad de un mayor par motor, por lo que se añadió una caja reductora con relación de reducción 1:3. La reductora, al igual que los soportes y bastidores de los motores, fue realizada en el taller mecánico del Centro de Investigaciones Tecnológicas de la Universidad. La disposición del conjunto se muestra en la Figura 8.4 izquierda. En las últimas pruebas realizadas, con el conductor sentado en el vehículo y en parado, el motor daba la potencia justa para girar la dirección. Aunque aún no se ha podido verificar, es de esperar que el par requerido en movimiento no sea tan elevado, y el motor no tenga ninguna dificultad para guiar la dirección.



Figura 8.4 Instalación de los actuadores en el vehículo.

Sobre el pedal de freno es normal que se lleguen a realizar fuerzas de aproximadamente 80 Kg que, al actuar el servofreno, se reducen a menos de la mitad. Para la actuación, se ha prolongado el brazo de palanca del pedal, de manera que se pueda aprovechar como reductora la relación entre el brazo de palanca del freno y el brazo del actuador. La relación entre ambos es de 1:4. Se ha instalado un motor similar al que actúa sobre el volante, de 7,8 N·m de par, que tira de la palanca a través de un hilo, arrollado al eje con un tambor de 26 mm de diámetro (Figura 8.4 derecha).

El tipo de actuación sobre el pedal del acelerador es igual al empleado sobre el freno. Sin embargo, la fuerza necesaria para desplazar este pedal es mucho menor. El propio pedal del acelerador ya realiza una reducción, y las fuerzas a vencer no son muy elevadas. Se ha instalado un motor de 2,8 N·m de par que actúa también a través de un hilo que se arrolla en un tambor de 36 mm de diámetro (Figura 8.4 derecha).

Para suministrar la alimentación eléctrica a los motores, se dispone de dos baterías de 24 V situadas en la parte trasera del vehículo. Estas

baterías son similares a las empleadas para la alimentación del circuito eléctrico del propio vehículo.

8.3.2. Algoritmo de Actuación.

Por el momento, se ha diseñado un algoritmo de actuación que, al no haber podido ser implementado ni comprobado, se trata más bien de un esquema a seguir. Para el desarrollo del algoritmo de actuación de los motores no se ha tenido en cuenta la dinámica de los mismos ni los posibles errores que cometan durante su actuación. También se han realizado una serie de supuestos que deben ser verificados y, si es necesario, corregidos oportunamente. Se ha supuesto que no se producen deslizamientos en el funcionamiento de los motores. Otra hipótesis latente en el planteamiento de la actuación es que el empleo de la Lógica Borrosa es capaz de “absorber” los posibles errores que se comentan en el posicionamiento de los actuadores. Como se puede observar, esta es una parte del trabajo que todavía está por desarrollar y que, seguramente, puede requerir bastante tiempo hasta la puesta a punto. En los siguientes párrafos se describen los elementos a tener en cuenta en la manipulación de los motores, y el posible algoritmo de actuación.

Cada uno de los motores instalados en el vehículo requiere dos señales para su control. Una de las señales corresponde al control del sentido giro, y la otra determina el ángulo girado por el motor. Al tratarse de motores paso a paso, el control del ángulo girado equivale al número de pasos que debe desplazarse el motor. La señal que se envía al módulo de accionamiento para indicar el sentido de giro es de +5V para un giro positivo, y -5V para negativo. En el caso del número de pasos se envía desde la tarjeta de adquisición una onda cuadrada de 5V de amplitud. Para formar esta onda, desde la tarjeta se van enviando señales de voltaje 0 ó 5, distanciadas entre sí 1 ms. Esto supone una frecuencia de 500 Hz (se necesita enviar dos pulsos de señal para dar un paso formando una onda cuadrada) que es la máxima frecuencia que se puede alcanzar antes de la saturación del motor, por lo que cada 2 milisegundos se avanzará un paso (1.8 °).

Como se recordará (apartado 8.2.2), para el cálculo de la velocidad se emplean diez pasos de tiempo. Por este motivo, se dispone de los datos necesarios para el control cada 10 ms. Otra limitación surge de la transferencia de datos entre la tarjeta de adquisición y Matlab. Se

comprobó que en el ordenador empleado, para asegurar la determinación del movimiento en tiempo real era necesario enviar la información de los sensores en paquetes de 60 datos. Esto corresponde a las muestras tomadas durante 20 ms y constituye una nueva limitación para la actuación. Sin embargo, también representa una ventaja para la actuación de los motores puesto que 20 ms representa una cantidad de tiempo suficiente para realizar un desplazamiento apreciable de los motores. Se realiza una actuación secuencial de los motores cada 20 milisegundos. En el algoritmo de control se había impuesto que el acelerador y el freno no podían ser manipulados simultáneamente. El algoritmo chequea si es necesario actuar sobre el freno. Si no lo es, pasa al acelerador. Si se ha actuado sobre el freno, se obvia la actuación sobre el acelerador. En el caso de que el desplazamiento requerido en el freno o el acelerador no agote los 20 ms, el tiempo restante se acumula para la actuación del volante. De esta manera, en 40 milisegundos se ha actuado una vez sobre la pareja acelerador-freno y el volante. La actuación de los motores es uno a uno, debido a que se utilizan las salidas síncronas de la tarjeta para poder contar el número de pasos. Así, no se produce solapamiento de órdenes en los canales de salida, lo que ocurre cuando el canal de salida tiene datos y se escribe en él.

Al ser actuaciones secuenciales cada 20 ms, el número máximo de pasos que puede dar el motor es de diez, que corresponde a 18 grados de giro del motor, en el caso de que tanto la pareja acelerador/freno como el volante necesiten moverse durante todo el tiempo. Como es lógico, en el caso del volante, el giro obtenido será tres veces menor, debido a la reductora instalada.

Por otro lado, hay que tener en cuenta el tipo de tarjeta utilizada para las señales de salida. Al emplear una tarjeta analógica que convierte las salidas digitales de la tarjeta DAP en analógicas, y que utiliza cuatro valores por cada valor que se quiere enviar, la frecuencia de envío de señales a los motores debe de ser cuatro veces mayor.

8.4. Pruebas y Resultados.

Para verificar el comportamiento del algoritmo de determinación del movimiento, se ha realizado una prueba consistente en que el vehículo dé la vuelta a uno de los edificios del campus, cuidando de que las posiciones de salida y llegada sean coincidentes. De esta manera, las desviaciones

obtenidas en los valores correspondientes a los desplazamientos en cada una de las direcciones, se pueden considerar directamente como el error cometido por el algoritmo.

Las Figura 8.5 y Figura 8.6 muestran los desplazamientos en la dirección del eje x y del eje y respectivamente. Como se puede observar, el error cometido en x es de 0,97 m sobre un desplazamiento total de 123,14 m. El error obtenido es pues del 0,78 %.

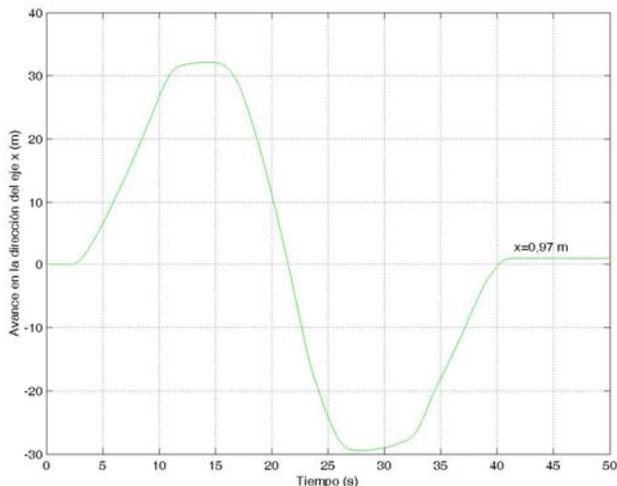


Figura 8.5 Movimiento en x según el algoritmo.

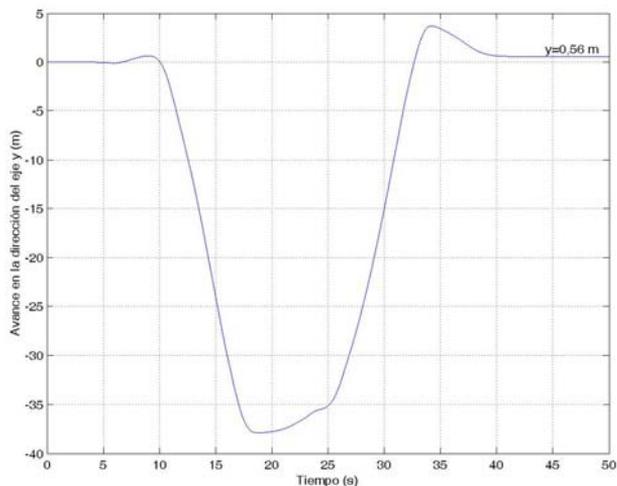


Figura 8.6 Movimiento en y según el algoritmo.

En la dirección y , el vehículo se detiene con un error de 0,56 m. El desplazamiento total es de 80,14 m. Por consiguiente, el error cometido es del 0,69 %. Como se observa, es próximo al cometido en el eje x .

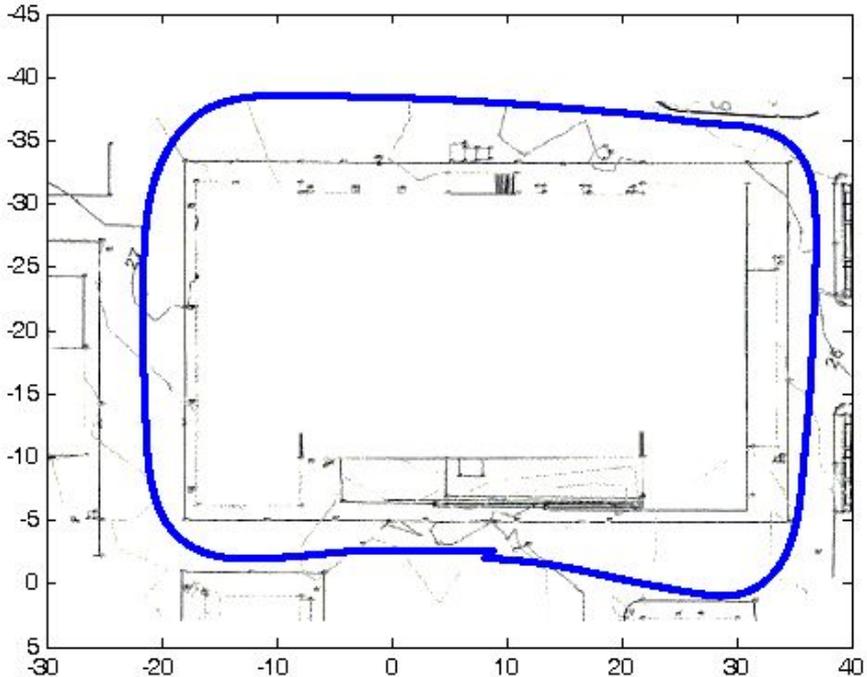


Figura 8.7 Trayectoria del vehículo obtenida por el algoritmo alrededor del edificio de la Facultad de Humanidades.

Finalmente, la Figura 8.7 muestra la trayectoria determinada por el algoritmo cuando el vehículo describe una vuelta completa alrededor del edificio. Como se puede observar, es congruente con la geometría del entorno. El punto de llegada es aproximadamente el de salida, salvo los errores que se han mencionado anteriormente.

Se ha repetido la maniobra dos veces más, obteniendo resultados similares.

9.

Conclusiones

9.1. Conclusiones.

Como se ha podido observar a lo largo de la exposición, la tesis se puede dividir en dos líneas: la referente al prototipo real y su instrumentación, y la correspondiente al modelo virtual y los trabajos realizados sobre él. Por este motivo, parece conveniente dividir en estos dos bloques tanto las conclusiones como las líneas de investigación que se derivan de cada uno de ellos.

En el apartado correspondiente al prototipo real, cabría resumir los resultados alcanzados como sigue:

- Se ha construido un prototipo real, con un chasis formado con barras tubulares, y elementos mecánicos adaptados a partir de los de vehículos comerciales.
- El prototipo se ha instrumentado con una tarjeta de adquisición de datos y un ordenador, resolviendo el suministro de electricidad mediante el empleo de un Sistema de Alimentación Ininterrumpido (SAI). Esta instrumentación posibilita el empleo del prototipo para la realización de experimentos en los que sea necesaria la toma de datos mediante sensores.

- Con vistas al desarrollo de algoritmos de conducción automática, la instrumentación del vehículo se ha realizado con un sensor de efecto Hall, un sensor de velocidad angular y un acelerómetro.
- Se ha desarrollado un algoritmo que, a partir de los datos de los sensores, permite obtener la posición, velocidad y aceleración del vehículo con una frecuencia de 100 Hz.
- El algoritmo de determinación del movimiento se ha verificado mediante maniobras con el vehículo, y se ha encontrado un error de un 1 % para trayectorias recorridas de entre 80 y 130 m.
- Con vistas a la conducción autónoma, se han instalado en el prototipo tres motores paso a paso para la actuación sobre el volante y los pedales de acelerador y freno. El control de los motores se realiza desde la tarjeta de adquisición.

En lo que se refiere al modelo virtual, los resultados conseguidos se resumen así:

- Se ha realizado un modelo virtual completo del prototipo. Se ha empleado un método desarrollado con anterioridad en el Laboratorio de Ingeniería Mecánica, consistente en una formulación aumentada de Lagrange de índice 3 con proyecciones de velocidades y aceleraciones, que incorpora técnicas de matrices dispersas, y la regla trapezoidal como integrador. Aunque la formulación dinámica es global (por consiguiente, fácil de implementar), y emplea un integrador implícito como es la regla trapezoidal (estable, también, ante problemas *stiff*), se alcanza el tiempo real con un paso de tiempo de 10 ms al simular el modelo completo y detallado del vehículo, programado en Fortran.
- Se han modelizado las fuerzas introducidas por cada uno de los elementos de vehículo. El peso del motor y sus características geométricas se han obtenido por métodos experimentales. El peso y la distribución de masas del chasis se han obtenido a partir de un modelo CAD. Las fuerzas del motor se han tomado de las curvas de par-revoluciones del motor real. Las fuerzas de frenado se han estimado a partir de los datos geométricos de los discos de freno del vehículo. Para la modelización de los neumáticos se ha empleado la *Fórmula Mágica* de Pacejka, con coeficientes suministrados por el

- fabricante. Los elementos de la suspensión se han estimado, y se han representado con muelles y amortiguadores de comportamiento lineal.
- El modelo del vehículo ha servido como base para desarrollar un simulador de conducción y una herramienta para el diseño de controladores. Se ha desarrollado una metodología completa para la creación de entornos de conducción, y se ha resuelto satisfactoriamente la interacción del vehículo con el entorno. Para los inputs se ha empleado un conjunto de volante y pedales del tipo de los utilizados en los juegos de ordenador. La salida gráfica se ha realizado a través del monitor del PC o, alternativamente, unas gafas de realidad virtual con sensor para la orientación de la cabeza.
 - En el simulador de conducción se han desarrollado tres entornos de conducción: uno con obstáculos (rampas, escaleras, túneles, etc.) para pruebas de robustez de la formulación; otro –basado en el circuito de Cataluña- que ha permitido comprobar la eficiencia del código al incluir la simulación simultánea de varios vehículos; y un último entorno que recrea parte del barrio de la Magdalena de Ferrol. Se ha verificado que el código es robusto y eficiente. La formulación permite además que todo el esfuerzo computacional relativo al simulador (dinámica del vehículo, gráficos, etc.) pueda ser llevado a cabo por un único PC sin características especiales.
 - El modelo del vehículo se ha conectado con Matlab para el desarrollo de controladores. Se han probado distintas metodologías para la conexión, resultando como más ventajoso el empleo de Matlab Engine.
 - Se han desarrollado los controladores para que el vehículo realice automáticamente dos maniobras sencillas: recorrer un tramo recto deteniéndose lo más próximo posible a un objetivo, y seguir una trayectoria sinusoidal como ejemplo de elusión de un obstáculo. Como inputs de los controladores se emplean posición, velocidad y aceleración, imitando los datos que proporcionan los sensores en el prototipo real. Los outputs son la posición de los pedales de acelerador y freno, así como la del volante. Como técnica de control se ha empleado Lógica Borrosa (Fuzzy Logic), aunque la metodología empleada es extensible a otras técnicas de control. El uso de Lógica Borrosa está justificado por el carácter altamente no lineal del sistema, permite evitar el empleo de un modelo simplificado del vehículo (como

el de la bicicleta), y es capaz de reproducir el comportamiento de un conductor debido al uso de expresiones de tipo lingüístico. Asimismo, Matlab proporciona una toolbox de Lógica Borrosa que facilita su empleo.

- En el proceso de diseño de controladores, el simulador proporciona un interface que facilita una valoración inicial del desarrollo de la maniobra y del realismo de los valores empleados como parámetros del controlador, permite disponer de una referencia para valorar la calidad de los controladores obtenidos, y posibilita la acción conjunta del conductor real y del controlador para aplicaciones de control parcial, como pueden ser todas aquéllas relacionadas con la asistencia a la conducción.
- La conexión entre el código del modelo del vehículo y Matlab mediante Matlab Engine ha resultado una herramienta útil durante la etapa de diseño de los controladores. El almacenamiento de los controles en matrices proporciona una solución eficaz para aquellas aplicaciones que requieren tiempo real.

Así pues, se concluye que la dinámica multicuerpo en tiempo real puede contribuir notablemente al desarrollo del sector automovilístico mediante su aplicación en la simulación y el control de automóviles. Y ello en tanto mayor medida en cuanto los métodos utilizados permitan simular modelos más detallados y maniobras más exigentes.

9.2. Futuras Líneas de Investigación.

Disponer de un vehículo real instrumentado proporciona numerosas posibilidades de estudio y experimentación, en un campo que actualmente vive un gran desarrollo debido a la búsqueda de mejoras en los sistemas de seguridad de los vehículos, tanto activa como pasiva. Algunas de las líneas más inmediatas, a partir del estado del vehículo, serían las siguientes:

- Como se ha visto, la conducción autónoma del vehículo se ha quedado sin completar. Para ello basta con la puesta a punto del algoritmo de actuación sobre los motores. Esto puede implicar numerosas modificaciones que afecten a distintas partes de este trabajo: puede suponer un rediseño de la actuación, de la comunicación entre el control y el vehículo, obligar al rediseño de los controladores para adaptarse a las limitaciones que puedan surgir,

etc. Por otro lado: en el diseño del vehículo virtual no se ha tenido en cuenta el comportamiento de los actuadores: quizá haya que incluir la modelización de los mismos y de su control.

- Otro paso evidente es la validación del modelo virtual del prototipo y de los controladores. Una diferencia de comportamiento entre el vehículo real y el virtual al realizar las maniobras descritas puede deberse a deficiencias del modelo, o a que en el diseño de los controladores se hayan pasado por alto algunas características y limitaciones de los actuadores.
- Un campo de interés que se ha incoado en este trabajo es el de la determinación del movimiento del vehículo. A partir de los buenos resultados obtenidos por el algoritmo, se puede estudiar su mejora, añadiendo nuevos sensores e integrando la señal de un GPS mediante filtros de Kalman. En el algoritmo desarrollado se parte de la suposición de que el vehículo discurre sobre un suelo completamente plano y, por este motivo, sólo se mide el ángulo girado alrededor del eje vertical. Aunque, efectivamente, los giros alrededor de los otros ejes puedan ser poco relevantes, sí pueden ser una fuente de error. Para ello, la solución sería instalar sensores de velocidad angular e integrar la señal tal como se ha hecho alrededor del eje z.
- Otra posibilidad que surge de forma inmediata al instrumentar la actuación sobre el volante, es la de eliminar la barra del mismo en la cadena de dirección cuando el vehículo es guiado por un conductor real. El interés de esta aplicación radica en la necesidad de representar adecuadamente la fuerza que experimenta el conductor en el volante. Disponer de un modelo del vehículo que puede ser ejecutado en tiempo real permite el cálculo de la fuerza en el volante y su recreación mediante motores de una forma realista.

El modelo del vehículo y el simulador de conducción también proporcionan un buen número de líneas a seguir. Además, el avance en estos ámbitos constituye también una forma muy gráfica de presentación del trabajo desarrollado en el Laboratorio. Dentro de este terreno, se puede hacer una nueva subdivisión. En primer lugar, se pueden citar las mejoras en el simulador de conducción:

- Mejora de la inmersión del conductor en el entorno. Para ello, se puede trabajar en la simulación del sonido del vehículo, y en la mejora del efecto de visión estereoscópica. Otro aspecto es la sustitución del

volante y los pedales por unos reales. Ello supone interponer una tarjeta de adquisición y control de datos, puesto que es necesario leer la posición de los dispositivos, y simular la fuerza que el vehículo ejerce sobre ellos, especialmente sobre el volante, todo ello manteniendo el sincronismo en tiempo real.

- Un paso más es la instalación de una plataforma de cinmática paralela que permita que el conductor experimente las aceleraciones del vehículo. Para ello, es necesario desarrollar el algoritmo que transforme las aceleraciones reales del vehículo a las que la plataforma (que es más limitada) pueda desarrollar. Este algoritmo es conocido con el nombre de algoritmo *washout*.
- Inclusión en el modelo de otros elementos, como puedan ser los *bushings* de la suspensión. La principal dificultad para este apartado recae en la necesidad de conocer el comportamiento de estos dispositivos, que es altamente no lineal. La rigidez que introducen los *bushings* en la simulación, puede suponer un desafío para la consecución de tiempo real.
- El simulador constituye un buen banco de pruebas para formulaciones dinámicas. En este apartado conviene no perder de vista el interés de aquellas formulaciones que permitan reducir el paso de tiempo (es habitual en los simuladores que éste sea de 1 ms) y que sean, a su vez, suficientemente eficientes como para seguir permitiendo que un mismo ordenador, sin características especiales, pueda encargarse de la gestión de todos los componentes del simulador (dinámica del vehículo, visualización del entorno, etc.).
- Dotar al simulador de una interfaz de comunicación con el usuario que le permita seleccionar los parámetros de la simulación: entorno gráfico, eventos atmosféricos, uso de algoritmos de control.

Finalmente, otra línea clara de desarrollo está constituida por todo lo relativo a las aplicaciones de control. En este apartado, las posibilidades más relevantes serían:

- La herramienta desarrollada de diseño de controladores permite probar otras técnicas de control, tanto del vehículo en su totalidad, como de control parcial del mismo. Sin embargo, en su forma actual, la herramienta requiere la participación del analista para ajustar los controladores, basándose en los resultados de simulaciones, en un

proceso iterativo. El objetivo es sustituir al analista por un procedimiento automático. En este sentido, son de especial interés las técnicas de control adaptativo, redes neuronales, algoritmos genéticos, o aprendizaje dirigido, utilizando la maniobra llevada a cabo por el conductor humano en el simulador como referencia para el ajuste del controlador.

- Desarrollo de algunas aplicaciones de control parcial, como pueden ser un sistema de frenado ABS, o un regulador de velocidad, que se encuentran entre los dispositivos comunes en los vehículos actuales.
- Por otro lado, los algoritmos de asistencia a la conducción están atrayendo gran atención para mejorar la maniobrabilidad y la seguridad. En esta dirección, se propone el desarrollo de un algoritmo predictivo, que saque partido al modelo altamente eficiente de vehículo disponible. El algoritmo constituye la base para la detección de pérdida de control y la determinación de las medidas correspondientes. De alguna manera, esta línea constituye un punto de convergencia de las tres áreas puesto que formulación, instrumentación y control convergen en el desarrollo del algoritmo.

Bibliografía

- ADAMS, "Running analysis in ADAMS/Car", Ref. manual, Mechanical Dynamics Incorporated, 2002.
- L. Alexander and M. Donath, "Differential GPS based control of a heavy vehicle", Technical Report of the Minnesota Department of Transportation, Minnesota, USA, 1999.
- P. Antos and J. Ambrosio, "A control strategy for vehicle trajectory tracking using multibody models", *Multibody System Dynamics*, vol. 11, pp. 365-394, 2004.
- F. Aparicio, J. Páez, F. Moreno, F. Jiménez and A. López, "Discussion of a new adaptive speed control system incorporating the geometric characteristics of the roadway", *International Journal of Vehicle Autonomous Systems*, vol. 3, n° 1, pp. 47-64, 2005.
- M. Arnold, B. Burgermeister and A. Eichberger, "Linearly implicit time integration methods in real-time applications: DAEs and stiff ODEs", *Multibody System Dynamics*, vol. 17, in press, 2007.
- H. Bae and J.Ch. Gerdes, "Parameter estimation and command modification for longitudinal control of heavy vehicles," Proc. Of the Int. Symp. Advanced Vehicle Control, Ann Arbor, MI, USA, 2000.
- E. Bakker and H.B. Pacejka, "A new tire model with an application in vehicle dynamics studies", *SAE Transactions: Journal of Passenger Cars*, vol. 98, pp. 451-439, 1989.

- P. Barber and N. Clarke, "Advanced collision warning systems," in *Proceedings of the 1998 IEEE Colloquium on Industrial Automation and Control: Application in the Automotive Industry*, London, U.K., pp. 2/1–2/9, 1998.
- E. Bayo and R. Ledesma, "Augmented Lagrangian and mass-orthogonal projection methods for constrained multibody dynamics", *Nonlinear Dynamics*, vol. 9, n° 1-2, pp. 113-130, 1996.
- E. Bertolazzi, F. Biral and M. Da Lio, "Real-time motion planning for multibody systems", *Multibody System Dynamics*, vol. 17, in press, 2007.
- M. Bertozzi and A. Broggi, "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection", *IEEE Transactions on Image Processing*, vol. 7, n° 1, pp. 62-81.1998.
- M. Bertozzi, A. Broggi and A. Fascioli. "Vision-based intelligent vehicles: State of the art and perspectives", *Robotics and Automation Systems*, vol. 32, pp. 1-16. 2000.
- D.M. Bevy, "Global Positioning System (GPS): a low-cost velocity sensor for correcting inertial sensor errors on ground vehicles", *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, n° 2, pp. 255-264, 2004.
- D.M. Bevy, J.Ch. Gerdes and Ch. Wilson "The use of GPS based velocity measurements for measurement of sideslip and wheel slip", *Vehicle System Dynamics*, vol. 38, pp. 127-147, 2002.
- B.L. Boada, M.J.L. Boada and V. Díaz, "Fuzzy-logic applied to yaw moment control for vehicle stability", *Vehicle System Dynamics*, vol. 43, n° 10, pp. 753-770, 2005.
- A. Bose and P. Ioannou, "Evaluation of the environmental effects of intelligent cruise control vehicles", *Transportation Research Record*, vol. 1774, pp. 90–97, 2001.
- D. Braid, A. Broggi and G. Schmiedel, "The TerraMax autonomous vehicle", *Journal of Field Robotics*, vol. 23, n° 9, pp. 693-708, 2006.

- A. Brazález, "Simulador de Metro Bilbao. Modelos matemáticos avanzados para la mejora de la formación de conductores", Tesis doctoral Universidad de Navarra, 2002.
- K. E. Brenan, S. L. Campbell and L. R. Petzold, *Numerical solution of initial-value problems in differential-algebraic equations*, Elsevier Science Publishing, 1989.
- A. Broggi, M. Bertozzi, A. Fascioli and G. Conte, *Automatic vehicle guidance: the experience of the ARGO autonomous vehicle*, World Scientific, 1999.
- A. Broggi, S. Nichele, M. Bertozzi and A. Fascioli, "Stereo vision-based vehicle detection", *Proceedings of the IEEE Intelligent Vehicles Symposium*, Detroit, Texas, USA, pp. 39-44, 2000.
- S. Choi and K. Hedrick, "Vehicle longitudinal control using an adaptive observer for automated highway systems," *Proceedings of American Control Conference*, Seattle, WA, USA, pp. 3106–3110, 1995.
- S. Choi and K. Hedrick, "Comparison of engine models and distance sensors for vehicle longitudinal control: Analysis and test results in PATH," *Proceedings of the Annual Meeting of ITS America*, Washington, DC, USA, pp. 61–68, 1995.
- P. Chakroborty and S. Kikuchi, "Evaluation of the general motors based car-following models and a proposed fuzzy inference model", *Transportation Research, Part C, Emerging Technologies*, vol. 7, pp. 209–235, 1999.
- C. Chan and H. Tan, "Feasibility analysis of steering control as a driver-assistance function in collision situations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 2, pp. 1–9, 2001.
- C.K. Chen and Y.C. Wang, "Fuzzy control for the antilock brake system", *Proceedings of the 1996 Asian Fuzzy Systems Symposium*, Kenting, Taiwan, pp. 67-72, 1996.
- D. Cho and J.K. Hedrick, "Automotive powertrain modeling for control", *Journal of Dynamic Systems, Measurement, and Control*, vol. 111, pp. 568–576, 1989.

- DADS, "Dynamic Analysis and Design System", Ref. manual, DADS 9.6, CADSI, 2001.
- R. Deering and D. Viano, "Critical success factors for crash avoidance countermeasure implementation", SAE Paper 94CO25, 1994.
- E. D. Dickmanns and A. Zapp, "A curvature-based scheme for improving road vehicle guidance by computer vision", *Proc. SPIE on Mobile Robots*, vol. 727, 1986.
- E. D. Dickmanns, R. Behringer, D. Dickmanns, T. Hildebrant, M. Mauer, F. Thomanek, and J. Shielhlen, "The seeing passenger car 'VaMoRs-P'", *Proceedings of International Symposium on Intelligent Vehicles*, Paris, France, 1994.
- E. D. Dickmanns. "Vehicles capable of dynamic vision", *International Joint Conference on Artificial Intelligence (IJCA-97)*, Nagoya, Japan, pp. 23-29, 1997.
- D. Dopico, "Formulaciones semi-recursivas y de penalización para la dinámica en tiempo real de sistemas multicuerpo", Tesis doctoral, Universidad de La Coruña, 2004.
- M. Druzhinina, A. Stefanopoulou and L. Moklegaard, "Compression braking control for heavy-duty vehicles", *2000 American Control Conf.*, Chicago, IL, USA, 2000.
- M. Druzhinina, L. Moklegaard and A. Stefanopoulou, "Speed gradient approach to longitudinal control of heavy-duty vehicles equipped with variable compression brake", *IEEE Transactions on Control Systems Technology*, vol. 10, pp. 209–220, 2002.
- M. Druzhinina, A. Stefanopoulou and L. Moklegaard, "Adaptive continuously variable compression braking control for heavy-duty vehicles", *Journal of Dynamic Systems, Measurement and Control*, vol. 124, pp. 406–414, 2002.
- M. Druzhinina and A. Stefanopoulou, "Speed control experiments for commercial heavy vehicles with coordinated friction and engine compression brakes", *Proc. American Control Conf.*, Anchorage, AK, USA, 2002.

- H. Dugoff, P.S. Fancher and L. Segel, "An analysis of tire traction properties and their influence on vehicle dynamic performance", SAE Paper 700377, FISITA/SAE, 1970.
- G. Ellis, *Control system design guide*, Academic Press, 1991.
- A. Eskandarian and S. Thiriez, "Collision avoidance using a cerebellar model arithmetic computer neural network", *Computer-Aided Civil Infrastructure Engineering*, vol. 13, pp. 303–314, 1998.
- J. D. Foley, A. van Dam, S.K. Feiner and J.F. Hughes, *Computer graphics. Principles and practices*, Addison-Wesley, 1997.
- R. French, Y. Noguchi and K. Sakamoto, "International competitiveness in IVHS: Europe, Japan and the United States", *Proceedings of the 1994 Vehicle Navigation and Information Systems Conference*, Yokohama, Japan, pp. 525-530, 1994.
- J.S. Freeman, G.W. Papelis, Y. Papelis, A. Tayyab, R.A. Romano and J.G. Kuhl, "The Iowa driving simulator: An implementation and application overview", SAE Paper 950174, 1995.
- R. Frezza, A. Saccon, D. Minen and C. Ortmann, "Smart Driver: a research project for closed loop vehicle simulation in MSC.ADAMS", *Multi-body Dynamics, Monitoring and Simulation Techniques III*, Rahnejat H. and Rothberg S. (eds.), 2004.
- J. García de Jalón, M. A. Serna and R. Avilés, "Computer method for kinematic analysis of lower-pair mechanisms. Part I: velocities and accelerations", *Mechanism and Machine Theory*, vol. 16, nº 5, pp. 543-556, 1981.
- J. García de Jalón, M. A. Serna and R. Avilés, "Computer method for kinematic analysis of lower-pair mechanisms. Part II: position problems", *Mechanism and Machine Theory*, vol. 16, nº 5, pp. 557-566, 1981.
- J. García de Jalón, J. Unda and A. Avello, "Natural coordinates for the computer-analysis of multibody systems", *Computer Methods in Applied Mechanics and Engineering*, vol. 56, nº 3, pp. 309-327, 1986.

- J. García de Jalón, J. Unda, A. Avello and J. M. Jimenez, "Dynamic analysis of 3-dimensional mechanisms in natural coordinates", *Journal of Mechanisms, Transmissions and Automation in Design*, vol. 109, n° 4, pp. 460-465, 1987.
- J. García de Jalón and E. Bayo, *Kinematic and dynamic simulation of multibody systems: The real-time challenge*, Springer-Verlag, 1994.
- O. Gehring and H. Fritz, "Practical results of a longitudinal control concept for truck platooning with vehicle to vehicle communication", *Proceedings of IEEE Conference on Intelligent Transportation Systems*, Boston, MA, USA, pp. 117–122, 1997.
- J.Ch. Gerdes and J.K. Hedrick, "Vehicle speed and spacing control via coordinated throttle and brake actuation", *Control Engineering Practice*, vol. 5, pp. 1607–1614, 1997.
- S. Germann and R. Isermann, "Nonlinear distance and cruise control for passenger cars", in *Proceedings of American Control Conference*, Seattle, WA, USA, pp. 3081–3085, 1995.
- Th.D. Gillespie, *Fundamentals of vehicle dynamics*, Society of Automotive Engineers, Inc., 1992.
- U. Gottwick, "The electrohydraulic brake revolutionizes the passenger car brake systems", *VDI Berichte*, (1494), pp. 287-303, 1999.
- R. Gutiérrez, "Cálculo de tensiones en componentes de sistemas móviles mediante dinámica de sistemas multicuerpo flexibles", Tesis doctoral, Universidad de La Coruña, 2003.
- D.A. Haessig and B. Friedland, "On the modeling and simulation of friction", *Journal of Dynamic Systems, Measurement, and Control*, vol. 113, pp. 354-362, 1991.
- Y. Hattori, E. Ono and S. Hosoe, "Optimum vehicle trajectory control for obstacle avoidance problem", *IEEE/ASME Transactions of Mechatronics*, vol. 11, n° 5, 2006.

- Y. Hayakawa, R. White, T. Kimura, and G. Naito, "Driver-compatible steering system for wide speed-range path following", *IEEE/ASME Transactions on Mechatronics*, vol. 9, pp. 544-552, 2004.
- R.G. Hebden, C. Edwards, and S.K. Spurgeon, "Automotive steering control in a split- μ manoeuvre using an observer-based sliding mode controller", *Vehicle System Dynamics*, vol. 41, n° 3, pp. 181-202, 2004.
- Ph. Heinzl, P. Lugner and M. Plöchl, "Stability control of a passenger car by combined additional steering and unilateral braking", *Supplement to International Journal on Vehicle System Dynamics*, vol. 37, 2003.
- G.J. Heydinger, M.K. Salani, W.R. Garrott and P.A. Grygier, "Vehicle dynamics modelling for the National Advance Driving Simulator", 2002.
- A. Hoess, W. Hosp and H. Rauner, "Comparison of radar and infrared distance sensors for intelligent cruise control systems", *Proceedings of SPIE - The International Society for Optical Engineering* 2511, pp. 208-216, 1995.
- R. Holve, P. Protzel and K. Naab, "Generating fuzzy rules for the acceleration control of an adaptive cruise control system", *Proceedings of Biennial Conference of North American Fuzzy Information Processing Society*, Berkeley, CA, USA, pp. 451-455, 1996.
- H. Holzmann, C. Halfmann, S. Germann, M. Würtenberger and R. Isermann, "Longitudinal and lateral control and supervision of autonomous intelligent vehicles," *Control Engineering Practice*, vol. 5, pp. 1599-1605, 1997.
- A. Hopgood, *Intelligent systems for engineers and scientists*, CRC Press, 2001.
- J. Huang and H.-S. Tan, "A low-order DGPS-based vehicle positioning system under urban environment", *IEEE/ASME Transactions of Mechatronics*, vol. 11, n° 5, 2006.
- F. Ibrahim, N. Al-Holou, T. Pilutti and M. Paulik, "Accurate INS position solution during GPS outages or degraded GPS", *Vehicle System Dynamics*, vol. 43, n° 9, pp. 633-652, 2005.

- S. Inagaki, I. Kshiro, and M. Yamamoto, "Analysis on vehicle stability in critical cornering using phase-plane method", *Proceedings of AVEC '94*, Tsukuba, Japan, 1994.
- P. Ioannou and H. Raza, "Vehicle following control design for automated highway systems", *Proceedings of 1997 IEEE Vehicular Technology Conference*, Phoenix, AZ, USA, pp. 904–908, 1997.
- P. Ioannou and H. Raza, "Vehicle following control design for automated highway systems", *IEEE Control System Magazine*, vol. 6, pp. 43–60, 1996.
- P. Ioannou and Z. Xu, "Throttle and brake control systems for automatic vehicle following", *PATH Research Rep. UCB-ITS-PRR-94-10*, 1994.
- H. Jula, E. Kosmatopoulos and P. Ioannou, "Collision avoidance analysis for lane changing and merging", *IEEE Transactions on Vehicular Technology*, vol. 49, pp. 2295–2308, 2000.
- I. Kageyama and H.B. Pacejka, "On a new driver model with fuzzy control", *Supplement to International Journal on Vehicle System Dynamics*, vol. 20, pp. 314-324, 1991.
- R.E. Kalman, "A new approach to linear filtering and prediction problems", *Journal of Basic Engineering*, vol. 82, pp. 35-45, 1960.
- S. Kato, S. Tsugawa, K. Tokuda, T. Matsui and H. Fujiri, "Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications", *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 3, pp. 155–161, 2002.
- M. Kawai, "Collision avoidance technologies", SAE Paper 94CO38, 1994.
- U. Kiencke and L. Nielsen, *Automotive control systems for engine, driveline and vehicle*, Springer-Verlag, Berlín, 2000.
- J. Kim, B.Y. Jeong, and D. Cho, "AUTOTOOL, a PC-based object-oriented automotive powertrain simulation tool", *Proceedings of the IEEE Conference on Intelligent Transportation Systems*, Boston, MA, USA, 753–758, 1997.

- M. Kuwahara, "Intelligent Transport Systems - Overview and Japanese systems", *Proceedings on ASEAN Infrastructure Planning & Management Workshop*, Bangkok, Thailand, pp.125-135, 1998.
- R. Labayrade, C. Royere, D. Gruyer, and D. Aubert, "Cooperative fusion for multi-obstacles detection with use of stereovision and laser scanner", *Proceedings of ICAR*, Coimbra, Portugal, pp. 1538–1543, 2003.
- A. Lawrence, *Modern inertial technology. Navigation, guidance, and control*, Springer-Verlag, Berlin, 1998.
- D. Lee, R. Attias, A. Puri, R. Sengupta, S. Tripakis and P. Varaiya, "A wireless token ring protocol for intelligent transportation systems", *Proceedings of the 4th IEEE International Conference on Intelligent Transportation Systems*, Oakland, USA, pp. 1152-1157, 2001.
- C. Liang and H. Peng, "Design and simulations of traffic-friendly adaptive cruise control algorithm", *Proceedings of ASME International Mechanical Engineering Congress and Exposition*, Anaheim, CA, USA, pp. 713–719, 1998.
- C. Liang and H. Peng, "Optimal adaptive cruise control with guaranteed string stability," *Vehicle System Dynamics*, vol. 31, pp. 313–330, 1999.
- C. Liang and H. Peng, "String stability analysis of adaptive cruise controlled vehicles", *International Journal for Mechanical Systems, Machine Elements and Manufacturing*, vol. 43, pp. 671–677, 2000.
- M. Liubakka, D. Rhode, J. Winkelmann and P. Kokotovic, "Adaptive automotive speed control", *IEEE Transactions on Automatic Control*, vol. 38, pp. 146–156, 1993.
- G. Lu and M. Tomizuka, "LIDAR sensing for vehicle lateral guidance: algorithm and experimental study", *IEEE/ASME Transactions of Mechatronics*, vol. 11, n° 6, 2006.
- X.-Y. Lu and J.K. Hedrick, "Real time estimation and compensation for tyre slip in longitudinal control", *17th IAVSD Symposium*, Copenhagen, 2001.

- X.-Y. Lu and J.K. Hedrick, "A panoramic view of fault management for longitudinal control of automated vehicle platooning", *Proc. of 5th Int. Symposium IMECE2002-32106*, New Orleans, LA, USA, 2002.
- X.-Y Lu, J.K. Hedrick and M. Drew, "ACC/CACC – Control design, stability and robust performance", *Proceedings of US Control Conference*, 8-10 May., 2002, Anchorage, AK, USA, pp. 4327-4332.
- X.-Y. Lu and J.K. Hedrick, "Impact of combined longitudinal, lateral and vertical control on autonomous road vehicle design", *International Journal of Vehicle Autonomous Systems*, vol. 2, pp. 40-70, 2004.
- X.-Y Lu, H. Tan, S. Shladover and K. Hedrick, "Nonlinear longitudinal controller implementation and comparison for automated cars", *Journal of Dynamics Systems, Measurement, and Control*, vol. 123, pp. 161–167, 2001.
- P. Lugner, *Horizontal Motion of Automobiles. Part 1: Theoretical and Practical Investigations. Dynamics of High-Speed Vehicles*, CISM 274, Springer Verlag, 1982.
- P. Lugner, "Theoretische Grenzen und Möglichkeiten der Fahrdynamik von PKW mit Allradantrieb", *Symposium Allradantrieb beim PKW, Reihe 12: Verkehrstechnik/Fahrzeugtechnik*, Vol. 81, 1986.
- P. Lugner and M. Plösch, "Modelling in vehicle dynamics of automobiles", *ZAMM Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 84, n° 4, pp. 219-236, 2004.
- P. Luque, D. Álvarez y C. Vera, *Ingeniería del automóvil. Sistemas y comportamiento dinámico*, Thomson, Madrid, 2004.
- O. Ma, "Contact dynamics modelling for the simulation of the space station manipulators handling payloads", *IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1252-1258, 1995.
- C.C. Macadam, "Understanding and modelling the human driver", *Vehicle System Dynamics*, vol. 40, pp. 101-134, 2003.

- E.H. Mamdani and S. Asslian, "An experiment in linguistic synthesis with fuzzy logic controller", *International Journal of Man-Machine Studies*, vol. 7, pp. 1-13, 1975.
- MAN-Project. Automatic Speed and Headway Distance. Control System for Buses and Trucks. http://www.unibwmuenchen.de/campus/LRT/LRT13/english/man_projekt.html.
- S. Matsumoto, H. Yamaguchi, H. Inoue and Y. Yasuno, "Braking force distribution control for improved vehicle dynamics", *Proceedings of AVEC '92*, Yokohama, Japan, 1992.
- R. Mayr, "Nonlinear vehicle distance control in longitudinal direction," *International Journal of Systems Science*, vol. 27, pp. 705–712, 1996.
- M. Mitschke, *Dynamik von Kraftfahrzeugen*, Band A: Antrieb und Bremsung (Springer Verlag, 1995); Band B: Schwingungen (Springer Verlag, 1997); Band C: Fahrverhalten (Springer Verlag, 1990).
- J. E. Naranjo, "Sistema de conducción automática de vehículos basado en lógica borrosa y sistemas globales de posicionamiento por satélite: programa Autopía", Tesis doctoral, Universidad Politécnica de Madrid, 2005
- J. E. Naranjo, C. González, R. García y T. de Pedro, "ACC+Stop&Go maneuvers with throttle and brake fuzzy control", *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, nº 2, pp. 213-225, 2006.
- J. Neumann, A. Manz, T. Ford and O. Mulyk, "Test results from a new 2 cm real time kinematics GPS positioning system", *Proceedings of the Institute of Navigation GPS Conference*, Kansas City, KS, USA, 1996.
- K. Oda, H. Takeuchi, M. Tsujii and M. Ohba, "Practical estimator for self-tuning automotive cruise control" *Proceedings of American Control Conference*, Boston, MA, USA, 1991, pp. 2066–2071.
- H. Peng and M. Tomizuka, "Preview control for vehicle lateral guidance in highway automation", *Journal of Dynamic Systems, Measurement, and Control*, vol. 115, pp. 679–686, 1993.

- M. Persson, F. Botling, E. Hesslow and R. Johansson, "Stop and go controller for adaptive cruise control", *Proceedings of the 1999 IEEE International Conference on Control Applications and IEEE International Symposium of Computer-Aided Control System Design*, Kohala Coast, HI, USA, pp. 1692–1697, 1999.
- H. Pham, M. Tomizuka and J.K. Hedrick, "Integrated manoeuvring control for automated highway systems based on a magnetic reference/sensing system", *PATH Research Rep. UCB-ITS-PRR-97-28*, 1997.
- D. A. Pomerleau, *Neural network perception for mobile robot guidance*, Kluwer Academic Publishers, 1993.
- D. A. Pomerleau and T. M. Jockem, "Rapidly adapting machine vision for automated vehicle steering", *IEEE Expert-Intelligent Systems and their Applications*, vol. 11, n° 2, pp. 19-27, 1996.
- W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, New York, 1992.
- M.T. Raharijaona, M.G. Duc and M.S. Mammar, "Linear parameter-varying control and H-infinity synthesis dedicated to lateral driving assistance", *Transactions of the Japan Society of Mechanical Engineers, Part C*, vol. 72, n° 7, pp. 2056-2064, 2006.
- R. Rajamani, S. Choi, B. Law, J.K. Hedrick, R. Prohaska and P. Kretz, "Design and experimental implementation of longitudinal control for a platoon of automated vehicles", *Journal of Dynamic Systems, Measurement, and Control*, vol. 122, pp. 470-476, 2000.
- R. Rajamani, H. Tan, B. Law and W. Zhang, "Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons", *IEEE Transactions on Control Systems Technology*, vol. 8, pp. 695–708, 2000.
- H. Raza, Z. Xu, B. Yang and P. Ioannou, "Modeling and control design for a computer-controlled brake system", *IEEE Transactions on Control Systems Technology*, vol. 5, pp. 279–296, 1997.

- RecurDyn, "User's Guide", Ref. manual, Functionbay Inc., 2004.
- G. Reymond, A. Heidet, M. Canry and A. Kemeny, "Validation of Renault's dynamic simulator for Adaptive Cruise Control experiments", *Proceedings of the Driving Simulation Conference*, Paris, France, pp. 181-191, 2000.
- J. Ryu and J.Ch. Gerdes, "Integrating inertial sensors with global positioning system (GPS) for vehicle dynamics control", *Journal of Dynamic Systems, Measurement, and Control*, vol. 126, pp. 243-254, 2004.
- SAE Standard. *Recommended practice for a serial control and communications vehicle network*, J1939, SAE International, Warrendale, PA, USA, 2005.
- P. Seiler, B. Song and J.K. Hedrick, "Development of a Collision Avoidance System", SAE Paper 98PC417, 1998.
- M. A. Serna, R. Aviles and J. García de Jalón, "Dynamic analysis of plane mechanisms with lower pairs in basic coordinates", *Mechanism and Machine Theory*, vol. 17, no. 6, pp. 397-403, 1982.
- W. Schiehlen and A. Fritz, "Nonlinear cruise control concepts for vehicles in convoy," *Supplement to International Journal on Vehicle System Dynamics*, vol. 33, pp. 256–269, 1999.
- R.S. Sharp, D. Casanova and P. Symonds, "A mathematical model for driver steering control, with design, tuning and performance results", *Vehicle System Dynamics*, vol. 33, pp. 289-326, 2000.
- Y. Shibahata, K. Shimada and T. Tomari, "The improvement of vehicle maneuverability by direct yaw moment control", *Proceedings of AVEC '92*, Yokohama, Japan, 1992.
- T. Shiiba and Y. Suda, "Evaluation of driver's behaviour with multibody-based driving simulator", *Multibody System Dynamics*, vol. 17, in press, 2007.
- M. Shimakage, H. Kawazoe, O. Sadano and T. Murakami, "Design of lane-keeping control with steering torque Input for a lane-keeping support system", SAE Paper 2001-01-0480, 2001.

- S. Shladover, "Review of the state of development of Advanced Vehicle Control Systems (AVCS)", *Vehicle System Dynamics*, vol. 24, pp. 551-595, 1995.
- S. Shladover, "Automated vehicles for highway operations (automated highway systems)", *Proc. ImechE, Part I: J. Systems and Control Engineering*, vol. 219, pp. 53-75, 2005.
- D. Shreiner, M. Woo and J. Neider, *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.4*, Addison-Wesley Professional, Boston, 2003.
- J.E. Shigley and C.R. Mischke, *Mechanical and engineering design*, McGraw-Hill, 6th edition, Singapore, 2001.
- SIMPACK, "Simpack Reference Guide", Release 8.6, Intec GmbH, 2003.
- G. Snook, *Real Time 3D Terrain Engines using C++ and DirectX*, Charles River Media, Hingham, MA, USA, 2003.
- H.-C. Sohn, K.-S. Hong and J.K. Hedrick, "Semiactive control of the MacPherson suspension system: Hardware-in-the-loop simulations", *Proceedings of IEEE International Conference on Control Applications*, Anchorage, AK, USA, pp. 982-987, 2000.
- M.Á. Sotelo, "Sistema de navegación global aplicado al guiado de un vehículo autónomo terrestre en entornos exteriores parcialmente conocidos", Tesis doctoral, Universidad de Alcalá, 2001.
- M. Sugeno and G.T. Kang, "Structure Identification of Fuzzy Model", *Fuzzy Sets and Systems*, vol. 28, n° 1, pp. 15-33, 1988.
- D. Swaroop, J.K. Hedrick, C. Chien and P. Ioannou, "Comparison of spacing and headway control laws for automatically controlled vehicles", *Vehicle System Dynamics*, vol. 23, pp 597-625, 1994.
- D. Swaroop and R. Huandra, "Intelligent cruise control design based on a traffic flow stability", *Vehicle System Dynamics*, vol. 30, pp. 319-344, 1998.

- D. Swaroop and K. Rajagopal, "Intelligent cruise control systems and traffic flow stability", *American Society of Mechanical Engineers, Dynamic Systems and Control Division (Publication) DSC 67*, pp. 373-380, 1999.
- D. Swaroop and S. Yoon, "Integrated lateral and longitudinal vehicle control for an emergency lane change maneuver design", *International Journal of Vehicle Design*, vol. 21, no. 2, pp. 161–174, 1999.
- F. Tahami, R. Kazemi and S. Farhanghi, "A fuzzy logic direct yaw-moment control system for all-wheel-drive electric vehicles", *Vehicle System Dynamics*, vol. 41, n° 3, pp. 203-221, 2004.
- M. Tai, P. Hingwe and M. Tomizuka, 2004, "Modeling and control of steering system of heavy vehicles for automated highway systems", *IEEE/ASME Transactions on Mechatronics*, vol. 9, pp. 609-618, 2004.
- T. Takahama, T. Kimura, E. Iwasaki and G. Naitou, "Robust vehicle detection for collision avoidance systems by using a LIDAR sensor and a vision sensor," *Proceedings of 10th World Congress and Exhibition of Intelligent Transportations Systems*, Madrid, Spain, 2003.
- H. Tan, R. Rajamani and W. Zhang, "Demonstration of an Automated Highway Platoon System", *Proceedings of 1998 American Control Conference*, Philadelphia, PA, USA, pp. 1823-1827, 1998.
- M. Tomizuka, J.K. Hedrick and H. Pham, "Integrated manoeuvring control for automated highway systems based on a magnetic reference/sensing system", *PATH Research Rep. UCB-ITS-PRR-95-12*, 1995.
- A. Touran, M. Brackstone and M. McDonald, "A collision model for safety evaluation of autonomous intelligent cruise control", *Accident Analysis and Prevention*, vol. 31, pp. 567–578, 1999.
- S. Tsugawa, T. Yatabe, T. Hirose and S. Matsumoto, "An automobile with artificial intelligence," *Proceedings of the 6th International Joint Conference on Artificial Intelligence (IJCAI)*, Tokyo, Japan, pp. 893–895, 1979.

- O. Vaculin, W.R. Krüger and M. Valasek, "Overview of coupling of multibody and control engineering tools", *Vehicle System Dynamics*, vol. 41, pp. 415-429, 2004.
- A. Vahidi, M. Druzhinina, A. Stefanopoulou, and H. Peng, "Simultaneous mass and time-varying grade estimation for heavy-duty vehicles," *Proceedings of American Control Conference*, Denver, CO, USA, pp. 4951–4956, 2003.
- A. Vahidi and A. Eskandarian, "Research advances in intelligent collision avoidance and adaptive cruise control", *IEEE Transactions on Intelligent Transportation Systems*, vol. 4, n° 3, pp. 143-153, 2003.
- A.T. Van Zanten, R. Erhardt and G. Pfaff, "VDC, the vehicle dynamics control system of Bosch", SAE Paper 950759, 1995
- P. Varaiya, "Smart cars on smart roads: problems of control", *IEEE Transactions on Automatic Control*, vol. 38, n° 2, pp. 195-207, 1993.
- P. Venhovens, K. Naab and B. Adiprastito, "Stop and go cruise control", *International Journal of Automotive Technology*, vol. 1, pp. 61–69, 2000.
- B. Wilson, "How soon to brake and how hard to brake: Unified analysis of the envelope of opportunity for rear-end collision warning", *Proc. International Technical Conference on Enhanced Safety of Vehicles*, Amsterdam, The Netherlands, 2001.
- J. Woll, "Radar based collision warning system", SAE Paper 94CO36, 1994.
- J. Woll, "Radar based adaptive cruise control for truck applications", SAE Paper 973 184, 1997.
- D. Yanakiev and I. Kanellakopoulos, "Speed tracking and vehicle follower control design for heavy-duty vehicles", *Vehicle System Dynamics*, vol. 25, pp. 251–276, 1996.
- D. Yanakiev and I. Kanellakopoulos, "Longitudinal control of automated CHV's with significant actuator delays", *Proc. CDC*, San Diego, CA, USA, 1997.

- D. Yanakiev, J. Eyre, and I. Kanellakopoulos, "Longitudinal control of HDVs: experimental evaluation", *PATH Research Rep. MOU 293*, 1998.
- K. Yi, M. Woo, S. Kim and S. Lee, "Study on a road-adaptive CW/CA algorithm for automobiles using HiL simulations," *JSME International Journal*, vol. 42, pp. 163–170, 1999.
- K. Yi, S. Lee and Y.D. Kwon, "An investigation of intelligent cruise control laws for passenger vehicles", *Proc. Instn. Mech. Engrs., Part D: J. Automobile Engineering*, vol. 215, pp. 159-169, 2001.
- K. Yi and J. Chung, "Nonlinear brake control for vehicle CW/CA systems," *IEEE/ASME Transactions on Mechatronics*, vol. 6, pp. 17–25, 2001.
- K. Youcef-Toumi, Y. Sasage, Y. Ardini and S. Huang, "Application of time delay control to an intelligent cruise control system," *Proceedings of American Control Conference*, Chicago, IL, USA, pp. 1743–1747, 1992.
- S. Yuan, J. Hu and R. Wang, "Electronic-control system of hydromechanical transmission", *Proceedings of IEEE International Vehicle Electronics Conference*, vol. 1, pp. 447-450, 1999.
- L.A. Zadeh, "Fuzzy Sets", *Information and Control*, vol. 8, pp. 338-353, 1965.
- L.A. Zadeh, "Fuzzy Logic=Computing with Words", *IEEE Transactions on Fuzzy Systems*, vol. 4, pp. 103-111, 1996.
- Q. Zhang, "A generic fuzzy electrohydraulic steering controller for off-road vehicles", *Proc. Instn. Mech. Engrs., Part D: J. Automobile Engineering*, vol. 217, pp. 791-799, 2003.
- W. Zhang, S. Shladover, R. Hall, and T. Plocher, "A functional definition of automated highway systems", TRB Paper 940 988, 1994.
- Y. Zhang, E. Kosmatopoulos, P. Ioannou, and C. Chien, "Autonomous intelligent cruise control using front and back information for tight vehicle following maneuvers," *IEEE Transactions on Vehicular Technology*, vol. 48, pp. 319–328, 1999.

Sh. Zheng, H. Tang, Zh. Han, and Y. Zhang, "Controller design for vehicle stability enhancement", *Control Engineering Practice*, vol. 14, n° 12, pp. 1413-1421, 2006.