UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE CIENCIAS FÍSICAS

DEPARTAMENTO DE ARQUITECTURA DE COMPUTADORES Y AUTOMÁTICA

TRABAJO DE FIN DE GRADO

Código de TFG: ACA03

Modelado y Simulación de Sistemas Dinámicos – Control de coche

Dynamical Systems Modelling and Simulation - Car Control

Supervisor:
José Manuel Velasco Cabo

Diego José Sánchez Martín

Doble Grado en Física y Matemáticas

Curso académico 2020-21

Convocatoria: Junio

Modelado y simulación de sistemas dinámicos - Control de coche

Resumen:

El Trabajo de Fin de Grado que recoge esta memoria consiste en la modelización y posterior simulado de un sistema dinámico concreto: un coche. En la sección 1 se describe cuál es la cuestión exacta a resolver y qué clase de modelo se ha empleado para ello. En la sección 2 quedan establecidos los objetivos, tanto generales como específicos, a los que aspira este trabajo. La sección 3 explica brevemente las herramientas que se utilizan dentro del proyecto, que se desarrolla en profundidad a lo largo de la sección 4. Este apartado es el más largo de la memoria, pues contiene tanto la explicación de los principios físicos que rigen el movimiento estudiado (secciones 4.1 y 4.2) como su implementación práctica (secciones 4.3 y 4.4). Finalmente, las conclusiones que se extraen del conjunto del trabajo se exponen en la sección 5.

Abstract:

The Final Thesis this document comprises consists in the modelling and subsequent simulation of a specific dynamical system: a car. In section 1 the exact question to be solved and the kind of model used are described. In section 2 the goals, both general and specific, that this thesis pursues are clearly stated. Section 3 is devoted to briefly explain the tools used for the project, which is more deeply developed along section 4. This is the longest part in the dissertation, as it includes the exposition of the physical principles that rule the movement studied (sections 4.1 y 4.2) as well as its practical implementation (sections 4.3 y 4.4). Lastly, the conclusions drawn from the thesis are discussed in section 5.

Índice

1. Introducción **3**

2. Objetivos **4**

3. Metodología **4**
3.1. Herramientas informáticas 4
3.2. Elección de vehículo a modelizar 5

4. Desarrollo **6**
4.1. Ecuaciones de movimiento: la velocidad 6
4.2. Ecuaciones de movimiento: la posición 10
4.3. Simulación del modelo 12
4.4. Hacia una conducción autónoma 14

5. Conclusiones **17**

Bibliografía **19**

Anexo 1: Imágenes de conducción autónoma **20**

Anexo 2: Código de MATLAB empleado **21**

1. Introducción

Desde muy temprana edad, siempre he sentido una gran fascinación por los automóviles. De hecho, recuerdo jugar con un coche teledirigido muchas mañanas de domingo. Me gustaba lanzarlo a diferentes velocidades por unas escaleras para ver si volcaba o no en función de cuánto impulso le estaba proporcionando yo a través del mando. En esta actividad no había afán destructivo, sino genuina curiosidad y exploración. Aunque en ese momento no le pusiera un nombre concreto, lo que estaba haciendo era, esencialmente, Física.

Resulta adecuado que, muchos años después, mi Trabajo de Fin de Grado de Ciencias Físicas (en adelante, TFG) verse acerca del control de un coche. Así, este trabajo consistirá principalmente en un proyecto técnico: la construcción de un modelo. Se entiende el concepto de "modelo" como una representación matemática de un sistema físico que nos permita razonar acerca de dicho sistema y realizar predicciones sobre su comportamiento [1]. Además, como la pretensión es "controlar" el sistema, el modelo, realizado en MATLAB, será interactivo. Esto quiere decir que se podrá actuar sobre él y reaccionará a las actuaciones. En este caso, el sistema físico de interés se trata de un automóvil que se desplaza por una carretera. Cabe destacar una serie de características acerca del tipo de modelo escogido:

- Es de representación externa, es decir, consideramos entradas y salidas sobre el sistema sin dirimir en detalle las relaciones internas que se producen entre las mismas. Se estudiarán, evidentemente, las ecuaciones dinámicas que explican el movimiento del coche, pero no se pondrá el foco sobre la mecánica del automóvil.
- Se utilizará una descripción matemática no lineal.
- Las propiedades del sistema serán temporalmente invariantes, considerando el tiempo de forma discretizada.
- Las señales de entrada serán cuantizadas, mientras que las de salida se permitirán continuas.
- En general, el sistema se considerará determinista. Sin embargo, también se incluirán elementos estocásticos en la última sección de la memoria.

El modelado y simulación de sistemas dinámicos con ayuda de un ordenador son centrales en la Física contemporánea. La computación dista de tratarse exclusivamente de una herramienta para el cálculo. Ofrece posibilidades muy amplias de creación de entornos virtuales para la experimentación. Esto es irrenunciable para ciertas disciplinas que, por su objeto de estudio, no pueden realizar experimentos directos. La Cosmología es ejemplo paradigmático de ello. También resulta de enorme utilidad cuando, como ocurre en este Trabajo de Fin de Grado, la experimentación directa sería demasiado costosa económicamente o potencialmente peligrosa. Así, podré modelizar el comportamiento de un coche sin necesidad de montarme en uno. Este acceder a lo inaccesible se encuentra en el núcleo de la motivación por la cual se ha estudiado Física desde el comienzo de la existencia de la Ciencia.

2. Objetivos

Los principales objetivos que se persiguen con la realización de este TFG se pueden dividir en dos grandes grupos. Por un lado, aquellos que se refieren al proyecto específico llevado a cabo se detallan a continuación:

1. Comprender y llevar a cabo el proceso requerido para la modelización de un sistema dinámico físico.
2. Analizar el comportamiento del sistema asociado al coche modelizado.
3. Presentar e interpretar los resultados obtenidos.
4. Entender las limitaciones del modelo propuesto y sus consecuencias sobre la validez de los resultados.
5. Inferir conclusiones acerca de la relación existente entre el modelo del coche y el sistema real del automóvil estudiado.

Además, existen también objetivos de carácter transversal que este trabajo permite alcanzar:

1. Ser capaz de redactar y defender una memoria de un trabajo académico.
2. Aplicar los conocimientos, habilidades y competencias adquiridos durante el Grado en Física a un problema concreto de esta disciplina.
3. Profundizar en las posibilidades que ofrecen las herramientas informáticas para un uso de tipo científico-técnico.

3. Metodología

Para la realización de este trabajo, se toma como base el libro proporcionado por los profesores Jiménez y Velasco en la bibliografía básica: "Virtual Reality and Animation for MATLAB and Simulink Users". En su sexto capítulo [6], este texto proporciona una iteración cero de una posible modelización para el coche que es nuestro objeto de estudio. Si bien se ha tomado dicha sección de este libro como punto de partida, la interpretación física que se hace en ella del movimiento del coche es muy limitada. Las ecuaciones propuestas son de tipo fenomenológico y no se explica su origen. Por este motivo, se ha implementado un modelo propio, que se detallará en la sección 4 del presente trabajo.

3.1. Herramientas informáticas

Puesto que se trata de un proyecto de carácter técnico, una parte muy importante de este TFG no aparece reflejada directamente en esta memoria. En efecto, gran parte de mi trabajo de los últimos meses ha estado dedicado a la simulación por ordenador del modelo. Para ello he empleado tres programas informáticos:

- MATLAB, herramienta de computación que permite realizar operaciones matemáticas con vectores y matrices, así como representaciones gráficas. Es el primer *software* científico con el que tuve contacto en la asignatura Laboratorio de Computación Científica y el que más he empleado a lo largo de mis años universitarios. Aunque es de pago, la Universidad Complutense de Madrid proporciona licencias de uso a sus estudiantes. Se ha usado la versión 2020a.
- Simulink, entorno de programación visual enfocado a la simulación de sistemas. Aunque estrictamente forma parte de MATLAB, sobre el que opera, merece mención expresa. Su manera de funcionamiento es más visual y está orientada a la modelización.
- *Virtual Reality Worlds*, programa específico destinado a la creación de entornos virtuales que está integrado en MATLAB y Simulink. Lo he empleado para visualizar el coche. En este caso, y dado que no es una funcionalidad que domine, he seguido de cerca las indicaciones de [6].

3.2. Elección de vehículo a modelizar

Con el fin de dotar al trabajo de una mayor dosis de realismo, decidí utilizar un coche real al que intentaría que se asemejase el modelo. Esto permitirá calibrar el comportamiento de las ecuaciones de movimiento, tal y como se explicará más adelante.

El vehículo elegido es el Tesla Model 3 Long Range AWD [10]. El motivo principal para escogerlo es que, en la última parte del trabajo, se simula una situación hipotética en la cual el coche del modelo circula de manera autónoma, sorteando los obstáculos que aparecen en su camino. La empresa automovilística que actualmente ofrece el servicio de conducción semiautomática más avanzado es Tesla. De entre sus modelos, se ha escogido el más vendido. Además, al tratarse de una berlina del segmento D, su forma se adecúa a la del coche de la simulación. En este sentido, cabe destacar que la aerodinámica del vehículo será tenida en cuenta en las ecuaciones que describen su movimiento, pero no el modelado geométrico del mismo, llevado a cabo en *Virtual Reality Worlds*, puesto que el diseño 3D preciso no es el objetivo de este trabajo.

A continuación, se muestra en una tabla la información técnica del vehículo que se empleará en el desarrollo del modelo, así como las abreviaturas que aparecerán en las expresiones matemáticas correspondientes. Todos los datos han sido extraídos del sitio web www.ultimatespecs.com [10].

Medida	Abreviatura	Valor
Masa[1]	m	1891 kg
Velocidad máxima	$v_{máx}$	233 km/h
Aceleración (0-100 km/h)	t_{0-100}	4,4 s
Distancia entre ejes	w	2,875 m
Radio de giro	r	11,8 m
Coeficiente de resistencia aerodinámica	C_X	0,23 kg/m

Cuadro 1: Especificaciones técnicas del Tesla Model 3 Long Range AWD (2021)

[1] Cabe destacar que la masa en vacío del Tesla Model 3 es 1831 kg según la página web referenciada. A esta cantidad se le han sumado 60 kg, mi propia masa, como conductor del vehículo. Esto no debería tener un efecto notable sobre los parámetros de rendimiento del coche, en este caso la velocidad máxima y el tiempo de aceleración hasta 100 km/h, puesto que los mismos se calculan con conductor dentro.

4. Desarrollo

4.1. Ecuaciones de movimiento: la velocidad

Una vez tenemos los datos numéricos relativos al coche objeto de estudio, debemos determinar cómo se mueve. Para ello, aplicaremos la Segunda Ley de Newton:

$$\sum_i \vec{F_i} = \frac{d\vec{p}}{dt}, \tag{1}$$

que establece la igualdad entre el sumatorio de las fuerzas $\vec{F_i}$ que actúan sobre una partícula y la derivada temporal de su momento lineal $\vec{p}$. Nótese que se ha tomado el coche como puntual sobre su centro de masas, aproximación que es válida ya que la distribución exacta de masa en el vehículo no tiene demasiada relevancia dado que no se van a simular colisiones. Además, se tiene la igualdad $\vec{p} = m\vec{v}$. Como la masa no va a variar a lo largo del movimiento considerado, podemos introducir la velocidad v expresamente en la derivada:

$$\sum_i \vec{F_i} = m\frac{d\vec{v}}{dt}. \tag{2}$$

A continuación, es necesario decidir qué fuerzas vamos a considerar que actúan sobre el movimiento del coche. A partir de las mismas se construye el modelo que se empleará en todo momento a lo largo del TFG. Estas fuerzas son:

- Fuerza motriz, que se denotará $\vec{F}_m$. Sea $\vec{u}_{\vec{v}}$ el vector unitario con misma dirección y sentido que la velocidad. Por simplicidad, la fuerza motriz podrá tomar tan solo tres valores: $F_m^+\vec{u}_{\vec{v}}$, si se pisa el acelerador; $-F_m^+\vec{u}_{\vec{v}}$, si se pisa el freno; y $\vec{0}$, si no se da ninguno de los dos casos anteriores. Así, tan solo se consideran los casos en los que el conductor acelera a fondo, frena a fondo o no actúa.

- Fuerza de resistencia aerodinámica [3], que aparece por la presencia de un fluido, en este caso aire, en el camino del vehículo. Se opone al movimiento. Toma un valor u otro en función de la forma del vehículo considerado y es proporcional al cuadrado de la velocidad: $\vec{F}_{aer} = -C_X\vec{v}|\vec{v}|$.

- Fuerza de rozamiento mecánico [7], que también se opone al movimiento. Es proporcional a la velocidad. La relación de proporcionalidad se establece a través de un coeficiente de rozamiento de forma que $\vec{F}_{roz} = -\mu\vec{v}$. Cabe destacar que este rozamiento incluye tanto el rozamiento de las ruedas con el suelo como cualquier otra fricción que ocurra en los engranajes del vehículo.

Juntando lo anterior en la expresión 2 y pasando la masa dividiendo al lado de las fuerzas, se obtiene una ecuación diferencial no lineal para la velocidad que toma la siguiente forma:

$$\frac{d\vec{v}}{dt} = \frac{1}{m}\left(\vec{F}_m - \mu\vec{v} - C_X\vec{v}|\vec{v}|\right). \tag{3}$$

Esta ecuación se resolverá numéricamente considerando un movimiento rectilíneo, de forma que la velocidad se pueda escribir en módulo. Para ello, se empleará un método de Runge-Kutta de cuarto orden [4], con condición inicial de velocidad nula. Cada paso iterativo, que sirve para pasar de una velocidad $v(i)$ en tiempo i a otra $v(i+1)$ en un tiempo i+1 separadas por un intervalo h de 0,1 s., se basa en el cálculo de cuatro coeficientes:

$$k_j = \frac{1}{m}\Big(F_m - \mu v_j(i) - C_X v_j(i)^2\Big) \quad con \;\; j = 1,2,3,4$$

$$\text{donde} \begin{cases} v_1(i) = v(i) \\ v_2(i) = v(i) + \dfrac{1}{2}hk_1 \\ v_3(i) = v(i) + \dfrac{1}{2}hk_2 \\ v_4(i) = v(i) + hk_3 \end{cases} \tag{4}$$

Estos coeficientes, que constituyen aproximaciones del incremento de la velocidad en momentos intermedios entre el tiempo i y el tiempo i+1, se combinan con diferentes pesos para dar lugar al valor estimado para la velocidad en la siguiente iteración:

$$v(i+1) = v(i) + \frac{1}{6}h\big(k_1 + 2k_2 + 2k_3 + k_4\big) \tag{5}$$

Cabe destacar que se ha escogido este método frente a otros conceptualmente más sencillos como el método de Euler por su mayor estabilidad [5]. Esto es especialmente relevante al trabajar con una ecuación diferencial no lineal, que podría presentar algún salto brusco.

Resulta interesante que, aunque se haya solucionado la ecuación diferencial planteada, nuestro resultado tiene todavía dos grados de libertad, que resultan del hecho de que las constantes F_m^+ y μ no están determinadas. Para asignarles valores numéricos, utilizaremos los datos conocidos del Tesla Model 3 reflejados en la sección 3.2. La primera referencia de la que disponemos es el tiempo que tarda nuestro vehículo en alcanzar los 100 km/h. Esto se traduce en que la integración anteriormente descrita debe arrojar una velocidad de 100 km/h a los $4,4$ segundos del momento de arranque.

Además, también conocemos cuál es la velocidad máxima del automóvil, lo cual nos proporciona una condición adicional sobre el coeficiente de rozamiento. Tendremos en cuenta que, para $v = v_{máx}$, la derivada temporal de la velocidad debe anularse. Despejando en la ecuación 7 se llega a

$$\mu = \frac{F_m^+}{v_{máx}} - C_X v_{máx}, \tag{6}$$

lo cual nos permite expresar en módulo la ecuación diferencial que rige el movimiento mientras el coche acelera en función de un único parámetro desconocido:

$$\frac{dv}{dt} = \frac{1}{m}\Big(F_m^+ - \big(\frac{F_m^+}{v_{máx}} - C_X v_{máx}\big)v - C_X v^2\Big) = \frac{1}{m}\Big(F_m^+\big(1 - \frac{v}{v_{máx}}\big) - C_X v^2\big(1 - \frac{v_{máx}}{v}\big)\Big), \tag{7}$$

donde el único valor desconocido es la fuerza motriz. Por tanto, para hallar su valor, basta con integrar la ecuación según el método convenido anteriormente para distintos valores posibles de dicha fuerza. Tomaremos aquel de ellos que cumpla alcanzar una velocidad de 100 km/h transcurridos 4,4 segundos. Esto se muestra en la figura siguiente.

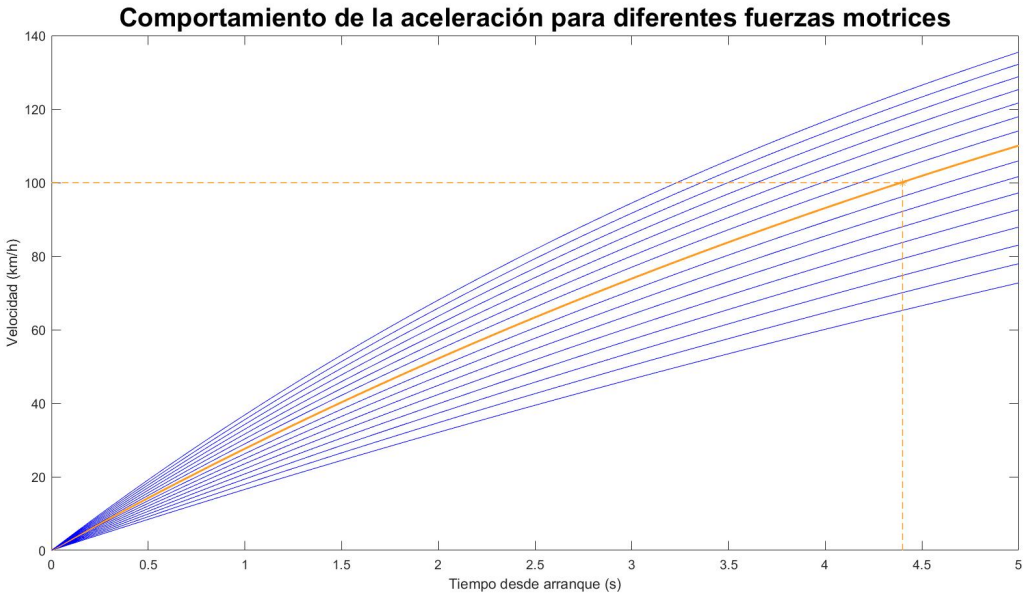

Figura 1: Velocidad que adquiere el vehículo en función del tiempo transcurrido desde el momento del arranque para diferentes fuerzas motrices, tomando valores entre 9000 y 21000 N con espacio de 800 N entre ellos. Se señala en color naranja la línea que corresponde a la fuerza elegida para la modelización.

La línea discontinua de color naranja indica el punto donde se produce esa llegada a los 100 km/h tras 4,4 segundos. Esto se consigue con la gráfica de integración resaltada, correspondiente a una fuerza motriz de 15400 N. Cabe destacar que este número es el resultado de integrar la ecuación diferencial para la velocidad utilizando muchos más posibles valores para la fuerza motriz de los que se muestran en la figura anterior. En concreto, se han empleado todos los valores de F_m^+ en el intervalo dado con un espaciado de 100 N entre los mismos, seleccionando de entre ellos aquel que da un resultado más cercano al deseado. Con el fin de obtener una mejor visualización, no se han incluido todas las líneas en la gráfica. En cualquier caso, haber llegado a este valor de $F_m^+ = 15400$ nos permite calcular el coeficiente de rozamiento a partir de la expresión 6, obteniendo $\mu = 223\ \frac{Ns}{m}$. Así, nuestros parámetros a determinar quedan:

$\mathrm{F}_m^+ = 15400\ N$	$\mu = 223\ \frac{Ns}{m}$

Ya con todos los valores definidos, nuestro modelo matemático está completado y podemos construir las representaciones gráficas del movimiento del vehículo.

En primer lugar, se representa la velocidad frente al tiempo para una prueba de aceleración corta. Esta consiste en mover el vehículo en línea recta durante 20 segundos desde el reposo manteniendo en todo momento pisado a fondo el acelerador, es decir, con la fuerza motriz actuando. Se obtiene la gráfica siguiente:

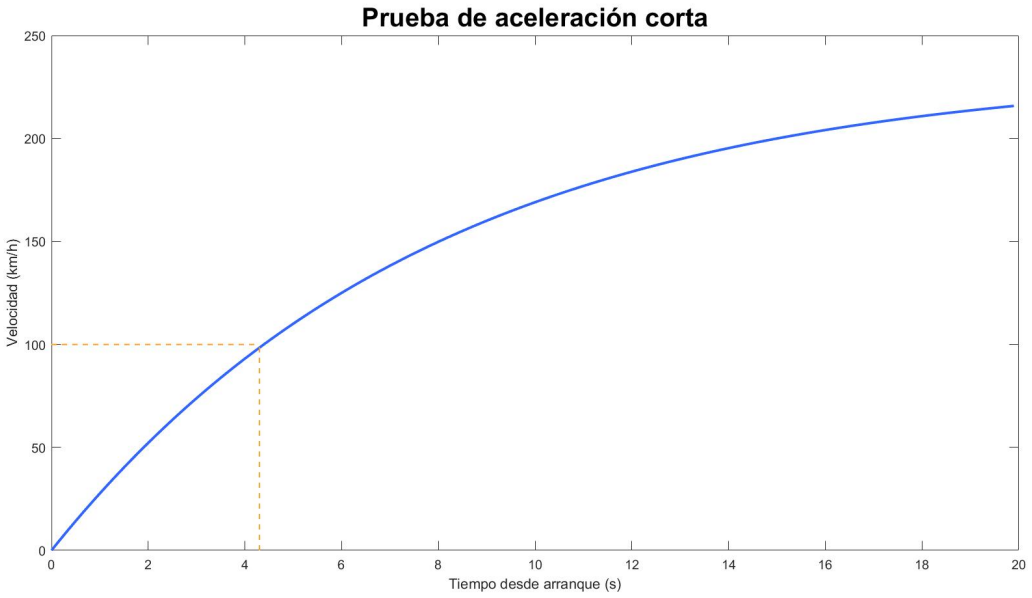

Figura 2: Prueba de aceleración de 20 segundos de duración para Tesla Model 3.

Se puede observar que el comportamiento de la velocidad es cuasilineal para tiempos cortos, puesto que los efectos del rozamiento son menos importantes. En segundo lugar, se hace una prueba de aceleración en las mismas condiciones pero con una duración total de 100 segundos.

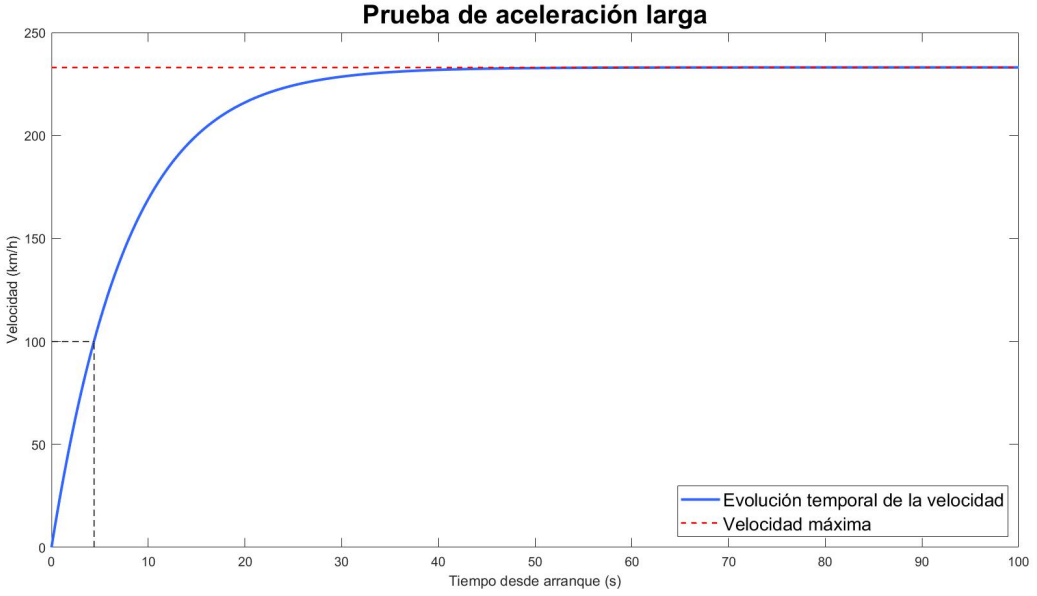

Figura 3: Prueba de aceleración de 100 segundos de duración para Tesla Model 3.

La velocidad se va comportando de forma progresivamente menos lineal a medida que va aumentando, lo cual es consecuencia de que las fuerzas que se oponen al movimiento crecen con v (rozamiento mecánico) y v^2 (rozamiento aerodinámico). De hecho, tal y como se quería modelar, encontramos un valor límite que actúa como asíntota horizontal. Está situado en 233 km/h, la velocidad máxima del vehículo.

4.2. Ecuaciones de movimiento: la posición

Una vez se ha obtenido la velocidad del vehículo, esta debe ser utilizada en cada iteración para calcular su posición, resolviendo la sencilla ecuación $\frac{dx(t)}{dt} = v(t)$. Para ello, aplicamos el método de Euler, que aproxima la derivada de la posición como constante e igual a la velocidad de la iteración anterior en un intervalo corto de tiempo. Así, se llega a $x(i+1) = x(i) + hv(i)$, donde cada uno de los valores $v(i)$ se obtiene a partir de la integración por el método de Runge-Kutta explicado anteriormente.

Se va a utilizar este modelo para simular una prueba muy habitual en coches de todo tipo: el cuarto de milla. De inspiración americana en las unidades, este test consiste en medir el tiempo que tarda un automóvil en recorrer una distancia de $402,34$ metros partiendo de parado y con el acelerador pisado a fondo en todo momento. Con los parámetros anteriormente discutidos, obtenemos la figura 3 para el coche modelizado. Se ha indicado en línea discontinua de color naranja el momento en el cual el vehículo alcanza la distancia objetivo de la prueba.

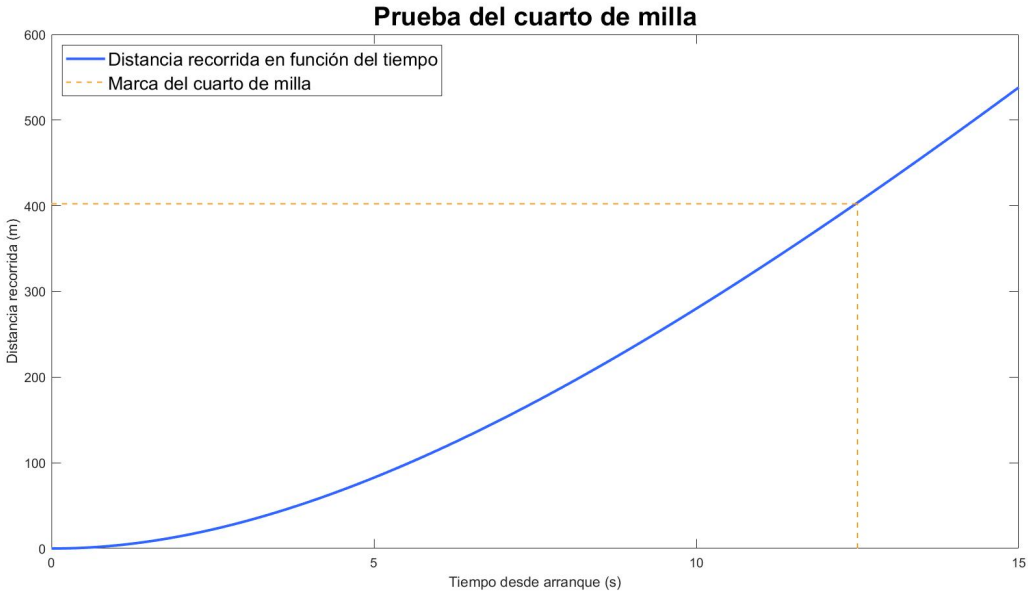

Figura 4: Prueba del cuarto de milla para el Tesla Model 3.

El tiempo obtenido en la prueba efectuada sobre el modelo es de $12,5$ segundos, momento en el cual el vehículo circulaba a $187,0$ km/h. Según datos del portal especializado MotorTrend [9], en su prueba de campo el tiempo de paso por el cuarto de milla para el Tesla Model 3 Long Range fue de $12,5$ segundos, alcanzado a $113,1$ millas por hora (o $182,0$ km/h). A la vista de estos resultados tan próximos (mismo tiempo de paso medido con precisión de una décima de segundo y menos de un 3 % de discrepancia en la velocidad), se puede afirmar que el modelo construido se ajusta muy bien, al menos circulando en línea recta, al coche real.

Sin embargo, el objetivo de este TFG no es simplemente mimetizar el comportamiento rectilíneo del vehículo, sino replicar su movimiento en un caso más general. Para poder construir una simulación, debemos ser capaces de describir su posición cuando el volante entra en escena. Para llevar esto a cabo, es imprescindible calcular cuánto es capaz de girar nuestro coche, es decir, cuál es el ángulo máximo que pueden desviarse las ruedas delanteras con respecto a la posición de su eje correspondiente a la marcha recta. Este dato no es proporcionado por el fabricante, por lo que resulta necesario calcularlo a partir de la información disponible: el radio de giro y la distancia entre los ejes del Tesla Model 3. El modelo geométrico que se presenta a continuación constituye nos servirá para ello.

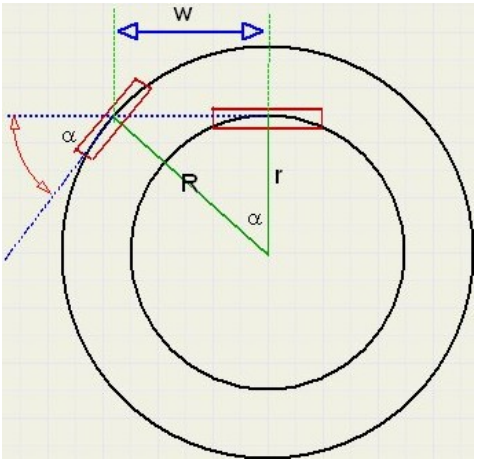

Figura 5: Diagrama esquemático del movimiento de giro de un coche en vista de planta (desde arriba). Imagen obtenida de http://www.davdata.nl/math/turning_radius.html

En la figura anterior se muestran las trayectorias seguidas por las ruedas del lado izquierdo de un coche que está dando una vuelta completa. Las direcciones de los ejes de ambas ruedas apuntan en todo momento al centro de giro, si bien los radios de las circunferencias que describen son diferentes, puesto que la rueda delantera, que provoca el giro, hace un recorrido más largo. Se puede observar que la perpendicularidad existente entre el radio de giro r y la distancia entre ejes w genera un triángulo rectángulo que permite establecer una relación trigonométrica sencilla utilizando la tangente del ángulo que describe la rueda delantera $\alpha_{máx}$ (α en el diagrama). Despejando obtenemos el parámetro buscado:

$$\tan\left(\alpha_{máx}\right) = \frac{w}{r} \quad \rightarrow \quad \alpha_{máx} = \arctan\left(\frac{w}{r}\right) \approx 0,24\pi \tag{8}$$

Aunque este valor no aparezca en las ecuaciones subsiguientes, disponer de él es esencial de cara a la simulación del modelo para limitar la capacidad de giro del vehículo. Al igual que la velocidad máxima, su cometido fundamental es adecuar el comportamiento del coche simulado al real.

Ya nos encontramos en situación de escribir las ecuaciones completas para la posición del vehículo. Utilizamos una matriz de rotación estándar para generalizar la solución ya expuesta en el comienzo de esta sección, de manera que la velocidad se distribuya entre los ejes x y z, puesto que el eje y se corresponde con la altura con respecto al suelo. Cabe destacar que esta elección de ejes viene motivada por la configuración por defecto de *Virtual Reality Worlds*. Así, se obtiene:

$$\begin{pmatrix} x(i+1) \\ y(i+1) \\ z(i+1) \end{pmatrix} = \begin{pmatrix} x(i) \\ y(i) \\ z(i) \end{pmatrix} + h \times \begin{pmatrix} \cos\beta(i) & 0 & \operatorname{sen}\beta(i) \\ 0 & 1 & 0 \\ -\operatorname{sen}\beta(i) & 0 & \cos\beta(i) \end{pmatrix} \times \begin{pmatrix} v(i) \\ 0 \\ 0 \end{pmatrix}. \tag{9}$$

Cabe destacar que el ángulo que aparece como argumento de las funciones trigonométricas de la matriz de rotación de la expresión anterior no es el ángulo α que se mide sobre las ruedas delanteras al girar éstas. La razón para esto es que las dos ruedas delanteras no giran exactamente lo mismo, ya que, en una curva, la rueda que circula por el exterior recorre una distancia algo mayor. La manera de tener en cuenta este efecto sobre la posición del conjunto del vehículo es sustituir α, controlable a través del volante, por un ángulo efectivo β. La expresión que relaciona estos dos conceptos depende de α y de la velocidad del coche. Siguiendo la fórmula experimental propuesta en [6], el cálculo realizado es:

$$\beta(i+1) = \beta(i) + v(i+1) \times \big(0,05 \times \alpha(i+1) + 0,9 \times \big[\alpha(i+1) - \alpha(i)\big]\big) \tag{10}$$

4.3. Simulación del modelo

Una vez desarrollado el modelo físico que describe el comportamiento dinámico del vehículo, se construye un entorno de simulación con el editor de *Virtual Reality Worlds* de Simulink. Puesto que el diseño gráfico no está entre los objetivos principales del presente TFG, se han seguido básicamente las indicaciones ofrecidas en [6], que resultan en un coche muy sencillo circulando por una carretera recta. Sí merece la pena destacar dos particularidades en este aspecto. Por un lado, se ha aumentado notablemente el ancho sugerido para la carretera. Esto permite una mayor maniobrabilidad del vehículo, lo cual será especialmente relevante en la sección posterior. Por otro, con el mero propósito de que la imagen del automóvil simulado fuera lo más parecida posible a un Tesla Model 3, se ha añadido color al cuerpo del coche: gris claro para las ruedas, simulando las llantas metálicas; rojo para el cuerpo central, como carrocería; y negro con transparencia en la parte superior, por ser esta donde se encuentran las lunas tintadas y el techo panorámico de cristal. El resultado se muestra a continuación:

Figura 6: Montaje comparativo entre la imagen del vehículo simulado por ordenador (izquierda) y una fotografía del Tesla Model 3 real (derecha).

Existen dos dificultades principales a la hora de reflejar el trabajo realizado hasta ahora con MATLAB en una simulación. En primer lugar, se debe llevar a cabo la implementación de las ecuaciones de movimiento en la simulación de manera iterativa. Esto se logra mediante el empleo del comando "world", que va actualizando lo que muestra la pantalla según los cálculos del ordenador vayan indicando. Además de lo ya expuesto anteriormente, en la simulación se ha añadido una particularidad al coeficiente de rozamiento mecánico. Puesto que el entorno virtual empleado consta de dos partes diferenciadas, carretera y césped, se ha decidido modificar el valor del coeficiente de rozamiento cuando el vehículo sale de la vía asfaltada. Así, se ha decidido tomar $\mu = 2000\frac{Ns}{m}$ para reflejar la dificultad del avance del coche fuera de la carretera. Este valor numérico no responde a un cálculo explícito, sino simplemente a la intención de que la simulación tenga comportamientos diferenciados en cada uno de los terrenos que muestra.

En segundo lugar, la dificultad principal que entraña una simulación de este tipo está relacionada con la interacción entre el usuario y el coche. Para que esta sea sencilla y efectiva, debe ocurrir en tiempo real y permitir cuatro acciones diferenciadas: acelerar, frenar, girar las ruedas a la derecha y girar las ruedas a la izquierda. Esto se corresponde con una intervención directa sobre los parámetros F_m y α de las ecuaciones anteriormente expuestas.

En este sentido, el texto de Khaled [6] propone la utilización de un *joystick* de la marca Logitech. Puesto que no dispongo del mismo, en un primer momento traté de implementar una entrada por teclado que permitiera controlar el vehículo simulado a través de las flechas de dirección. Para ello, experimenté con varias combinaciones de código incluyendo las funciones "waitforbuttonpress" y "getkey", así como el bloque de Simulink que recoge entradas de teclado. Ninguno de estos métodos resultó completamente efectivo, ya que *Virtual Reality Worlds* emplea las flechas para ajustar la vista de cámara, lo cual hace que la integración entre el teclado y el control de la simulación sea problemática. En el mejor de los casos, conseguí que el entorno simulado avanzase a saltos, congelándose en el intervalo de tiempo entre la pulsación de una tecla y la siguiente. Esto resultaba tremendamente incómodo y, sobre todo, no reflejaba adecuadamente el trabajo previo de estudio de las ecuaciones de movimiento que sigue el coche, motivo por el cual se desechó como opción.

Finalmente, Ricardo Martín Aldazábal, compañero que también está realizando su TFG sobre sistemas dinámicos y con el que había compartido pesquisas acerca de métodos de intracción con la simulación por teclado, dio con la solución. Existe un programa externo a Matlab llamado SCP Toolkit que permite instalar controladores en un ordenador de sistema operativo Windows para conectar al mismo un mando *Dualshock3*, de *PlayStation*. Tras instalar este programa, además de algunos otros requeridos para hacerlo funcionar[2], pude emplear mi viejo mando de *PlayStation* para manejar el coche simulado. Si bien, evidentemente, esto no puede plasmarse sobre una memoria en papel, en la defensa del TFG haré una demostración, que me permitirá enseñar tanto el manejo básico de la dirección del coche como el detalle del coeficiente de rozamiento mecánico que cambia entre la carretera y el césped.

[2] A título informativo, estos programas complementarios fueron Microsoft .NET Framework 4.5, Microsoft Visual C++ 2010 Redistributable Package, Microsoft Visual C++ 2013 Runtime y DirectX Runtime.

4.4. Hacia una conducción autónoma

Si hace 50 años se hubiera hecho una encuesta acerca de cómo cambiarían los vehículos en el futuro, seguramente una de las respuestas más repetidas habría sido "coches voladores". Si bien parece que todavía estamos lejos de disponer de una tecnología que lo permita, sí que parece mucho más posible en el medio plazo que los automóviles funcionen de manera autónoma. En enero de 2020, el Departamento de Transporte de Estados Unidos publicó un informe titulado *Ensuring American Leadership in Automated Vehicle Technologies* [2]. En este documento se recoge la estrategia que pretende seguir el gobierno estadounidense para fomentar la investigación enfocada a conseguir la automatización de los vehículos de transporte privado.

Además de la importancia que tendría para el posicionamiento internacional del país que las empresas que desarrollasen este tipo de vehículos fueran estadounidenses, el informe destaca dos grandes posibles beneficios de la conducción autónoma. Por un lado, el impacto económico sobre ciertas actividades sería muy positivo. Asi, por ejemplo, el sector agrario podría operar de forma mucho más productiva si los tractores no necesitasen de un conductor para funcionar. Por otra parte, se podría dar una reducción sustancial de la siniestralidad en carreteras. Personalmente, considero que este último es el factor más interesante y que puede llevar al apoyo de los poderes públicos para el desarrollo de tecnologías de conducción autónoma.

De hecho, si atendemos a las cifras más recientes presentadas por Tesla, correspondientes al primer trimestre de 2021, los accidentes son menos frecuentes cuando sus vehículos circulan con los sistemas de asistencia a la conducción activados [11]. En resumen, sus datos para Estados Unidos muestran que:

- Un automóvil genérico sufre de media un accidente por cada 484.000 millas recorridas.
- Un automóvil marca Tesla sufre de media un accidente por cada 978.000 millas recorridas en conducción manual.
- Un automóvil marca Tesla sufre de media un accidente por cada 4.190.000 millas recorridas con Autopilot activado (el sistema de asistencia a la conducción más avanzado de la compañía).

Por tanto, se podría concluir que el sistema de conducción autónoma de Tesla reduce entre 4 y 9 veces el riesgo de accidente. No obstante, estos números deben ser tomados con ciertas reservas. Es evidente que a cualquier empresa automovilística le interesa que sus potenciales clientes consideren que sus coches son muy seguros. En cualquier caso, las cifras anteriores sí que nos indican una posible tendencia para el futuro cercano: un ordenador puede llegar a conducir de manera más segura que un ser humano.

Todo lo descrito anteriormente me sirvió como motivación para plantear el último apartado de mi TFG. Así, quiero que mi aportación original final sea diseñar un sistema sencillo de conducción autónoma que permita al coche reaccionar a obstáculos en el entorno de la simulación. Estos obstáculos tomarán la forma de conos de color naranja, similares a los empleados para las señales circunstanciales en carretera, que aparecerán en la trayectoria del vehículo. El objetivo es que el coche reduzca su velocidad, esquive el cono y continúe la marcha sin necesidad de intervención por mi parte en la simulación.

El primer problema con el que me encontré para simular la conducción autónoma es cómo hacer que los obstáculos "aparezcan". Al fin y al cabo, si hubiera colocado conos en lugares concretos de la carretera, estaría realizando un ejercicio de guiado del vehículo, pero no existiría reacción como tal. Por tanto, surge la necesidad de introducir en la simulación elementos de carácter estocástico. A continuación detallo cómo he gestionado esto para proporcionar cierta dosis de imprevisibilidad a la simulación.

La función de MATLAB que he empleado es "randn", que al ejecutarse devuelve un número perteneciente a una distribución normal de varianza la unidad. De esta forma, en cada iteración del modelo, además de recalcular la posición y velocidad del vehículo, se obtiene un número aleatorio p_a en las condiciones anteriores. Si este número cumple una condición dada, se ejecuta una instrucción sobre el entorno virtual que hace aparecer un cono en el camino del vehículo. En mi caso, opté por trabajar con la condición $p_a > 2$. En la distribución de la cual se están tomando los valores de p_a, esto ocurre con una probabilidad aproximada del $2,3\,\%$, puesto que corresponde a que p_a pertenezca a la cola derecha, estando situado a más de dos desviaciones típicas de la media.

Así, en cada iteración hay una pequeña probabilidad de que aparezca un obstáculo, lo cual se corresponde con un experimento de Bernoulli. Como cada iteración transcurre en $0,1$ segundos, en realidad la aparición del cono no se demora mucho en el tiempo. La suma de distribuciones de Bernoulli correspondientes a cada iteración genera una distribución binomial [8]. Podemos calcular la probabilidad de que haya aparecido el cono en el camino del vehículo tras haber esperado un cierto tiempo teniendo en cuenta que para cada iteración se tiene $P_{Cono\ aparece\ en\ iteración\ i} = p \times (1-p)^{i-1}$, con $p = 0,023$. Los resultados se muestran en la siguiente figura.

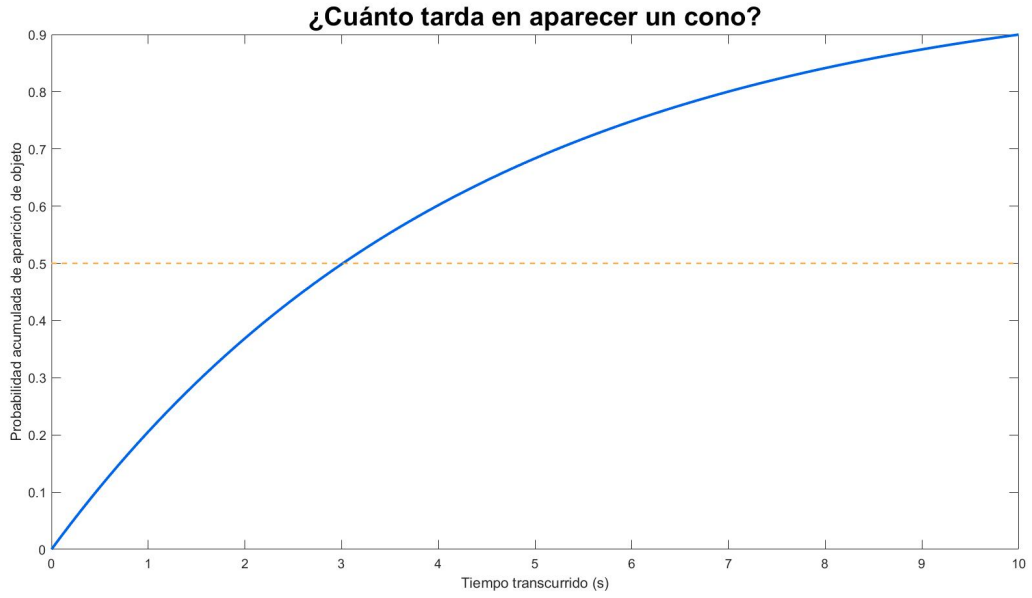

Figura 7: Probabilidad en tanto por uno de que, para cierto tiempo dado t, haya aparecido ya el cono en la simulación. La línea horizontal naranja indica la probabilidad 1/2.

Tal y como se puede observar en el gráfico, más de la mitad de las veces el cono aparece tras una espera inferior a tres segundos. De hecho, en el $90\,\%$ de las ocasiones esta demora es inferior a 10 segundos. Estos datos nos sirven para comprobar que la simulación tendrá una duración razonable, siendo breve pero no demasiado rápida.

Una vez aparece el cono, el coche aminora su velocidad y gira las ruedas hacia el centro de la calzada para cambiar de carril. Este giro es progresivo, puesto que en cada iteración existe un tope para la variación del ángulo α que aparece en la ecuación 10. Mientras esto ocurre, en cada iteración el coche mantiene controlada tanto su posición con respecto al centro de la calzada (tomado en $x = 0$) como su orientación con respecto al sentido de la carretera. Esto permite que, cuando el centro geométrico del automóvil supera este punto, se invierta el sentido de giro para no abandonar la calzada. Igualmente, una vez se ha recuperado la dirección deseada, el coche deja de girar y continúa hacia delante.

Con el fin de mostrar que esta secuencia puede repetirse en condiciones diferentes, cuando se supera el obstáculo vuelve a ponerse en marcha el generador de números aleatorios. De la misma manera que se ha descrito anteriormente, acabará apareciendo un nuevo cono interponiéndose en el camino del vehículo. El coche de la simulación, de manera autónoma, realizará los mismos giros en sentido contrario para superarlo. Se puede ver una muestra de todo este proceso en el anexo, que se encuentra al final de este memoria. Se ha ubicado allí porque ocupa una página completa y podría suponer un impedimento para la lectura. Dicho anexo consta de una serie de diez imágenes tomadas durante la simulación que permiten hacerse una idea del comportamiento del vehículo ante la aparición de obstáculos sucesivos. Durante la defensa del TFG, para una mejor visualización por parte del tribunal, esta simulación se efectuará en directo.

Esta última sección recoge una simulación muy sencilla de conducción autónoma. Se podrían haber incluido más obstáculos para que el vehículo repitiera los mismos movimientos, pero el objetivo principal no era tanto ver al coche hacer movimientos de zigzag continuados, sino mostrar una versión simplificada de conducción sin intervención humana. Esto se ha logrado teniendo como componente fundamental la generación de números aleatorios por parte de MATLAB, de manera que los conos no pudieran ser previstos por el coche hasta su aparición.

Cabe destacar que el Anexo 2 que acompaña a esta memoria recoge la parte fundamental del código de MATLAB empleado. Está dividido en tres secciones: ecuaciones del movimiento en línea recta, empleadas para construir las figuras 2, 3 y 4; simulación de la conducción manual, que utiliza *joystick*; y simulación de la conducción autónoma, con la aparición sucesiva de conos. Dicho código va acompañado de indicaciones para faciliar su lectura, la cual puede ser útil con el fin de clarificar cómo se ha llevado a la práctica la implementación de todo lo anteriormente descrito para las diferentes simulaciones.

5. Conclusiones

El presente TFG ha consistido en el modelado y simulación del movimiento de un vehículo. Esto ha requerido de la ejecución ordenada de una serie de pasos. Así, se ha planteado el problema y el tipo de modelo que se usaría como solución en la sección 1. Posteriormente, se han definido y motivado una serie de ecuaciones diferenciales en la sección 4.1, que han sido resueltas mediante las herramientas descritas en el apartado 3.1. A partir de estas soluciones, y tras aplicar en 4.2 los ajustes necesarios para introducir los giros de volante, se ha construido una simulación en dos escenarios diferenciados: el control estándar de un coche (sección 4.3) y su conducción autónoma (sección 4.4).

Uno de los pilares sobre los que se asienta el trabajo es la información presente en la sección 3.2. Más allá de que la elección del Tesla Model 3 haya supuesto una excusa para introducir la conducción autónoma como uno de los elementos de la simulación, la razón de ser de cualquier modelo es asemejarse lo más posible a aquello que intenta reproducir. Precisamente, este el el motivo por el cual cobran tanta importancia los valores numéricos reales del vehículo elegido. A partir de ellos hemos podido ajustar los parámetros libres del modelo, tal y como ha quedado reflejado en la tabla presente en el apartado 4.1. De esta manera se ha reproducido el tiempo de aceleración desde 0 hasta 100 km/h y la velocidad máxima del coche. Sin embargo, lo verdaderamente destacable es el resultado obtenido para la prueba del cuarto de milla, donde se ha llegado a valores muy próximos a los ofrecidos por el Tesla Model 3 en pruebas reales. Esta adecuación es consecuencia de un buen ajuste del modelo pues, al contrario que para la velocidad máxima, no se ha incluido directamente en el mismo *ad hoc*. Es por esto precisamente que una predicción tan cercana al comportamiento real supone un gran éxito del modelo empleado.

También considero que otro de los puntos fuertes de este trabajo ha sido el desarrollo de las simulaciones. Aunque su diseño a nivel gráfico sea básico, permiten replicar adecuadamente en un entorno virtual cómo se mueve el vehículo. En la primera simulación, el control del movimiento del coche con un mando estándar de *PlayStation* permite interactuar muy fácilmente con el modelo. Así, se pueden comprobar de manera sencilla características como el diferente coeficiente de rozamiento mecánico en función del terreno por el que se circula o la limitación de velocidad para el coche cuando se mueve marcha atrás. La segunda simulación supone una primera aproximación al funcionamiento de un vehículo autónomo. La escritura del código que permite al coche sortear los obstáculos me ha resultado un pequeño pero interesante desafío.

Dicho esto, es necesario mencionar también aquellos aspectos que podrían ser mejorados del trabajo. En numerosas ocasiones, se tiene una idea en mi opinión equivocada sobre los objetivos de la Física. Se habla de la capacidad de medir valores concretos con exactitud, cuando en realidad a lo que aspira la Ciencia es a acotar el error cometido en la medición. De esta forma, y con cierta dosis de juego lingüístico, se podría afirmar que lo más importante no es tanto saber cuánto mide aquello que se está estudiando, sino cuánto *no* mide. Aplicando esto a mi TFG, resulta muy importante destacar las limitaciones más importantes del modelo empleado, es decir, aquellos factores que no se están teniendo en cuenta a la hora de describir el movimiento del coche simulado. Solamente así se puede entender hasta qué punto los resultados obtenidos a través de él son válidos y proponer formas de mejorarlo.

En primer lugar, cabe destacar que la fuerza motriz que impulsa al coche cuando éste acelera a fondo, si bien efectivamente es casi constante, no adquiere su valor de forma instantánea en $t = 0$. De hecho, esto sería contraproducente, pues podría provocar que las ruedas del vehículo

patinasen. Por ello, lo que ocurre en realidad es que la fuerza proporcionada por el motor crece de forma rápida pero continua desde 0 hasta un valor pico, que se ha denotado F_m^+, que se mantiene constante. En cualquier caso, el hecho de que los motores eléctricos entreguen el par total muy rápidamente en comparación con el orden de tiempo característico de la dinámica del vehículo posibilita modelizar esto como una función escalón. Además, los ajustes realizados a partir de las especificaciones reales del Tesla Model 3 contribuyen a que esta precisión técnica sea poco importante de cara al comportamiento de la velocidad del coche. El modelo empleado tampoco permite pisar el acelerador a medias, pero esto no supone un problema para controlar el coche con el *joystick*, tal y como se ha comprobado en la simulación.

En segundo lugar, se debe tomar la ausencia de consideración del efecto de la gravedad como crítica más relevante al realismo del modelo. Por simplicidad, se ha decidido no incluir rampas en el entorno de simulación, que es completamente plano. Si se hubiera querido tener en cuenta, habría sido necesario incluir en las ecuaciones de la dinámica del vehículo presentadas en la sección 4.1 un término de la forma $-mg\,\mathrm{sen}\,(\eta)$, siendo g la aceleración de la gravedad y η la inclinación del terreno. El signo negativo indica que esta fuerza, consecuencia de la componente horizontal del peso, se opone al movimiento si la pendiente η es positiva (cuesta arriba) y contribuye a aumentar la velocidad en caso contrario (cuesta abajo). De hecho, una consecuencia interesante de esto es que posibilita rebasar la velocidad máxima homologada.

Por último, resulta evidente que el apartado final de la sección correspondiente al desarrollo del trabajo, en la que se explica la simulación de conducción autónoma, es muy simple. En realidad, la dificultad principal de diseñar algoritmos de conducción es conseguir que su capacidad de reacción ante elementos inesperados sea adecuada. Esto es complicado de reproducir en un entorno creado por ordenador, que carece de sorpresas en sentido estricto. En el caso de la simulación que nos ocupa, este carácter imprevisto se ha tratado de introducir a través de los números aleatorios que generan la aparición de obstáculos, lo cual es bastante limitado. De todas formas, debe entenderse que la pretensión de mi TFG es solamente mostrar una iteración cero del diseño de un método de conducción autónoma.

Como cierre a esta memoria, querría expresar que considero que este Trabajo de Fin de Grado constituye una finalización muy satisfactoria de mis estudios de Física. En ocasiones, la acumulación de diferentes asignaturas no deja tiempo para profundizar en ellas, problema que sufrimos especialmente los estudiantes de doble grado. Considero que esto no ha ocurrido en el TFG, que me ha permitido trabajar a mi propio ritmo, aplicando de manera directa contenidos de distintas materias. Las principales, dada la temática elegida, han sido Mecánica Clásica y Sistemas Dinámicos y Realimentación. Sin embargo, también hay en este trabajo elementos destacables de otras como Laboratorio de Computación Científica, Ecuaciones Diferenciales o Laboratorio de Física. Por todo ello, estoy contento con el resultado final.

Bibliografía

[1] Karl Johan Astrom y Richard M. Murray. *Feedback Systems, an introduction for scientists and engineers*. Princeton University Press, 2019, pág. 73. ISBN: 978-0691193984.

[2] National Science Technology Council y United States Department of Transportation. *Ensuring American Leadership in Automated Vehicle Technologies*. U.S. Government, 2020.

[3] Thomas D. Gillespie. *Fundamentals of Vehicle Dynamics*. Society of Automotive Engineers, Inc., 1992, pág. 47. ISBN: 978-1560911999.

[4] Sigal Gottlieb, David Ketcheson y Shi Wang Chu. *Strong stability preserving Runge-Kutta and multistep time discretizations*. World Scientific Publishing Co. Pte. Ltd., 2011, pág. 12. ISBN: 978-9814289269.

[5] Sigal Gottlieb, David Ketcheson y Shi Wang Chu. *Strong stability preserving Runge-Kutta and multistep time discretizations*. World Scientific Publishing Co. Pte. Ltd., 2011, pág. 13. ISBN: 978-9814289269.

[6] Nassim Khaled. *Virtual Reality and Animation for MATLAB and Simulink Users*. Springer-Verlag London Limited, 2012, págs. 49-50. ISBN: 978-1-4471-2329-3.

[7] Charles Kittel, Walter D. Knight y Malvin A. Ruderman. *Mecánica - Berkeley Physics Course - Volumen 1*. Editorial Reverté, S.A., 1968, pág. 205. ISBN: 84-291-4021-2.

[8] William Mendenhall, Robert J. Beaver y Barbara M. Beaver. *Introducción a la probabilidad y estadística*. Cengage Learning Editores S.A., 2010, pág. 208. ISBN: 978-607-481-466-8.

[9] MotorTrend. *TESTED: The Tesla Model 3 Long Range Dual Motor is Quicker Than You Think*. 2019. URL: https://www.motortrend.com/cars/tesla/model-3/2018/2018-tesla-model-3-long-range-dual-motor-first-test-review/#:~:text=Nothing%20demonstrates%20that%20new%20reality,376%20lb%2Dft%20of%20torque. (visitado 09-05-2021).

[10] Ultimate Specs. *Tesla Model 3 Long Range AWD Specs*. 2021. URL: https://www.ultimatespecs.com/car-specs/Tesla/122589/2021-Tesla-Model-3-Long-Range-AWD.html (visitado 29-04-2021).

[11] Inc. Tesla. *Tesla Vehicle Safety Report*. 2021. URL: https://www.tesla.com/es_ES/VehicleSafetyReport?redirect=no (visitado 06-05-2021).

Anexo 1: Imágenes de conducción autónoma

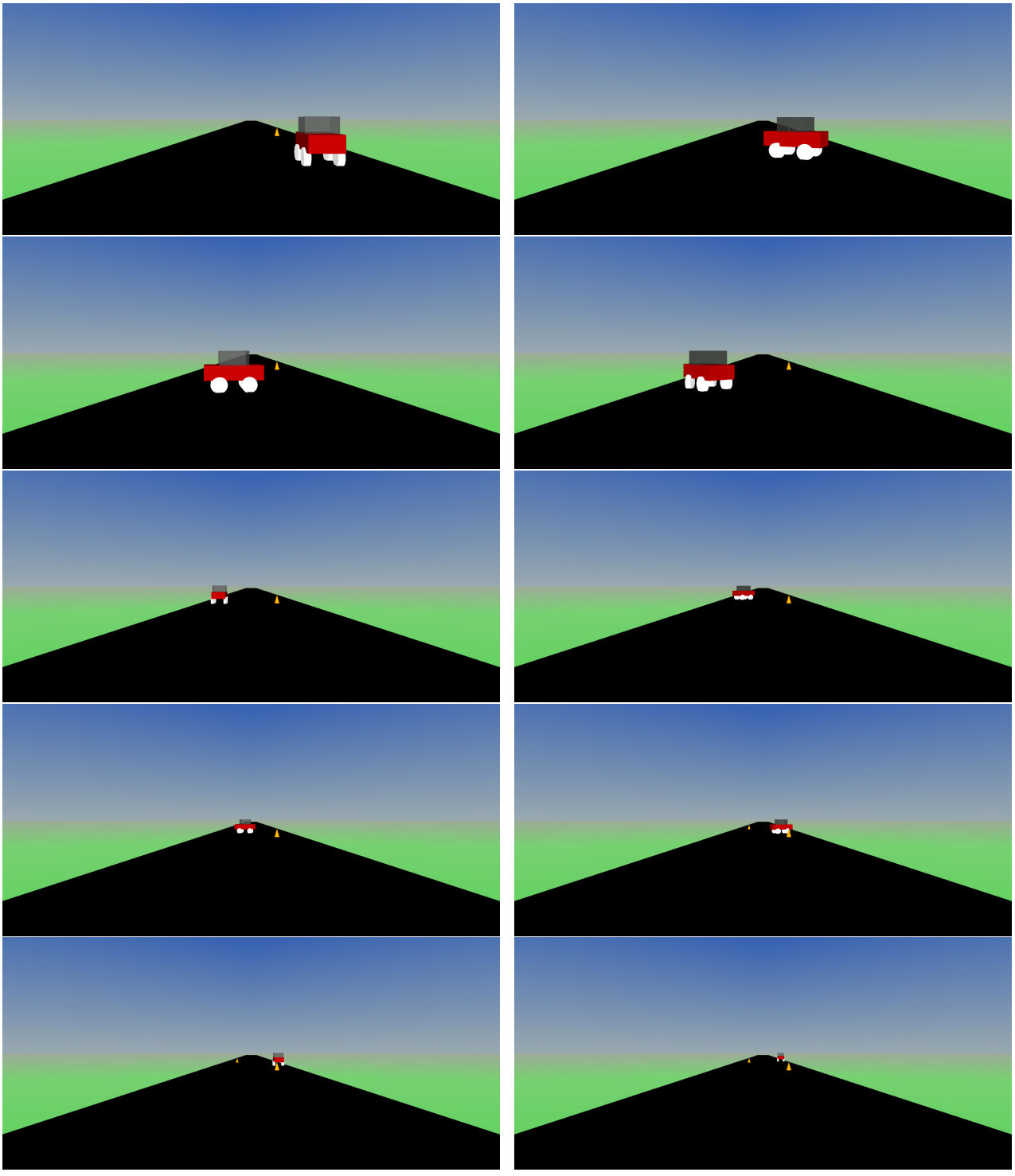

Figura 8: Secuencia de imágenes que muestran el coche de la simulación en conducción autónoma. Observándolas de izquierda a derecha y de arriba a abajo, representan cronológicamente cómo el vehículo sortea dos conos que aparecen en la calzada, todo ello sin necesidad de intervención en la simulación por parte del usuario.

Anexo 2: Código de MATLAB empleado

Ecuaciones de movimiento para construcción de gráficas

```
%Datos del coche
m=1831+60;
mu1=223;
Cx=0.23;
Fimp1=15400;
h=0.1;

%Condiciones iniciales
Position_Car_i=[0;0;0];
V_i=0;
betta_i=pi;
theta_i=0;
ts = 0:h:100;
y = zeros(1,length(ts));
y(1) = V_i;
x = zeros(3,length(ts));
x(:,1)=Position_Car_i;

for i=1:(length(ts)-1)
    k_1 = Fcar(Fimp1, mu1,y(i), m, Cx);
    k_2 = Fcar(Fimp1, mu1,y(i)+0.5*h*k_1, m, Cx);
    k_3 = Fcar(Fimp1, mu1,y(i)+0.5*h*k_2, m, Cx);
    k_4 = Fcar(Fimp1, mu1,y(i)+h*k_3, m, Cx);

    y(i+1) = y(i) + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h; %main equation
    if y(i+1)<-6
        y(i+1)=-6;
    end

    theta_i1=theta_i;
    if theta_i1>pi*0.24
        theta_i1=pi*0.24;
    elseif theta_i1<-pi*0.24
        theta_i1=-pi*0.24;
    end

    if (betta_i^2-pi^2)<0.001
        Velocity_Car1=[1 0 0;0 1 0;0 0 1]*[0;0;y(i)];
    else
        Velocity_Car1=[cos(betta_i) 0 sin(betta_i);0 1 0;
        -sin(betta_i) 0 cos(betta_i)]*[0;0;y(i)];
    end

    x(:,i+1)=x(:,i)+h.*Velocity_Car1;
    betta_i1=betta_i+0.05*theta_i1*y(i)+0.9*(theta_i1-theta_i)*y(i);
```

```
    betta_i=betta_i1;
    theta_i=theta_i1;
end
ykm=y*3.6;

%Se elaboran las figuras con los datos del movimiento
figure()
plot(ts(1:200),ykm(1:200), 'LineWidth',2,'Color',[50,100,255]/255)
hold on
plot([0,4.3],[100,100],'--','LineWidth',1,'Color',[255,154,32]/255)
plot([4.3,4.3],[0,100],'--','LineWidth',1,'Color',[255,154,32]/255)
xlabel('Tiempo desde arranque (s)')
ylabel('Velocidad (km/h)')
title('Prueba de aceleraci n corta', 'FontSize', 20)

figure()
plot(ts,ykm, 'LineWidth',2,'Color',[50,100,255]/255)
hold on
plot([0,100],[233,233],'-r','LineWidth',1.25)
plot([0,4.4],[100,100],'-k','LineWidth',0.75)
plot([4.4,4.4],[0,100],'-k','LineWidth',0.75)
xlabel('Tiempo desde arranque (s)')
ylabel('Velocidad (km/h)')
legend('Evoluci n temporal de la velocidad', 'Velocidad m xima',
'Location','southeast', 'FontSize', 14)
title('Prueba de aceleraci n larga', 'FontSize', 20)

figure()
plot(ts(1:151),x(3,1:151), 'LineWidth',2,'Color',[50,100,255]/255)
hold on
plot([0,12.5],[402.34,402.34],'--','LineWidth',1,'Color',[255,154,32]/255)
plot([12.5,12.5],[0,402.34],'--','LineWidth',1,'Color',[255,154,32]/255)
xlabel('Tiempo desde arranque (s)')
ylabel('Distancia recorrida (m)')
title('Prueba del cuarto de milla', 'FontSize', 20)
legend('Distancia recorrida en funci n del tiempo',
'Marca del cuarto de milla','Location','northwest', 'FontSize', 14)

function Ffin=Fcar(F,mu,v,m,Cx)
Ffin=(1/m)*(F-mu*v-Cx*v*abs(v));
```

Simulación en conducción manual

```
%Se inicia el mundo virtual
world = vrworld('Car3.WRL');
open(world);
fig = view(world, '-internal');

%Datos del modelo
m=1831+60;
mu1=223;
Cx=0.23;
Fimp1=15400;
delta_t=0.1;

%Condiciones iniciales
Position_Car_i=[0;0;0];
V_i=0;
betta_i=pi;
F_i=0;
theta_i=0;

%Se introduce el joystick
joy = vrjoystick(1);

for t=0:0.1:100
    %Comando pause para que el entorno no vaya demasiado deprisa
    pause(.05);

    %El joystick entra mediante un vector de dos componentes:
    %a(1) para girar el volante, con a(1)=-1 para izquierda
    y a(1)=1 para derecha.
    %a(2) para acelerar y frenar, con a(1)=-1 si joystick hacia arriba
    y a(1)=1 hacia abajo.

    a = axis(joy, [1 2]);

    theta_i1=theta_i -0.01*round(a(1));

    if theta_i1>pi*0.24
        theta_i1=pi*0.24;
    elseif theta_i1<-pi*0.24
        theta_i1=-pi*0.24;
    end

    F_i1=-Fimp1*round(a(2));
    x=Position_Car_i(1);

    %Ajuste del coeficiente de rozamiento si el coche sale de la carretera
    if x^2>16
```

```
        mu1=2000;
    else
        mu1=223;
    end

    %Runge-Kutta
    k_1 = Fcar(F_i1, mu1,V_i, m, Cx);
    k_2 = Fcar(F_i1, mu1,V_i+0.5*h*k_1, m, Cx);
    k_3 = Fcar(F_i1, mu1,V_i+0.5*h*k_2, m, Cx);
    k_4 = Fcar(F_i1, mu1,V_i+h*k_3, m, Cx);

    V_i1 = V_i + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h; %main equation
    if V_i1<-6
        V_i1=-6;
    end

    Velocity_Car1=[cos(betta_i) 0 sin(betta_i);0 1 0;
    -sin(betta_i) 0 cos(betta_i)]*[0;0;V_i1];
    Position_Car_i1=Position_Car_i+delta_t.*Velocity_Car1;
    betta_i1=betta_i+0.05*theta_i1*V_i1+0.9*(theta_i1-theta_i)*V_i1;

    %Transformaciones introducidas en entorno virtual
    world.transform_car.translation=(Position_Car_i1)';
    world.transform_car.rotation=[0 1 0 betta_i1];
    world.front_right_wheel.rotation=[1 0 0 theta_i1];
    world.front_left_wheel.rotation=[1 0 0 theta_i1];

    %Nuevas condiciones iniciales
    Position_Car_i=Position_Car_i1;
    V_i=V_i1;
    betta_i=betta_i1;
    theta_i=theta_i1;
    F_i=F_i1;

    %Se dibuja en entorno virtual lo anterior
    vrdrawnow;
end
```

Simulación en conducción autónoma

```
%Se inicia el mundo virtual
world = vrworld('car3_cone1.WRL');
open(world);
fig = view(world, '-internal');

%Datos del modelo
m=1831+60;
mu1=223;
Cx=0.23;
Fimp1=15400;
C=0;
p=0;
q=0;
v_obj=0;
delta_t=0.1;
h=delta_t;

%Condiciones iniciales
Position_Car_i=[3;0;0];
V_i=0;
betta_i=pi;
F_i=0;
theta_i=0;
a=[0  0];

for t=0:0.1:300
    theta_i1=theta_i-0.04*round(a(1));
    if theta_i1>pi*0.24
        theta_i1=pi*0.24;
    elseif theta_i1<-pi*0.24
        theta_i1=-pi*0.24;
    end

    F_i1=Fimp1*round(a(2));
    x=Position_Car_i(1);

    if x^2>16
        mu1=2000;
    else
        mu1=223;
    end

    k_1 = Fcar(F_i1, mu1,V_i, m, Cx);
    k_2 = Fcar(F_i1, mu1,V_i+0.5*h*k_1, m, Cx);
    k_3 = Fcar(F_i1, mu1,V_i+0.5*h*k_2, m, Cx);
    k_4 = Fcar(F_i1, mu1,V_i+h*k_3, m, Cx);
```

```
V_i1 = V_i + (1/6)*(k_1+2*k_2+2*k_3+k_4)*h; % main equation
if V_i1<-6
    V_i1=-6;
end

Velocity_Car1=[cos(betta_i) 0 sin(betta_i);0 1 0;
-sin(betta_i) 0 cos(betta_i)]*[0;0;V_i1];
Position_Car_i1=Position_Car_i+delta_t.*Velocity_Car1;
betta_i1=betta_i+0.05*theta_i1*V_i1+0.9*(theta_i1-theta_i)*V_i1;

% Transformaciones introducidas en entorno virtual
world.transform_car.translation=(Position_Car_i1)';
world.transform_car.rotation=[0 1 0 betta_i1];
world.front_right_wheel.rotation=[1 0 0 theta_i1];
world.front_left_wheel.rotation=[1 0 0 theta_i1];

% Inicialmente los conos se encuentran debajo de la carretera
if t==0
    world.Cone1.translation=[2 -1 -10];
    world.Cone2.translation=[-2 -1 -10];
end

if C==0 & t>30
    v_obj=(50/3.6)/30;
    p=randn;
elseif C==1
    v_obj=(50/3.6)/80;
end

% Se pisa el acelerador si no se ha llegado a la velocidad objetivo
if V_i1<v_obj
    a(2)=1;
else
    a(2)=0;
end

% Aparece el primer cono
if p>2
    world.Cone1.translation=[3 0.5 Position_Car_i1(3)-20];
    a(1)=-1;
    a(2)=-1;
    t_paso=t+50;
    C=1;
    p=0;
end

% Ajuste del giro del coche para sortear el cono
if C==1 & x<0
    a(1)=1;
```

```
        if (betta_i1-pi)^2<0.01
            a(1)=0;
            betta_i1=pi;
            theta_i1=0;
            C=2;
            v_obj=(50/3.6)/30;
        end
    end

    if C==2 & t>t_paso
        q=randn;
    elseif C==3
        v_obj=(50/3.6)/80;
    end

    %Aparece el segundo cono
    if q>2
        world.Cone2.translation=[-3 0.5 Position_Car_i1(3)-20];
        a(1)=1;
        a(2)=-1;
        C=3;
        q=0;
    end

    if C==3 & x>0
        a(1)=-1;
        if (betta_i1-pi)^2<0.01
            a(1)=0;
            betta_i1=pi;
            theta_i1=0;
            v_obj=(50/3.6)/20;
            C=4;
        end
    end

    %Nuevas condiciones iniciales
    Position_Car_i=Position_Car_i1;
    V_i=V_i1;
    betta_i=betta_i1;
    theta_i=theta_i1;
    F_i=F_i1;

    %Se dibuja en el entorno virtual lo anterior
    vrdrawnow;

end
```