



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

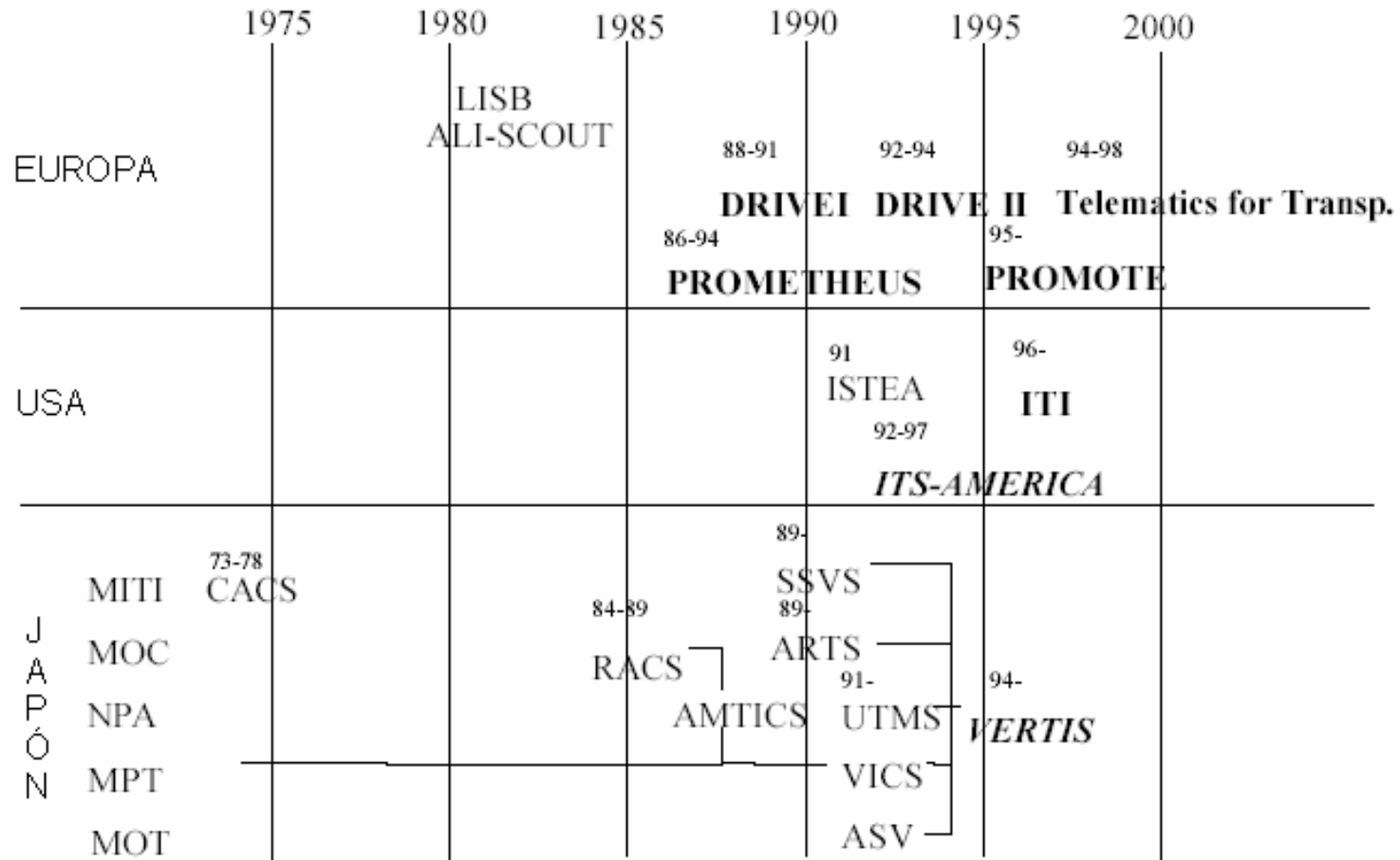
Miguel Ángel Naya Villaverde

Tesis para optar al grado de
Doctor Ingeniero Industrial

Índice

1. Estado del arte.
2. El prototipo real.
3. El modelo virtual del prototipo.
4. El simulador de conducción.
5. Módulo para desarrollo de controladores: maniobras.
6. Instrumentación.
7. Conclusiones y futuras líneas de investigación.

Proyectos internacionales



Vehículos inteligentes

- Primer vehículo autónomo construido en Japón 1977 en el proyecto CACS (Prof. Tsugawa).
- NavLab (Pomerleau)
- VaMP (Dickmanns)
- ARGO (Broggi)
- AUTOPIA (García Rosa)

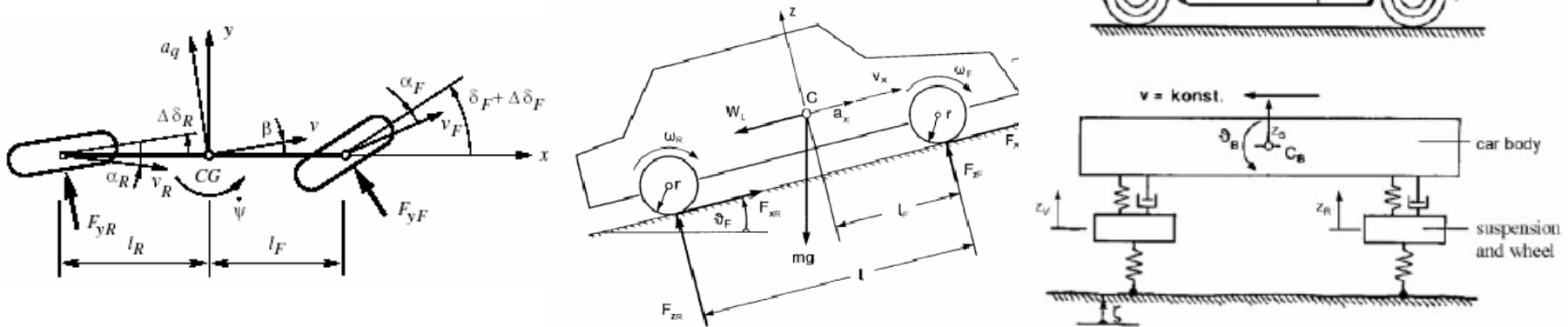


Vehículos inteligentes

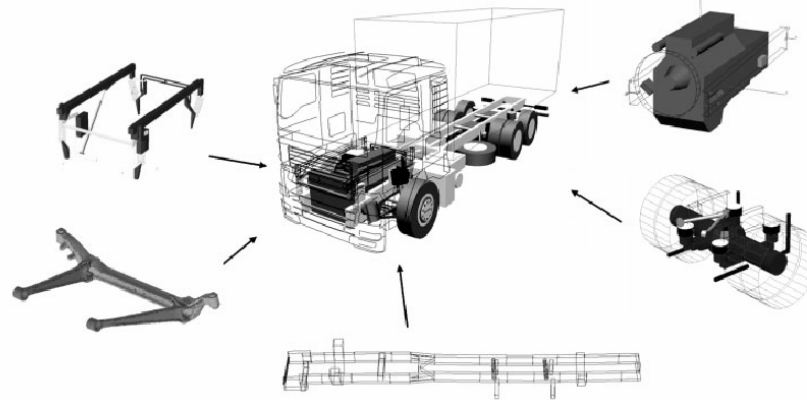
- Control Longitudinal.
 - Cruise Control (CC).
 - Adaptive Cruise Control (ACC).
 - Cooperative Adaptive Cruise Control (CACC).
 - "Stop & Go".
 - Automated Vehicle Following.
 - Sistemas de Alarma y Asistencia al Conductor.
- Control Lateral.
 - ESP
- Control Vertical.

Modelos dinámicos

- Modelos simplificados.



- Programas MBS (ADAMS, RecurDyn, SIMPACK, DADS).



Dinámica multicuerpo y control de vehículos

- Las simulaciones MBS de la dinámica de vehículos ayuda al diseño y a la evaluación de
 - Vehículos.
 - Sistemas de control de vehículos.
- Reduce la necesidad de elaborar prototipos físicos/experimentos.
- Simulación MBS aplicada a sistemas de control de vehículos:
 - Diseño de la estrategia de control.
 - Simulaciones con HIL: ajuste y evaluación del comportamiento de ECU.
 - Control óptimo: se realizan muchas simulaciones en cada paso de tiempo para minimizar el control.
 - Aplicaciones a bordo: sistemas de alarma al conductor (observador de estados)/actuación automática (diseño de trayectorias).
- Simuladores de conducción.
- Relevancia de la eficiencia de la simulación.

Aplicaciones multicuerpo

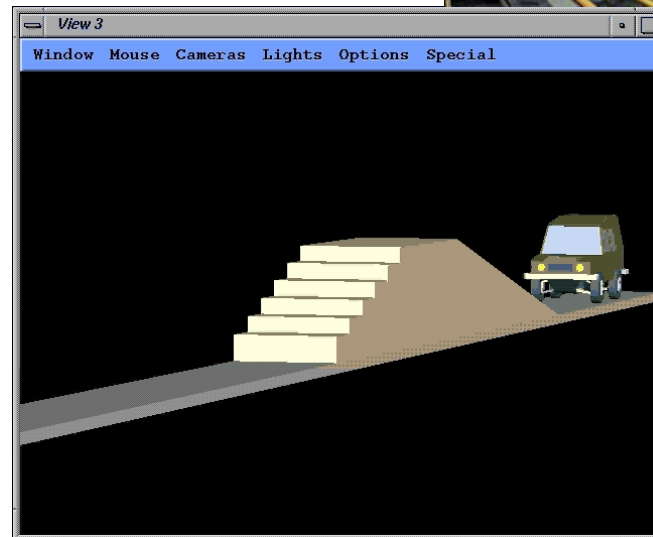
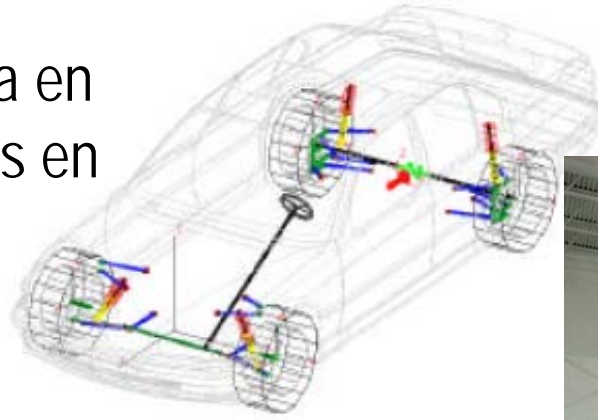
- Pasos de tiempo en la simulación de la dinámica de vehículos entre 0,5 y 2 ms o inferiores.
- Integradores explícitos o semi-implícitos de paso fijo (Euler explícito).
- ODE no *stiff*.

Pero

- Suspensiones, sistemas de dirección y/o *bushings*, conducen a ecuaciones diferenciales de tipo *stiff*.

Objetivos

- La simulación de la dinámica en tiempo real tiene gran interés en el campo de la automoción.
- Es posible emplear una formulación que:
 - Permita realizar modelos detallados de automóviles.
 - Sea eficiente: permitir simulaciones en tiempo real con Human y/o Hardware-in-the-loop.
 - Sea robusta: pueda abordar simulaciones de maniobras exigentes.



Formulación LIM

Ecuaciones del movimiento: formulación aumentada de Lagrange de índice 3.

$$\mathbf{M} \ddot{\mathbf{q}} + \Phi_{\mathbf{q}}^T \alpha \Phi + \Phi_{\mathbf{q}}^T \lambda^* = \mathbf{Q}$$

$$\lambda_{i+1}^* = \lambda_i^* + \alpha \Phi_{i+1} \quad i = 0, 1, 2, \dots$$

$$\begin{array}{l} \dot{\mathbf{q}}_{n+1} = \frac{2}{\Delta t} \mathbf{q}_{n+1} + \hat{\mathbf{q}}_n \\ \ddot{\mathbf{q}}_{n+1} = \frac{4}{\Delta t^2} \mathbf{q}_{n+1} + \hat{\ddot{\mathbf{q}}}_n \end{array} \quad \left| \quad \begin{array}{l} \hat{\mathbf{q}}_n = -\left(\frac{2}{\Delta t} \mathbf{q}_n + \dot{\mathbf{q}}_n \right) \\ \hat{\ddot{\mathbf{q}}}_n = -\left(\frac{4}{\Delta t^2} \mathbf{q}_n + \frac{4}{\Delta t} \dot{\mathbf{q}}_n + \ddot{\mathbf{q}}_n \right) \end{array} \right.$$

Integrador numérico: regla trapezoidal.

Paso de tiempo: 10 ms.

Proyecciones de velocidades y aceleraciones.

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

Formulación LIM vs aplicaciones MBS

Aplicaciones tradicionales en tiempo real:

Formulaciones **topológicas**.

Esquemas de integración **explícitos**.

Implementación realizada:

Formulación **global**.

Esquema de integración **implícito**.

Ventajas:

Fácil de implementar.

Robusto.

1. Estado del arte.
2. El prototipo real.
3. **El modelo virtual del prototipo.**
4. El simulador de conducción.
5. Módulo para desarrollo de controladores: maniobras.
6. Instrumentación.
7. Conclusiones y futuras líneas de investigación.

El prototipo real

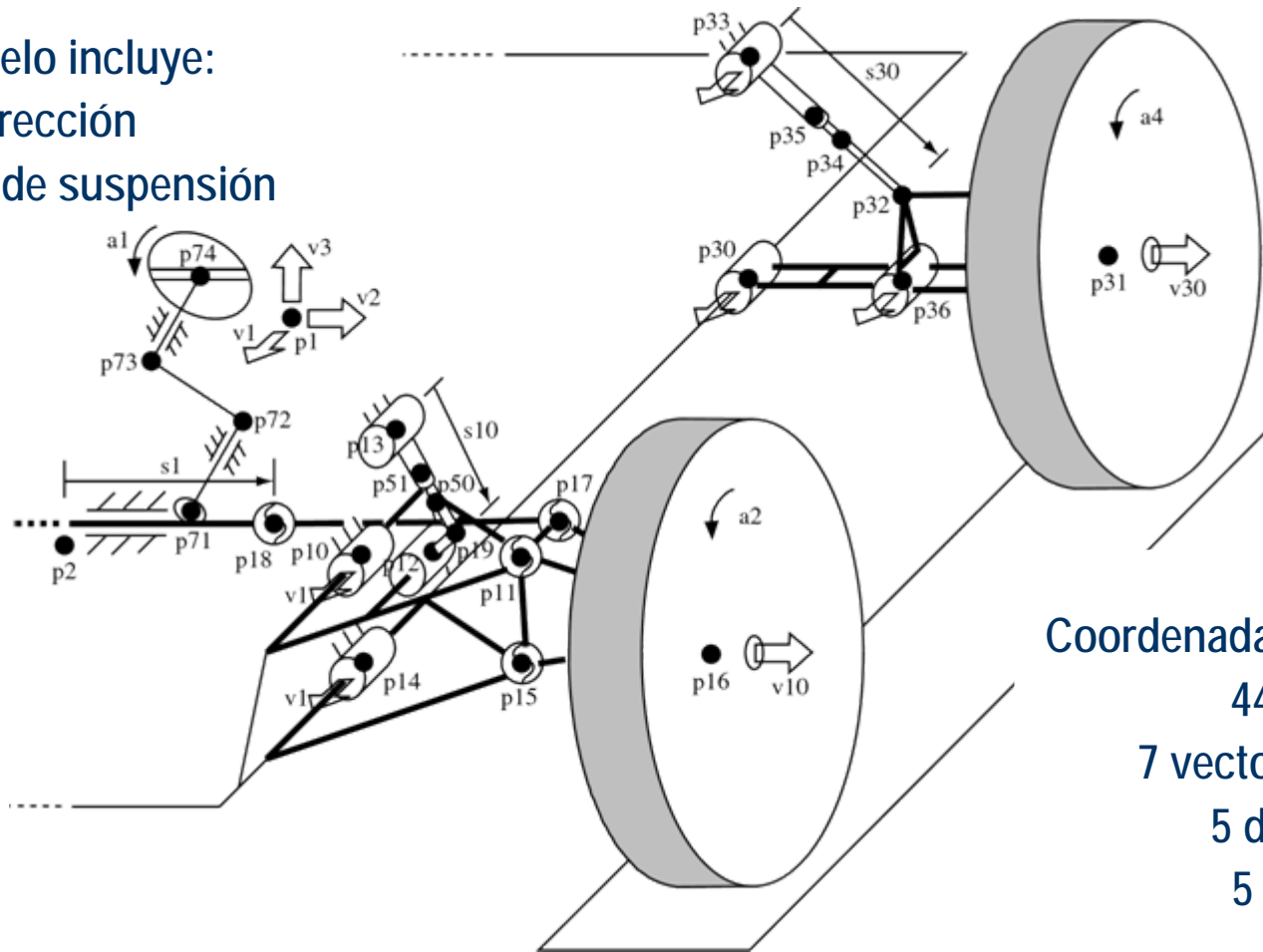


El modelo virtual del prototipo (i)

El modelo incluye:

-dirección

-sistemas de suspensión



Coordenadas Naturales (163):

44 puntos

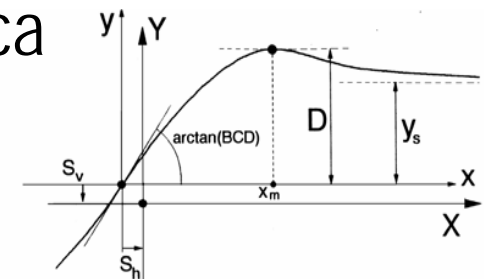
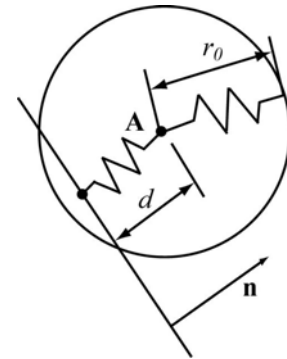
7 vectores unitarios

5 distancias

5 ángulos

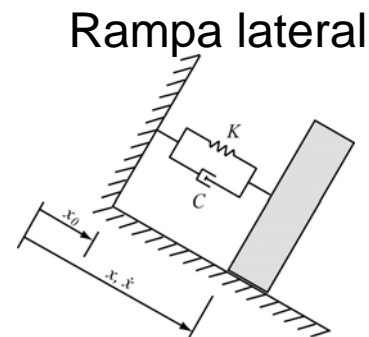
El modelo virtual del prototipo (ii)

- Algoritmo de detección del punto de contacto.
- Fuerza vertical y de compresión: muelle lineal.
- Contacto neumático-suelo: Fórmula Mágica con parámetros proporcionados por un fabricante de neumáticos.
- Peculiaridades de la fuerza lateral.



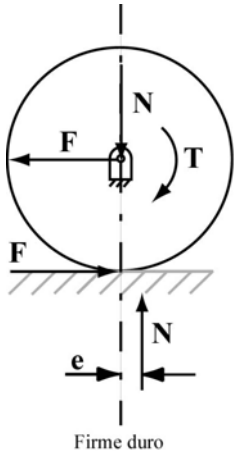
Ajuste a bajas velocidades

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



El modelo virtual del prototipo (iii)

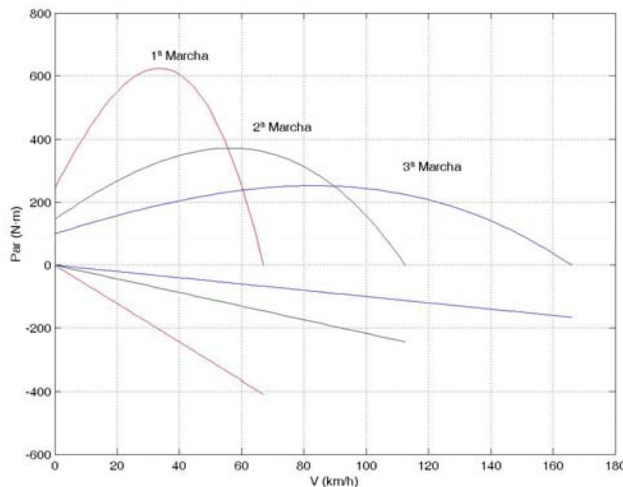
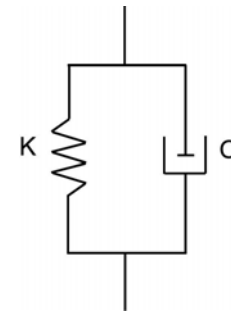
- Resistencia a la rodadura.



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

Superficie		
Cemento	Dureza media	Arena
0,015	0,08	0,30

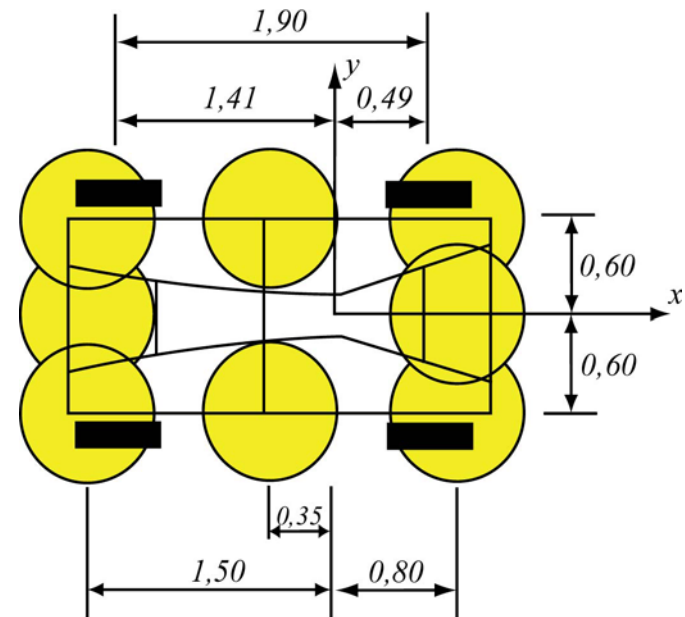
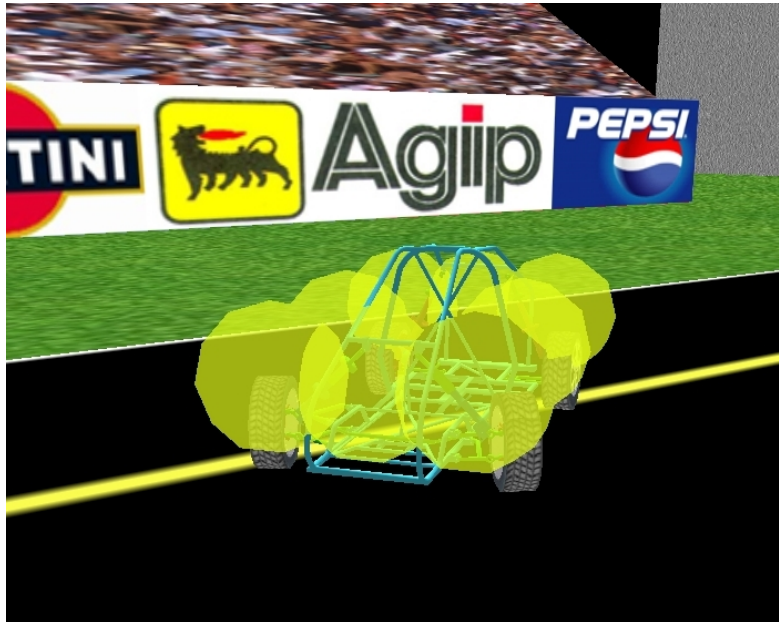
- Fuerzas en las suspensiones: muelles y amortiguadores lineales.



- Fuerzas de la transmisión: Curvas de par-velocidad del motor y relaciones de velocidades (cambio automático).
- Fuerzas de frenado: estimadas a partir de la geometría de los discos y las propiedades del freno. Ajuste a bajas velocidades.

El modelo virtual del prototipo (iv)

- Módulo de colisiones.

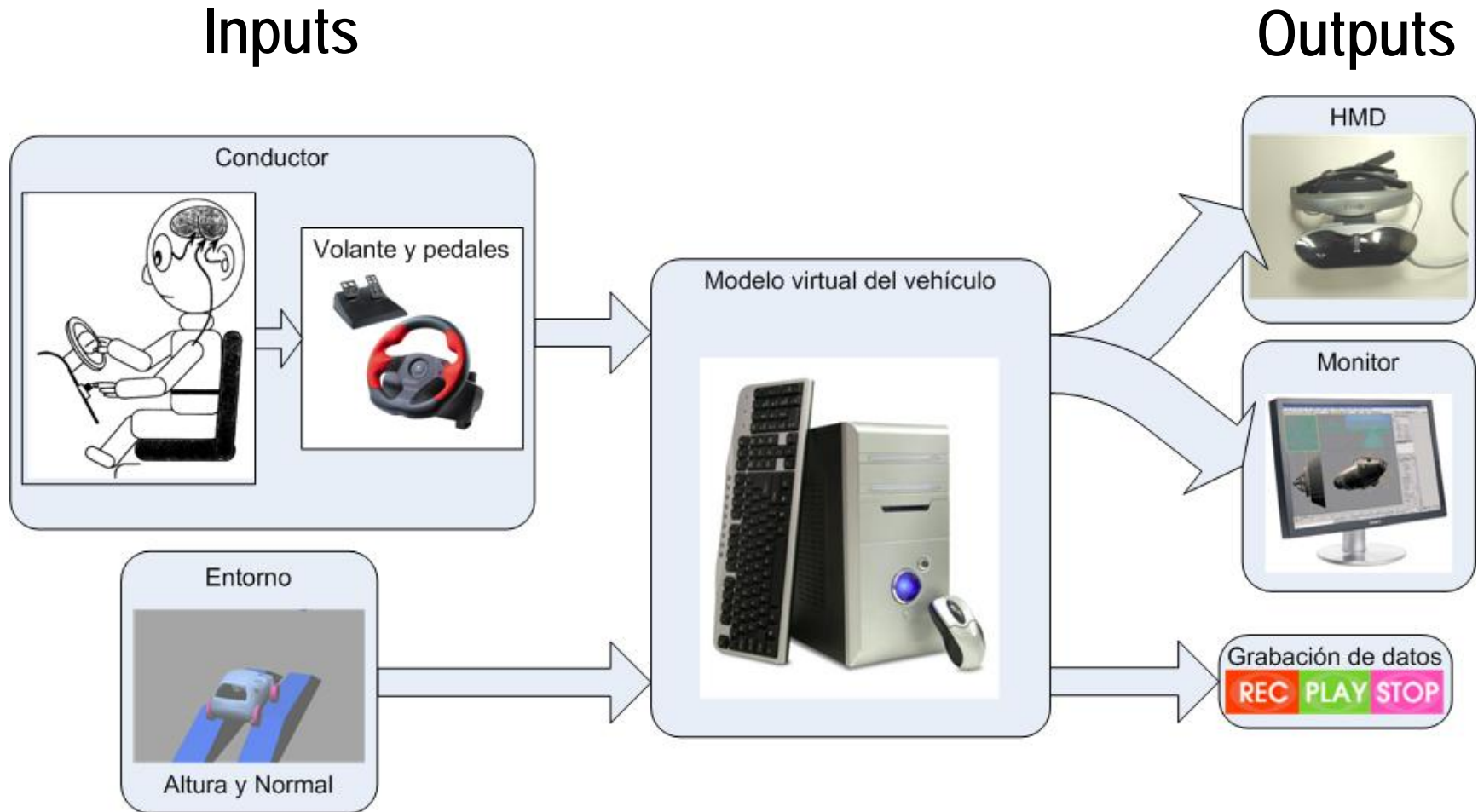


- Resistencia aerodinámica.

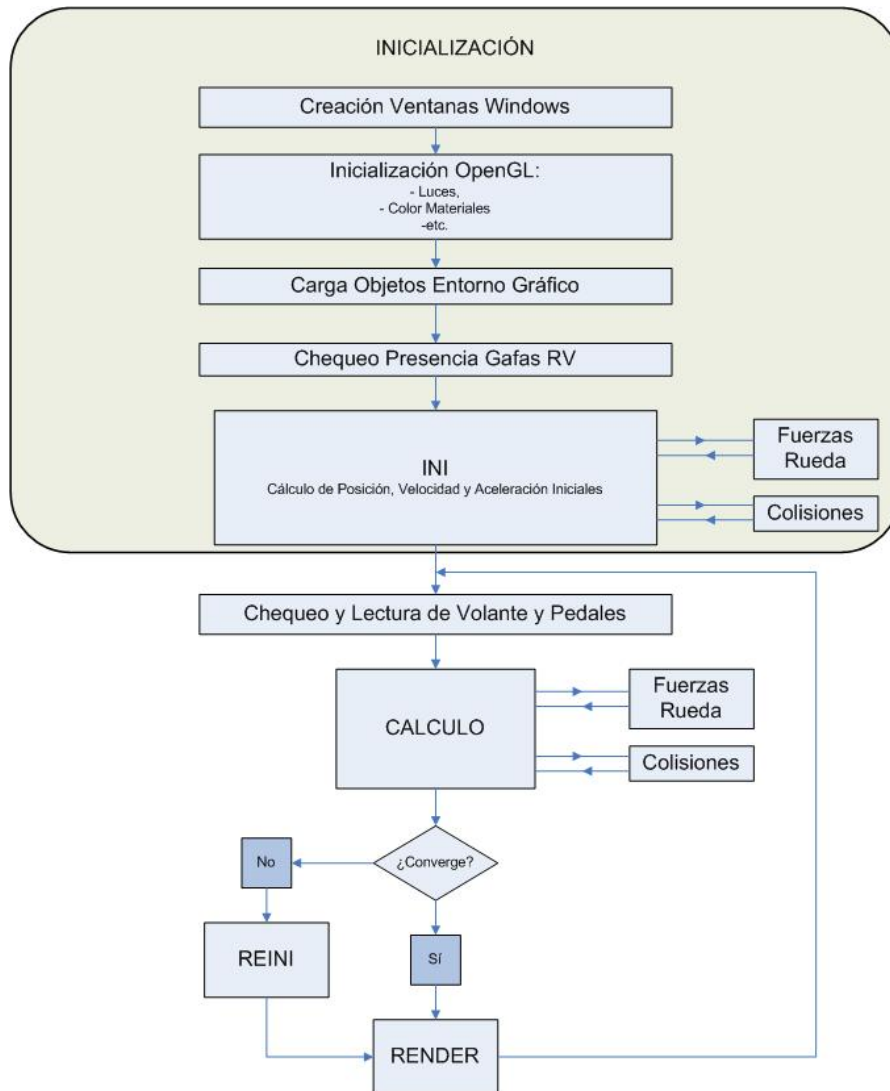
$$F_x = \frac{1}{2} \rho C_x S v^2 \quad C_x = 0,35 \quad \rho = 1,225 \text{ kg} / \text{m}^3$$

1. Estado del arte.
2. El prototipo real.
3. El modelo virtual del prototipo.
4. **El simulador de conducción.**
5. Módulo para desarrollo de controladores: maniobras.
6. Instrumentación.
7. Conclusiones y futuras líneas de investigación.

Simulador de conducción



Desarrollo del programa



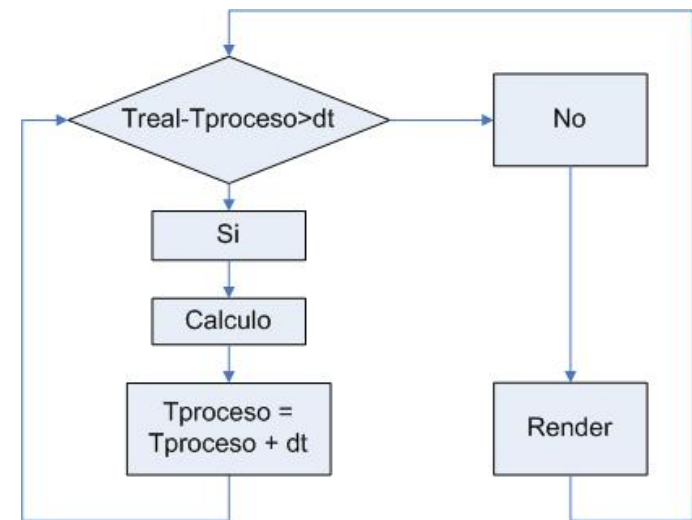
Paso de integración: 10 ms.

Tolerancias:

- 10^{-14} m en problema de posición inicial.

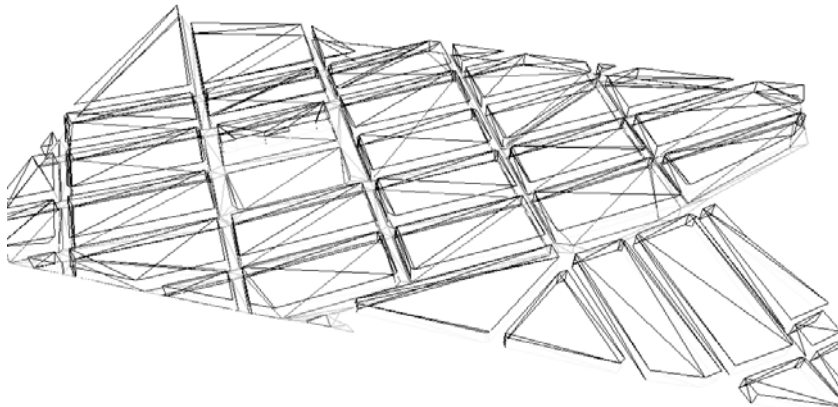
- 10^{-5} m en cálculo dinámico.

Sincronización temporal de la simulación.



Inputs

- Entornos generados mediante OpenGL.
- Superficies triangularizadas.



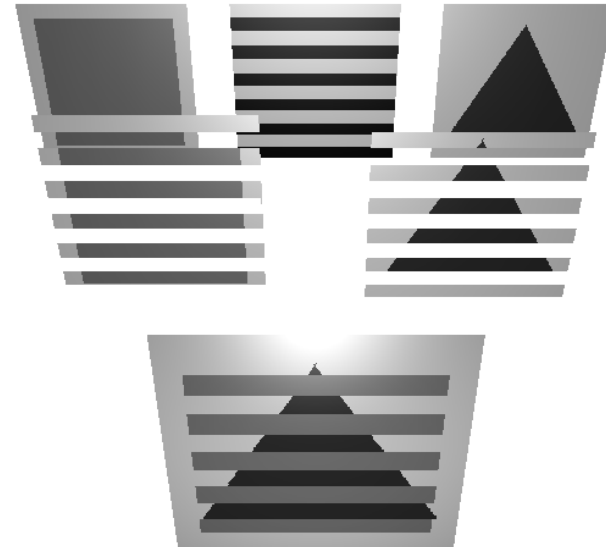
- Almacenamiento en un fichero DXF.
- Algoritmo para la construcción del entorno con lectura de:
 - Triángulo con el que se contacta.
 - Normal.

- Volante y pedales similares a los empleados en videojuegos.
- Selección de cámaras mediante botones.
- Cambio automático, posibilidad de cambio en los botones del volante.



Outputs

- Posible empleo de HMD con visión estereoscópica.



- Ilusión estereoscópica mediante: doble salida gráfica o mediante entrelazamiento de imágenes.
- Posibilidad de grabación y reproducción de maniobras.

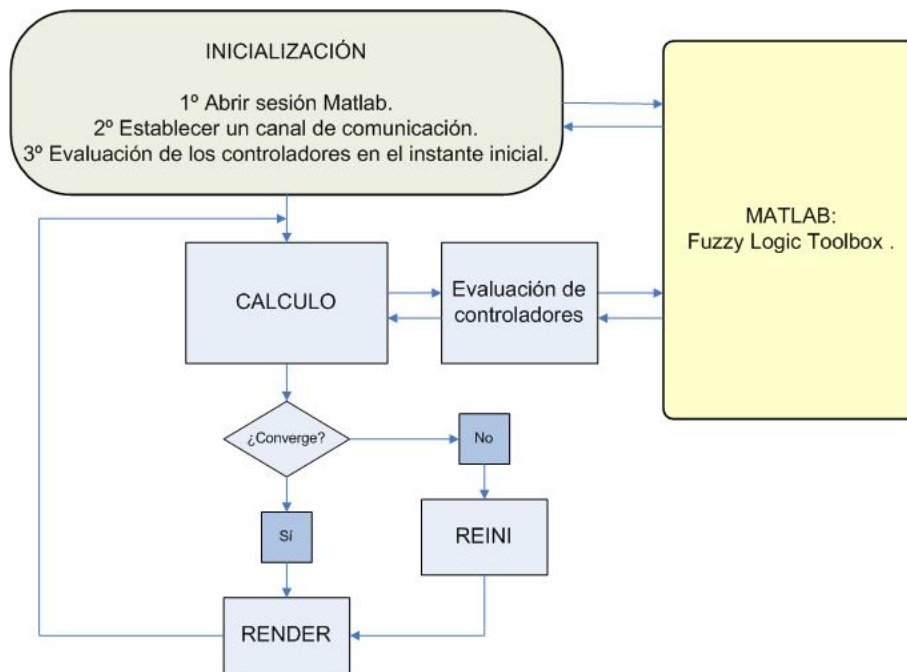
1. Estado del arte.
2. El prototipo real.
3. El modelo virtual del prototipo.
4. El simulador de conducción.
5. **Módulo para desarrollo de controladores: maniobras.**
6. Instrumentación.
7. Conclusiones y futuras líneas de investigación.

Módulo para diseño de controladores

- Tipo de control del vehículo: Seguimiento de trayectoria.
- Técnica de control: Lógica Borrosa.
 - El vehículo es un sistema altamente no-lineal.
 - No se requieren modelos simplificados del vehículo.
 - La lógica borrosa imita el comportamiento humano.
 - Matlab dispone de una toolbox de lógica borrosa.
- El módulo de diseño debe ser computacionalmente eficiente:
 - El modelo computacional del vehículo es un programa eficiente en Fortran.
 - Dos posibles métodos de conexión Fortran-Matlab:
 - Matlab Engine.
 - MEX files.
- El controlador es ajustado manualmente, repitiendo las simulaciones hasta alcanzar el comportamiento deseado.

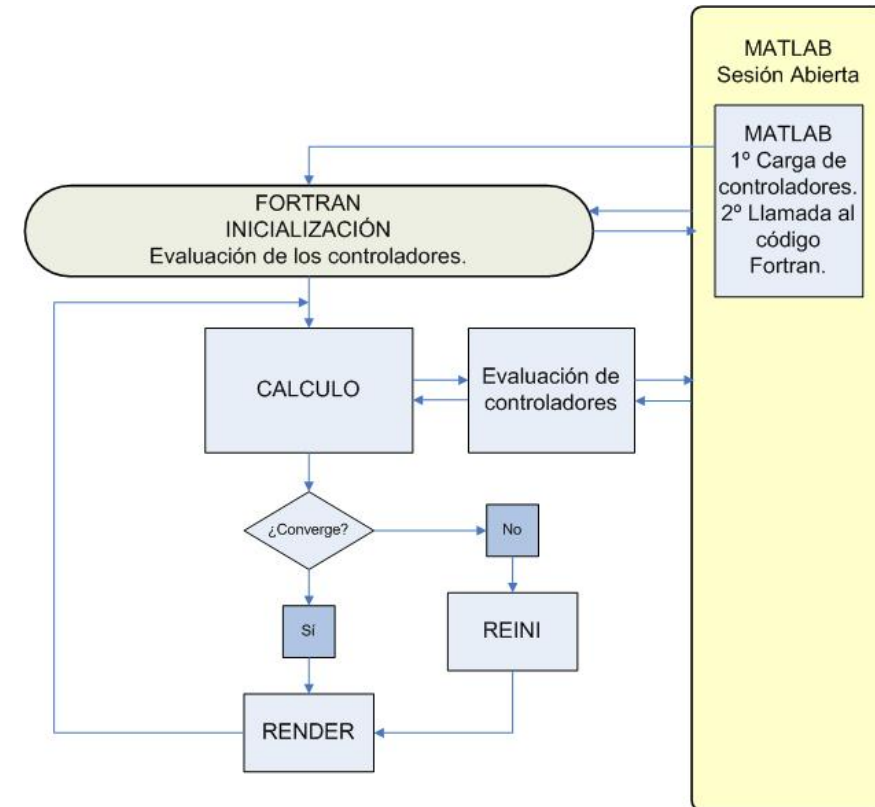
Matlab Engine vs ficheros MEX

Matlab Engine



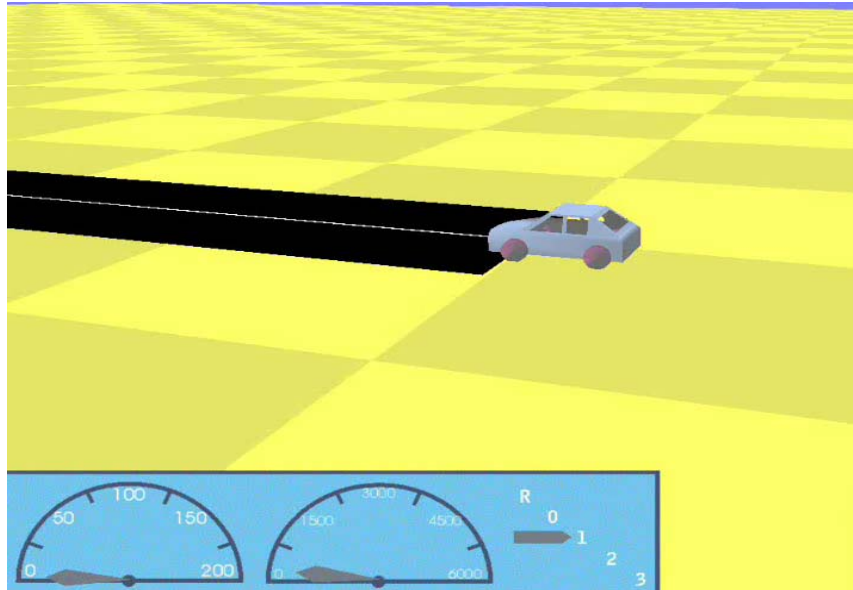
Matlab Engine: el programa en Fortran se conecta a una sesión de Matlab.

Ficheros MEX



Ficheros MEX: Matlab llama a los ficheros MEX generados a partir del programa en Fortran.

Maniobra 1: línea recta



Maniobra: parte del reposo – durante 20 m seguir una línea recta – frenada.

Objetivo: detenerse lo más cerca posible de la meta en el menor tiempo posible.

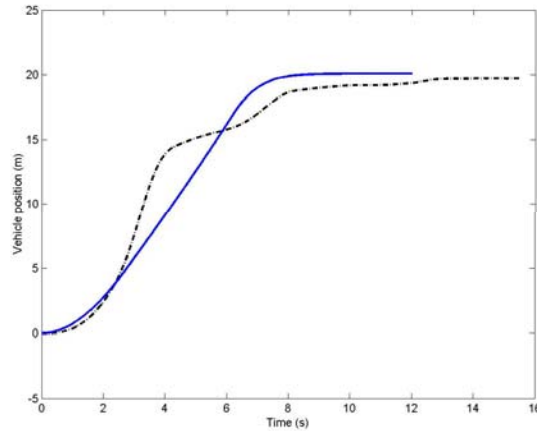
Restricciones: máxima velocidad de 5 m/s; máxima aceleración 3 m/s².

Variables del controlador: posición, velocidad y aceleración.

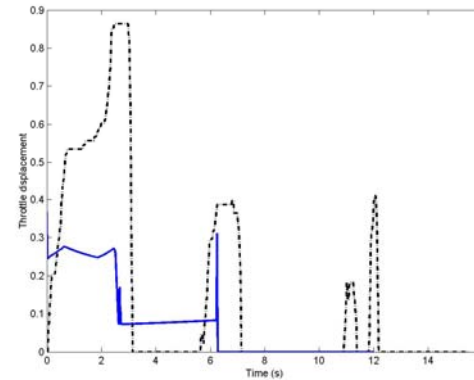
Actuación: acelerador/freno (evitando la acción simultánea sobre ambos).

Maniobra 1: comparativa

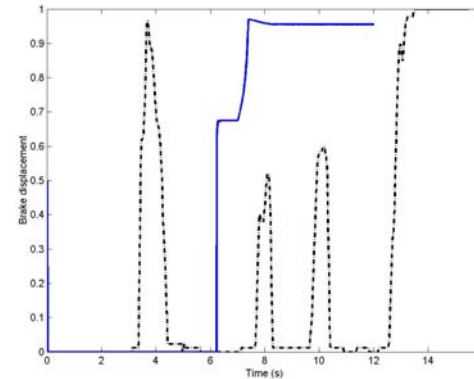
Posición



Comportamiento acelerador



Comportamiento freno

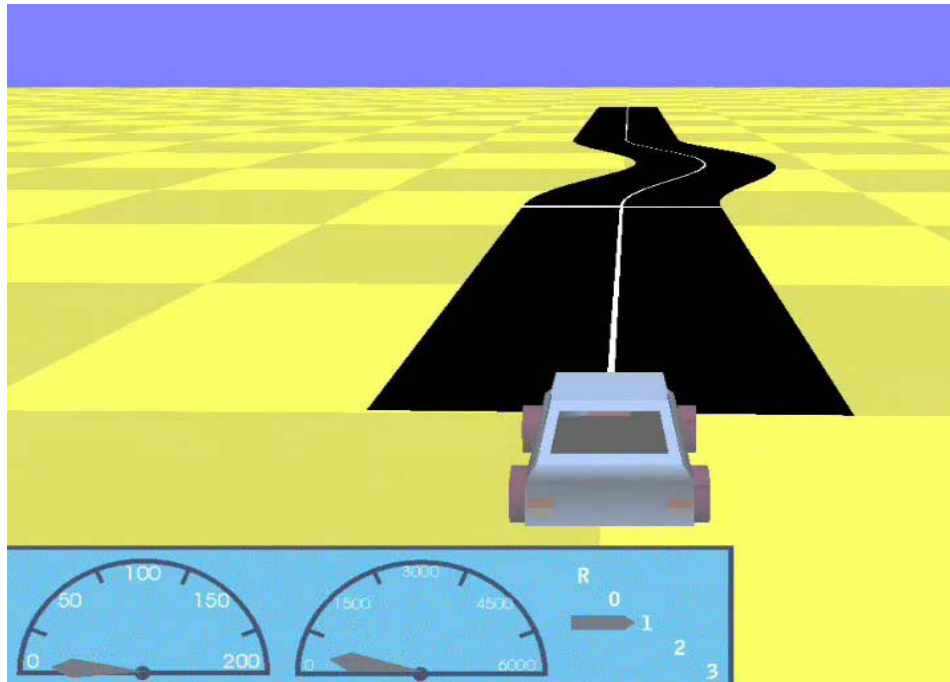


Valores característicos

	Tiempo Maniobra (s)	Error (m)	Maxima velocidad (m/s)
Conductor	13.1	-0.314	8.11
Controlador	8.6	0.081	3.7

Maniobra 2: elusión de un obstáculo

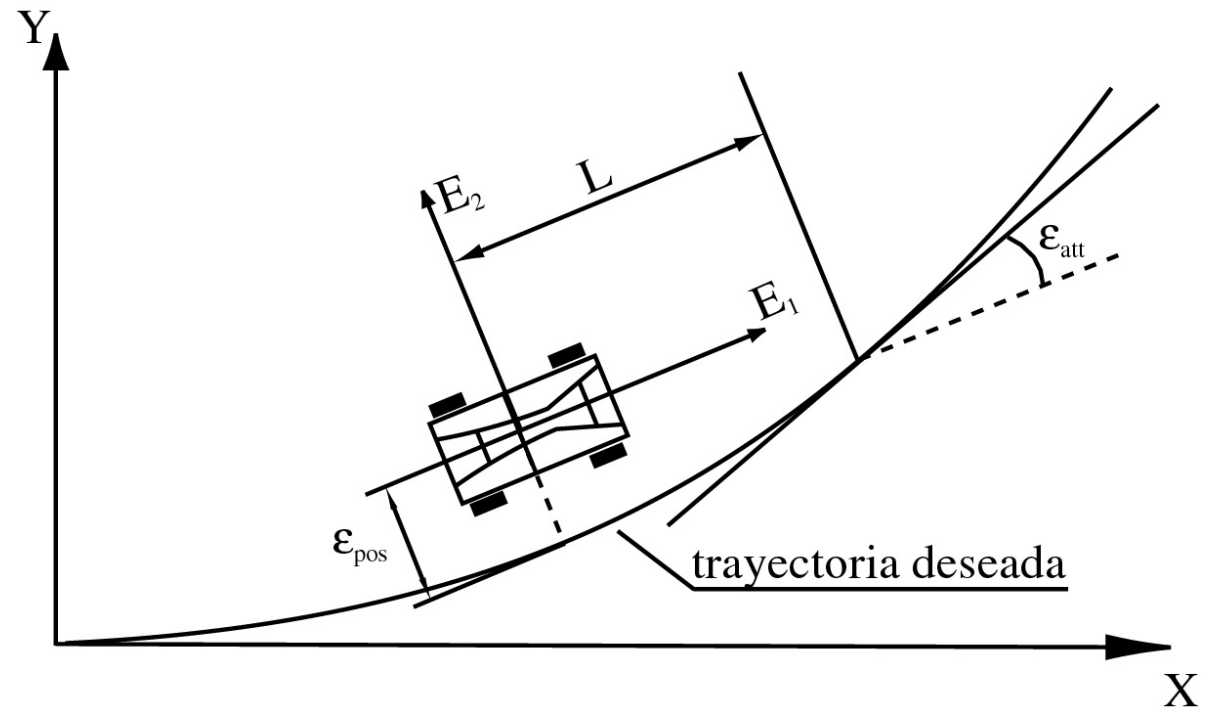
- Maniobra: parte del descanso – tramo recto inicial de 20 m – período sinusoidal completo de 1.75 m amplitud – vuelta a la línea recta.
- Objetivo: error mínimo y mínimo tiempo de maniobra.
- Restricciones: velocidad máxima de 8 m/s.



Maniobra 2: elusión de un obstáculo

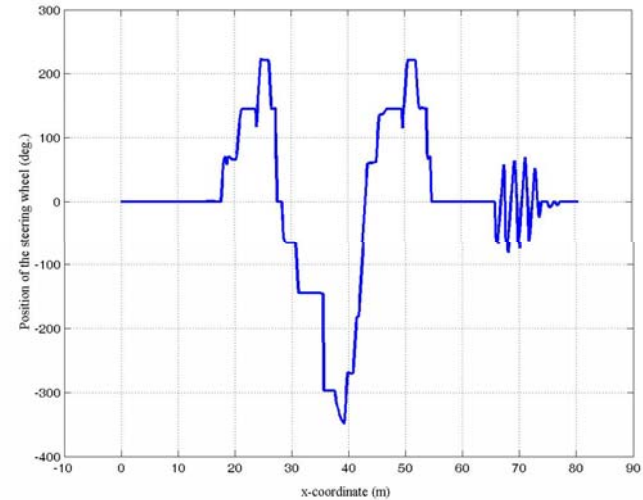
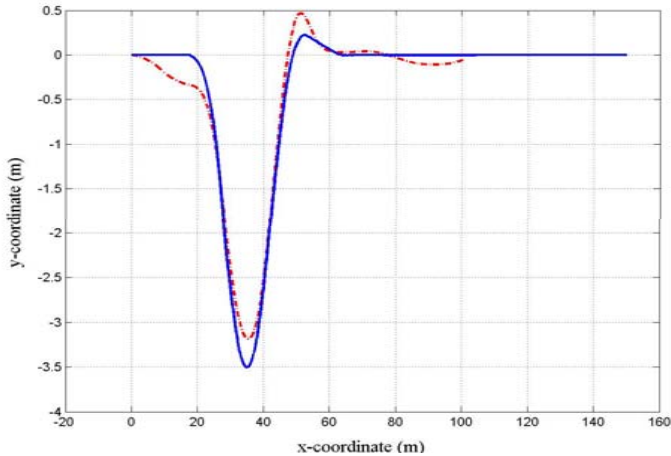
Dos controladores diseñados

- Volante: variables de entrada
 - Error en posición
 - Error angular
- Acelerador/freno: sensores
 - Error
 - Velocidad
 - Aceleración



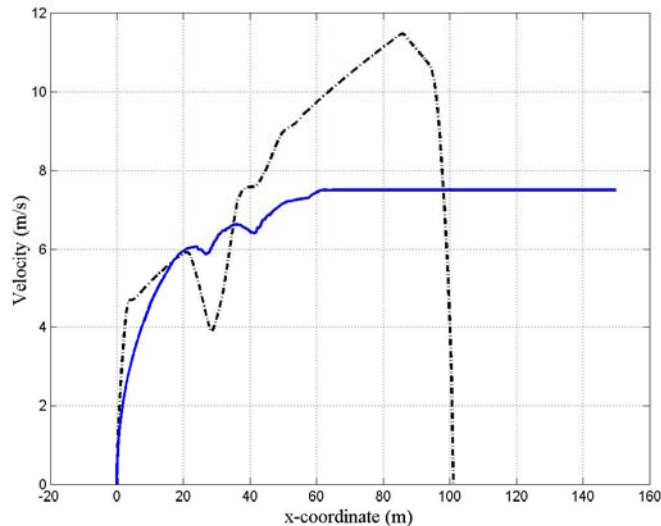
Maniobra 2: comparativa

Trayectoria



Comportamiento del volante

Comportamiento de la velocidad



	Tiempo (s)	Error max. pos. (m)	Error medio pos. (m)	Max. Velocidad (m/s)
Conductor	14.05	0.455	0.1294	7.55
Controlador	14.91	0.205	0.0656	6.55

Valores característicos

Eficiencia

- Diseño: Matlab Engine.
 - Tiempo-CPU*: incluye el tiempo para abrir una sesión de Matlab.
 - Tiempo-CPU **: empleando una sesión ya abierta de Matlab.
 - Tiempos empleando ficheros MEX ligeramente superiores.

Maniobra	Tiempo (s)	Tiempo-CPU* (s)	Tiempo-CPU** (s)
1	12	18.00	11.99
2	24	36.14	30.45

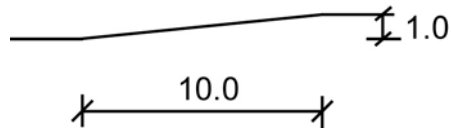
- Aplicación: almacenamiento del controlador diseñado en matrices.

Maniobra	Tiempo (s)	Tiempo-CPU (s)
1	12	4.87
2	24	11.75

Tiempos calculados en Intel Pentium 4 , 2.0 GHz, 1 Gbyte of RAM

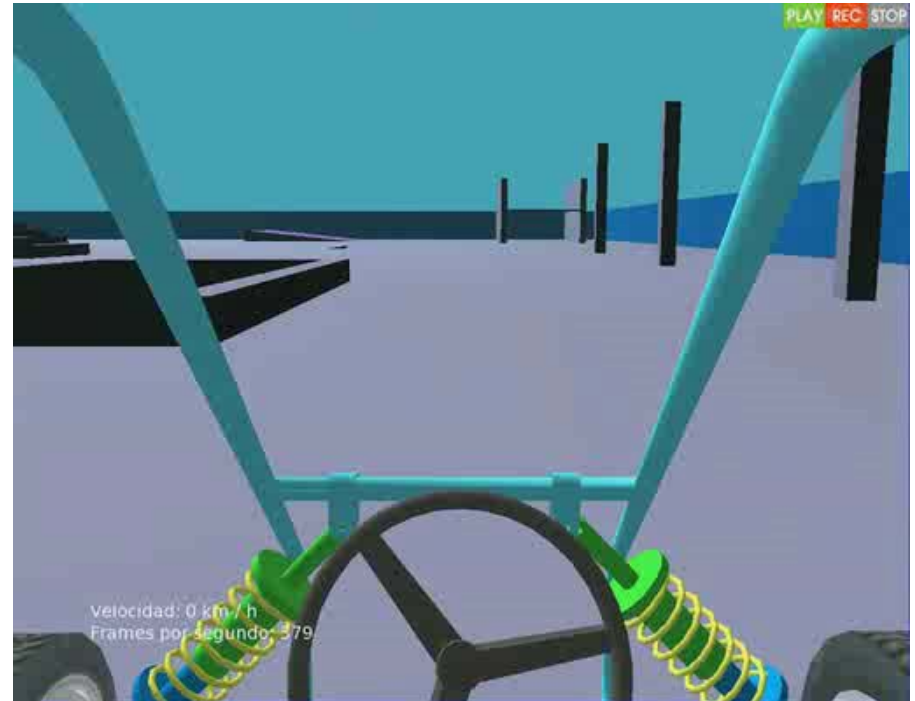
Robustez del simulador

Rampa

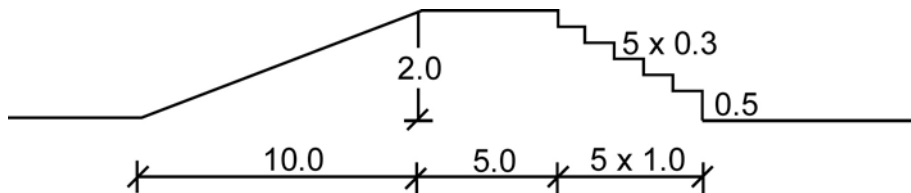


$$v_0 = 11 \text{ m/s}$$

$$F_{\text{máx}} = 15.675 \text{ N}$$



Descenso de escaleras



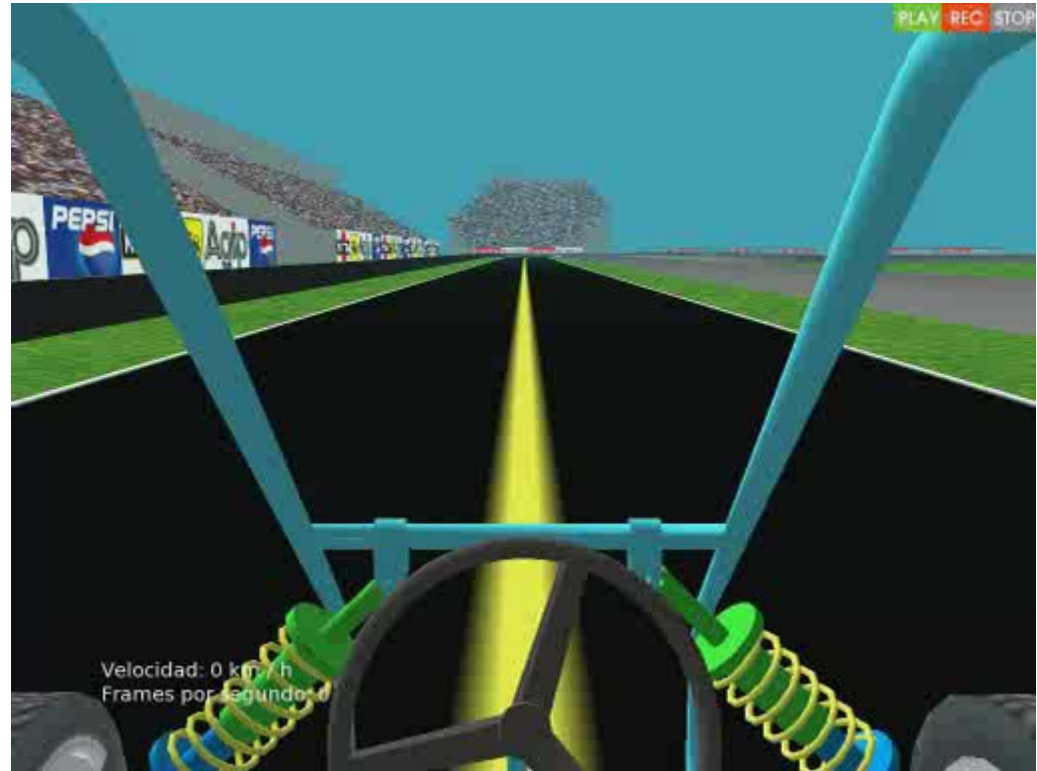
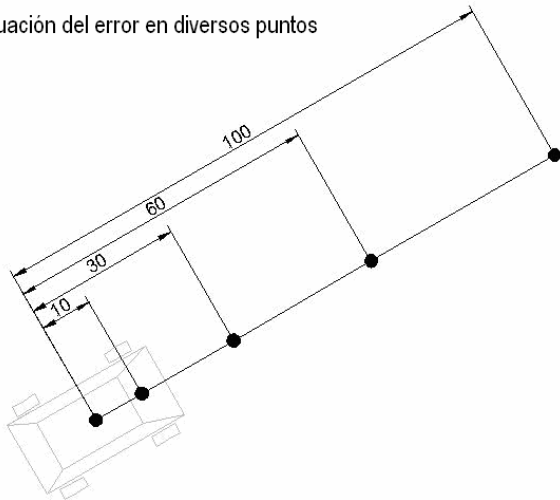
$$v_0 = 1 \text{ m/s}$$

$$F_{\text{máx}} = 14.070 \text{ N}$$

Eficiencia del simulador

- 1 vehículo guiado por conductor y los 2 restantes con controladores.
- Controladores similares a los empleados en la maniobra 2 pero sin tener en cuenta la aceleración.
- Algoritmo de determinación de la velocidad máxima en el tramo.

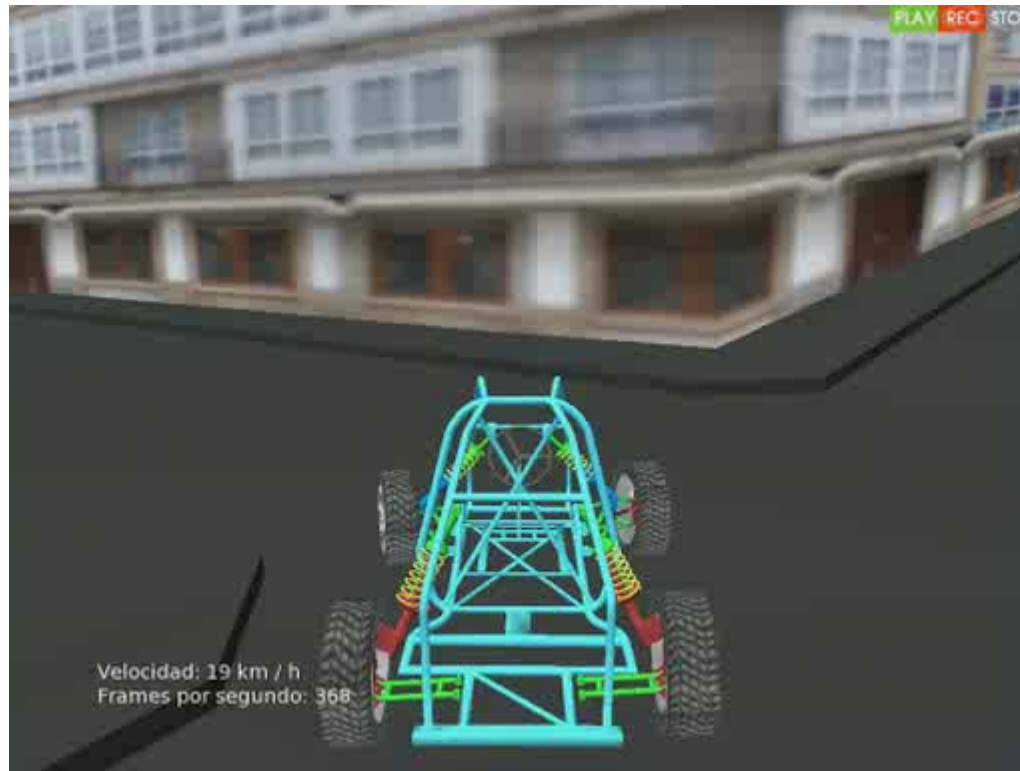
Evaluación del error en diversos puntos



Simulación de varios vehículos en tiempo real en un
Intel Pentium 4, 2.0 GHz, 1 Gbyte of RAM

Entorno realista

- Datos topográficos del barrio de la Magdalena de Ferrol.

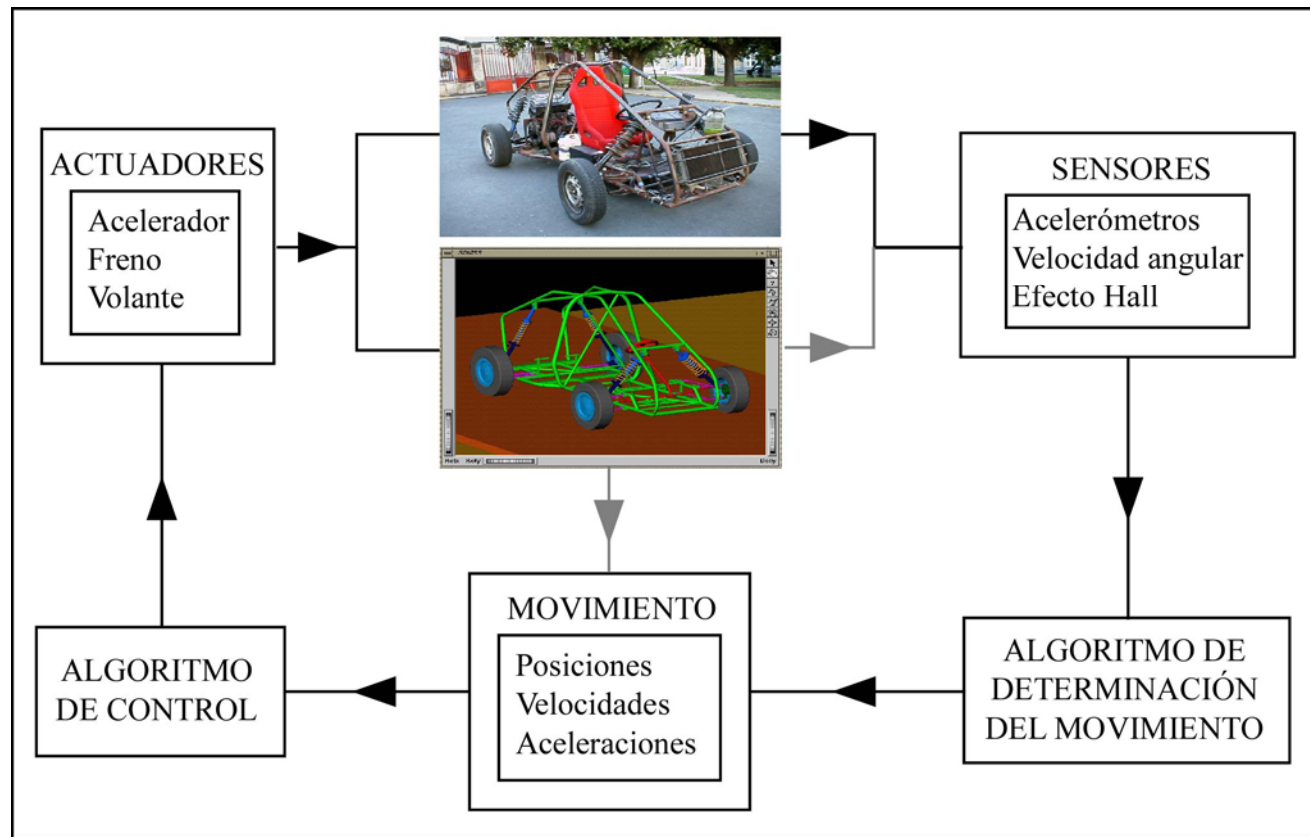


- Alturas de los edificios y aceras extrudidos en OpenGL.
- Texturas de edificios.

1. Estado del arte.
2. El prototipo real.
3. El modelo virtual del prototipo.
4. El simulador de conducción.
5. Módulo para desarrollo de controladores: maniobras.
6. **Instrumentación.**
7. Conclusiones y futuras líneas de investigación.

Instrumentación del prototipo

- Motivación: disponer de una herramienta para verificar que los controles desarrollados son aplicables al vehículo real.



Sensores y actuadores



Sensores

#1 Acelerómetro: aceleración longitudinal.

#2 Sensor de velocidad angular: giro del vehículo alrededor del eje vertical local.

Sensor de efecto Hall: determinar el giro de la rueda y el desplazamiento del vehículo.



Actuadores

Volante: motor paso a paso 7,8 N y reductora 1:3

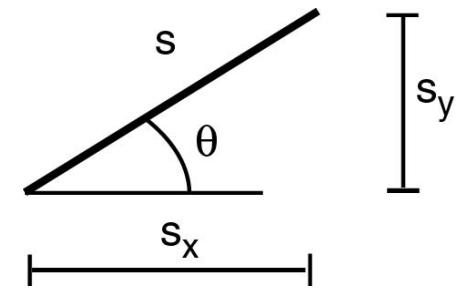
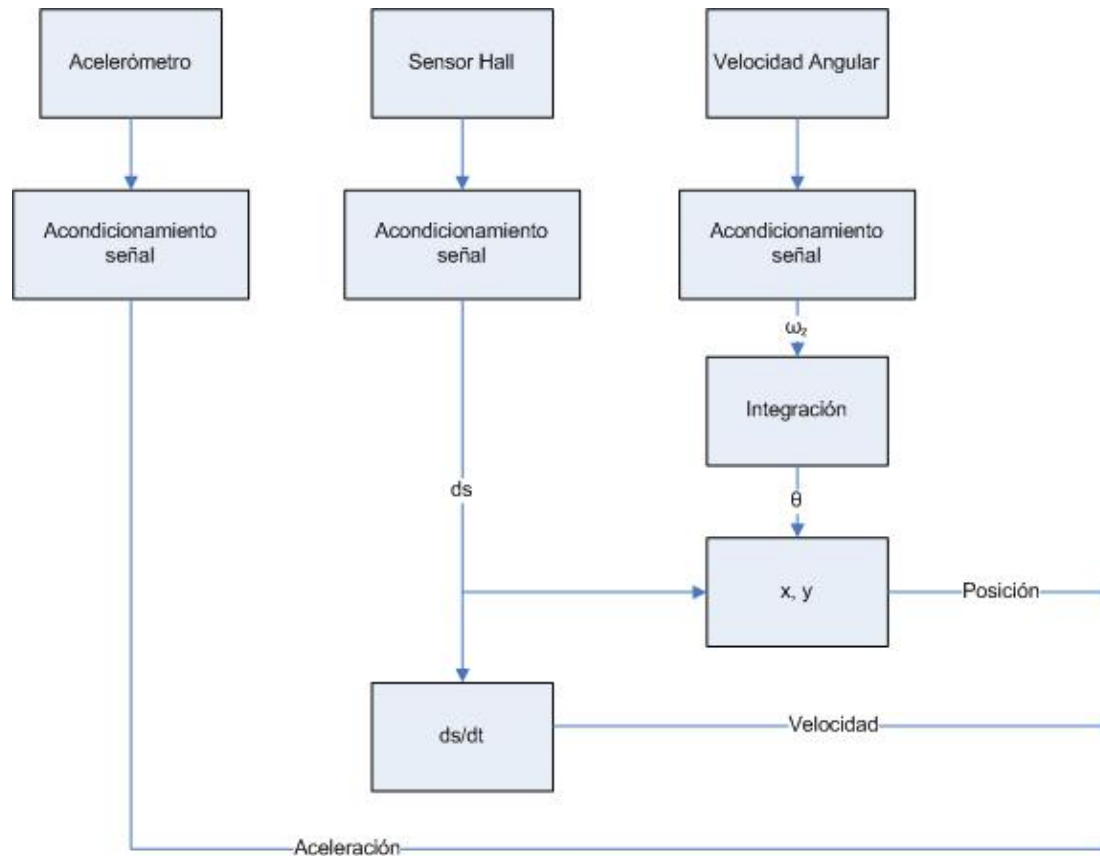
Freno: motor paso a paso 7,8 N y palanca 1:4.

Acelerador: motor paso a paso 2,4 N.



Determinación del movimiento

- Método odométrico a partir de los datos de los sensores inerciales.

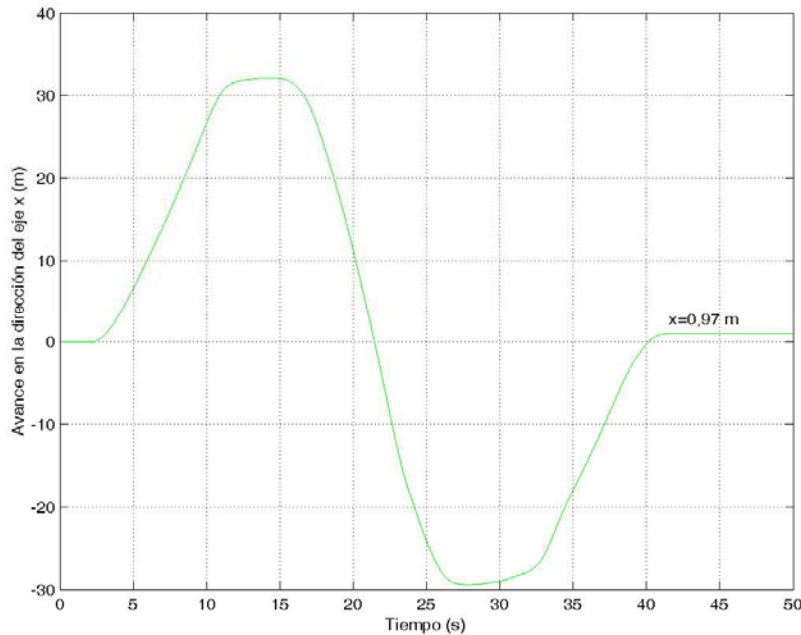


s , lecturas del sensor Hall durante 10 ms.

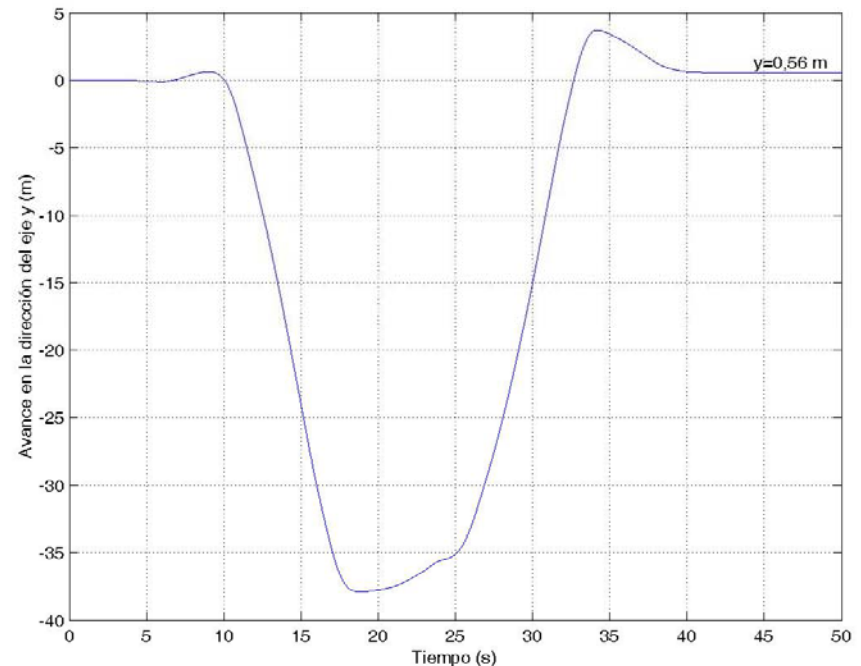
θ , integración del sensor de velocidad angular.

Pruebas y resultados

- Vuelta completa alrededor del edificio de la Facultad de Humanidades del Campus de Ferrol.

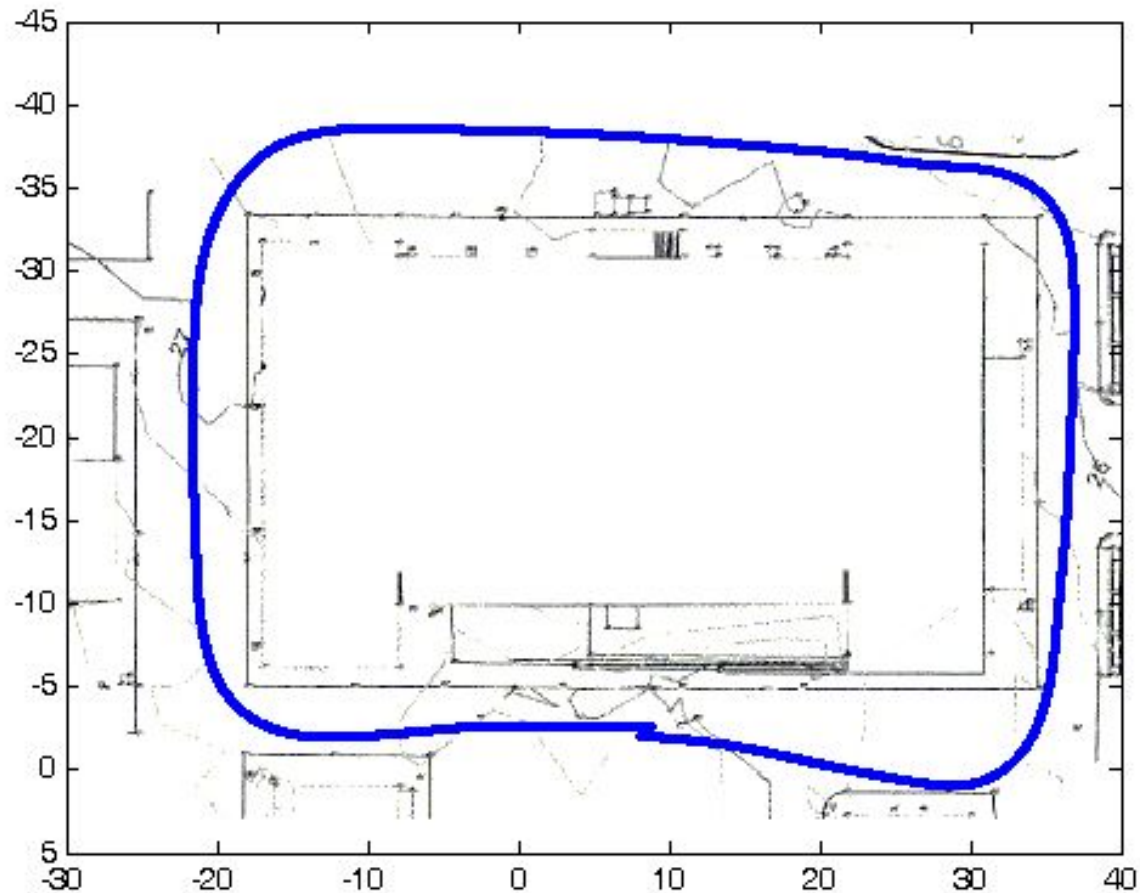


Movimiento en x según el algoritmo:
error de 0,97 m sobre 123,14 m (0,78 %).



Movimiento en y según el algoritmo:
error de 0,56 m sobre 80,14 m (0,69 %).

Pruebas y resultados



Conclusiones

- Se ha construido un prototipo real, instrumentado con sensores y actuadores. Se ha desarrollado un algoritmo para la determinación del movimiento mediante sensores inerciales con un 1 % de error.
- El modelo virtual del prototipo muy detallado ha dado lugar a un simulador de conducción, una herramienta para el diseño de controladores, con diversos entornos, etc.

La dinámica multicuerpo en tiempo real y, en concreto, la formulación expuesta es una herramienta que puede contribuir al desarrollo del sector automovilístico mediante su aplicación a la simulación y el control de automóviles: los métodos utilizados permitan simular modelos más detallados y maniobras más exigentes.

Futuras líneas de investigación

- Prototipo real:
 - Completar la conducción autónoma.
 - Validación del modelo virtual y de los controladores.
 - Mejora del algoritmo de determinación del movimiento (inclusión de GPS mediante filtros de Kalman).
 - Desarrollo de un sistema de "steering by wire",...
- Modelo virtual:
 - Mejora de la inmersión (sonido, mejora visión estereoscópica).
 - Introducción de una plataforma de cinemática paralela.
 - Más detalle en algunos elementos (bushings, transmisión...).
 - Pruebas de formulaciones dinámicas.
 - Mejora del interfaz con el usuario.
- Control:
 - Desarrollo de otras técnicas de control.
 - Asistencia al usuario
 - Control parcial
 - Observador de estados...