

Trabajo de Fin De Grado
Grado en Ingeniería de Tecnologías Industriales

Diseño de interfaz en Matlab para optimizar la
subida y bajada en cuesta

Autor: Jose Antonio Rodríguez Rodríguez

Tutor: Johan Wideberg

**Dpto. de Ingeniería y Ciencia de los Materiales y
del Transporte
Universidad de Sevilla**

Sevilla, 2020



Trabajo de Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

Diseño de interfaz en Matlab para optimizar la subida y bajada en cuesta

Autor:

Jose Antonio Rodríguez Rodríguez

Tutor:

Johan Wideberg

Profesor titular

Dpto. de Ingeniería y Ciencia de los Materiales
y del Transporte

Universidad de Sevilla

Sevilla, 2020

Trabajo de Fin de Grado: Diseño de interfaz en Matlab para optimizar la subida y bajada en cuesta

Autor: Jose Antonio Rodríguez
Rodríguez

Tutor: Johan Wideberg

El tribunal nombrado para juzgar el Trabajo arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2020

El Secretario del Tribunal

A mi familia
A mis maestros

Agradecimientos

A mis padres, por haber confiado siempre en mí y animarme a no tirar nunca la toalla confiando siempre en mí. Por su cariño, por apoyarme en todos los sentidos, haciendo que nunca me faltara de nada y queriendo siempre lo mejor para mí, apoyando cada decisión que tomaba y ayudándome cada día a ser mejor como persona y a alcanzar mis objetivos y metas, porque sin ellos esto no habría sido posible, este trabajo de fin de grado también es de ellos.

A mi abuela Rafa y mi abuelo Pepe, y a mi tía Rocío y mi primo Carlos. Por estar siempre ahí, interesándose por como me iba en la carrera en todo momento y apoyándome desde la distancia con su cariño y apoyo, este trabajo también va por vosotros.

A Nahikari, por haber estado ahí siempre a pesar de la distancia, mostrándome su cariño constantemente y motivándome a seguir adelante, este trabajo también va por ti.

A todos los docentes y profesores que me han guiado a lo largo de todo el grado de Ingeniería en Tecnologías Industriales. En especial, a los de la rama de Mecánica-Maquinas que me han hecho apasionarme por este campo tan bonito de la Ingeniería. Especial mención a Johan Wideberg, mi tutor en este trabajo de fin de grado, que a pesar de las circunstancias tan difíciles en las que nos encontrábamos, ha estado ahí para resolverme cualquier duda que tuviera.

Jose Antonio Rodríguez Rodríguez

Sevilla, 2020

Resumen

En este trabajo de fin de grado se ha diseñado una interfaz de usuario para optimizar la subida y bajada en cuesta de un automóvil con motor de combustión interna. Para llevar a cabo la interfaz se han tenido en cuenta los conceptos teóricos relativos a la dinámica longitudinal del automóvil, el sistema de transmisión del automóvil y el consumo específico de combustible. El diseño de la interfaz se ha llevado a cabo en el software MATLAB, es por ello que se explicarán los conceptos más importantes de este software en lo que a programación de objetos se refiere.

La interfaz diseñada constará de una ventana inicial, un menú y tres ventanas gráficas adicionales más a las cuales se accede a través del menú. En cada una de estas tres ventanas adicionales el usuario puede calcular algunos de los parámetros más importantes en la optimización de la subida y bajada en cuesta de un automóvil como son la pendiente máxima superable, la velocidad máxima admisible y el consumo óptimo de combustible.

Como muestra del funcionamiento de la interfaz se obtendrán resultados en hasta tres modelos de automóviles para cada uno de los parámetros citados, y se analizarán dichos resultados para ver qué sucede y obtener unas conclusiones finales.

Abstract

In this end-of-degree work a user interface has been designed to optimize the climb and descent in slope of a car with internal combustion engine. The interface has taken into account theoretical concepts relating to the longitudinal dynamics of the car, the car transmission system and the specific fuel consumption. The interface design has been carried out in the MATLAB software, which is why the most important concepts of this software will be explained in terms of object programming.

The designed interface will consist of an initial window, a menu and three additional graphical windows which are accessed through the menu. In each of these three additional windows the user can calculate some of the most important parameters in the optimization of the climb and descent in slope such as the maximum surmountable slope, the maximum permissible speed and the optimal fuel consumption.

As a sample of the operation of the interface, results will be obtained in up to three car models for each of the above parameters, and these results will be analysed to see what happens and to obtain final conclusions.

Índice

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Índice	xv
Índice de Tablas	xviii
Índice de Figuras	xx
Notación	xxii
1 Introducción	1
1.1 <i>Motivación y objetivos</i>	1
1.2 <i>Reseña histórica</i>	1
1.3 <i>Estructura del trabajo</i>	3
2 El Modelo Dinámico	5
2.1 <i>Sistema de referencia</i>	5
2.2 <i>Equilibrio dinámico longitudinal</i>	6
2.3 <i>Pendiente máxima superable</i>	8
2.3.1 <i>Tracción trasera</i>	9
2.3.2 <i>Tracción delantera</i>	11
2.3.3 <i>Tracción en los dos ejes</i>	12
3 El sistema de transmisión en el automóvil	15
3.1 <i>Curvas de potencia y par</i>	16
3.2 <i>Relaciones de transmisión</i>	17
3.3 <i>Relación entre el par motor y la fuerza de tracción</i>	17
3.4 <i>Relación entre la velocidad del motor y la del automóvil</i>	18
3.5 <i>El freno motor</i>	18
3.6 <i>Hipótesis previas al cálculo de la velocidad máxima admisible en pendientes</i>	19
3.6.1 <i>Pendiente ascendente</i>	19
3.6.2 <i>Pendiente descendente</i>	20
3.7 <i>Velocidad máxima admisible en una pendiente</i>	20
4 El problema del consumo de combustible	23
4.1 <i>Consumo específico de combustible</i>	23
4.2 <i>Mapa de consumo</i>	24
4.3 <i>Consumo óptimo de combustible</i>	24
5 Interfaz gráfica en MATLAB	27

5.1	<i>Introducción a las interfaces de usuario en MATLAB</i>	27
5.2	<i>El sistema Handle Graphics de MATLAB</i>	27
5.3	<i>UIControls y sus propiedades</i>	28
5.4	<i>Manipulación de las propiedades de los objetos gráficos: get y set</i>	30
5.5	<i>Callback Functions: Qué son y para qué sirven</i>	31
5.6	<i>MATLAB GUIDE: Ventajas y desventajas</i>	32
5.7	<i>Conceptos previos de programación</i>	34
5.7.1	Bucles y bifurcaciones	34
5.7.2	Mensajes al usuario mediante órdenes	35
6	Implementación de la interfaz gráfica de MATLAB	38
6.1	<i>Pantalla de inicio</i>	38
6.2	<i>Menú</i>	40
6.3	<i>Opción pendiente</i>	40
6.4	<i>Opción velocidad</i>	41
6.5	<i>Opción consumo de combustible</i>	43
6.6	<i>Funciones auxiliares</i>	45
6.6.1	Ventana gráfica de pendiente	45
6.6.2	Ventana gráfica de velocidad	46
6.6.3	Ventana consumo de combustible	51
6.6.4	Funciones auxiliares comunes	54
7	Ejemplo de la interfaz gráfica	56
7.1	<i>Cálculo de la pendiente máxima superable</i>	57
7.2	<i>Cálculo de la velocidad máxima admisible</i>	59
7.3	<i>Cálculo del consumo óptimo de combustible</i>	62
7.4	<i>Obtención de resultados para otros vehículos</i>	65
7.4.1	Pendiente máxima superable	67
7.4.2	Velocidad máxima admisible	68
7.4.3	Consumo óptimo de combustible	69
8	Conclusiones y líneas de mejora	72
8.1	<i>Análisis de resultados</i>	72
8.1.1	Pendiente máxima	72
8.1.2	Velocidad máxima	73
8.1.3	Consumo óptimo de combustible	73
8.2	<i>Líneas futuras de mejora</i>	73
	Referencias	76
	ANEXOS	78
	<i>Anexo A. Código de MATLAB para el diseño de la interfaz para optimizar la subida y bajada en cuesta</i>	78
	<i>Anexo B. Código de MATLAB de las funciones auxiliares para el funcionamiento de la interfaz</i>	90

Índice de Tablas

Tabla 2-1. Valores típicos del coeficiente de resistencia por rodadura para distintos tipos de neumáticos	7
Tabla 2-2. Algunos valores del coeficiente de arrastre en algunos vehículos	8
Tabla 2-3. Valores característicos de la adherencia para algunas superficies	9
Tabla 6-1. Código ASCII de cada letra que representa un color en la orden plot	48
Tabla 7-1. Especificaciones técnicas Mini Cooper 5 puertas 136 CV 2020	57
Tabla 7-2. Tabla de resultados del cálculo de la pendiente máxima superable para el caso del Mini Cooper	59
Tabla 7-3. Tabla de resultados del cálculo de la velocidad máxima admisible para el caso del Mini Cooper	62
Tabla 7-4. Tabla de resultados del cálculo del consumo óptimo de combustible para el caso del Mini Cooper	65
Tabla 7-5. Especificaciones técnicas SSANGYONG REXTON D22DTR 4x2 Premium 181CV de 201866	
Tabla 7-6. Especificaciones técnicas AUDI A7 Sportback 50 TDI 210kW quattro triptron. 286CV de 2020	67
Tabla 7-7. Tabla de resultados del cálculo de la pendiente máxima superable para el caso del Ssanyong Rexton	68
Tabla 7-8. Tabla de resultados del cálculo de la pendiente máxima superable para el caso del Audi A7	68
Tabla 7-9. Tabla de resultados del cálculo de la velocidad máxima admisible para el caso del Ssanyong Rexton	69
Tabla 7-10. Tabla de resultados del cálculo de la velocidad máxima admisible para el caso del Audi A7	69
Tabla 7-11. Tabla de resultados del cálculo del consumo óptimo de combustible para el caso del Ssanyong Rexton	69
Tabla 7-12. Tabla de resultados del cálculo del consumo óptimo de combustible para el caso del Audi A7	70

Índice de Figuras

Figura 1-1. Primer automóvil de la historia, Benz Patent Motorwagen	2
Figura 1-2. Imagen de cómo es un automóvil hoy en día	2
Figura 2-1. Sistema de referencia ligado al vehículo según ISO 8855 y SAE670e	6
Figura 2-2. Fuerzas que aparecen en el movimiento de un automóvil en una pendiente ascendente	6
Figura 2-3. Fuerzas que aparecen en un neumático	9
Figura 2-4. Diagrama de fuerzas de un automóvil con tracción trasera	10
Figura 2-5. Diagrama de fuerzas de un automóvil con tracción delantera	11
Figura 2-6. Diagrama de un automóvil con tracción en los dos ejes	12
Figura 3-1. Esquema del sistema de transmisión en un automóvil	15
Figura 3-2. Curvas de potencia y par deseadas	17
Figura 3-3. Gráfica que relaciona el par motor con la velocidad de giro del motor	19
Figura 4-1. Tabla con los valores del flujo másico de combustible	24
Figura 5-1. Organización sistema Handle Graphics	28
Figura 5-2. Ventana de inicio GUIDE	32
Figura 5-3. Menú de GUIDE	33
Figura 6-1. Pantalla de inicio de la interfaz de usuario	39
Figura 6-2. Menú de la interfaz de usuario	40
Figura 6-3. Ventana pendiente	41
Figura 6-4. Ventana velocidad	43
Figura 6-5. Ventana consumo de combustible	44
Figura 6-6. Ventana que pide al usuario los datos del motor y de la caja de cambios	47
Figura 6-7. Ventana que pide al usuario los datos de la caja de cambios y a qué velocidad circula el vehículo	52
Figura 7-1. Mini Cooper 5 puertas 136 CV 2020	56
Figura 7-2. Ejemplo de introducción de datos en la ventana pendiente	58
Figura 7-3. Solución de la pendiente máxima superable para el caso de pendiente ascendente	58
Figura 7-4. Solución de la pendiente máxima superable en el caso de pendiente descendente	59
Figura 7-5. Ejemplo de introducción de datos en la ventana velocidad	60
Figura 7-6. Ventana que se abre al pulsar sobre el botón destinado a introducir los datos del sistema de transmisión y el motor	60
Figura 7-7. Gráfica que relaciona la fuerza de tracción con la velocidad del vehículo en función de las marchas	61
Figura 7-8. Cuadro de diálogo explicativo ventana velocidad	61
Figura 7-9. Solución de la velocidad máxima admisible para el caso de pendiente ascendente	61
Figura 7-10. Solución de la velocidad máxima admisible en el caso de pendiente descendente	62
Figura 7-11. Ventana que se abre al pulsar sobre el botón destinado a calcular la tabla de par motor y velocidad	

de giro	63
Figura 7-12 Ventana que contiene la tabla de par motor y velocidad de giro	63
Figura 7-13 Cuadro de diálogo explicativo ventana consumo de combustible	64
Figura 7-14 Mapa de consumo de combustible	64
Figura 7-15. Solución del consumo óptimo de combustible para el caso de pendiente ascendente	65
Figura 7-16. Solución del consumo óptimo de combustible para el caso de pendiente descendente	65
Figura 7-17. SSANGYONG REXTON D22DTR 4x2 Premium 181CV de 2018	66
Figura 7-18. AUDI A7 Sportback 50 TDI 210kW quattro triptron. 286CV de 2020	67

Notación

f_r	Coefficiente de resistencia de rodadura
N_1	Fuerza perpendicular a la superficie en el eje delantero
N_2	Fuerza perpendicular a la superficie en el eje trasero
m	Masa del automóvil
g	Aceleración de la gravedad
$\text{sen}(\alpha)$	Seno del ángulo de la pendiente
$\text{cos}(\alpha)$	Coseno del ángulo de la pendiente
$\text{tan}(\alpha)$	Tangente del ángulo de la pendiente
F_{aero}	Fuerza aerodinámica
F_{resist}	Fuerza resistente
ρ	Densidad del aire
C_D	Coefficiente de resistencia aerodinámica
v	Velocidad del vehículo
a_x	Aceleración del vehículo
μ	Coefficiente de adherencia
h	Altura del centro de masas del vehículo
l	Distancia entre los ejes delantero y trasero de un vehículo
a	Distancia entre el centro de masas y el eje delantero del vehículo
b	Distancia entre el centro de masas y el eje trasero del vehículo
P_{mot}	Potencia del motor de combustión
T_{mot}	Par motor
T_{rueda}	Par producido en las ruedas del vehículo
ω_m	Velocidad de giro del motor
ω_s	Velocidad de giro a la salida de la caja de cambios
ω_{rueda}	Velocidad de giro de las ruedas del vehículo
r	Relación de transmisión
r_d	Relación de transmisión del diferencial
η	Eficiencia de la transmisión
\dot{m}	Flujo másico de combustible

1 INTRODUCCIÓN

1.1 Motivación y objetivos

El objetivo principal de este trabajo es el de crear una interfaz gráfica de usuario para optimizar la subida y bajada en cuesta de cualquier tipo de automóvil de motor de combustión interna una vez se conocen las principales características de este. La interfaz gráfica diseñada permitirá al usuario tanto conocer la pendiente máxima que puede superar su automóvil como la velocidad máxima que este puede alcanzar en ella. Además de esto, se puede conocer el consumo de combustible que tendrá el vehículo en una pendiente determinada a una cierta velocidad, por lo que esta interfaz gráfica será de gran ayuda para que el usuario pueda ver cómo varían el consumo de combustible, la velocidad máxima admisible y la pendiente máxima superable dependiendo del tipo de automóvil que se estudie.

Para la simulación dinámica del automóvil existen ya algunos softwares como es el caso de CARSIM. Sin embargo, de entrada, este programa puede llegar a ser difícil de usar para un usuario inexperto debido a la multitud de datos que hay que introducir y a la cantidad de parámetros diferentes que puede llegar a calcular. Es por ello que con la interfaz gráfica que se va a diseñar en este trabajo se quiere que el usuario tenga una primera toma de contacto en el cálculo de algunos de los parámetros que varían en un automóvil en el ascenso o descenso de estos por distintas pendientes. Todo ello de una forma muy simplificada y sencilla, para no dar lugar a confusión.

Dicha interfaz se ha realizado a través del software MATLAB®, el cual se estudia en varias asignaturas de este grado que hacen ver la gran utilidad de este software para su uso en las diferentes áreas de la ingeniería y que ha servido para llevar a cabo la creación de la interfaz gráfica, objeto de estudio en este trabajo. Para ello, se necesita un gran conocimiento de la programación en MATLAB® orientada a objetos, algo en lo que ha ayudado en gran parte el hecho de cursar la asignatura Matemática Computacional por parte del autor de este documento.

Para llevar a cabo la interfaz, en primer lugar, hay que estudiar la dinámica del automóvil, más concretamente la dinámica longitudinal y las cuestiones que derivan de esta, siendo este el punto de partida del trabajo. Este estudio se ha visto favorecido gracias a que en la asignatura de Automóviles ya se ven algunas nociones acerca de la dinámica longitudinal, una disciplina realmente reciente tal y como se mostrará en la reseña histórica.

Así mismo, también es necesario conocer el consumo de combustible del automóvil. Esta cuestión es una de las más importantes para intentar optimizar la marcha de un automóvil tanto en cuesta ascendente como descendente. La meta de dicho tema de estudio es intentar conseguir un mapa de consumo que intente representar lo más fielmente posible la realidad, y a través de él y con ciertos parámetros obtener la marcha óptima del vehículo, tal y como se verá en la sección que tratará este tema más en profundidad. Para la construcción del citado mapa, haría falta explicar en profundidad qué sucede dentro de un motor de combustión interna, cuestión que no se tratará en profundidad, ya que para ello habría que hacer un estudio detallado de este tipo de motores y esto excede el alcance de este trabajo, cuyo objetivo principal como ya se ha remarcado es el de crear una interfaz gráfica.

1.2 Reseña histórica

Aunque la aparición del automóvil como tal no ocurriese hasta finales del siglo XIX, el pensamiento de que un vehículo se autopropulsara ya llevaba varios años en la mente de diversas personas. Un claro ejemplo de ello fue la idea que tuvo el suizo J.H. Genevois en el año 1760, quién propuso el montaje de un sistema compuesto por un muelle y un molino de viento, (el cual haría que el muelle se enrollara, provocando el movimiento del

vehículo), con la idea de propulsar un carruaje. Nueve años más tarde, en 1769, se le atribuye a Cugnot de Lorraine la construcción del primer automóvil de vapor, aunque el gran apogeo de este tipo de autopropulsión no llegó hasta principios del siglo XIX, más concretamente entre 1831 y 1838.

La gran revolución en la autopropulsión de vehículos llegaría a finales del siglo XIX, tal y como se ha mencionado antes. Esto fue debido a la invención del motor de combustión interna, un éxito que se le atribuye a Carl Benz y Gottlieb Daimler, quienes lograron hacer funcionar sus motores en 1885 y 1886 respectivamente. Además el primero de ellos patentó el que es conocido como el primer automóvil propulsado por un motor de combustión interna de la historia. Este vehículo estaba formado por tan solo tres ruedas, por lo que tenía aspecto de triciclo como se muestra en la siguiente figura.

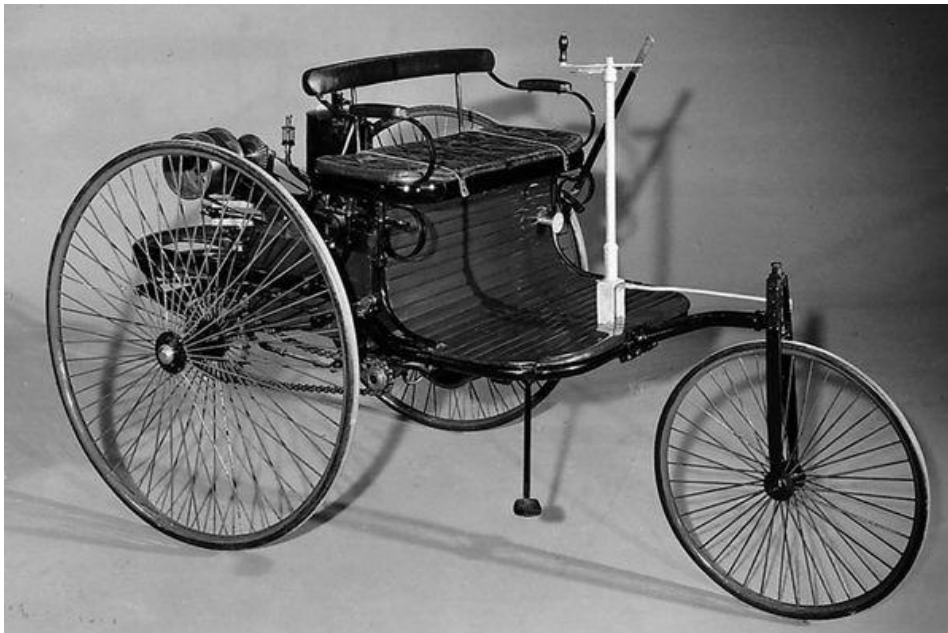


Figura 1-1. Primer automóvil de la historia, Benz Patent Motorwagen

Fuente: <https://www.pinterest.es/pin/354658539378085935/>

Aunque este automóvil patentado por Benz no tuvo un buen número de ventas fue al comienzo del siglo XX, entre los años 1904 y 1908, cuando se vivió un gran auge en la construcción de automóviles y es que una gran multitud de marcas se dedicaban a ello. Entre estas marcas se encontraban ya industrias de gran renombre en la actualidad como Peugeot y Renault en Francia o Alfa Romeo, Ferrari y Fiat en Italia. Desde entonces el automóvil ha ido cambiando a lo largo de los años, tanto en diseño como en prestaciones.



Figura 1-2. Imagen de cómo es un automóvil hoy en día

Fuente: <https://motor.elpais.com/actualidad/mejor-coche-del-ano-en-el-mundo-2019/>

Paralelamente a ello, como no podía ser de otra manera, se ha ido desarrollando el concepto de la dinámica del

automóvil, una de las ciencias sobre la que profundizaremos en este trabajo. Como ya se dijo, esta es una ciencia relativamente nueva, ya que tuvo su mayor desarrollo entre mediados y finales del siglo pasado. El problema de la dinámica vehicular surgió cuando una vez el automóvil estaba implantado, se quería conseguir una mayor velocidad en éste. Por tanto, al ser los automóviles más veloces aparecían grandes problemas a la hora de girar y frenar a esa velocidad, es ahí donde entró en juego la dinámica del automóvil.

Desde que se introdujo la idea de la dinámica del movimiento en el giro de curvas por Lanchester en 1907, la dinámica vehicular empezó a evolucionar de forma imparable introduciendo por primera vez conceptos como el shimmy, el ángulo de deriva o términos como sobrevirador o subvirador, desconocidos hasta la época. Así mismo, se realizaron numerosos estudios acerca de las fuerzas que aparecían en el neumático o las fuerzas aerodinámicas que tenían lugar en el automóvil. Todo ello hasta que en el año 1956 se crea el primer departamento de dinámica de vehículos en EEUU. A partir de ahí la dinámica vehicular siguió evolucionando hasta llegar a ser tal y como la conocemos hoy en día.

En esta evolución fueron de gran ayuda la aparición de los primeros ordenadores analógicos, los cuales permitieron por primera vez simular la respuesta dinámica de un automóvil. Además, el avance en los modelos analíticos dinámicos fue importante para representar más fielmente la dinámica del vehículo.

1.3 Estructura del trabajo

En el segundo capítulo se tratará en detalle la dinámica longitudinal del automóvil, esta es la cuestión esencial sobre la que derivan casi todas las cuestiones que habrá que resolver en la interfaz, por lo que hay que conocer bastante bien esta área de conocimiento y establecer un modelo dinámico antes de ponerse a programar en el software. Además dicho capítulo también contendrá algunas cuestiones relativas a la dinámica longitudinal como son la pendiente máxima superable de un vehículo en subida.

Justo después de ver el modelo dinámico del automóvil en una pendiente, se pasará a explicar el sistema de transmisión presente en un automóvil, así como otros conceptos relacionados con el como la relación de transmisión o las curvas de potencia y par. Esto servirá además para establecer unas relaciones entre el par motor y la fuerza de tracción y entre la velocidad del motor y la del automóvil, necesarias para realizar la gráfica que relacione la fuerza de tracción con la velocidad del automóvil. En ese tercer capítulo también se explicará otro concepto como el freno motor que será esencial cuando el automóvil se encuentre en pendiente descendente. Finalmente se explicará cómo se calcula la velocidad máxima que puede experimentar un automóvil en una pendiente determinada.

El cuarto capítulo estará dedicado íntegramente al consumo de combustible y a todos los conceptos que proceden de él. Algunos de los temas que se tratarán serán el consumo específico de combustible o los mapas de consumo específico de combustible, uno de los temas más importantes de este capítulo, ya que posteriormente su implementación en MATLAB es un tanto compleja.

Una vez vistos los dos conceptos teóricos necesarios para empezar a programar la interfaz gráfica de usuario, se tratará el entorno gráfico en MATLAB en el quinto capítulo. En éste se verán los objetos gráficos más importantes con los que cuenta el software a la hora de crear una interfaz, a la vez que se tratan otros conceptos de gran importancia como son las Callbacks Functions o funciones de llamada, y las sentencias get y set. Además se verán las pautas que se han seguido a la hora de empezar a programar y el por qué de esas elecciones, llegando a introducir algunos conceptos de programación.

Después de esto, está todo listo para empezar el diseño de la interfaz gráfica de usuario para la optimización de la subida y bajada en cuesta. Por tanto en el sexto capítulo, paso a paso, se explicará cómo se ha ido creando la interfaz, sus distintas ventanas gráficas y las distintas funciones auxiliares para llevar a cabo cada uno de los cálculos necesarios para dicha optimización. Cuando ya esté bien explicada la construcción de la interfaz gráfica que se ha creado para este trabajo, se procederá a mostrar un ejemplo de su funcionamiento en el séptimo capítulo, el cual posiblemente sea el más gráfico de todos.

Por último en el octavo capítulo se expondrán las conclusiones sacadas de este trabajo de fin de grado, analizando los resultados que se obtendrán utilizando la interfaz diseñada en diferentes modelos de automóviles, así como las líneas de mejora futuras a seguir en un posible futuro diseño de otra interfaz a partir del diseño de la que se va a diseñar en este trabajo de fin de grado.

2 EL MODELO DINÁMICO

La dinámica del automóvil, como ya se ha remarcado en la introducción del presente documento es uno de los conceptos claves para llevar a cabo este trabajo de fin de grado. Es por ello, que en este capítulo se va a explicar en profundidad los conceptos dinámicos necesarios para llevar a cabo la interfaz gráfica de usuario. Unos conceptos en los que intervienen las leyes de la Mecánica Clásica, tal y como se verá más adelante.

Las fuerzas a las que tiene que hacer frente un vehículo son tres: las procedentes de la interacción entre el neumático y la carretera, las debidas a la resistencia que ejerce el entorno en el que se encuentra (fuerzas aerodinámicas) y por último las que aparecen debido al efecto de la gravedad. Las más relevantes en la dinámica vehicular son las primeras, aquellas que se producen en el neumático, constituyendo estos una parte fundamental en el estudio de la dinámica del automóvil. Es por ello por lo que es coherente pensar que el modelo dinámico se tiene que hacer respecto a las fuerzas que aparecen en las ruedas del automóvil, por lo que en un principio habrá que establecer un sistema de referencia para visualizar estas fuerzas.

En primer lugar, el lector debe saber que hay tres campos de la dinámica presentes en el movimiento del automóvil. Por una parte está la dinámica lateral, imprescindible para entender problemas tan importantes como el vuelco de los vehículos y el balanceo. Por otra parte, está la dinámica vertical del vehículo, necesaria para diseñar correctamente el sistema de suspensión del vehículo. Finalmente, se encuentra la dinámica longitudinal, la cual va a ser objeto de estudio en este documento, ya que nos permite conocer las fuerzas de tracción presentes en el vehículo así como las fuerzas resistivas que este tiene que vencer.

2.1 Sistema de referencia

El lector debe saber que hay tres campos de la dinámica presentes en el movimiento del automóvil. Por una parte está la dinámica lateral, imprescindible para entender problemas tan importantes como el vuelco de los vehículos y el balanceo. Por otra parte, está la dinámica vertical del vehículo, necesaria para diseñar correctamente el sistema de suspensión del vehículo. Finalmente, se encuentra la dinámica longitudinal, la cual va a ser objeto de estudio en este documento, ya que nos permite conocer las fuerzas de tracción presentes en el vehículo así como las fuerzas resistivas que este tiene que vencer.

En este subcapítulo se tratará de explicar como se asocian cada uno de los tres campos de la dinámica vehicular (dinámica lateral, dinámica lateral y dinámica longitudinal) a unos ejes tridimensionales que se definirán. Estos ejes, que podrían pertenecer a un sistema de referencia fijo a tierra, irán referenciados a un sistema el cual se encuentra ligado al automóvil. Esta elección se debe a que como se verá más adelante, en el caso de la dinámica longitudinal, el hecho de tener un sistema de referencia ligado al vehículo simplificará las ecuaciones en el equilibrio de fuerzas.

El sentido de los ejes del sistema de referencia ligado al automóvil está definido según dos normas: la ISO 8855 y la SAE 670e. En este documento se sigue el criterio que se usa en la norma ISO 8855, la cual define los ejes xyz tal y como aparecen en la siguiente figura:

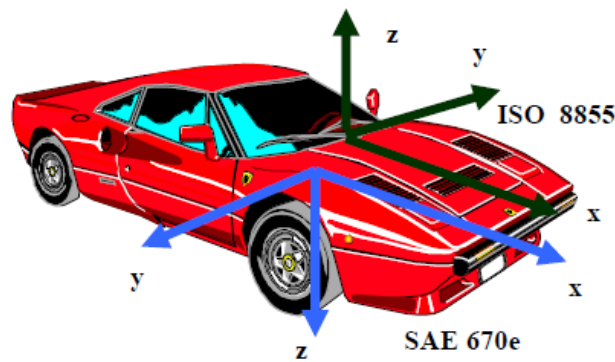


Figura 2-1. Sistema de referencia ligado al vehículo según ISO 8855 y SAE670e [1]

Como se muestra en la figura 2-1, en el sistema de referencia según la norma ISO 8855, el eje x es aquel que tiene el mismo sentido que la dirección de movimiento que sigue el vehículo, el eje y es aquel que se encuentra perpendicular al eje x en plano horizontal del automóvil y el eje z es el eje vertical, perpendicular al eje x e y ya que los tres ejes del sistema de referencia son ortogonales entre sí. Así mismo, dichos ejes son positivos en el sentido (referente a la norma ISO 8855) mostrado en la figura.

Ahora solo queda definir a qué eje está referida cada una de las fuerzas que tratan cada uno de los campos de la dinámica vehicular. La fuerza longitudinal se dará en el eje x y las fuerzas lateral y vertical tendrán lugar en los ejes y y z, respectivamente. Por tanto en este documento solo nos interesará lo que suceda en el plano x-z ya que la dinámica que vamos a estudiar es la de las fuerzas que se producen en el sentido de avance del vehículo, es decir la dinámica longitudinal.

2.2 Equilibrio dinámico longitudinal

En este subcapítulo se va a realizar el equilibrio de las fuerzas que aparecen a lo largo del movimiento del automóvil en carretera. Las fuerzas que se van a tener en cuenta van a ser solamente las longitudinales, que son las únicas que interesan para llevar a cabo el diseño de la interfaz gráfica de usuario que se va a desarrollar en el presente trabajo de fin de grado.

Por tanto quedan descartadas tanto las fuerzas verticales como las fuerzas laterales que pueden llegar a aparecer en un automóvil, por lo que las fuerzas existentes durante el movimiento del mismo serán las que aparecen en la figura de abajo, que son aquellas que se ven si se mirase al automóvil desde su perfil. Estas fuerzas se verán con más detalle a continuación.

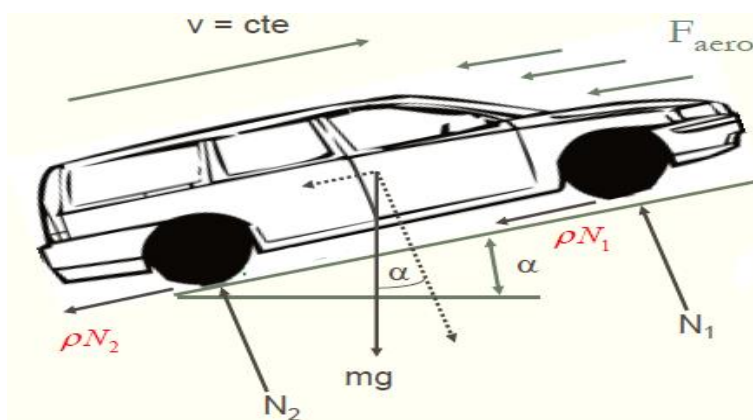


Figura 2-2. Fuerzas que aparecen en el movimiento de un automóvil en una pendiente ascendente [5]

Antes de hacer el equilibrio denotar que ρ será nombrado a partir de ahora como f_r y representa el parámetro denominado coeficiente de resistencia por rodadura. Ahora realizando los correspondientes equilibrios en los ejes x y z definidos según el sistema de referencia anteriormente explicado, quedan las siguientes expresiones:

$$\sum F_X = 0 \rightarrow f_r N_1 + f_r N_2 + mg \sin(\alpha) + F_{aero} = F_{resist} \quad (2-1)$$

$$\sum F_Z = 0 \rightarrow N_1 + N_2 = mg \cos(\alpha) \quad (2-2)$$

A continuación, llevando la expresión número 2-2 a la 2-1, se queda una sola ecuación que es la siguiente:

$$mg(\sin(\alpha) + f_r \cos(\alpha)) + F_{aero} = F_{resist} \quad (2-3)$$

La expresión anterior constituye la expresión de las fuerzas que se oponen al movimiento de un automóvil, por lo que a partir de ahora será llamada fuerza resistiva, tal y como ya indicaba su subíndice. Si se mira la anterior ecuación se observa que las fuerzas opuestas al movimiento se dividen en tres, aunque también se explicará otra que aparece en la realidad, por tanto estas son:

- Resistencia por pendiente: Esta fuerza es debida a la componente longitudinal del peso, ya que hay que recordar la existencia de la fuerza de la gravedad, y que cuando el vehículo se encuentra subiendo una pendiente hay que realizar un esfuerzo mayor debido a esta. Sin embargo, cuando el automóvil está descendiendo una pendiente, esta fuerza deja de oponerse al movimiento y ayuda a que este se desarrolle. Por lo tanto la fuerza resistiva debida a la pendiente tendría los siguientes valores dependiendo del valor de la pendiente a la que se enfrente el automóvil en cuestión.

$$F_{pend} = \begin{cases} > 0 & \text{si } \alpha > 0 \\ 0 & \text{si } \alpha = 0 \\ < 0 & \text{si } \alpha < 0 \end{cases} \quad (2-4)$$

- Resistencia por rodadura: Esta componente de las fuerzas que se oponen al movimiento en el movimiento de un vehículo es debida al contacto que ejerce el neumático con la carretera, por lo tanto esta resistencia dependerá de algunas características intrínsecas del mismo. Más concretamente dependerá exclusivamente del valor que tenga el denominado coeficiente de resistencia por rodadura (f_r), así como del peso del vehículo y del valor de la pendiente, tal y como aparece a continuación.

$$F_{rod} = mg f_r \cos(\alpha) \quad (2-5)$$

Tipo de neumático	Coefficiente de resistencia por rodadura
Neumáticos de automóvil de baja resistencia y neumáticos de camión sobre carretera lisa	0.006-0.01
Neumáticos ordinarios de automóvil sobre losas de piedra	0.01-0.015
Neumáticos ordinarios de automóvil sobre hormigón	0.02
Neumáticos ordinarios de automóvil sobre alquitrán o asfalto	0.03-0.035
Neumáticos ordinarios de automóvil sobre tierra, barro o arena	0.055-0.065

Tabla 2-1. Valores típicos del coeficiente de resistencia por rodadura para distintos tipos de neumáticos [9]

- Resistencia aerodinámica: Esta fuerza resistiva es consecuencia de la interacción entre el aire que golpea al automóvil cuando este se mueve y el área de este golpeada por el aire. Cabe lógico pensar que cuanto mayor sea la velocidad de circulación del automóvil esta resistencia será más grande. Además de la velocidad de movimiento, también influye el tamaño de área que choca con el aire, ya que cuanto mayor sea esta mayor será la resistencia aerodinámica desarrollada. Para calcular este tipo de resistencia en este documento se ha hecho la hipótesis de que la velocidad del viento es despreciable (ya que en caso contrario esta velocidad se le debería restar a la velocidad del vehículo).

Por último este tipo de resistencia también depende del coeficiente de resistencia aerodinámica del vehículo (C_D), también llamado coeficiente de arrastre, así como de la densidad del aire.

$$F_{aero} = \frac{1}{2} \rho C_D v^2 \quad (2-6)$$

Vehículo	C_D
VW Polo	0.37
Ford Escort	0.36
Opel Vectra	0.29
BMW 520i	0.31
Mercedes 300SE	0.36

Tabla 2-2. Algunos valores del coeficiente de arrastre en algunos vehículos [5]

- Fuerza de inercia: La última componente de las fuerzas que se oponen al movimiento en un automóvil es causa de la aceleración o deceleración que este experimenta a lo largo de su recorrido, en el diseño de la interfaz esta fuerza no se va a tener en cuenta pero recalcar que en la realidad si que aparece. Por lo tanto su valor será positivo si el vehículo está acelerando, negativo si está decelerando y nulo si mantiene su velocidad constante, esto se expone a continuación.

$$F_{iner} = m g a_x \quad (2-7)$$

De esta manera quedan descritas las fuerzas resistivas que aparecen en el ascenso de una pendiente y que la fuerza de tracción ofrecida por el vehículo tendrá que vencer, tal y como se verá en el siguiente capítulo.

Solamente queda explicar como es la fuerza resistiva que aparece en el descenso de pendientes, una expresión en la que solo cambiará el signo de una de las componentes vistas anteriormente. Esta es la componente de la resistencia por pendiente, que en este caso irá a favor del movimiento, por lo que la expresión resultante queda tal y como se muestra a continuación:

$$m g (f_r \cos(\alpha) - \text{sen}(\alpha)) + \frac{1}{2} \rho C_D v^2 = F_{resist} \quad (2-8)$$

Como el valor de la componente de la resistencia por pendiente es mucho mayor que el de la resistencia por rodadura y el de la resistencia aerodinámica (hasta una cierta velocidad) esta deberá ser compensada por la fuerza del freno motor que se opone al movimiento y que también se verá en el siguiente capítulo. Queda así expresado el modelo dinámico sobre el que se basará la interfaz de usuario que se va a diseñar en este trabajo de fin de grado.

2.3 Pendiente máxima superable

En el anterior subcapítulo ya se vio la influencia que puede llegar a tener la pendiente a la hora de calcular las fuerzas que se oponen al movimiento de un vehículo, puesto que el valor de esta influya en la resistencia por pendiente y en la resistencia por rodadura. Llegado el momento, el lector podría llegar a plantearse que puede existir un valor de pendiente para el cual el vehículo no pudiera ascender la misma, y efectivamente esto ocurre. Una vez el automóvil supera su, denominada, capacidad de subida de pendientes, este patinaría y no podría seguir avanzando.

Por lo tanto, este subcapítulo se encargará de ver el método por el cual se puede llegar a obtener el valor máximo de la pendiente que puede llegar a superar el vehículo. Para ello primero se tendrá que explicar el concepto de adherencia, más concretamente el coeficiente de adherencia. También denominado en ocasiones coeficiente de rozamiento estático es uno de los conceptos físicos por excelencia a la hora de tratar dos superficies que se encuentran en contacto.

El coeficiente de adherencia en el caso de los automóviles se define como la relación existente entre la fuerza

paralela al plano, de la superficie de contacto del neumático con la carretera, y la fuerza perpendicular a dicho plano. Para que no se llegue a producir deslizamiento entre el neumático y la carretera el valor del cociente producido al dividir las fuerzas paralela y perpendicular, anteriormente mencionadas, debe ser menor que el coeficiente de adherencia dado. Dado que en un neumático se producen las siguientes fuerzas:

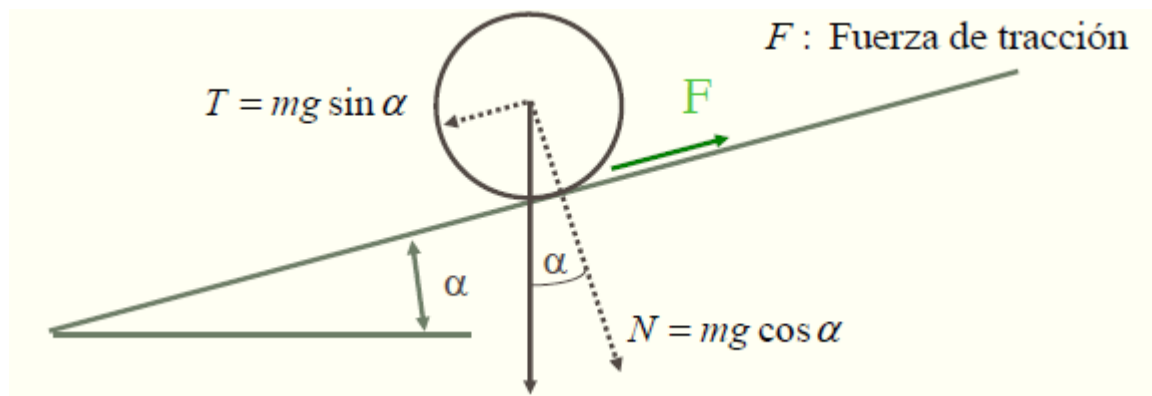


Figura 2-3. Fuerzas que aparecen en un neumático [5]

Al llevar a cabo la relación del coeficiente de adherencia con las fuerzas paralelas y perpendiculares a la superficie de contacto entre el neumático y la carretera queda la siguiente expresión:

$$\frac{mg \operatorname{sen}(\alpha)}{mg \operatorname{cos}(\alpha)} < \mu \quad (2-9)$$

Operando un poco la anterior ecuación da lugar a la expresión del denominado coeficiente de adherencia aprovechado.

$$\tan(\alpha) < \mu \quad (2-10)$$

Superficie	Valor máximo de μ
Asfalto seco	0.8-0.9
Hormigón seco	0.8-0.9
Asfalto húmedo	0.5-0.7
Hormigón húmedo	0.8
Grava	0.6
Nieve	0.2
Hielo	0.1

Tabla 2-3. Valores característicos de la adherencia para algunas superficies [10]

Una vez entendido el concepto de coeficiente de adherencia, para calcular la capacidad de subida de pendientes que tiene un vehículo se debe de tener en cuenta el tipo de tracción que tiene este. Así pues para obtener la pendiente máxima superable se distinguirá entre si el vehículo está dotado con tracción delantera, con tracción trasera o con tracción en ambos ejes. Además se hace saber que en el cálculo de la pendiente máxima superable la fuerza debido a la resistencia aerodinámica se desprecia ya que hay que tener en cuenta las fuerzas que actúan directamente sobre los neumáticos y la fuerza de resistencia aerodinámica es debida a la resistencia del aire que tiene que vencer el área del automóvil como ya es sabido.

2.3.1 Tracción trasera

Si la tracción en el automóvil es trasera, el diagrama de fuerzas queda de la siguiente forma:

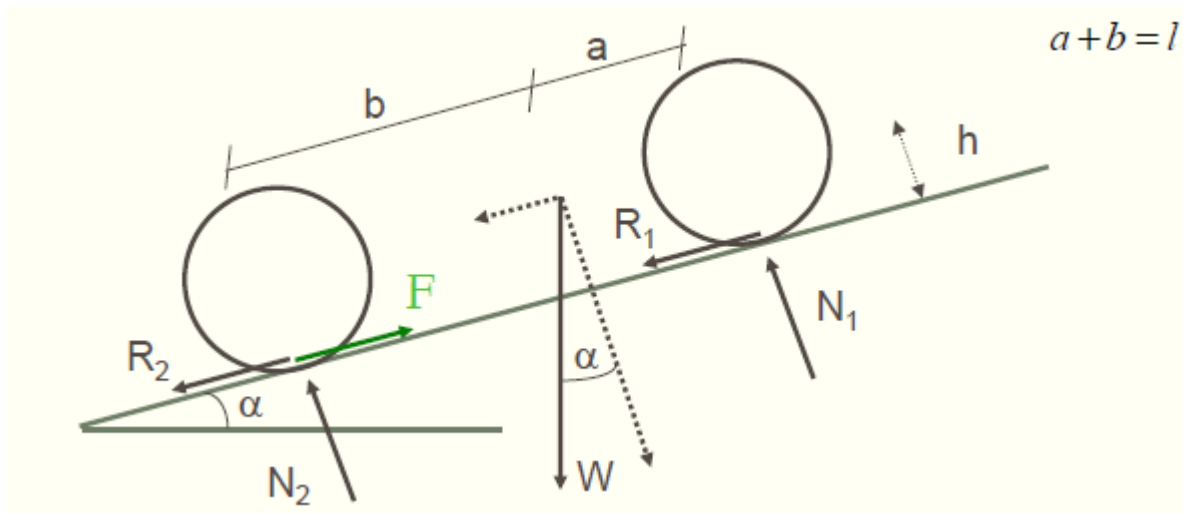


Figura 2-4. Diagrama de fuerzas de un automóvil con tracción trasera [5]

Antes de seguir con el equilibrio, denotar que W equivale a la fuerza que aparece debido al peso del coche y que tanto R_1 como R_2 equivale a la fuerza por rozamiento ($f_r N$) que surge del contacto entre el neumático y la carretera. Además se observa que la fuerza por resistencia aerodinámica no influirá en el cálculo de la pendiente máxima superable y que la fuerza F que aparece es la debida a la tracción del vehículo y que según el equilibrio de fuerzas deberá ser igual a la fuerza resistiva al movimiento del automóvil (ecuación 2-3).

A continuación se realiza el equilibrio de fuerzas en los ejes x y z ya definidos y se hace equilibrio de momentos respecto al punto situado en el eje sin tracción, en este caso el eje delantero. Al hacer esto, además de las ecuaciones 2-1 y 2-2 vistas anteriormente, resulta la siguiente ecuación referente al equilibrio de momentos:

$$\sum M_1 = 0 \rightarrow N_2 l - hmg \sin(\alpha) - amg \cos(\alpha) = 0 \quad (2-11)$$

Ahora se despejan la fuerza paralela a la superficie de contacto entre el neumático y la carretera (F) y la normal a esta (N_2), quedando la primera de las expresiones mencionadas de la misma forma que la ecuación 2-3 (quitando la fuerza debida a la resistencia aerodinámica) y la otra tal y como aparece a continuación:

$$N_2 = \frac{hmg \sin(\alpha) + amg \cos(\alpha)}{l} \quad (2-12)$$

El próximo paso es utilizar la relación existente entre el coeficiente de adherencia y las fuerzas paralelas y perpendiculares a la superficie de contacto del neumático con la carretera, relación que ya se ha explicado en este subcapítulo. Se recuerda que esta relación establecía que el cociente entre las fuerzas paralelas y perpendiculares que aparecían en el neumático debía ser menor que el coeficiente de adherencia (en este caso se utilizará la igualdad, ya que más adelante se quiere obtener el valor concreto de la pendiente máxima superable). Por tanto, aplicando la relación a las ecuaciones 2-3 (caracterizada para el caso en el que la fuerza de resistencia aerodinámica es nula) y 2-12 que corresponden respectivamente a las fuerzas paralelas y perpendiculares que tienen lugar en el neumático perteneciente al eje de tracción queda la siguiente expresión:

$$\frac{F}{N_2} = \mu = l \frac{\sin(\alpha) + f_r \cos(\alpha)}{h \sin(\alpha) + a \cos(\alpha)} \quad (2-13)$$

Operando un poco la ecuación anterior se queda:

$$\tan(\alpha) = \frac{f_r l - a \mu}{h \mu - l} \quad (2-14)$$

Despejando el valor del ángulo α correspondiente a la pendiente que está ascendiendo el vehículo, se obtendría el valor máximo de la misma que el vehículo podría ascender sin patinar, si este tuviese tracción trasera.

Caracterizando ahora la fuerza paralela que aparecería en el neumático si este estuviese descendiendo una pendiente la relación anterior quedaría con el siguiente aspecto:

$$\tan(\alpha) = \frac{f_r l + a\mu}{h\mu + l} \quad (2-15)$$

De manera análoga al caso de la pendiente ascendente, despejando el valor de α se obtiene el valor máximo de la pendiente descendente que podría descender el vehículo sin que sus neumáticos patinaran.

2.3.2 Tracción delantera

Si la tracción en el automóvil es delantera, el diagrama de fuerzas queda de la siguiente forma:

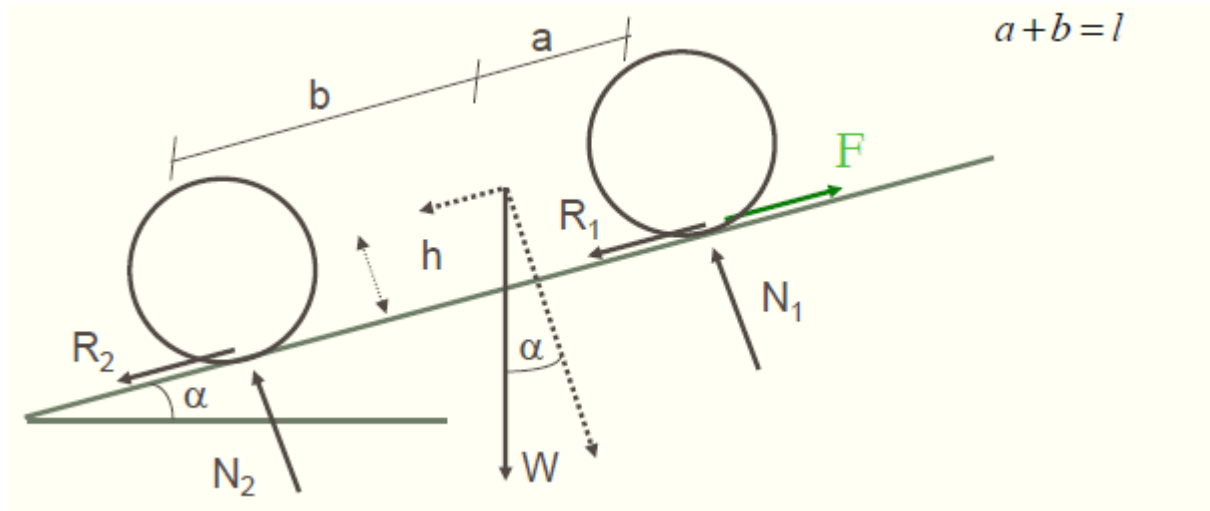


Figura 2-5. Diagrama de fuerzas de un automóvil con tracción delantera [5]

A continuación se realiza el equilibrio de fuerzas en los ejes x y z ya definidos y se hace equilibrio de momentos respecto al punto situado en el eje sin tracción, en este caso el eje trasero. Al hacer esto, resultan además de las ecuaciones 2-1 y 2-2 como ya pasaba en el caso de la tracción trasera, la siguiente ecuación relativa al equilibrio de momentos desde el eje sin tracción, en este caso el trasero.

$$\sum M_2 = 0 \rightarrow N_1 l + hmg \sin(\alpha) - bmg \cos(\alpha) = 0 \quad (2-16)$$

Ahora se despejan la fuerza paralela a la superficie de contacto entre el neumático y la carretera (F), que tendrá la misma forma que la ecuación 2-3 caracterizada para una fuerza de resistencia aerodinámica despreciable, y la perpendicular a esta (N_1), que queda con el siguiente aspecto:

$$N_1 = \frac{bmg \cos(\alpha) - hmg \sin(\alpha)}{l} \quad (2-17)$$

El próximo paso, al igual que ya se hizo en el caso de que el automóvil tuviese tracción trasera, es utilizar la relación existente entre el coeficiente de adherencia y las fuerzas paralelas y perpendiculares a la superficie de contacto del neumático con la carretera. Del mismo modo que en el caso de la tracción trasera, se sustituyen las ecuaciones de las fuerzas paralelas y normales a la superficie, resultando la siguiente expresión:

$$\frac{F}{N_1} = \mu = l \frac{\sin(\alpha) + f_r \cos(\alpha)}{b \cos(\alpha) - h \sin(\alpha)} \quad (2-18)$$

Operando un poco la ecuación anterior se queda:

$$\tan(\alpha) = \frac{b\mu - f_r l}{h\mu + l} \quad (2-19)$$

Despejando el valor del ángulo alpha correspondiente a la pendiente que está ascendiendo el vehículo, se obtendría el valor máximo de la misma que el vehículo podría ascender sin patinar, si este tuviese tracción delantera.

Caracterizando ahora la fuerza paralela que aparecería en el neumático si este estuviese descendiendo una pendiente la relación anterior quedaría con el siguiente aspecto:

$$\tan(\alpha) = \frac{f_r l + b\mu}{l - h\mu} \quad (2-20)$$

De manera análoga al caso de la pendiente ascendente, despejando el valor de α se obtiene el valor máximo de la pendiente descendente que podría descender el vehículo sin que sus neumáticos patinaran.

2.3.3 Tracción en los dos ejes

Si la tracción en el automóvil se produce en los dos ejes, el diagrama de fuerzas queda de la siguiente forma:

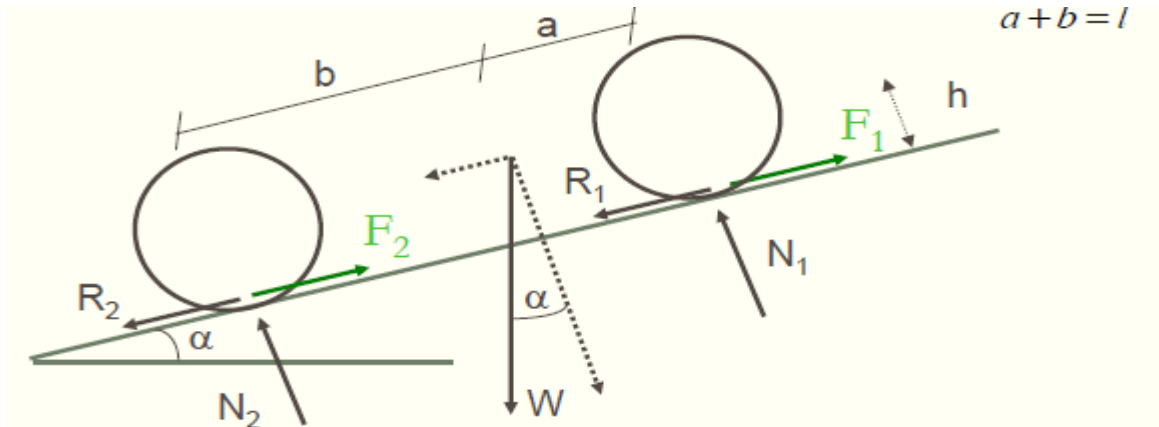


Figura 2-6. Diagrama de un automóvil con tracción en los dos ejes [5]

A continuación se realiza el equilibrio de fuerzas en los ejes x y z ya definidos (resultando la ecuación 2-3 caracterizada para el caso en el que se desprecia la resistencia por fuerza aerodinámica) y se hacen los equilibrios de momentos respecto al eje delantero en primer lugar, y en segundo lugar respecto al eje trasero. Al hacer esto, resultan las ecuaciones 2-16 y 2-11 respectivamente.

Ahora se despejan las fuerzas paralelas a la superficie de contacto entre el neumático y la carretera ($F_1 + F_2$, que serán iguales a la ecuación 2-3 cuando se desprecia la fuerza aerodinámica) y las normales a esta (por una parte N_1 y por otra N_2) que tienen el aspecto de las ecuaciones 2-17 y 2-12 respectivamente.

El próximo paso, al igual que ya se hizo en los casos en los que el automóvil tenía tracción trasera o delantera, es utilizar la relación existente entre el coeficiente de adherencia y las fuerzas paralelas y perpendiculares a la superficie de contacto del neumático con la carretera. Sin embargo, en esta ocasión hay una pequeña diferencia en la forma de proceder, y es que la capacidad de subida de pendientes ahora dependerá además de si las fuerzas de tracción (F_1 y F_2) son iguales o no. Por lo tanto:

- Si $F_1 = F_2$:

Se sustituyen las ecuaciones de las fuerzas paralelas y normales a la superficie, resultando las siguientes expresiones para la fuerza F_1 y F_2 respectivamente:

$$\frac{F_1}{N_1} = \mu = l \frac{\text{sen}(\alpha) + f_r \cos(\alpha)}{2(b \cos(\alpha) - h \text{sen}(\alpha))} \quad (2-21)$$

$$\frac{F_2}{N_2} = \mu = l \frac{\text{sen}(\alpha) + f_r \cos(\alpha)}{2(h \text{sen}(\alpha) + a \cos(\alpha))} \quad (2-22)$$

Operando un poco, las ecuaciones anteriores se quedan:

$$\tan(\alpha_d) = \frac{2b\mu - f_r l}{2h\mu + l} \quad (2-23)$$

$$\tan(\alpha_t) = \frac{f_r l - 2a\mu}{2h\mu - l} \quad (2-24)$$

Denotar que los subíndice d y t que aparecen junto al ángulo α significan delantero y trasero respectivamente, indicando así el ángulo que se alcanza dependiendo de cuál sea la fuerza de tracción a la que se le atribuye.

Despejando el valor de los ángulos α_d y α_t correspondientes a la pendiente que está ascendiendo el vehículo,

se obtendría un valor cuando la fuerza de tracción estuviese en el eje delantero (α_d) y otro para cuando la fuerza de tracción estuviese en el eje trasero (α_t). Como en este caso la fuerza de tracción está presente en los dos ejes simultáneamente, el valor de la pendiente máxima superable por el vehículo sería el valor mínimo entre los valores de α_d y α_t .

Caracterizando ahora las fuerzas F_1 y F_2 que aparecerían en el neumático si este estuviese descendiendo una pendiente las expresiones 2-23 y 2-24 quedarían respectivamente de la siguiente forma:

$$\tan(\alpha_d) = \frac{f_r l + 2b\mu}{l - 2h\mu} \quad (2-25)$$

$$\tan(\alpha_t) = \frac{f_r l + 2a\mu}{2h\mu + l} \quad (2-26)$$

Análogamente al caso en el que la pendiente era ascendente se despejan los valores de α_d y α_t y el valor mínimo de estos es el correspondiente al valor de la pendiente máxima superable para el caso de pendientes descendentes.

- Si F_1 y F_2 son distintos:

Los valores tanto de F_1 como de F_2 serían iguales al valor del coeficiente de adherencia, por lo que el coeficiente de adherencia aprovechado tendría la siguiente expresión:

$$\tan(\alpha) = \mu - f_r \quad (2-27)$$

Despejando el valor del ángulo α correspondiente a la pendiente que está ascendiendo el vehículo, se obtendría el valor máximo de la misma que el vehículo podría ascender sin patinar, si este tuviese tracción en los dos ejes y la fuerza de tracción en ambos ejes fuese distinta. En esta situación, en el caso de que el automóvil estuviese descendiendo una pendiente el ángulo α de descenso se calcularía de la misma forma.

Por último se hace saber que para el diseño de la interfaz de usuario, se ha optado por suponer que en el caso de que la tracción del vehículo sea en los dos ejes, F_1 y F_2 sean iguales.

3 EL SISTEMA DE TRANSMISIÓN EN EL AUTOMÓVIL

Como bien es sabido, el sistema de autopropulsión más común en el automóvil es el motor de combustión interna, capaz de dotar a estos de la energía necesaria para su desplazamiento. Para ello, el motor se encarga de transformar la energía del combustible en energía mecánica para así conseguir el desplazamiento del vehículo. A su vez, los citados motores de combustión interna, que constituyen la fuente de potencia del automóvil, tienen que entregar a las ruedas la energía mecánica en forma de giro, para que posteriormente éstas debido al par motor que se produce también en el eje de salida del motor creen una fuerza de tracción longitudinal.

Esto, el motor de combustión interna lo consigue proporcionando a las ruedas la velocidad de giro de su eje de salida. Sin embargo, esta velocidad de giro tendrá que pasar por un grupo de componentes hasta llegar a las ruedas motrices, además deberá de sufrir algunas variaciones en el camino que transcurre desde el motor hasta las ruedas del automóvil, para que la velocidad de giro de las estas, así como el par motor transmitido por el motor, pueda variar dependiendo de las condiciones a las que se enfrente el automóvil. A ese grupo de componentes anteriormente citado se le denomina cadena cinemática o más habitualmente sistema de transmisión, y está presente en cada automóvil hoy en día. Un ejemplo de elemento de transmisión lo encontramos en la caja de cambios, que puede ser manual o automática. Esta junto al embrague, las juntas, el grupo cónico y el diferencial conforman el sistema de transmisión mecánica de un automóvil.

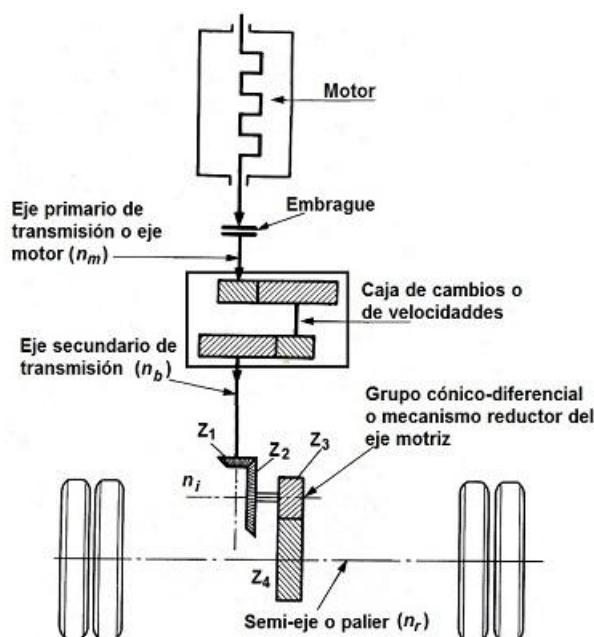


Figura 3-1. Esquema del sistema de transmisión en un automóvil [7]

Además de servir para transmitir la potencia desde el motor hasta las ruedas motrices, el sistema de transmisión del automóvil es necesario para que sea posible la opción de modificar la velocidad de giro de las ruedas, anteriormente citada, así como invertir su sentido de giro. Esto último es de gran ayuda para que el vehículo se adapte perfectamente a las condiciones del terreno por el que se circule en un determinado momento. En otras palabras, una caja de cambios actúa simultáneamente como un transformador de velocidad de giro y como un convertidor mecánico de par.

Es lógico que cada motor tenga un régimen de velocidades angulares de funcionamiento, por lo que tiene una

velocidad angular máxima y otra mínima. Ahora bien, si solo pudiésemos tener una sola relación de transmisión en la caja de cambios, es decir una relación fija, el coche solo tendría una velocidad mínima y otra máxima. Esto no es útil ya que el automóvil deberá satisfacer un gran rango de velocidades, lo que es posible introduciendo más de una relación de transmisión.

Otro problema que puede plantearse, es que con la primera relación de transmisión, se fija una determinada velocidad de la que no se podría bajar a no ser que se apagase el motor, además de no ser posible pasar de una marcha a otra de una forma que no sea brusca. Por ello es necesario el uso del embrague que actúa como un interruptor mecánico entre el motor y la caja de cambios, haciendo que estos se desenlacen al activarlo.

Por otra parte, las juntas universales son necesarias ya que la posición absoluta entre la caja de cambios y las ruedas va a variar a lo largo del trayecto que realice el automóvil. Es por ello que se necesita un elemento de direccionamiento que se encargue de transmitir la potencia del eje de salida de la caja de cambios hasta el eje de las ruedas, ya que estos no van a estar alineados. En concreto, estas juntas son llamadas homocinéticas porque tienen la peculiaridad de que la velocidad de giro es igual en su eje de entrada que en su eje de salida.

Una vez llegados al eje de las ruedas motrices del vehículo, la velocidad de giro tiene que ser transmitida hasta las ruedas. Esto se lleva a cabo mediante el diferencial, que además de ser el elemento repartidor en el sistema de transmisión, hace posible que cada rueda motriz tenga una velocidad de giro distinta, a la vez que cada una mantiene el mismo par. Esto es esencial, ya que por ejemplo en un paso por curva las ruedas interiores y exteriores no recorren la misma distancia.

De esta forma queda definida la función de cada uno de los elementos que componen un sistema de transmisión mecánica común en cualquier automóvil. El siguiente paso será explicar cómo varían la velocidad y el par transmitidos por el motor de combustión interna hacia las ruedas motrices.

3.1 Curvas de potencia y par

Generalmente en un motor el par motor o torque transmitido a través de su eje de salida varía según la velocidad de giro en la que este se encuentre. A su vez, como el par motor varía en función de la velocidad angular del motor, la potencia del mismo también lo hará, ya que esta, como bien es sabido, viene dada por la siguiente expresión:

$$P = T \cdot \omega \quad (3-1)$$

Es por ello, por lo que surgen este tipo de curvas, de las cuales se van a hablar en este subcapítulo. Así pues, las curvas de potencia y par no son más que una representación gráfica de la variación de la potencia y el par motor máximos transmitidos por el motor en función de la velocidad de giro a la que se encuentre este. Este tipo de curvas se obtienen a través de diferentes ensayos con el motor, siendo el más habitual mediante un freno hidráulico.

La curva de potencia y par ideal es aquella en la cual la potencia sería constante independientemente de la velocidad a la que se encuentre girando el motor en un determinado instante. En otras palabras, la potencia máxima del automóvil debería ser constante sea cual sea la velocidad del motor. Esto se puede conseguir en diversos tipos de motores, incluso en el tipo de motor que se trata al hablar de automóviles (motor de combustión interna). En estas las curvas de potencia y par motor difieren un tanto de la ideal, pero esto se puede solucionar gracias a la caja de cambios que deberá tener relaciones de transmisión variables tal y como se verá en el siguiente subcapítulo.

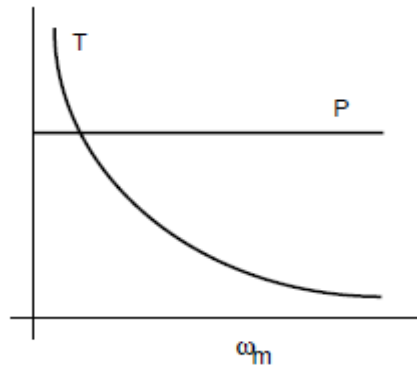


Figura 3-2. Curvas de potencia y par deseadas [2]

Por último recalcar que mientras la velocidad del vehículo solo depende de la velocidad de giro del motor así como de las relaciones de transmisión existentes en la caja de cambios, la potencia se transmite a las ruedas mediante una relación de transmisión fija, que en este caso es el diferencial anteriormente visto.

3.2 Relaciones de transmisión

Como ya se ha visto desde el principio del capítulo en el que nos encontramos, es necesario que el automóvil posea una caja de cambios en la cual se encuentren una serie de relaciones de transmisión variables para que este tenga un correcto funcionamiento y pueda desplazarse en un amplio rango de velocidades.

Habitualmente, en la teoría la relación de transmisión entre dos engranajes se entiende como la división de la velocidad angular del eje de salida y la velocidad angular del eje de entrada. Se entiende que en el caso de una caja de cambios la velocidad angular del eje de salida correspondería con la que llegaría hasta las ruedas del automóvil (antes de pasar por el diferencial del sistema de transmisión) y la velocidad angular del eje de entrada se corresponde con la velocidad que tiene el motor. Por lo que es comprensible pensar que en las primeras marchas (también llamadas marchas cortas), la velocidad del motor siempre será mayor que la que se transmita a las ruedas. Es por ello que la relación de transmisión en el caso de las marchas cortas será menor de 1 e irá aumentando hasta llegar a 1 (marcha directa) e incluso a ser mayor que la unidad.

Sin embargo, mirando varias fichas técnicas de diferentes automóviles se puede ver como los fabricantes toman las relaciones de transmisión como la división de la velocidad angular del motor entre la velocidad angular a la salida de la caja de cambios. Debido a esto vemos como en las marchas cortas el valor de la relación de transmisión es ampliamente mayor que la unidad, además este va decreciendo conforme avanza el número de marcha hasta llegar incluso a estar por debajo de la unidad en las últimas marchas. Por este motivo, para llevar a cabo la interfaz gráfica el autor de este documento ha decidido tomar la relación de transmisión de igual forma que lo hacen los principales fabricantes de automóviles, por lo que esta relación tiene la siguiente forma:

$$r = \frac{\omega_m}{\omega_s} \quad (3-2)$$

Llegados a este punto, ya se intuye cual es el funcionamiento de la relación de transmisión de la caja de cambios dependiendo de la marcha en la que nos encontremos en un instante determinado. De esta forma las primeras marchas o marchas cortas se encargarán de reducir la velocidad que el motor transmite a las ruedas a la vez que actuarán como multiplicadoras de par, por lo que un automóvil en las primeras marchas es capaz de ejercer una gran fuerza de tracción. Por el contrario las últimas marchas o marchas largas, se encargarán de multiplicar la velocidad que el motor proporciona a las ruedas además de reducir el par que llega hasta las ruedas.

3.3 Relación entre el par motor y la fuerza de tracción

Explicadas ya las partes que componen el sistema de transmisión de un vehículo convencional, así como qué es una relación de transmisión, ha llegado el momento de establecer varias relaciones existentes en un

automóvil. En este apartado, más concretamente, se explicará la relación existente entre el par motor, que se produce en el eje de salida del motor, y la fuerza de tracción que mueve al vehículo.

Antes de ello, es necesario establecer cómo se relaciona el par motor a la salida del motor y el par transmitido al eje de las ruedas. Para saber qué relación tienen estos dos pares, hay que tener en cuenta la expresión de la potencia que ya se vio en el subcapítulo dedicado a curvas de potencia y par.

Caracterizando el término 3-1, que expresa la ecuación de la potencia, por una parte para el eje de salida del motor, y por otra parte para el eje de las ruedas nos quedan las dos expresiones siguientes:

$$P_{mot} = T_{mot} \cdot \omega_m \quad (3-3)$$

$$P_{Rueda} = T_{Rueda} \cdot \omega_{Rueda} \quad (3-4)$$

A continuación se divide el término 3-4 entre el término 3-3 y queda la siguiente ecuación:

$$\eta = \frac{T_{Rueda}}{T_{motor}} \cdot \frac{1}{r \cdot r_d} \quad (3-5)$$

En la anterior ecuación queda definida la relación entre el par motor a la salida del motor y el par motor en el eje de las ruedas. Siendo η el rendimiento de la transmisión y r la relación de transmisión entre el eje de salida del motor y el eje a la salida de la caja de cambios, tal y cómo se ha definido con anterioridad.

Conociendo ya esa relación, solo basta definir la expresión que relaciona la fuerza de tracción con el par que tiene lugar en el eje de las ruedas, la cual aparece a continuación:

$$T_{Rueda} = F \cdot R \quad (3-6)$$

Sustituyendo la anterior expresión en la ecuación 3-5 se obtiene lo siguiente:

$$F = \frac{T_{motor} \cdot r \cdot r_d \cdot \eta}{R} \quad (3-7)$$

En dicha ecuación se observa finalmente la relación entre el par motor transmitido por el motor y la fuerza de tracción o resistente (dependiendo de cómo sea la pendiente) del vehículo.

3.4 Relación entre la velocidad del motor y la del automóvil

Si en el apartado anterior se establecía la relación entre el par motor y la fuerza de tracción del automóvil, en la sección en la que nos encontramos en este momento se establecerá la relación entre la velocidad del motor de combustión interna del automóvil y la velocidad que tiene el vehículo.

En primer lugar, hay que saber cuál es la expresión que relaciona la velocidad angular de la rueda con su velocidad lineal. Como se puede observar a continuación esta relación es la siguiente:

$$v_{Rueda} = \omega_{Rueda} \cdot R \quad (3-8)$$

Una vez se conoce dicha ecuación solo basta poner la velocidad de la rueda en función de la velocidad de giro del motor (esto se obtiene a través de la definición que se ha hecho anteriormente de la relación de transmisión vista en la ecuación 3-2). Finalmente se obtiene lo siguiente:

$$v_{Rueda} = \frac{\omega_{motor} \cdot R}{r \cdot \eta} \quad (3-9)$$

La expresión anterior constituye, por tanto, la relación entre la velocidad angular de giro de motor y la velocidad de las ruedas, que viene a ser la velocidad del vehículo.

3.5 El freno motor

Como ya se ha visto en los subcapítulos anteriores, el motor es capaz de ofrecer un par motor a las ruedas, y consecuentemente una fuerza de tracción que ayuda a que el vehículo pueda vencer las resistencias que aparecen al subir una pendiente. Pero, además de todo esto el motor también puede ofrecer un par que se

oponga al movimiento de avance del automóvil. Esto sucede, por ejemplo, cuando el conductor pasa de la cuarta a la tercera marcha o a otra marcha más baja en una pendiente descendente.

La acción en la que el motor genera un par motor en oposición al movimiento del vehículo se conoce como freno motor, y es realmente útil sobre todo en pendientes descendentes prolongadas, ya que en dichas pendientes, los frenos del automóvil pueden llegar a calentarse mucho, lo que conlleva un posterior deterioro y acortamiento de su vida útil. Llegados a este punto, hay que señalar que el freno motor no es un componente físico en sí del automóvil, es decir, es tan solo un concepto teórico, aunque se ha decidido explicar este en el sistema de transmisión del automóvil debido a la influencia que va a tener para calcular la velocidad en pendientes descendentes.

Cabe recalcar que el freno motor bajo ningún concepto puede sustituir al uso del sistema de frenado, ya que el freno motor no es más que un sistema con el que se puede llegar a reducir la velocidad del automóvil sin necesidad de pisar el pedal de freno, por lo que su uso aumenta la vida útil del sistema de frenado. Sin embargo, hay situaciones en las que se necesita el uso de los frenos ya que no es suficiente con el freno motor, es por ello por lo que este es tan solo un sistema complementario al sistema de frenado.

Como en este trabajo de fin de grado no se van a estudiar las fuerzas de frenado que aparecen en un vehículo debido al sistema de frenado, las únicas fuerzas procedentes del vehículo que se van a oponer al movimiento van a ser las creadas por el par resistente generado por el freno motor. Estas fuerzas van a ser de gran importancia en el estudio de las pendientes descendentes, tal y como se verá más adelante.

3.6 Hipótesis previas al cálculo de la velocidad máxima admisible en pendientes

Antes de explicar cómo se puede calcular la velocidad máxima a la que un coche puede ascender o descender una determinada pendiente, hay que establecer una serie de hipótesis básicas que el autor de este documento ha establecido para que los cálculos sean más sencillos y que no se introduzcan otras cuestiones que dificultarían la implementación de la interfaz gráfica de usuario que se desea diseñar en este trabajo de fin de grado. Se diferenciará entre la hipótesis que hay que tomar diferenciando si la pendiente es ascendente o descendente ya que en el primer caso interviene el par motor mientras que en el segundo interviene el par resistente debido al freno motor.

3.6.1 Pendiente ascendente

La primera hipótesis que se establece para simplificar los cálculos y el problema del ascenso de pendientes es la forma que tendrá la curva que relaciona el par del motor con su velocidad angular. Y es que, esta tendrá forma de impulso, es decir, tendrá unos valores mínimos y máximos de velocidad angular y un valor máximo de par motor, siendo siempre el mínimo par motor existente de valor nulo. La forma de esta gráfica se observa mejor en la siguiente figura.

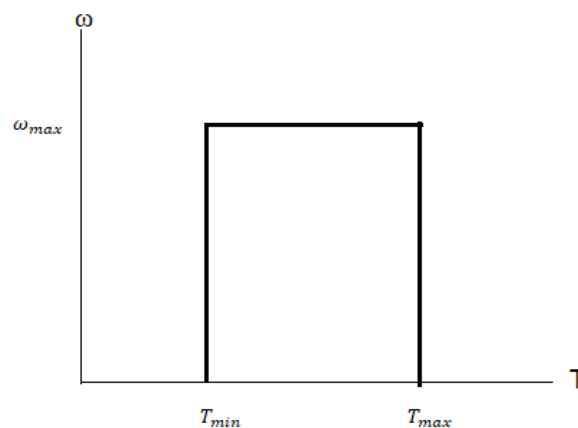


Figura 3-3. Gráfica que relaciona el par motor con la velocidad de giro del motor

Fuente: Elaboración propia

Con esta hipótesis se simplifica sobre todo la solicitud de datos al usuario, ya que si la gráfica no tuviese esta forma se deberían pedir más valores, a parte de los máximos y los mínimos citados en el párrafo anterior (incluso se le podría llegar a pedir al usuario que ingresara la forma que tendría la gráfica del par motor y la velocidad angular del motor), y esto puede no resultar efectivo y dar lugar a fallos o incluso a modelos muy alejados de la realidad al no conocerse el aspecto real de la curva.

Otra de las hipótesis que hay que realizar antes del cálculo de la máxima velocidad admisible en pendientes ascendentes está relacionada con la relación que se establecía entre el par motor y la fuerza de tracción, cuya forma se puede ver en la ecuación 3-7. La hipótesis que se hace esta vez es que la eficiencia de la transmisión es de un 100% (en otras palabras no se producen pérdidas entre el eje de entrada de la caja de cambios y el eje de las ruedas), además la relación fija del diferencial será de 1 (ya que suponemos que la pendiente no tendrá curvas y no hará falta variar el par de las ruedas motrices), por lo que se simplifican mucho los cálculos. Cabe resaltar que la relación fija del diferencial no afectará a la relación entre la velocidad del motor y la velocidad del automóvil porque la velocidad de giro del motor se transmite tal y como sale, una vez pasa por la caja de cambios, hasta las ruedas, cosa que no pasaba en términos de potencia.

Por último para el cálculo de la velocidad máxima admisible se despreciará el término de las fuerzas de inercia en las fuerzas resistentes al movimiento.

3.6.2 Pendiente descendente

En el caso de la pendiente descendente en primer lugar se va a suponer que el valor del par resistente que puede transmitir el motor mediante el freno motor va a ser del mismo valor que el par motor que el mismo motor podía transmitir.

Añadir que las hipótesis simplificativas para el caso de pendiente ascendente son igualmente válidas en el caso de pendiente descendente. Recaltar además que la velocidad que se calcule en la pendiente descendente es aquella para la cual se podría controlar la velocidad del vehículo tan solo levantando el pie del acelerador, sin necesidad de pisar el pedal de freno.

3.7 Velocidad máxima admisible en una pendiente

Este subcapítulo como su nombre indica estará dedicado a explicar cómo se calcula la velocidad máxima que un automóvil puede alcanzar en una pendiente, ya sea esta ascendente o descendente. Para ello se han tenido en cuenta conceptos teóricos relativos al equilibrio dinámico que se hacía en el capítulo anterior del modelado dinámico. Además en este subcapítulo también se utilizan, como no podía ser de otra forma, las relaciones, anteriormente obtenidas, entre el par motor y la fuerza de tracción y entre la velocidad de giro del motor y la velocidad del automóvil. A todo ello hay que sumarle las hipótesis simplificativas vistas en el subcapítulo anterior.

En primer lugar hay que tener en cuenta el equilibrio dinámico del automóvil y la expresión de la fuerza resistente que se producía en el vehículo, la cual se obtenía gracias al equilibrio y que tenía la forma expresada en la ecuación número 2-3.

Ahora parece lógico pensar que el automóvil podrá ascender una pendiente siempre que la fuerza de tracción que este sea capaz de dar sea mayor o igual que la fuerza resistente. Por lo tanto será necesario obtener los valores de dicha fuerza en función de la relación de transmisión que tenga la marcha en la que se encuentre el automóvil. Así mismo habrá que realizar la misma operación para obtener los valores de velocidad máxima y mínima de cada marcha. Debido a esto será necesario conocer el valor de par motor máximo transmitido por el motor, así como los valores máximos y mínimos de la velocidad de giro que este pueda alcanzar.

Con todos los datos expuestos en el párrafo anterior, se obtendría una gráfica similar a la que aparece a continuación, en la cual se representa en el eje y la fuerza de tracción que puede llegar a ofrecer el vehículo dependiendo de la marcha en la que se encuentre y en el eje x las velocidades máximas y mínimas del vehículo (expresadas en m/s) disponibles en cada marcha.

Por otra parte, con el equilibrio dinámico se debe obtener otra gráfica en la que se represente la fuerza resistente, debida a las fuerzas presentes en la subida de la pendiente, en función de la velocidad del vehículo,

obteniendo una gráfica similar a la siguiente. En dicha gráfica se mostrarán las fuerzas resistentes que aparecerán en función de la pendiente que tengamos, todo ello con unos datos característicos del vehículo concretos.

Una vez se tengan ambas gráficas habrá que superponer una encima de otra para observar cuál es la velocidad máxima admisible del vehículo dependiendo de la pendiente en la que se encuentre. Para obtener esta velocidad máxima solo habrá que observar cuál es aquella velocidad máxima en la que la fuerza de tracción es mayor o igual que la fuerza resistente. Esta velocidad máxima alcanzable por el vehículo en una pendiente determinada será por tanto aquella que cumpla esta condición y se dé en la mayor marcha posible (en el caso que se cumpla en varias marchas).

4 EL PROBLEMA DEL CONSUMO DE COMBUSTIBLE

En este capítulo se va a tratar uno de los problemas más importantes a optimizar en un automóvil, y de gran interés para el usuario a la hora de adquirir un nuevo vehículo. Este problema al que se hace referencia no es otro que el del consumo de combustible, el cual depende de un amplio abanico de factores. Además este consumo no es constante a lo largo de todo el trayecto que el conductor realice, ya que sobre él influirá enormemente las condiciones de conducción.

De este problema parte también el concepto de conducción eficiente tan desarrollado hoy en día. Un concepto que está además íntimamente ligado a muchos de los contenidos vistos en el capítulo anterior, relacionados con el sistema de transmisión del automóvil. Esto se debe a que algunos estudios han podido probar que el consumo de combustible puede disminuir si el conductor hace un buen uso en todo momento de las relaciones de transmisión de las que dispone la caja de cambios de su automóvil así como del freno motor (en caso de pendientes descendentes).

Como es un tema bastante amplio, el autor de este documento decidió simplificar muchos de los parámetros que intervienen a la hora de calcular el consumo de combustible del automóvil ya que tratar algunos de ellos era bastante complejo y excedía los objetivos de este trabajo de fin de grado. De esta forma, en este trabajo de fin de grado no se explicará en profundidad el ciclo de Otto que es el que se desarrolla en los motores de combustión interna y del cual parten las ecuaciones que se irán viendo a lo largo de este capítulo. Esto es debido, a que en el caso de que se expusiera este ciclo habría que tratar conceptos termodinámicos un tanto complejos, los cuales escapan al alcance de este trabajo cuyo objetivo primordial se recuerda que es el de diseñar una interfaz gráfica para optimizar la subida y bajada en cuesta, todo ello basándose en la dinámica longitudinal del automóvil y algunos conceptos relativos a la transmisión del automóvil.

4.1 Consumo específico de combustible

El consumo específico de combustible (Brake Specific Fuel Consumption) es el parámetro esencial a la hora de calcular el consumo de combustible que tiene un automóvil. Este se define como la división entre el flujo másico de combustible que se produce en el motor, medido en g/s, y la potencia suministrada por él mismo medida en kW. Como el consumo específico se suele expresar en g/kW·h su expresión queda de la siguiente forma:

$$BSFC = \frac{\dot{m}}{P_{motor}} \quad (4-1)$$

A dicha expresión se ha llegado a través del ciclo termodinámico que ha tenido lugar en el motor de combustión, el cual no se va a exponer porque escapa a los objetivos del presente documento tal y como se dijo anteriormente. El problema a la hora de calcular el consumo específico de combustible está en cómo obtener los datos del flujo másico de combustible en el motor, ya que esto suele obtenerse mediante ensayos con un dinamómetro en un banco de ensayos adecuado. Sin embargo, indagando en internet, el autor de este documento encontró una tabla de valores de flujos másicos para una velocidad angular y un par motor dado los cuales el autor de dicho estudio había obtenido de la forma ya citada. Dicha tabla es la que se va a usar para optimizar el consumo de combustible en la interfaz gráfica, aunque tiene un par de inconvenientes que se exponen a continuación.

El primero es que, quizás esta no sea una forma realista de obtener el consumo específico en un determinado motor de un vehículo ya que el flujo másico de combustible es distinto en cada motor, sin embargo es una

buena forma para empezar a hacer un diseño ya que la realización de ensayos con un motor excedía el alcance de este trabajo. El segundo inconveniente es que en dicha tabla solo se muestra el valor del flujo másico de combustible hasta un par con un valor de 171.6 N·m, un valor un tanto bajo teniendo en cuenta el par motor de los automóviles que hay hoy en día en el mercado. Esto se intentará subsanar de una forma que se explicará en los siguientes subcapítulos pero desde ya se adelanta que va a ser uno de los grandes inconvenientes del mapa de consumo y por tanto el consumo real del combustible variará un poco respecto al que se obtenga en la interfaz.

Una vez se tengan cada uno de los valores del flujo másico de combustible para un par y una velocidad angular determinados, se construirá una especie de mapa que se explica en el siguiente sub-capítulo. Con el mapa y conociendo el par motor y la velocidad angular del motor en cada marcha se calculará la marcha más óptima para circular por una determinada pendiente, que será aquella que ofrezca un consumo mínimo.

4.2 Mapa de consumo

El mapa de consumo muestra el consumo específico existente en un determinado motor en función de una velocidad angular y carga (medida en bar) determinadas. Aunque también existen mapas de consumo que muestran el consumo específico de combustible en función de la velocidad angular y el par. Este último es el que se quiere obtener en este documento.

Para ello se ha hecho uso de una tabla de valores encontrada en la página x-engineer.org en la que se muestran los valores del flujo másico de combustible obtenido en un banco de ensayos para un determinado motor y para unas velocidades de giro y par en concreto. Esta tabla se muestra a continuación.

Fuel mass flow rate [g/s]	Engine torque [Nm]	15.6	31.2	46.8	62.4	78	93.6	109.2	124.8	140.4	156	171.6
	MEP [bar]	1.03	2.06	3.09	4.12	5.16	6.19	7.22	8.25	9.28	10.31	11.34
Engine speed [rpm]	500	0.1389	0.2009	0.2524	0.3006	0.3471	0.4264	0.4803	0.5881	0.5881	0.6535	0.7188
	1000	0.2777	0.3659	0.4582	0.5587	0.6453	0.7792	0.8977	1.0325	1.1762	1.3069	1.4376
	1500	0.4166	0.5538	0.7057	0.8332	0.9557	1.0733	1.2127	1.3428	1.5438	1.9604	2.1564
	2000	0.5391	0.7188	0.9116	1.0913	1.2497	1.4115	1.5552	1.7774	2.029	2.3851	2.8752
	2500	0.633	0.8658	1.0904	1.2906	1.5111	1.6786	1.944	2.2217	2.4995	2.8997	3.594
	3000	0.7106	0.9949	1.2718	1.5193	1.7888	2.0878	2.3671	2.6661	2.9993	3.5286	4.3128
	3500	0.7433	1.0806	1.3722	1.7839	2.2013	2.549	2.8817	3.1562	3.5507	4.1739	5.0316
	4000	0.9475	1.2938	1.729	2.2087	2.5648	2.9993	3.3391	3.6855	4.2932	4.8355	5.7504
	4500	1.1027	1.6026	2.1525	2.5877	2.9957	3.4184	3.8852	4.4108	5.0151	5.6238	6.4692
	5000	1.5519	2.091	2.573	3.0222	3.4715	3.8717	4.4998	5.0642	5.7781	6.4528	7.188
	5500	1.8868	2.5517	3.1537	3.6479	4.0882	4.4206	5.2203	5.8941	6.55	7.2329	7.9068
	6000	2.0584	2.8817	3.5286	4.0775	4.5578	5.1165	5.6948	6.43	7.1455	7.8414	8.6256

Figura 4-1. Tabla con los valores del flujo másico de combustible [8]

Mediante esta tabla y a través de MATLAB se obtendrá el mapa de consumo específico de combustible con el cual se calculará cuál es el consumo óptimo de cualquier tipo de vehículo con motor de combustión interna que el usuario desee obtener. La forma de obtener este consumo óptimo, así como la marcha óptima en la cual se alcanza, se expone en siguiente subcapítulo.

Sin embargo esta tabla tiene una serie de limitaciones ya expuestas dentro de este mismo capítulo, la más importante es el valor límite de par de 171.6 Nm. Este valor es demasiado bajo para los vehículos que encontramos en el mercado hoy en día, es por ello que en el siguiente subcapítulo también se explicará la forma de proceder de el usuario si el valor del par motor en alguna de las marchas es superior a 171.6 Nm.

4.3 Consumo óptimo de combustible

Este subcapítulo se va a dedicar a explicar cómo se calcula el consumo óptimo de combustible en un automóvil que asciende o desciende una determinada pendiente a una velocidad constante. Este consumo irá íntimamente ligado a la relación de cambio en la que se encuentre la caja de cambios del vehículo por lo que el

cálculo del consumo óptimo de combustible también conlleva obtener cuál es la marcha en la cual el automóvil tiene un consumo menor.

Lo primero que hay que calcular es el momento que tiene que ejercer el motor dependiendo de la velocidad y la pendiente dada por el usuario. Es por eso que hay que establecer cuál es la fuerza mínima que tienen que aportar las ruedas, dependiendo de si la pendiente es ascendente o descendente:

- Si la pendiente es ascendente la fuerza que tienen que ejercer las ruedas es de tracción y esta deberá de ser como mínimo del valor de las fuerzas resistivas que se oponen al movimiento del vehículo en el ascenso de una pendiente (esto ya se vio en el capítulo 2). El valor de estas fuerzas resistivas dependerá de la velocidad constante a la que ascienda el automóvil la pendiente, así como del valor de la misma, ambos datos dados por el usuario.
- Si la pendiente es descendente, la fuerza que ejercen las ruedas es resistiva, ya que en descenso se iba a controlar la velocidad del vehículo con el freno motor debido a que la componente de la fuerza resistiva por pendiente en este caso va a favor del movimiento del vehículo (esto también se explicó en el capítulo 2). Puesto que esto es así esta fuerza de frenado como mínimo deberá tener el valor de las fuerzas resistivas, que en esta ocasión irán a favor del movimiento una vez se realice la suma de todas ellas.

Una vez calculada la fuerza que ofrecen las ruedas del vehículo, se obtiene el valor del par motor que se produce en las ruedas. Esto se hace de manera sencilla, a través de la ecuación 3-6 vista en el capítulo 3.

Ahora solo queda saber cuál es la velocidad de giro de la rueda, algo trivial conociendo la velocidad a la que asciende el vehículo (para ello se hace uso una vez más de una ecuación vista en el capítulo relativo al sistema de transmisión del automóvil, más concretamente la ecuación 3-8).

Cuando ya se hayan obtenido tanto el par producido en la rueda, así como la velocidad de giro de esta, solo queda obtener el par motor y la velocidad de giro del motor dependiendo de la marcha en la que se encuentre la caja de cambios. Esto se consigue a través de las relaciones de cambios, la relación del diferencial y el rendimiento de la transmisión.

Para obtener el consumo óptimo de combustible del automóvil en cuestión solo queda señalar en el mapa de consumo correspondiente (en el caso que ocupa a este trabajo de fin de grado solo se dispone de un mapa de consumo) cada pareja de par motor y velocidad de giro del motor en función de la marcha. Si el par motor obtenido en los cálculos es superior a 171.6 Nm, este se supondrá que tiene el valor máximo al que se puede llegar en la gráfica, es decir su valor pasará a ser de 171.6 Nm. Se procederá de igual forma si la velocidad de giro del motor excede en alguna de las marchas a la máxima de la gráfica, que tiene un valor de 6000 rpm, o si por el contrario es menor que la velocidad de giro mínima de la gráfica, esta es de 500 rpm. Esta suposición es la que hará que el consumo de combustible que se obtenga a través de la interfaz no se asemeje tanto como se desearía a la realidad.

La marcha a la que pertenezca la pareja con la que se obtenga el menor consumo será la marcha óptima de consumo de combustible del vehículo en la pendiente dada para la velocidad especificada. Este consumo real se calcula convirtiendo el consumo específico de combustible (en g/KWh) obtenido del mapa de consumo a consumo del vehículo (en L/100 Km). Esta conversión se realiza a través de varios parámetros tales como la potencia del motor, la densidad del combustible y la velocidad constante a la que circula el automóvil en cuestión.

5 INTERFAZ GRÁFICA EN MATLAB

5.1 Introducción a las interfaces de usuario en MATLAB

Como bien es conocido por los usuarios de MATLAB, este software posee una gran cantidad de funciones de gran utilidad en el campo científico y tecnológico. Es una de las herramientas de las herramientas más eficaces para todo ingeniero, ya que estamos hablando de uno de los softwares más potentes del mercado capaz de realizar cálculos de cualquier tipo.

Además de esto, otra de las aplicaciones con la que cuenta MATLAB es la posibilidad de crear tu propia interfaz gráfica de usuario, algo que vamos a tratar en este trabajo de fin de grado tal y como el lector ya sabrá llegados a este punto.

Alguien que vea el término de “Interfaz Gráfica de Usuario” por primera vez se puede preguntar qué es dicho término. Pues bien, una interfaz gráfica de usuario o GUI (Graphics User Interface) en inglés no es más que un medio de comunicación entre el usuario y el ordenador, en nuestro caso. La creación de GUIs en MATLAB te permite incluir en las mismas todo tipo de herramientas como botones, menús, menús deslizantes entre otras, lo que posibilita una gran interacción entre el usuario y el programa, todo ello integrado en un entorno gráfico. De esta manera, siempre que el usuario pulse sobre alguna de estas herramientas el programa realizará alguna acción determinada por el propio programador a través de la llamada a una cierta función denominada Callback Function o función de llamada. Sobre este tipo de funciones hablaremos más adelante, en un subcapítulo dedicado a su funcionamiento y forma.

En MATLAB, a la hora de programar las citadas interfaces gráficas de usuario se pueden seguir dos caminos distintos. Uno es la de la herramienta GUIDE (Graphics User Interface Design Environment), una opción que te ofrece el propio software de una forma muy simple, tan solo necesitas escribir la palabra “guide” en su ventana de comandos o Command Window. Por otra parte, tienes la opción de diseñarlo todo tú mismo empezando desde cero y escribiendo todo el código necesario para generar la GUI. En otro de los subcapítulos de este capítulo dedicado a las interfaces gráficas, se detallará cuál es la forma que ha elegido el autor de este documento de programar su GUI, así mismo se darán las causas que han llevado a la toma de esta decisión.

5.2 El sistema Handle Graphics de MATLAB

La idea de este subcapítulo es la de hacer entender al lector de este documento, cómo MATLAB organiza el control y diseño de todos sus objetos gráficos en el entorno del propio software. Todo ello está englobado en un complejo sistema denominado Handle Graphics, del cual se dará una introducción que ayude a comprender el trasfondo que tiene lo que el usuario ve en la pantalla de su ordenador una vez inicia la interfaz gráfica de MATLAB.

Para empezar a entender como organiza MATLAB a sus objetos gráficos hay que saber que los distintos objetos gráficos (un ejemplo de éstos puede ser una gráfica que generamos mediante una determinada orden como puede ser la orden plot), que existen en el software se organizan mediante una estructura firmemente jerarquizada. Dicha estructura divide a los objetos según su clase, como si de familias se tratase, quizás es por ello por lo que dicha estructura recibe el nombre de Parent-Child Estructure. Lo que en otras palabras quiere decir que cada ‘familia’ de objetos tendrá una especie de objeto ‘padre’, y así sucesivamente, siendo el objeto principal el objeto raíz o root que es con el que da comienzo todo, ya que es en otras palabras es la ventana principal del software. Esta explicación de entrada puede parecer un tanto difícil de entender, pero como se verá en la siguiente imagen es más sencillo de lo que parece.



Figura 5-1. Organización sistema Handle Graphics [6]

Como se denota en la imagen después del objeto raíz, vendría el objeto figure que se corresponde con las diversas ventanas gráficas que se abren al realizar ciertas acciones en MATLAB (es lo que se abre al crear una gráfica o al inicializar una GUI en concreto). Seguido de estas hay dos tipos de objetos, por un lado están los axes y por otro los UIObjects. En un objeto tipo figure puede haber uno o más ejes o axes en inglés, según se desee graficar en dicha ventana, a su vez dentro del objeto tipo axes hay diversos objetos de los cuales resaltamos los Core Objects. Estos tipos de objetos son los que contienen diversos objetos como las líneas o el texto que puedes observar cada vez que creas una nueva gráfica. Estas gráficas a su vez pueden ser creadas con diversas órdenes, por ejemplo plot, que es la más común. Esta orden se enmarca en el grupo de Linear Series de la familia Plot Objects que pertenece a su vez a la familia de objetos tipo axes, tal y como se ha podido observar. Además de los Plot Objects y los Core Objects, en los objetos de tipo axes también hay otras dos familias más, que no se suelen usar con tanta frecuencia como los anteriores pero que también es importante remarcar. Estos son los Annotation Objects y los Group Objects, que como su nombre indica sirven para realizar anotaciones y para controlar diversos grupos de objetos gráficos, respectivamente.

Por otra parte, como se ha dicho en el párrafo anterior, en el objeto tipo figure también existen otro grupo de objetos, que son los que se van a utilizar en su mayoría en la creación de la interfaz gráfica, de ahí la importancia en entender como MATLAB los controla y organiza. Estos son los denominados UIObjects (de User Interface Objects), de los cuales existen varios grupos como UIButtongroup (para crear un grupo de botones), UIMenu (para crear un menú), o en los que se va a centrar el siguiente subcapítulo, los UIControl.

Teniendo ya una breve noción básica de cómo se organiza el sistema de Handles Graphics en MATLAB y por consiguiente cómo controla el software los distintos objetos gráficos de los que dispone, el lector ya está preparado para empezar a aprender los distintos elementos de los que está compuesta una interfaz gráfica de usuario en MATLAB.

5.3 UIControls y sus propiedades

Como se ha visto en el subcapítulo anterior el sistema Handle Graphics de MATLAB se compone de distintas

familias que se rigen bajo una estructura jerarquizada llamado Parent-Child. Así mismo hemos tratado brevemente los UIObjects, que son una de las dos familias de las que se componen los objetos tipo figure. Los User Interface Objects, como ya se dijo son los objetos más importantes a la hora de diseñar una interfaz gráfica de usuario en MATLAB, y en este subcapítulo se va a tratar el tipo más importante que hay de esta familia de objetos: los UIControls, de los cuales se verán los distintos tipos que existen, así como sus propiedades.

Los UIControls son la principal herramienta a la hora de crear una interfaz gráfica interactiva, ya que como se podrá observar más adelante, la mayoría de los objetos de esta familia resultarán familiares y es que están presentes en cualquier App que se utilice hoy en día. Antes de describir brevemente los distintos grupos de UIControls, se va a explicar cómo se invocan en MATLAB, lo cual se hace de la siguiente forma:

```
uicontrol('Style','tipo de UIControl','Propiedad del UIControl',Valor  
de la propiedad)
```

En la sentencia se observa como mediante la propiedad 'Style' de los propios UIControls se puede crear cualquier tipo de estos, los cuales se exponen a continuación junto a sus propiedades más importantes y redactando entre paréntesis la opción que habría que poner en 'Style' para invocarlos:

- Check Boxes ('checkbox'): Estos tipos de controles se usan para dotar al usuario de la posibilidad de habilitar o desactivar funciones individuales dentro de una GUI. Son las denominadas 'casillas de verificación', cuya apariencia es la de pequeño cuadrado en el cual cuando pulsas sobre él se produce una marca en el mismo. En estas la acción de marcar una de estas casillas no influye sobre el estado de las demás, en el caso de que las hubiese. De esta manera cuando está marcada su propiedad 'Value' pasa de ser 0 a ser 1.
- Radio Buttons ('radiobutton'): Su funcionamiento es similar al de las casillas de verificación, sin embargo en este tipo de control los botones no son independientes entre sí, ya que al pulsar sobre un radio button que no estuviera marcado, cambia el valor de uno que si lo estuviera. Consecuentemente cambia su propiedad 'Value', la cual solo podrá tener un valor de 1 en uno de los radio buttons que conformen el grupo de estos en una GUI.
- Edit Text Boxes ('edit'): Uno de los controles más útiles en una app, ya que permiten al usuario del programa una gran interacción. Son utilizados habitualmente para pedir a la persona que use la interfaz un determinado dato, el cual podrá introducir a través del propio teclado y que quedará recogido en la propiedad 'String'.
- Static Text Labels ('text'): Este tipo de control proporciona al usuario la capacidad de crear cajas de texto en la propia interfaz. El usuario podrá modificar el texto en el interior de la caja de texto a través de la propiedad 'String', pero una vez iniciado el programa no existirá interacción con dicha caja, al contrario que el anterior tipo de control.
- List Boxes ('listbox'): En este caso, se trata de una lista de acciones, de las cuales el usuario puede seleccionar una solo o varias, es decir no es excluyente pulsar sobre una acción de las contenidas en la lista. La lista de acciones se puede modificar a través de su propiedad 'String' y cada línea se corresponde con un valor en la propiedad 'Value', la cual si tan solo hubiese un elemento en la lista adoptaría un valor de 1, y en el caso de tener más tendría forma de vector con los valores sucesivos a la unidad en cada índice del vector.
- Push Buttons ('pushbutton'): Estos pueden ser, fácilmente, el tipo de control más usado al diseñar una GUI interactiva. Este elemento de las interfaces no es más que un botón con el que el usuario puede interactuar pulsando sobre él, acto seguido el Push Buttons llamará a una función denominada Callback Function y realizará una determinada acción, según sea la finalidad de la GUI.
- Toggle Buttons ('togglebutton'): Es muy parecido al anterior, pero con una peculiaridad, y es que al pulsarlo, se queda en ese estado hasta que vuelvas a pulsar sobre él. En otras palabras, tiene un

funcionamiento similar al que sucede cuando pulsas el play al reproducir una canción en cualquier app de música.

- **Sliders** (`'slider'`): Como su propio nombre indica es un deslizador, el cual está incluido en una barra. Con este tipo de control el usuario puede modificar una propiedad de la interfaz gráfica que está manejando con tan solo deslizarse por la barra del slider, que tendrá un valor mínimo y uno máximo.
- **Pop-up Menus** (`'popupmenu'`): Similares a los List Boxes anteriormente mencionados pero con algunas diferencias. Las principales son que la lista de acciones no aparece directamente en la interfaz, sino que pulsando se despliega dicha lista. Además este tipo de control tan solo deja que el usuario elija una opción de la lista desplegable.

Al declarar un UIControl en concreto, en la sentencia se veía cómo además se podía cambiar cualquier propiedad que este tuviese. A continuación se exponen las propiedades más importantes y comunes (salvo las Callback Functions que tienen un subcapítulo a parte dedicado a ellas), las cuales servirán de ayuda para desarrollar este trabajo de fin de grado:

- **Position**: Es la primera propiedad que se debe modificar ya que es la que determina en qué posición de la ventana gráfica está situado el UIControl que se haya declarado. El valor que hay que darle a esta propiedad es un vector formado por cuatro elementos. Los dos primeros elementos se refieren a las coordenadas dónde se va a situar el origen del UIControl en la ventana gráfica. Por otra parte, los otros dos elementos del vector se refieren a las dimensiones que va a tener dicho control.
- **BackgroundColor**: Esta es una propiedad básicamente visual y estética, no cambia nada del UIControl, solo su apariencia. El valor que habrá que pasarle por tanto será un determinado color ya sea en formato color con su nombre en inglés o un vector de tres elementos que se corresponde con el color que se le quiera dar.
- **Tag**: Como su nombre indica es la etiqueta que se le da al UIControl que se crea. Esta es una de las propiedades más importantes ya que a través de ella es como MATLAB ve al UIControl, y siempre que se quiera hacer referencia a él se le tendrá que llamar por su 'nombre' que será el que se le haya dado previamente en el campo de esta propiedad.
- **String**: Esta propiedad ya se ha mencionado anteriormente cuando se han visto los distintos tipos de UIControl presentes en MATLAB. Se corresponde con el texto que va a ver el usuario en un cierto tipo de control cuando este inicialice la interfaz gráfica. Sin embargo también podría tratarse de un dato que el usuario introduzca, tal y como vimos en el control de tipo edit text boxes.
- **FontName**: Se corresponde con el tipo de letra que tendrá la propiedad String cuando aparezca en pantalla.
- **FontSize**: Se corresponde con el tamaño de letra que tendrá la propiedad String cuando aparezca en pantalla.
- **Visible**: Esta propiedad es de gran importancia, sobre todo en ventanas gráficas, ya que puede darse el caso de que al programar una GUI sea necesario el uso de varias de ellas y llegado un punto puede que alguna ventana gráfica de las creadas ya no sea necesaria. Gracias a esta propiedad podemos hacer que estas desaparezcan dándole el valor de `'off'`, llegado el punto. Por tanto esta propiedad solo admite dos valores, que son los de encendido y pagado (On-Off).

5.4 Manipulación de las propiedades de los objetos gráficos: get y set

Vistas las propiedades más importantes que tienen los diferentes elementos de los que puede estar compuesta una GUI, resulta interesante explicar cómo se pueden manipular dichas propiedades, aunque se les haya dado ya un valor cuando se ha creado el objeto.

Las dos órdenes con las que se pueden tratar las propiedades de los objetos gráficos son get y set, de las cuales

se verá a continuación su funcionamiento. En primer lugar tenemos get, una orden que es muy útil si se quiere obtener el valor de una determinada propiedad de un objeto gráfico en concreto, y que se invoca mediante la siguiente sentencia:

```
get(Nombre del objeto gráfico, 'Propiedad de la cual se quiere obtener su valor')
```

Por otra parte tenemos la orden set, que se utiliza para cambiar una determinada propiedad del objeto gráfico en cuestión por un valor en concreto. Esta orden tiene la siguiente notación:

```
set(Nombre del objeto, 'Propiedad a la cual se le quiere cambiar el valor', Nuevo valor de la propiedad)
```

Cabe recalcar que además de estas dos órdenes, existe otra forma de manipular las propiedades de los objetos gráficos. Esto es mediante la notación propia de las estructuras, ya que al crear un determinado objeto, las propiedades de este se guardan como un campo de estructura. Para poder obtener las propiedades de esta forma, en primer lugar se tendrá que crear una variable tipo estructura que almacene todas las propiedades del objeto gráfico. A continuación se accederá a cualquier propiedad mediante la notación de estructuras. Esto se entiende mejor con las líneas de código que aparecen más abajo

```
Nombre_variable=Objeto gráfico;  
Propiedad_que_se_quiere_obtener=Nombre_Variable.Nombre_Propiedad;  
Nombre_Variable.Nombre_Propiedad=Valor_Propiedad_se_desea_modificar;
```

La segunda línea del código que se acaba de mostrar haría la misma función que la orden get y la tercera y última línea del código mostrado sería lo mismo que usar la orden set.

5.5 Callback Functions: Qué son y para qué sirven

Anteriormente, en el subcapítulo en el que se veían las propiedades más importantes de los objetos gráficos, faltaba por explicar una de ellas que posiblemente sea la más importante en el caso de algunos objetos de tipo UIControl. Estas son las denominadas funciones de llamada o Callback Functions, que se tratarán en este subcapítulo.

Lo primero que hay que explicar es que las propiedades de los objetos gráficos se pueden dividir en dos: las propiedades de ajuste o setting properties y las propiedades evento-funcionales, entre las que se enmarcan las funciones de llamada a las que hacemos referencia en el contenido de este subcapítulo.

Las propiedades de ajuste son las que hemos visto en los subcapítulos anteriores, las cuales hemos visto que no sirven para mucho más que para modificar ciertos aspectos de un determinado objeto gráfico, así como para obtener su valor. En otras palabras, estas no influyen en la interacción que el usuario pueda tener con la interfaz gráfica.

Sin embargo, esto no ocurre con las propiedades evento-funcionales, ya que lo que suceda tras la interacción del usuario con el programa (un ejemplo de ello puede ser cuando el usuario pulsa sobre un botón de la GUI), dependerá completamente de ellas. Ahí reside la importancia de las Callback Functions, las cuales suelen estar presentes sobre todo en objetos como los Push Buttons.

Todas las funciones de llamada deben de tener la siguiente forma:

```
function nombre_de_la_función(objeto,evento)
```

Como se puede observar, las Callback Functions deben de tener dos argumentos de entrada siempre. Uno de ellos es objeto que se corresponde con la etiqueta que tiene el objeto gráfico desde el cual se llama a la función de llamada. Es decir mediante este argumento, se le pasa a la Callback Function toda la estructura correspondiente al objeto gráfico que realice la llamada, por lo tanto se puede manipular cualquier propiedad

del mismo. Por otra parte en evento se transmite a la función de llamada la información relativa a la causa que ha originado la llamada a dicha función.

Además de estos dos argumentos de entrada, las Callback functions pueden tener varios más, que se les pase directamente desde el objeto gráfico, para ello habrá que crear un campo de estructuras llamando en primer lugar a la función e introduciendo a continuación los argumentos que se le desea pasar a la misma, tal y como aparece a continuación:

```
{@Callback_function,argumento_de_entrada1,argumento_de_entrada2}
```

El campo de estructuras anterior se ubicará al lado de la propiedad de Callback Function que tienen ciertos objetos UIControls, tal y como se vio en la definición de cómo crear un determinado objeto de control. Remarcar que la mayoría de funciones de llamada no necesitan argumentos de entrada adicionales, más allá de objeto y evento. Por lo tanto su llamada se realizará a través del nombre de la propia función precedida del símbolo de arroba.

5.6 MATLAB GUIDE: Ventajas y desventajas

Una de las principales preguntas que se puede hacer el usuario a la hora de empezar a diseñar una interfaz gráfica en MATLAB es por dónde empezar. En relación a dicha cuestión una de las opciones que nos ofrece MATLAB es el uso de una herramienta denominada GUIDE (Graphics User Interface Development Environment). Esta herramienta nos permite crear una interfaz gráfica de usuario de una manera sencilla, pero no todo van a ser ventajas en esta opción que nos ofrece el software. Es por ello, que en este subcapítulo se detallarán las ventajas y desventajas que tiene el uso de GUIDE en el diseño de una interfaz de usuario, así como la opción por la que se ha optado finalmente para diseñar la interfaz que se quería conseguir en este trabajo de fin de grado.

Lo primero que hay que explicar es cómo el usuario puede acceder a la herramienta GUIDE, esto es muy sencillo ya que como se dijo en la introducción del capítulo solo tiene que escribir la orden `guide` en la ventana de comandos de MATLAB (Command Window). Una vez haga esto y pulse la tecla enter, el usuario verá como en su pantalla aparece la ventana de inicio de GUIDE, en ella se da la opción de abrir una GUI ya existente o crear una nueva desde cero (la opción llamada Blank GUI (Default)).

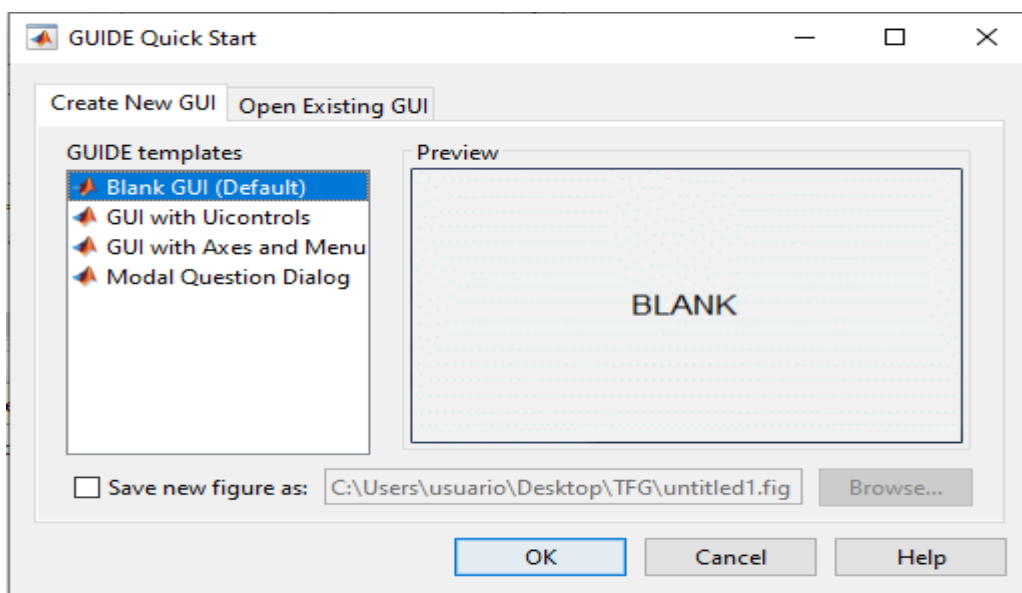


Figura 5-2. Ventana de inicio GUIDE

Haciendo click con el ratón sobre la última opción mencionada aparece el menú de GUIDE o entorno de diseño de la GUI.

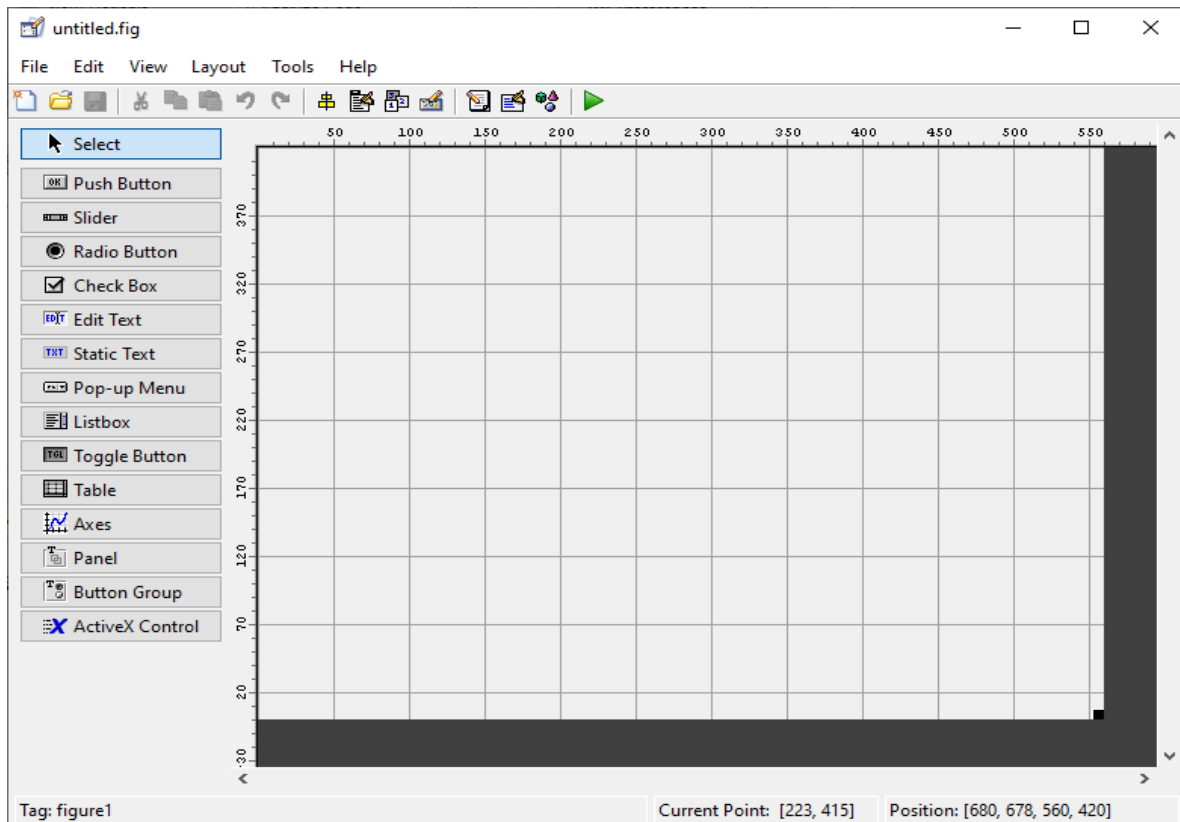


Figura 5-3. Menú de GUIDE

En la figura 5-2 se observa la ‘paleta’ existente en el menú de GUIDE (en el margen izquierdo), la cual está compuesta de los distintos UIControls de los que puede estar compuesta una interfaz de usuario. Al lado de esta ‘paleta de UIControls’ se encuentra lo que sería el objeto figura principal de la GUI, del cual se puede cambiar su tamaño si se quisiera. Para colocar los diferentes UIControls en él tan solo hay que hacer click sobre el que se desee situar en el objeto figura, y a continuación con el puntero del ratón en la figura, dejar pulsado el botón izquierdo del mismo para darle el tamaño deseado.

Una vez tengamos colocados todos los UIControls de los que se va a componer nuestra interfaz gráfica, podemos manipular sus distintas propiedades a través de la opción “Property Inspector”. A esta opción se llega pulsando sobre cualquier UIControl, del cual se quiera cambiar sus propiedades, con el botón derecho del ratón.

Realizando la misma acción que se citaba en el párrafo anterior también se muestra una de las opciones más importantes a la hora de utilizar GUIDE para el diseño de interfaces gráficas. Esta opción es la de “View Callbacks”, con la que al hacer click sobre ella nos aparece el código de Matlab asociado a cada una de las funciones de llamada que tiene los UIControls de la interfaz creada.

De esta forma cuando se crea una interfaz a través de la herramienta GUIDE se generan dos tipos de archivos: un archivo .m que contiene las funciones de llamada de los distintos botones del programa, en otras palabras este archivo es el que controla lo que pasa en el programa cuando el usuario interactúa con él. Y un archivo .fig que es el que contiene los objetos gráficos de la interfaz.

Visto todo esto parece que hasta un usuario inexperto podría crear una interfaz gráfica de forma sencilla dada la facilidad de situar los diferentes UIControls que ofrece la herramienta GUIDE, lo que constituye su principal ventaja respecto a la programación típica de una interfaz. Pero no todo es tan maravilloso como aparenta el entorno GUIDE, ya que al observar el código de MATLAB que se obtiene al diseñar la interfaz a través de dicha herramienta, se ve que puede ser muy complicado de entender para un usuario inexperto del software. Además también se observa como aparecen muchas líneas innecesarias de comentario, las cuales pueden hacer tediosa la lectura del código y dificultar la operación de búsqueda de posibles errores que pueda tener la interfaz en su ejecución.

Es por todo ello, que el autor de este trabajo de fin de grado, así como de la interfaz gráfica de usuario fruto del

mismo, decidió escribir todo el código de la interfaz desde cero. De esta manera el programador tiene todo el control sobre el programa, pudiendo modificar todos los elementos del mismo a su gusto y le será más fácil encontrar los distintos errores que vayan surgiendo a la hora del diseño de la interfaz.

5.7 Conceptos previos de programación

En este capítulo se explicarán las decisiones tomadas por el autor de este documento a la hora de programar la interfaz así como se verán algunos conceptos relativos a la programación en MATLAB. La primera decisión que ha tomado el programador es la de generar todo el código de la interfaz por sí mismo, éste también ha decidido que las funciones de llamada que necesite utilizar durante la redacción del código que da lugar a la interfaz sean subfunciones en vez de funciones anidadas.

La principal diferencia entre las subfunciones y las funciones anidadas es que las subfunciones son funciones que, aunque se encuentren en el mismo fichero que la función principal, están escritas a parte de la función principal pudiendo solamente ser llamadas por esta u otras subfunciones distintas. Mientras que las funciones anidadas son funciones que empiezan y acaban dentro de función principal. Es decir, las subfunciones tienen su propio espacio de trabajo, mientras que las funciones anidadas pueden llegar a compartir variables con la función principal.

Una de las principales desventajas del uso de las subfunciones es que, en algunas ocasiones, tienes que hacer a algunas variables globales, ya que estas pueden estar presentes tanto en la función principal como en la subfunción. Sin embargo, su escritura queda de una forma más clara a la hora de programar, al contrario que las funciones anidadas, que algunas veces pueden dar lugar a confusión al encontrarse escritas dentro de la función principal. De todas maneras, es un tema subjetivo a la hora de programar y todo dependerá de la forma de verlo que tenga el programador.

Otro de los aspectos a tener en cuenta antes de empezar a diseñar la interfaz gráfica es cómo se le pueden pasar a las funciones de llamada otras variables, en el caso de que fueran necesarias, a parte de las estrictamente necesarias. Si las Callback Functions se programaran mediante funciones anidadas no habría problema, ya que estas pueden compartir algunas de sus variables con la función principal. Pero, en la programación de funciones de llamada mediante subfunciones hay que proceder de otro modo, más concretamente hay dos formas diferentes de llevar esta tarea a cabo. Una de esas formas, en realidad, ya se vio en el subcapítulo dedicado a las Callback Functions, el cual era darle como valor a dicha propiedad, en el UIControl en concreto, una estructura que tuviese en un campo la función y en los demás campos las variables necesarias. Sin embargo, existe otra forma, la cual vamos a explicar en este subcapítulo a continuación.

En este caso el paso de variables se va a realizar a través de las órdenes disponibles en MATLAB, `guihandles` y `guidata`. Por una parte la instrucción `guihandles` guarda las variables del objeto activo en el momento de su uso en una variable tipo estructura en la cual cada uno de sus campos equivale a las distintas variables anteriormente mencionadas. Por otra parte `guidata`, sirve para añadir más campos a la estructura ya existente por el uso de `guihandles`, además de ofrecerte la posibilidad de obtener, en la función de llamada, el valor de alguna de las variables del objeto gráfico desde el cual se ha llamado a dicha función, ya que puede darse el caso de que se necesiten alguna de esas variables.

Por ello es importante darle un nombre a cada objeto gráfico que creamos en la interfaz de usuario (mediante la propiedad 'tag', tal y como se ha visto en subcapítulos anteriores). Con esta manera se puede sustituir la forma que se había explicado de pasar determinadas variables a las funciones de llamadas. De hecho es la que el autor de la interfaz gráfica correspondiente a este trabajo de fin de grado ha elegido para desarrollar dicha interfaz. Esta elección se ha tomado teniendo en cuenta la gran cantidad de objetos gráficos que va a poseer el programa, ya que de esta forma es menos complicado.

A continuación se mostrarán algunos bucles y bifurcaciones útiles a la hora de programar y algunos comandos que ofrecen mensajes al usuario.

5.7.1 Bucles y bifurcaciones

Los bucles y las bifurcaciones son elementos de gran importancia a la hora de programar. Por una parte los

bucles permiten repetir una misma acción tantas veces como el programador necesite, ahorrando con ello líneas de código. Y por otra parte están las bifurcaciones que permiten realizar o no una acción dependiendo de una condición que se establece a la entrada de las mismas.

En la interfaz gráfica que se va a diseñar en este trabajo de fin de grado se van a utilizar un tipo de bucle y dos tipos de bifurcaciones que pasan a explicarse a continuación.

- **Bucle for:** Con esta orden el programador puede repetir una determinada acción tantas veces como elementos tenga el vector con el que se recorre el bucle. Para que se entienda mejor se expone a continuación la sintaxis para emplear este tipo de bucle.

```
for variable=vector
    acción;
end
```

Al cambiar el valor de la variable en cada paso del bucle, esta se podrá usar como índice de un vector al cual se le quiera ir cambiando el valor de sus componentes a lo largo del bucle, tal y como sucederá más adelante en el diseño de la interfaz.

- **Bifurcación tipo if:** Este tipo de bifurcación realiza una acción si la condición que se le pone a la entrada de la misma es cierta y otra o ninguna acción si esta condición es falsa. En su definición tiene la siguiente sintaxis.

```
if condición_lógica
    acción_1;
else
    acción_2;
end
```

Este tipo de bifurcaciones además puede tener condiciones múltiples con el uso de la orden elseif dentro del propio if.

- **Bifurcación tipo switch:** Este tipo de bifurcación tiene un funcionamiento parecido a la bifurcación de tipo if, sin embargo tiene algunas diferencias que se van a exponer. La mayor diferencia es que la bifurcación tipo switch actúa en función del valor de una variable y dependiendo de cuántos valores tenga esta variable podrá adoptar más o menos casos. Su sintaxis es la siguiente.

```
switch variable
case 1
    acción_1;
case 2
    acción_2;
case n
    acción_n;
end
```

5.7.2 Mensajes al usuario mediante órdenes

En este sub-capítulo se va a enseñar al lector que es posible crear ventanas gráficas a partir de ciertas órdenes, sin necesidad de recurrir al uso de objetos gráficos para crearlas. Esto es realmente útil ya que en ocasiones, crear nuevas ventanas puede resultar muy tedioso y a veces la declaración de estas resulta innecesaria debido a su simpleza. Es por ello que a continuación se mostrarán algunas de las órdenes para declarar ventanas con mensajes al usuario que se van a usar en el diseño de la interfaz gráfica.

- **msgbox:** Esta orden abre una ventana que da un mensaje al usuario que habitualmente sirve como ayuda o para guiarle a que realice una acción en concreto. Su declaración es la siguiente:

```
msgbox('Mensaje que se desea que aparezca','Título de la ventana')
```

- `inputdlg`: Con esta orden se le puede pedir al usuario todos los datos necesarios para realizar una determinada acción. Es realmente útil cuando es necesario pedir una gran cantidad de datos. Su declaración es la siguiente:

```
msgbox('Dato que se solicita','Título de la ventana',Número de  
líneas que tiene que ocupar cada dato)
```

Señalar que si los datos que se solicitan son más de uno, estos deberán aparecer en formato de celda.

- `warndlg`: Esta orden sirve para advertir al usuario de que está realizando una acción de manera incorrecta o que está introduciendo un valor por encima del límite permitido de dicho valor. Su declaración es la siguiente:

```
warndlg('Mensaje que se desea que aparezca','Título de la  
ventana')
```


6 IMPLEMENTACIÓN DE LA INTERFAZ GRÁFICA DE MATLAB

Teniendo ya claros todos los conceptos expuestos en el anterior capítulo dedicado íntegramente a explicar los aspectos más importantes de las interfaces gráficas en MATLAB, así como algunos conceptos previos de programación necesarios para llevar a cabo dicha interfaz. Además de conocer los conceptos teóricos explicados en los capítulos segundo y tercero de este documento, llegó la hora de empezar a explicar cómo se ha llevado a cabo el diseño de la interfaz para optimizar la subida y bajada en cuesta de un vehículo, objetivo final de este trabajo de fin de grado.

En principio, la interfaz gráfica se va a dividir en tres niveles, correspondiendo cada nivel a una ventana gráfica distinta. En el primer nivel se encuentra la portada, que se corresponde con la pantalla de inicio de la interfaz, y en el segundo nivel el menú en el que el usuario deberá elegir cuál va a ser la incógnita del problema de optimización (la incógnita puede ser la velocidad, el consumo de combustible o la pendiente a la que se enfrenta el vehículo). Finalmente en el último nivel se encuentra la ventana en la que el usuario deberá introducir los datos del vehículo que se le pidan (dependiendo de la incógnita que haya seleccionado previamente), además en dicha ventana gráfica se mostrará la solución tal y como explicaremos más adelante. Todos los códigos de MATLAB que dan lugar a la generación de cada una de las ventanas gráficas que se expondrán a continuación están incluidos en el anexo A.

6.1 Pantalla de inicio

Este es el punto de partida de la interfaz para optimizar la subida y bajada en cuesta, que el autor de este documento ha diseñado para este trabajo de fin de grado. Esta ventana gráfica no es más que la portada de la interfaz de usuario, por lo que solamente contiene un objeto tipo figure, un objeto tipo axes (del que posteriormente se explicará su uso) y un objeto de tipo UIControl, más concretamente un push button, que servirá para dar comienzo a la GUI diseñada. Además hay varios objetos de tipo texto para indicar el título del trabajo de fin de grado, así como el nombre del autor del mismo y del tutor.

En primer lugar, la propiedad Position del objeto figure tendrá el valor de la propiedad ScreenSize de la pantalla del ordenador del usuario que inicialice la interfaz gráfica, dicho valor se obtiene a través de la orden get. Ahora cabe preguntarse, qué nombre tiene el objeto que da lugar a la pantalla del ordenador, este recibe el nombre de 0, ya que es el objeto que da origen a todo, también denominado objeto raíz (Root). Por lo tanto las primeras líneas que aparecerán en el código de la interfaz serán las siguientes:

```
close('all')
global tam_screen
tam_screen=get(0,'ScreenSize');
```

Como se puede observar en tam_screen se ha guardado el valor del tamaño de la pantalla del ordenador del usuario (que puede variar de un ordenador a otro), además esta variable se ha declarado como global porque tal y como veremos más adelante será necesaria en otras ventanas gráficas dentro de la interfaz. Además la primera línea de código corresponde a una orden que le indica a MATLAB que cierre todas las ventanas gráficas que haya abiertas.

Además de cambiar la propiedad Position del objeto figure, hay otras propiedades que se deben cambiar como la de la barra de menú (Menubar) que suele aparecer cuando generamos este tipo de objetos, a la cual habrá que darle el valor de 'none'. Otra propiedad que se cambiará será la denominada Numbertitle, que corresponde al número que se le da al objeto figure, el cual no se quiere que aparezca. Por lo tanto se le dará el valor de apagado (off).

Por otra parte, se necesita generar un objeto de tipo axes para albergar el logo de la Escuela Técnica Superior de Ingeniería (ETSI) que se va a fijar en la pantalla de inicio de la interfaz. Una vez creado este tipo de objeto, habrá que hacer uso de dos órdenes que nos ofrece MATLAB para tratar imágenes en formato jpg. Estas órdenes son `imread` e `image`, la primera de ellas se usa para leer la imagen que el programador desee que sea visible en su GUI, mientras que `image` es la orden usada para visualizar dicha imagen previamente importada mediante `imread`. Esto es posible gracias a que `imread` importa la imagen y hace que MATLAB la vea como una matriz de tres dimensiones compuesta por números, para que posteriormente mediante la orden `image`, se represente la imagen en unos ejes en concreto.

Una vez se definan los ejes con sus propiedades correspondientes, se aplican estas dos órdenes que se han citado mediante las siguientes líneas de código:

```
haxes=imread('tituloLogoGrande.jpg');
image(haxes);
axis off
```

Se puede observar como al objeto axes creado se le ha dado el nombre de `haxes`, además en la última línea se ha hecho invisible la visibilidad de los ejes del mismo objeto. Finalmente en la siguiente figura se puede observar cómo ha quedado esta ventana.

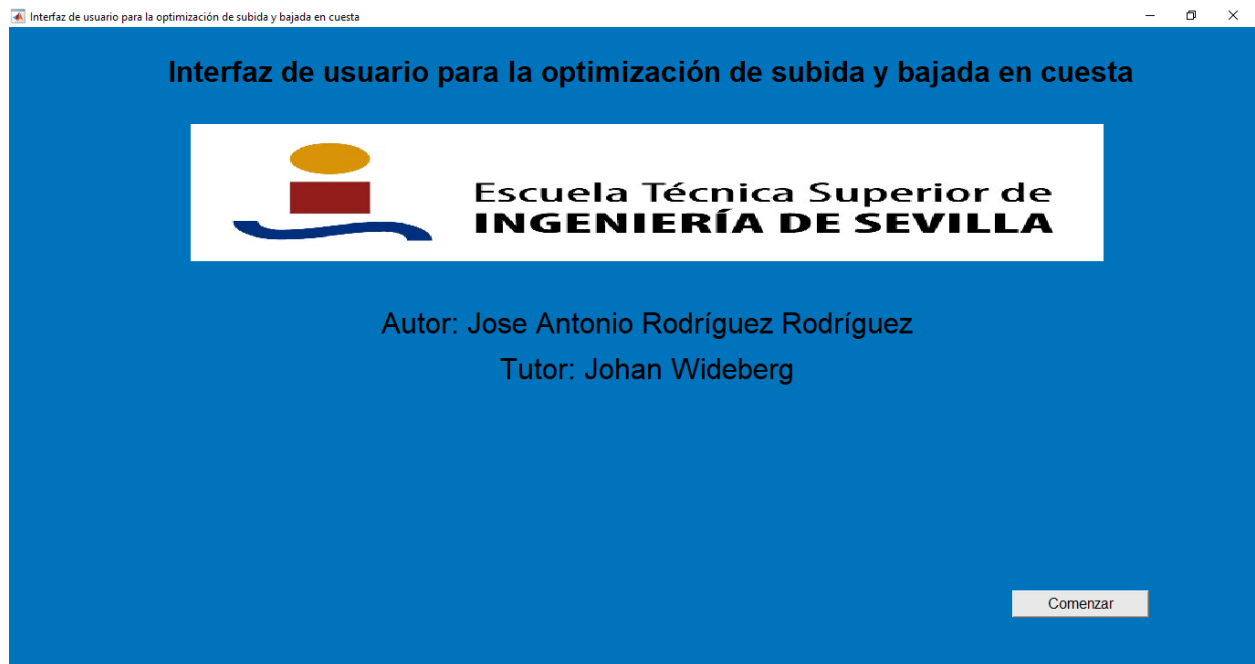


Figura 6-1. Pantalla de inicio de la interfaz de usuario

Antes de pasar a explicar la siguiente ventana gráfica hay que recordar que antes de hacer esto debemos crear una estructura del objeto gráfico mediante `guihandles`, y posteriormente guardar esta información en `guidata`, tal y como se verá en las siguientes líneas de código:

```
global data_portada
data_portada=guihandles(hfig);
guidata(hfig,data_portada)
```

Resaltar que se ha definido a `data_portada` como una variable global ya que esa información va a servir en una ventana posterior a la que se va a abrir cuando el usuario pulse sobre el botón COMENZAR (ver figura no se cuanto).

6.2 Menú

Esta ventana gráfica se abre una vez el usuario pulse sobre el pushbutton definido en la pantalla de inicio, aquel en el que pone Comenzar. Al realizar dicha acción se abrirá una nueva ventana gráfica que es similar al menú que puede aparecer en cualquier aplicación en la que el usuario podrá elegir cuál quiere que sea la incógnita del problema.

Se podrá elegir entre la velocidad (si se elige esta opción la interfaz calculará cuál es la velocidad máxima a la que el automóvil puede ascender una determinada pendiente), el consumo de combustible (si se elige esta opción la interfaz calculará cual es la marcha óptima que minimiza el consumo de combustible y cuál es el valor de dicho consumo) y la pendiente (si se elige esta opción la interfaz calculará qué pendiente máxima puede ascender o descender el automóvil).

Para llevar a cabo el diseño de esta ventana gráfica tan solo es necesario declarar tres pushbuttons diferentes, uno para cada incógnita que se puede elegir. Además de un cuadro de texto para que el usuario pueda ver lo que le está pidiendo ese menú que se acaba de desplegar. Por último, y no menos importante, habrá que volver a utilizar las órdenes `guihandles` y `guidata`. A continuación se puede ver como quedaría esta ventana gráfica.

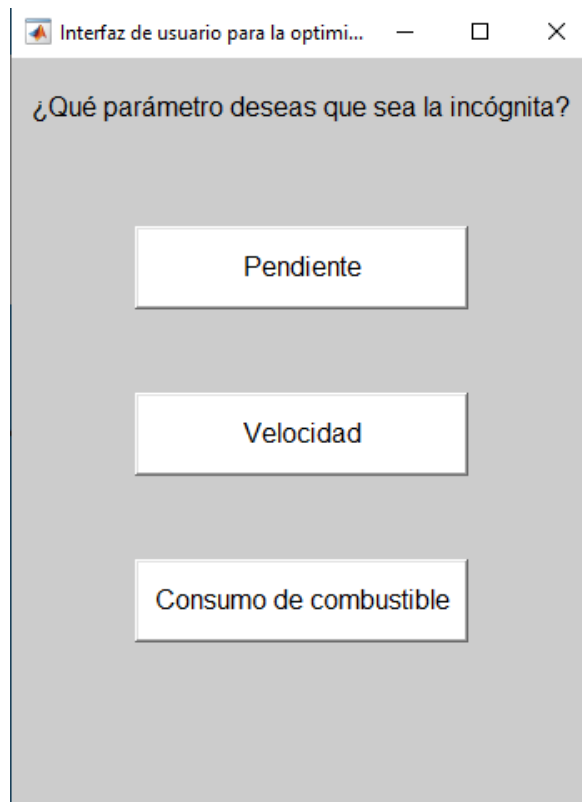


Figura 6-2. Menú de la interfaz de usuario

6.3 Opción pendiente

Esta ventana gráfica se despliega si el usuario pulsa sobre el pushbutton que corresponde a la pendiente en la ventana del menú de la interfaz gráfica. A continuación se detallarán los aspectos más importantes de esta nueva sub-función, a la cual se le llamó mediante la propiedad de función de llamada presente en el pushbutton definido para ello. Resaltar que lo primero que hace esta sub-función es cerrar tanto la pantalla de inicio como la de menú, esto lo hace dándole el valor de apagado a la propiedad `Visible` de ambas pantallas.

Esta ventana gráfica es en la que menos datos se le solicitan al usuario de todas las ventanas gráficas posibles que se podrían abrir mediante la pantalla de menú, tal y como se verá más adelante. Para llevar a cabo esta ventana gráfica hay que basarse en los conceptos teóricos vistos en el subcapítulo de la pendiente máxima superable, el cual se encontraba dentro del capítulo del modelo dinámico. En esa misma sub-sección se veía

cómo la pendiente máxima que un vehículo puede superar dependía exclusivamente de la tracción que tuviera el vehículo en cuestión y de algunos de los parámetros del mismo. Se recuerda que la tracción existente en el automóvil podía ser de tres tipos distintos: trasera, delantera o tracción total (la tracción en este caso estaba presente tanto en el eje delantero como en el trasero).

Para el desarrollo de esta pantalla han sido necesarios varios objetos de tipo `UIControls`, englobados, como ya viene siendo habitual en el diseño de la interfaz, en un objeto `figure`. Como pasará en las ventanas velocidad y consumo de combustible, en esta ventana también estarán presentes los objetos tipo `static text` así como los `edit text`. Aunque esta vez, como ya se ha especificado en párrafos anteriores serán menos, pues para calcular la máxima pendiente superable no son necesarios tantos datos como los que hacían falta para calcular la máxima velocidad, así como el consumo de combustible del automóvil. Los datos necesarios en esta ocasión serán los siguientes: por una parte están algunos parámetros del vehículo como la altura, la distancia entre los ejes delantero y trasero (a la que a partir de ahora se le llamará longitud) y la relación de carga y por otra parte están otros parámetros como la adherencia máxima y la resistencia a la rodadura.

Para cada uno de los parámetros anteriores hará falta incluir un `UIControl` de tipo `static text`, además de su correspondiente objeto de texto editable para que el usuario pueda introducir los datos necesarios. También se incluye en dicha ventana un `UIControl`, en este caso, de tipo `pop-up menú` en que se le pedirá al usuario que elija el tipo de tracción que tiene su automóvil y hacia dónde va (subida o bajada), por lo que dispondrá de un menú desplegable con hasta seis opciones (ya que hay tres tipos de tracción diferentes, tal y como se ha comentado con anterioridad, para cada caso de subida o bajada de pendiente).

Una vez escrito todo el código referente a la programación del diseño de la ventana gráfica, esta quedará tal y como aparece a continuación.

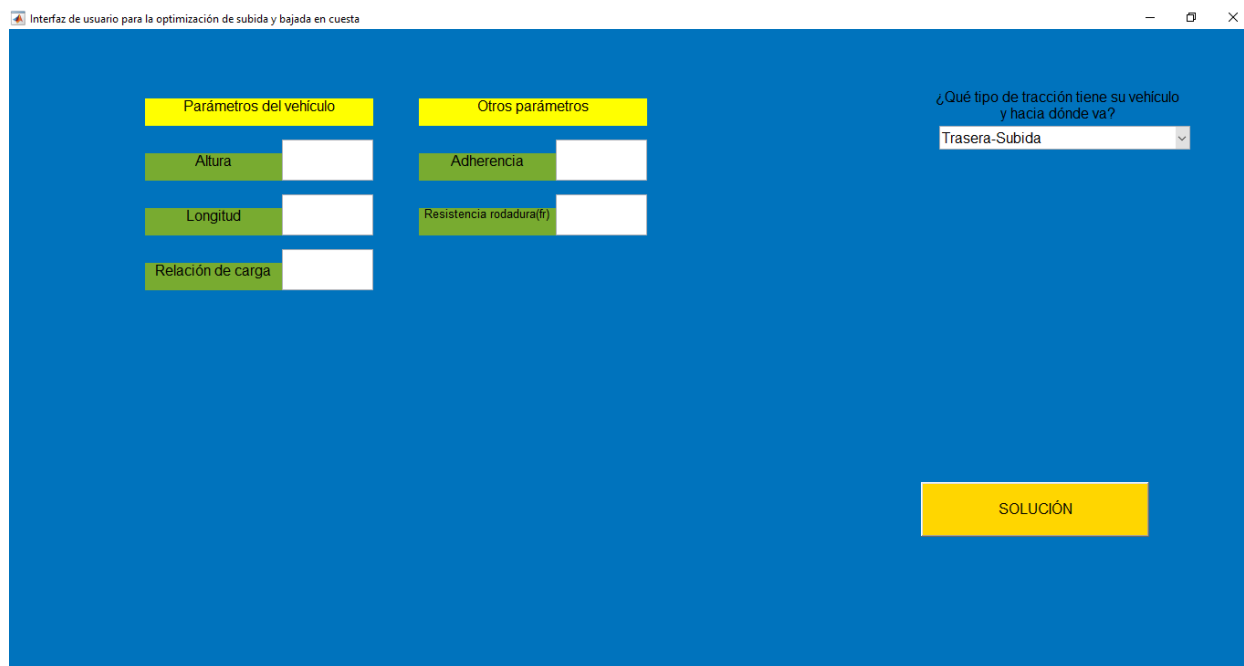


Figura 6-3. Ventana pendiente

Se observa como también se ha incluido el ya habitual pushbutton de `SOLUCIÓN`, objeto el cual al pulsar sobre él con el ratón mostrará la pendiente máxima que el automóvil, definido por el usuario, puede alcanzar.

6.4 Opción velocidad

En este sub-capítulo se explica de qué objetos gráficos está compuesta la ventana gráfica que se abre cuando el usuario pulsa sobre el pushbutton correspondiente a la velocidad, en la pantalla de menú. El objetivo primordial de esta pantalla es el de calcular la velocidad máxima que puede alcanzar el vehículo en una pendiente ascendente o descendente. Al igual que ocurría en la ventana pendiente y ocurrirá en la ventana de consumo de combustible, lo primero que hace esta sub-función es cerrar tanto la pantalla de inicio como la

de menú, dándole el valor de apagado a la propiedad Visible de ambas pantallas.

Esta vez para saber qué datos hay que pedirle al usuario, hay que tener en cuenta los conceptos teóricos desarrollados en el capítulo relativo al sistema de transmisión del automóvil, así como los conceptos dinámicos ya desarrollados en el capítulo dedicado a la dinámica del automóvil.

Teniendo en cuenta esto, serán necesarios varios parámetros del automóvil como la altura, la distancia entre los ejes delantero y trasero, la masa o la relación de carga del mismo, además de varios parámetros aerodinámicos como la densidad del aire, el coeficiente de resistencia aerodinámica, o el área frontal del automóvil. También harán falta otros tipos de parámetros como la resistencia por rodadura.

Además de todos los parámetros citados anteriormente, al usuario también se le deberá de pedir los datos relativos a la caja de cambios del automóvil, así como del motor del mismo. Por último se le debe solicitar al usuario que indique de qué tipo de tracción está dotado el vehículo y hacia dónde se dirige el vehículo (si está ascendiendo la pendiente o descendiendo esta) y dar el valor de la pendiente a la que se enfrenta.

De entrada parece que hay que pedirle una gran cantidad de datos al usuario y que puede haber la posibilidad de que no haya espacio en la pantalla para solicitar tal número de datos. Sin embargo el diseñador de la interfaz ha diseñado la misma para pedir los datos en tres bloques. Por tanto habrá un primer bloque en el que se le solicita al usuario que ingrese los valores de los parámetros referentes al vehículo y a la aerodinámica, expuestos anteriormente. Este bloque estará formado por objetos de tipo UIControl, más concretamente por static texts y edit texts, los cuales indicarán el nombre del parámetro y la caja donde el usuario deberá ingresar el valor del mismo, respectivamente. También se incluirá en este bloque, la petición al usuario del ingreso del valor que tiene la pendiente.

En otro bloque la pantalla de la opción velocidad, pedirá al usuario que ingrese el tipo de tracción que tiene su vehículo y hacia dónde se dirige este. Esta solicitud la hará preguntando lo anteriormente expuesto mediante un cuadro de texto, y las opciones de la cuestión las expondrá en un menú desplegable mediante un objeto pop-up menú. El usuario indicará la opción que desee elegir pulsando con el ratón sobre una de las opciones disponibles.

El tercer y último bloque de entrada de datos por pantalla se realizará a través de un pushbutton, en el cual se le pondrá al usuario que pulse para introducir los datos necesarios para calcular la fuerza de tracción. Estos datos, como ya se vio en el capítulo relativo al sistema de transmisión en el automóvil, son los siguientes: el par máximo transmitido por el motor, las velocidades de giro máximas y mínimas del motor, el número de marchas presentes en la caja de cambios del automóvil, el valor de las mismas, y por último el radio de las ruedas del automóvil. La solicitud de todos estos datos al usuario se realizará mediante la orden inputdlg vista previamente. Una vez el usuario haya introducido todo lo que se le pide en la ventana que se despliega al usar la orden inputdlg y pulse el botón ok de dicha ventana, se le mostrará la gráfica que relaciona la fuerza de tracción o la fuerza resistente debida al freno motor (esto depende de si la pendiente es ascendente o descendente) con la velocidad del vehículo según la marcha en la que este se encuentre. Así mismo, se abrirá un cuadro de texto, mediante la orden msgbox, indicando qué muestra la gráfica y animando al usuario a qué pulse sobre el botón solución (que se verá a continuación) para ver cuál es la velocidad máxima que puede alcanzar su automóvil en la pendiente dada.

Por último se ha incluido un objeto UIControl de tipo pushbutton de solución, el cual al pulsarlo mostrará un cuadro de texto con la máxima velocidad que puede alcanzar el automóvil que el usuario haya diseñado en una determinada pendiente dada también por el usuario.

Cuando ya se ha programado el código de MATLAB con el cual se definen todos los objetos gráficos descritos en los párrafos anteriores, la pantalla de la opción velocidad se muestra con el siguiente aspecto.

Figura 6-4. Ventana velocidad

6.5 Opción consumo de combustible

En este sub-capítulo se explica de qué objetos gráficos está compuesta la ventana gráfica que se abre cuando el usuario pulsa sobre el pushbutton correspondiente al consumo de combustible, en la pantalla de menú. El objetivo principal de esta pantalla es la de ofrecer al usuario la posibilidad de calcular el consumo óptimo de combustible y en qué marcha se alcanza para un determinado vehículo que circula a una cierta velocidad constante por una determinada pendiente. Cabe decir que lo primero que hace esta sub-función es cerrar tanto la pantalla de inicio como la de menú, esto lo hace dándole el valor de apagado a la propiedad Visible de ambas pantallas.

Para saber qué datos hay que pedirle al usuario, hay que tener en cuenta los conceptos teóricos desarrollados en el capítulo relativo al sistema de transmisión del automóvil, los conceptos dinámicos ya desarrollados en el capítulo dedicado a la dinámica del automóvil, así como aquellos conceptos estudiados en el capítulo que abordaba el problema del consumo de combustible en automóviles.

Teniendo en cuenta esto, serán necesarios varios parámetros del automóvil como la altura, la distancia entre los ejes delantero y trasero, la masa o la relación de carga del mismo, además de varios parámetros aerodinámicos como la densidad del aire, el coeficiente de resistencia aerodinámica, o el área del automóvil. También harán falta otros tipos de parámetros como la resistencia por rodadura además se debe pedir la densidad del combustible del vehículo.

Así mismo, al usuario también se le deberá de pedir los datos relativos a la caja de cambios del automóvil, así como del motor del mismo, sin olvidarse que en esta pantalla el usuario deberá indicar la velocidad (constante) a la que quiere calcular el consumo de combustible de su automóvil en una determinada pendiente. Por último se le debe solicitar al usuario que indique de qué tipo de tracción está dotado el vehículo y hacia dónde se dirige el vehículo (si está ascendiendo la pendiente o descendiendo esta) y dar el valor de la pendiente a la que se enfrenta.

Como ya sucedía en la pantalla de la opción velocidad, la solicitud de datos al usuario se dividirá en tres bloques. De dicha forma habrá un primer bloque en el que se le solicita al usuario que ingrese los valores de los parámetros referentes al vehículo y a la aerodinámica, así como otro tipo de parámetros expuestos anteriormente. Este bloque estará formado por objetos de tipo UIControl, más concretamente por static texts y edit texts, los cuales indicarán el nombre del parámetro y la caja donde el usuario deberá ingresar el valor del mismo, respectivamente. También se incluirá en este bloque, la petición al usuario del ingreso del valor que tiene la pendiente.

En otro bloque la pantalla de la opción consumo de combustible, pedirá al usuario que ingrese el tipo de tracción que tiene su vehículo y hacia dónde se dirige este. Esta solicitud la hará preguntando lo anteriormente expuesto mediante un cuadro de texto, y las opciones de la cuestión se expondrá en un menú desplegable mediante un objeto pop-up menú. El usuario indicará la opción que desee elegir pulsando con el ratón sobre una de las opciones disponibles.

El tercer y último bloque de entrada de datos por pantalla se realizará a través de un pushbutton, en el cual se le pedirá al usuario que pulse para introducir los datos relativos a la velocidad y al sistema de transmisión (se sobreentiende que también se le pedirá al usuario que introduzca datos referentes al motor del automóvil). Estos datos serán los siguientes: el par máximo transmitido por el motor, las velocidades de giro máximas y mínimas del motor, el número de marchas presentes en la caja de cambios del automóvil, el valor de las mismas, la relación del diferencial, el rendimiento de la transmisión, la velocidad a la que circula el vehículo y por último el radio de las ruedas del automóvil. La solicitud de todos estos datos al usuario se realizará mediante la orden `inputdlg` vista previamente. Una vez el usuario haya introducido todo lo que se le pide en la ventana que se despliega al usar la orden `inputdlg` y pulse el botón ok de dicha ventana, se le mostrará una tabla en la que aparecerán los valores del par motor o par resistente (según cuál sea el sentido de la pendiente) y la velocidad angular para cada marcha de la que disponga la caja de cambios del vehículo. Así mismo, se abrirá una nueva ventana, indicando qué representa la tabla y animando al usuario a que pulse sobre el botón continuar que está presente en esa misma ventana para que el usuario pulse en el mapa de consumo de combustible (que se desplegará después de que el usuario pulse el botón continuar), dependiendo del valor que tenga en cada marcha el par y la velocidad angular del motor. Es decir, el usuario deberá pulsar sobre el mapa de consumo tantas veces como marchas tenga la caja de cambios del automóvil (acto seguido deberá pulsar la tecla enter tal y como se verá más adelante).

Por último se ha incluido un objeto UIControl de tipo pushbutton de solución, el cual al pulsarlo mostrará un cuadro de texto con cuál sería el consumo de combustible óptimo para el automóvil que haya definido según la velocidad a la que circule por la pendiente introducida. Además mostrará cuál es la marcha adecuada para que se de este consumo óptimo.

Cuando ya se ha programado el código de MATLAB con el cual se definen todos los objetos gráficos descritos en los párrafos anteriores, la pantalla de la opción consumo de combustible se muestra con el siguiente aspecto.

Figura 6-5. Ventana consumo de combustible

6.6 Funciones auxiliares

En este subcapítulo se verán cada una de las funciones auxiliares de las que está compuesta la interfaz gráfica de usuario diseñada. La mayoría de ellas se corresponden con las funciones de llamada de los distintos pushbutton de los que consta dicha interfaz, mientras que otras son funciones auxiliares a las propias funciones de llamada, por lo que todas estas funciones auxiliares serán sub-funciones dentro del código de la interfaz. Estas se irán explicando en función de a qué ventana, de las tres que se pueden abrir a partir de la pantalla de menú, pertenezcan dichas funciones. Así mismo el último sub-apartado tratará las funciones auxiliares que sean comunes a más de una de estas ventanas gráficas. Señalar que los códigos de MATLAB correspondientes a estas funciones auxiliares se encuentran en el anexo B.

6.6.1 Ventana gráfica de pendiente

La ventana pendiente al ser más simple que las demás solo cuenta con una función auxiliar, además de otra función auxiliar que es común a las otras ventanas (a esta se le llama una vez el usuario elige qué tipo de tracción tiene su automóvil) y que citaremos más adelante.

6.6.1.1 `resolv_alpha.m`

Esta función recibe el nombre de `resolv_alpha` y es llamada una vez el usuario pulsa el botón SOLUCIÓN. Básicamente lo que hace es resolver la ecuación con la que se obtiene la pendiente máxima que puede ascender, o por la que puede descender, un automóvil. Por lo tanto lo primero que tiene que hacer es obtener los datos que se le han pedido al usuario a través de las cajas de texto editable que estaban presentes en la interfaz.

Para ello habrá que obtener los datos (en forma de variable estructura) de la ventana gráfica desde la cual se llama a la función. Esto se hace a través de `guidata` tal y como se explicó en el subcapítulo dedicado a los conceptos previos de programación. Dicha orden y la declaración de la sub-función se ven en las siguientes líneas:

```
function resolv_alpha(objeto,~)
data=guidata(objeto);
```

En la sentencia de la función `resolv_alpha` se ve cómo hay que pasarle a esta la variable `objeto` que es aquella que se le pasa una vez se llama a esta y que se corresponde con los datos de la ventana gráfica desde la que se le ha llamado. Además se observa que todos los objetos gráficos definidos en la ventana pendiente, desde la cual se ha llamado a la sub-función `resolv_alpha`, se han guardado como campos en una variable estructura a la que se le ha dado de nombre `data`. Ahora solo bastará con obtener los valores de los parámetros del automóvil definidos por el usuario en los objetos de tipo texto editable. Esto se consigue a través de la propiedad `String`, de la que se obtiene el valor mediante la sentencia `get` como ya se pudo ver anteriormente en este documento. Sin embargo, una vez se obtiene este valor está en formato cadena, por lo que habrá que convertirlo en un número para su uso correcto. Dicha acción se realiza con la orden `str2double`, que convierte las cadenas en números. Todo esto se expone en la siguiente línea de código que refleja lo que hay que escribir en MATLAB para conseguir cada uno de los valores ingresados por el usuario.

```
Nombre_variable=str2double(get(data.Tag,'String'));
```

Señalar que `Tag` hace referencia al nombre que se le ha dado al objeto gráfico determinado mediante la propiedad `tag` cuando se realizó su definición en la ventana pendiente. Una vez se haya obtenido el valor de cada uno de los parámetros exigidos al usuario, se deberá obtener el valor que el usuario ha elegido del menú desplegable del objeto pop-up menú. Esto se consigue a través de la propiedad `Value` del pop-up menú que tendrá un valor de 1 si el usuario elige la primera fila del menú desplegable, 2 si elige la segunda y así sucesivamente. Para obtener dicha propiedad se procede de la misma forma que se hace para obtener las propiedades de los objetos gráficos mediante la sentencia `get`.

Una vez se tenga este valor, se programará una bifurcación de tipo `switch` donde cada uno de sus casos (`case`) dependan del valor de la propiedad `Value` del pop-up menú obtenida previamente. En cada uno de estos casos

se obtendrá el ángulo de la pendiente máxima que el vehículo puede superar en función del tipo de tracción que este tenga y si la pendiente es ascendente o descendente, que habrá sido elegido anteriormente por el usuario. Todo lo que se ha explicado en el párrafo presente se refleja en las siguientes líneas de código:

```
switch traccion
    case 1
        alpha_rad=atan((fr*L-nu*landa*L)/(L-nu*h));
    case 2
        alpha_rad=atan((nu*landa*L+fr*L)/(nu*h+L));
    case 3
        alpha_rad=atan((nu*(1-landa)*L-fr*L)/(L+nu*h));
    case 4
        alpha_rad=atan(fr*L+(nu*(1-landa)*L)/(L-nu*h));
    case 5
        alpha_d_rad=atan((2*nu*(1-landa)*L-fr*L)/(L+2*nu*h));
        alpha_t_rad=atan((fr*L-2*nu*landa*L)/(2*nu*h-L));
        alpha_rad=min(alpha_d_rad,alpha_t_rad);
    case 6
        alpha_d_rad=atan(fr*L+(2*nu*(1-landa)*L)/(L-2*nu*h));
        alpha_t_rad=atan((2*nu*landa*L+fr*L)/(2*nu*h+L));
        alpha_rad=min(alpha_d_rad,alpha_t_rad);
end
```

Los casos 1 y 2 corresponden al caso de que un automóvil tenga tracción trasera y los cálculos que se realizan en ellos se corresponden con las ecuaciones 2-14 y 2-15 respectivamente. Los casos 3 y 4 tratan el caso en el que el vehículo está dotado de tracción delantera y sus ecuaciones correspondientes son las 2-19 y 2-20 respectivamente. Por último los casos 5 y 6 tienen que ver con el caso en el que el automóvil tenga tracción en los dos ejes, en esa ocasión sus ecuaciones son las ya vistas en el subcapítulo de pendiente máxima superable y que son la 2-23 y 2-24 por una parte (caso ascendente) y la 2-25 y 2-26 por otra (caso descendente).

Una vez se salga de la bifurcación anterior se deberá pasar el ángulo α obtenido de radianes a grados, además de pasarlo en forma de cadena para incluirlo en el mensaje que se mostrará por pantalla gracias a la función auxiliar `muestra_sol` que se verá más adelante. De esta forma cuando el usuario pulse sobre el botón solución, se abrirá una ventana gráfica que contendrá un mensaje con la pendiente máxima que puede superar el automóvil. Esta acción se lleva a cabo llamando a la función auxiliar citada (`muestra_sol`) desde la función en la que nos encontramos.

6.6.2 Ventana gráfica de velocidad

La ventana velocidad cuenta con dos funciones auxiliares, una de ellas necesaria para pedir al usuario los datos relativos al motor del automóvil y el sistema de transmisión, necesarios para poder representar la gráfica que relaciona la fuerza de tracción que ejerce el vehículo con la velocidad que este ofrece en función de las marchas, gráfica que se lleva a cabo a través de esta función también. La otra función auxiliar sirve para calcular la velocidad máxima que el automóvil, definido por el usuario, puede alcanzar en una pendiente determinada.

Además la ventana velocidad cuenta con otras funciones auxiliares que son comunes a otras ventanas, es por ello por lo que se expondrán en el sub-capítulo dedicado a las funciones auxiliares comunes a varias ventanas.

6.6.2.1 graf_vel_f.m

La función auxiliar `graf_vel_f` se encargará tanto de pedir al usuario los datos del motor y del sistema de transmisión del automóvil como de obtener la gráfica que relaciona la fuerza de tracción o resistente (en función de si la pendiente es ascendente o descendente) con la velocidad del vehículo, tal como se indicó anteriormente. Se puede pensar en un primer momento, que la función `graf_vel_f` es una función

independiente a los datos que haya introducido el usuario en la ventana velocidad, sin embargo esto no es así, ya que se necesita saber si el vehículo está subiendo o bajando la pendiente indicada. Es importante saber la respuesta a esta cuestión ya que como se indicó en el capítulo del sistema de transmisión del automóvil, si el vehículo se encontraba descendiendo la pendiente, el par resistente máximo que iba a ofrecer el freno motor iba a ser la mitad del par motor máximo que tiene el motor.

Lo primero que hace esta función auxiliar, la cual es llamada a través del pushbutton diseñado en la ventana velocidad para pedir los datos necesarios para calcular la fuerza y la velocidad tal y como se indicó en su diseño, es solicitar al usuario los siguientes datos: el par máximo transmitido por el motor, las velocidades de giro máximas y mínimas del motor, el número de marchas presentes en la caja de cambios del automóvil, el valor de las mismas, y por último el radio de las ruedas del automóvil. Esta acción se realiza a través de la orden `inputdlg`, la cual como se explicó anteriormente abre una ventana en la que se pide al usuario que introduzca los datos ya descritos, y que para los valores solicitados tendrá el siguiente aspecto:

The image shows a MATLAB dialog box with the following fields and labels:

- Introduzca el máximo par motor de su motor:
- Introduzca la mínima velocidad angular de giro de su motor(en rad/s):
- Introduzca la máxima velocidad angular de giro de su motor(en rad/s):
- Introduzca el número de marchas de su caja de cambios:
- Introduzca las relaciones de marchas de su caja de cambios(en forma de vector):
- Introduzca el radio de rueda de su vehículo:

Buttons: OK, Cancel

Figura 6-6. Ventana que pide al usuario los datos del motor y de la caja de cambios

Recordar que la orden `inputdlg` guarda los datos introducidos por el usuario en una variable tipo celda con el tamaño del número de datos pedidos al usuario, además de que estos datos se encuentran en forma de cadena. Por lo tanto, para obtener estos se deberá pasar de formato cadena a formato numérico (mediante la orden `str2double`, con la excepción del valor de las relaciones de transmisión que al ser un vector hay que realizarlo mediante la orden `str2num`), los datos contenidos en cada una de las celdas de la variable tipo celda. A continuación se muestra un ejemplo de cómo llevar a cabo esto.

```
w_min=str2double(f{2,1});
```

Denotar que `f` es el nombre que se le ha dado a la variable celdas que contiene todos los datos introducidos por el usuario a través de la orden `inputdlg`. Además de estos datos, se debe obtener el en qué sentido va el coche, lo cual se conoce a través de la propiedad `Value` del pop-up menú programado en la ventana gráfica de velocidad, que tendrá un valor de 1 si el usuario elige la primera fila del menú desplegable, 2 si elige la segunda y así sucesivamente. Si el valor de la fila se corresponde con aquella en la que se indica que el automóvil va en descenso (esto se preguntará con una bifurcación tipo `if`), el valor del par motor se verá reducido a su mitad.

Una vez obtenidos todos estos datos, se pasa a calcular la fuerza de tracción que puede llegar a ofrecer el vehículo en cada marcha, así como la velocidad máxima y mínima del mismo en cada una de las marchas. Esto se calcula mediante las relaciones que ya se vieron en los sub-capítulos dedicados a las relaciones realizar esta acción en MATLAB se programa un bucle `for` que guarde todos los valores en los vectores fuerza, velocidad máxima y velocidad mínima, tal y como se muestra a continuación.

```

global F
global v_max
F=zeros(1,n);
v_min=zeros(1,n);
v_max=zeros(1,n);
for ii=1:n
    F(ii)=M*marchas(ii)/R;
    v_min(ii)=w_min*R/marchas(ii);
    v_max(ii)=w_max*R/marchas(ii);
end

```

Se observa como los vectores de fuerza y velocidad máxima se definen como globales, ya que harán falta en otra función auxiliar más adelante. Así mismo se muestra como se inicializan los vectores necesarios para dibujar la gráfica, y que después del bucle for estos tendrán los valores asociados a cada una de las marchas de las que consta la caja de cambios del automóvil.

Con los vectores anteriormente calculados, ya se puede dibujar la gráfica que relaciona la fuerza de tracción con la velocidad del vehículo según la marcha. La gráfica se dibujará con otro bucle for haciendo uso también de la orden `hold on` con la cual se irán superponiendo las distintas gráficas que se vayan creando a cada paso del bucle. A cada marcha se le pondrá un color distinto para que quede lo más visiblemente claro qué marcha es cada una mediante una leyenda que se incluirá en la gráfica.

Para realizar esto último, el diseñador de la interfaz ha decidido crear un vector de números cuya primera componente es el código ASCII de la letra que en la orden `plot` se corresponde con el color azul, esta es la `b`. La segunda componente del vector es el salto que hay entre el código ASCII de la letra `b` y la letra `c` (es lógico que este salto vale 1), dicha letra `c` representa el color cian, y así sucesivamente con los distintos colores que se le puede dar a una gráfica con la orden `plot`, los cuales se detallan a continuación con sus correspondientes códigos ASCII según la letra del abecedario que haya que escribir para darle ese determinado color.

Letra	Color al que está asociada	Código ASCII
b	Azul	98
c	Cian	99
g	Verde	103
k	Negro	107
m	Magenta	109
r	Rojo	114
w	Blanco	119
y	Amarillo	121

Tabla 6-1. Código ASCII de cada letra que representa un color en la orden `plot`

Además de esto se ha creado una variable llamada `texto` que es de tipo celda, que servirá para almacenar el texto que se pondrá en la leyenda de la gráfica. Finalmente el código de esta parte de algoritmo queda de la siguiente forma:

```

color=[98 1 4 4 2 5 5 2];
c=0;
texto=cell(1,n);
for ii=1:n

```

```

x=[v_min(ii) v_min(ii) v_max(ii) v_max(ii)];
y=[0 F(ii) F(ii) 0];
c=c+color(ii);
texto(ii)=strcat({num2str(ii)},{ 'a marcha' });
plot(x,y,char(c));
hold on
end

```

Después de esto se pone la leyenda ya fuera del bucle con la orden `legend` a la que se le pasará la variable `texto`, además se pondrán los títulos a los ejes. Por último se mostrará un mensaje por pantalla que indique al usuario el contenido de la gráfica, así como la sugerencia de pulsar el botón solución si quiere ver cuál es la velocidad máxima que puede alcanzar el automóvil que ha definido en la pendiente que ha elegido.

6.6.2.2 calc_fuerza_resist.m

La función auxiliar `calc_fuerza_resist` tiene como objetivo mostrar al usuario cuál es la velocidad máxima admisible (entendiéndose por esta velocidad aquella para la cual el vehículo se mantiene estable sin necesidad de pisar el pedal del freno en el caso de pendientes descendentes), que puede adoptar el vehículo que él mismo ha definido mediante sus parámetros característicos. Esta función actúa una vez que el usuario pulsa sobre el botón SOLUCIÓN.

Lo primero que hay que obtener son los datos de la ventana gráfica desde la cual se llama a la función (dichos datos estarán guardados en forma de estructura). Esto se hace a través de `guidata` tal y como se explicó en el subcapítulo dedicado a los conceptos previos de programación. Dicha orden y la declaración de la sub-función se ven en las siguientes líneas:

```

function calc_fuerza_resist(objeto,~)
data=guidata(objeto);

```

En la sentencia de la función se ve cómo hay que pasarle a esta la variable `objeto` que es aquella que contiene los datos de los objetos gráficos presentes en la ventana velocidad (que es desde la cual se llama a esta sub-función). Además se observa que todos los objetos gráficos definidos en la pantalla velocidad, se han guardado como campos en una variable estructura a la que se le ha dado de nombre `data`. Ahora solo bastará con obtener los valores de los parámetros del automóvil definidos por el usuario en los objetos de tipo texto editable. Esto se consigue a través de la propiedad `String`, de la que se obtiene el valor mediante la sentencia `get` como ya se pudo ver anteriormente en este documento. Sin embargo, una vez se obtiene este valor está en formato cadena, por lo que habrá que convertirlo en un número para su uso correcto. Dicha acción se realiza con la orden `str2double`, que convierte las cadenas en números. Todo esto se expone en la siguiente línea de código que refleja lo que hay que escribir en MATLAB para conseguir cada uno de los valores ingresados por el usuario.

```

Nombre_variable=str2double(get(data.Tag,'String'));

```

Señalar que `Tag` hace referencia al nombre que se le ha dado al objeto gráfico determinado mediante la propiedad `tag` cuando se realizó su definición en la ventana velocidad. Por otra parte para saber si la pendiente es ascendente o descendente, tan solo habrá que obtener el valor de la propiedad `Value` del pop-up menú diseñado en la ventana velocidad, tal y como ya se especificó.

Después de conseguir todos los datos procedentes de la ventana gráfica desde la cual se llamó a esta función auxiliar, se declaran los vectores `F` y `v_max` (anteriormente calculados en la función `graf_vel_f`) como variables globales, ya que harán falta para el cálculo de la velocidad máxima, objetivo último de esta función.

Dependiendo del tipo de pendiente en la que se encuentre el automóvil, el algoritmo de búsqueda de la velocidad máxima será uno u otro. Para el caso en el que el automóvil esté subiendo la pendiente, el algoritmo será el siguiente:

```

n=length(v_max);
v=0:0.01:v_max(n);

```



```

requis=zeros(1,n);
if traccion==1 || traccion==3

F_resist=m*g*(sin(alpha*pi/180)+fr*cos(alpha*pi/180))+0.5*ro*A*cd*v.^2
;
    for ii=1:n
        requis(ii)=length(F_resist(F_resist<=F(ii)));
    end
    indice_vel=requis(requis~=0);
    if length(indice_vel>1)
        indice_vel=max(indice_vel);
    end
    posi=find(requis==indice_vel);
    sol=v(indice_vel);
    if sol>v_max(posi)
        sol=v_max(posi);
    elseif sol<v_max(posi-1)
        sol=v_max(posi-1)
    end
    sol_km=sol*3600/1000;
    cad=num2str(sol_km);

```

Se observa que antes de programar el algoritmo de búsqueda hay que declarar o inicializar una serie de variables que son necesarias para llevar a cabo el propio algoritmo. La primera variable tiene el nombre de n y se corresponde con la longitud del vector de velocidades máximas, después se declara la variable v que es un vector cuyos valores van de 0 a el último valor del vector de velocidades máximas, con un paso entre cada valor de dicho vector declarado de 0.01. Por último se inicializa un vector en el que se irán guardando las longitudes de los vectores de los valores que cumplen un requisito que a continuación se explicará, este vector que se inicializa recibe el nombre de `requis`.

A continuación se define el vector `F_resist`, el cual, como su nombre indica, almacenará los valores de la fuerza resistente (debida a las fuerzas que se producen el automóvil en el ascenso de una pendiente) en función de la velocidad, la cual está almacenada en el vector `v` anteriormente declarado.

El requisito que se debe cumplir es que para que el vehículo pueda seguir avanzando la fuerza de tracción proporcionada por el motor a las ruedas debe de ser mayor que la fuerza resistente que se produce en el ascenso de la pendiente. Esta condición se establece dentro del bucle `for` que se repetirá tantas veces como marchas tenga la caja de cambios del automóvil, tal y como se muestra en el trozo de código expuesto antes. Como habrá muchos valores en `F_resist` que cumplan esta condición en cada marcha, se va guardando la longitud del vector que los almacena en `requis`, en cada paso del bucle. Esta longitud se entiende que irá siendo menor conforme se vayan incrementando las marchas, pero puede haber casos en los que esto no sea así.

Aquellas posiciones de `requis` en las que su valor sea distinto de 0 corresponderán a las marchas denominadas límite (el valor de elementos dentro de ellas que cumplen esa condición se almacena en `indice_vel`), ya que en esas marchas se llega a un punto en el que la fuerzas que se oponen al movimiento sean mayores que la fuerza de tracción que pueda ejercer el vehículo. Como puede haber más de una marcha límite, mediante una bifurcación `if` se seleccionará la que sea menor, es decir aquella en las que haya menos valores que cumplan la condición vista en el párrafo anterior.

En `posi` se almacena la marcha en la que se alcanza esta velocidad máxima en la pendiente y en solución el valor máximo de esta velocidad que es el que interesa. Este valor se deberá pasar a Km/h tal y más tarde pasar a formato cadena, tal y como aparece en las líneas de código anteriores.

De manera análoga en el caso de que la pendiente fuese descendiente se procedería del mismo modo con la diferencia de que `F_resist` pasa a llamarse `F_descend` y su valor tendrá que ser el contrario a las fuerzas resistivas que venían reflejadas en la ecuación 2-8, ya que en este caso va a ir a favor del movimiento hasta una determinada velocidad y va a ser equilibrado con la fuerza resistente del freno motor. Por ello la condición este caso es que los valores de `F_descend` sean mayores o iguales que los valores de `F` obtenidos esta vez de la

fuerza resistente que ejerce el freno motor.

Una vez obtenida la velocidad máxima en pendiente descendente o ascendente y esta se encuentre en formato cadena se formará un mensaje con ella que se le pasará a la función auxiliar `muestra_sol`, para que una vez el usuario pulse sobre el botón solución aparezca el mensaje con la velocidad máxima en la pendiente definida por él mismo.

6.6.3 Ventana consumo de combustible

La ventana de consumo de combustible cuenta con hasta tres funciones auxiliares, una de ellas construirá una tabla en la que se mostrarán las parejas de par motor y velocidad de giro del motor para cada marcha de la caja de cambios del automóvil. Esto se realizará pidiéndole, en primer lugar, al usuario los datos relativos al motor del automóvil y el sistema de transmisión, así como la velocidad a la que circula el vehículo. Otra de las funciones auxiliares con las que cuenta esta ventana se empleará para construir el mapa de consumo a través del cual se calculará el consumo óptimo de combustible señalando sobre él las parejas de puntos presentes en la tabla en el mapa de consumo, tal y como ya se explicó en capítulos anteriores. La última función auxiliar presente en esta ventana sirve para calcular el consumo de combustible óptimo y la marcha en la cual se alcanza en función de los puntos anteriormente señalados.

Además la ventana de consumo de combustible cuenta con otras funciones auxiliares que son comunes a otras ventanas, es por ello por lo que se expondrán en el sub-capítulo dedicado a las funciones auxiliares comunes a varias ventanas.

6.6.3.1 `tabla_par_w.m`

La función auxiliar `tabla_par_w` se ocupará tanto de pedir al usuario los datos del sistema de transmisión del automóvil y la velocidad constante a la que circula el vehículo, como de obtener la tabla que contiene a las parejas de velocidad de giro y par motor (o resistente si se trata de una pendiente descendente) según la marcha en la que se encuentre el vehículo. Se puede pensar en un primer momento, que la función `graf_vel_f` es una función independiente a los datos que haya introducido el usuario en la ventana velocidad, sin embargo esto no es así, ya que se necesita saber si el vehículo está subiendo o bajando la pendiente indicada. Es importante saber la respuesta a esta cuestión ya que como se indicó en el capítulo del sistema de transmisión del automóvil, si el vehículo se encontraba descendiendo la pendiente, el par resistente máximo que iba a ofrecer el freno motor iba a ser la mitad del par motor máximo que tiene el motor.

Lo primero que hace esta función auxiliar, la cual es llamada a través del pushbutton diseñado en la ventana consumo de combustible para pedir los datos necesarios para calcular las parejas de par motor y velocidad de giro del motor tal y como se indicó en su diseño, es solicitar al usuario los siguientes datos: la velocidad a la que circula el automóvil, el número de marchas del que consta su caja de cambios, la relación de transmisión de estas marchas, la relación de transmisión del diferencial y la eficacia del sistema de transmisión. Esta acción se realiza a través de la orden `inputdlg`, la cual como se explicó anteriormente abre una ventana en la que se pide al usuario que introduzca los datos ya descritos, y que para los valores solicitados tendrá el siguiente aspecto:

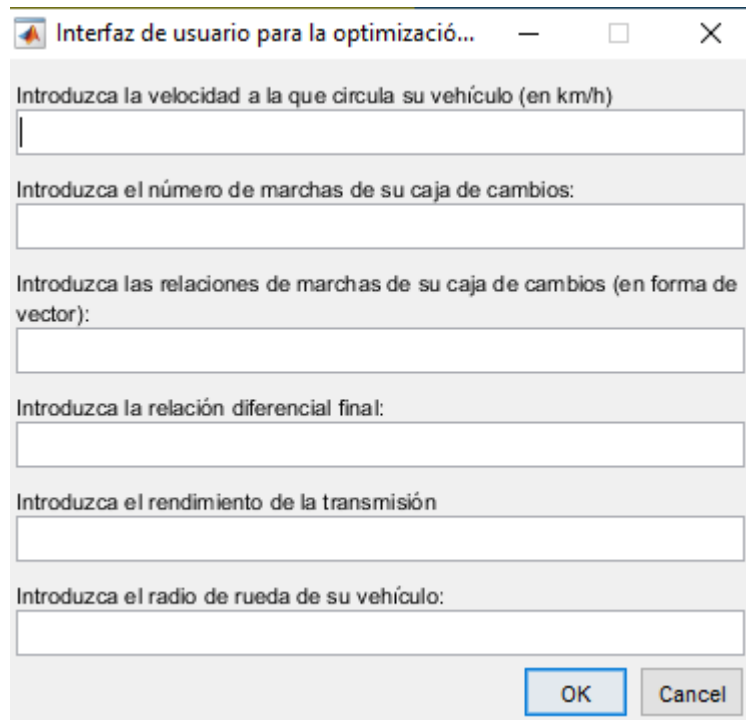


Figura 6-7. Ventana que pide al usuario los datos de la caja de cambios y a qué velocidad circula el vehículo

Además de estos datos, la función auxiliar `tabla_par_w` tiene que obtener los datos de la ventana gráfica desde la cual se llama a la función. Esto se hace a través de `guidata` tal y como se explicó en el subcapítulo dedicado a los conceptos previos de programación. Con todos estos datos ya obtenidos es hora de calcular las parejas de par y velocidad de giro, que se almacenarán en una tabla.

Esta acción se realiza calculando la fuerza de tracción mínima (en el caso de pendiente ascendente) o la fuerza resistente mínima (en el caso de pendiente descendente). Estas fuerzas son iguales a las fuerzas resistentes que se oponen al movimiento caracterizadas para la pendiente ascendente y la pendiente descendente respectivamente.

Una vez calculadas estas fuerzas, a través de la ecuación 3-6 y el radio de la rueda del vehículo se obtiene el par que posee la rueda para que con las ecuación 3-7 se calcule el par motor presente en el motor en cada una de las marchas.

Se procede de manera análoga para obtener las velocidades de giro del motor a través en primer lugar de la ecuación 3-8 y en segundo lugar de la ecuación 3-9. Todo lo explicado en los párrafos anteriores se ve reflejado en las siguientes líneas de código.

```

if traccion==1 || traccion==3 || traccion==5

F_calc=m*g*(sin(alpha*pi/180)+fr*cos(alpha*pi/180))+0.5*ro*A*cd*(v_con
s*1000/3600)^2;
else
    F_calc=m*g*(sin(alpha*pi/180)-fr*cos(alpha*pi/180))-
0.5*ro*A*cd*(v_cons*1000/3600)^2;
end
M=F_calc*R;
w_rueda=((v_cons/3.6)/R)*(60/(2*pi));
global tabla
tabla=zeros(n,2);
filas=cell(1,n);
format long
for ii=1:n
    tabla(ii,1)=M/(marchas(ii)*rr*nu_trans);

```

```

        tabla(ii,2)=w_rueda*marchas(ii)*rr;
        filas(ii)=strcat({num2str(ii)},{ 'a marcha' });
    end

```

Denotar que v_{cons} es la velocidad constante a la que circula el vehículo y M es el par que se produce en la rueda. La variable `tabla` más tarde se mostrará en otra ventana gráfica, junto a un `pushbutton` que habrá que pulsar para poder marcar las parejas de par motor y velocidad de giro obtenidas, en el mapa de consumo el cual se va a mostrar su diseño en la función auxiliar siguiente.

6.6.3.2 selecciona_consumo.m

Esta función en primer lugar construye el mapa de consumo de combustible y en segundo lugar deja al usuario pulsar sobre dicho mapa las parejas de par motor y velocidad de giro del motor que ha obtenido en la función auxiliar `tabla_par_w`. De hecho a esta función se le llama una vez el usuario pulse sobre el botón continuar presente en la función auxiliar `tabla_par_w`.

Esta función tiene en cuenta la tabla de flujo másico de combustible (4.1) para llevar a cabo el consumo específico de combustible (BSFC) que formará parte del mapa de consumo. Los datos relativos al flujo másico se cargarán desde un fichero llamado `datos_flujo_masico.dat`. De esta forma se procede a construir el mapa de consumo con las siguientes líneas de código.

```

function selecciona_consumo
w_motor=(500:500:6000);
par_motor=[15.6 31.2 46.8 62.4 78 93.6 109.2 124.8 140.4 156 171.6];
flujo_mas=load('datos_flujo_masico.dat');
w_rad_s=w_motor*pi/30;
Pot_motor=w_rad_s'*par_motor./1000;
BSFC=(flujo_mas./Pot_motor)*3600;

[c,h]=contour(w_motor,par_motor',BSFC',20);
clabel(c,h)
grid on
xlabel('Engine speed [rpm]')
ylabel('Engine torque [Nm]')
title('Mapa del consumo de combustible del vehículo')
valores=clabel(c,h,'manual');
for ii=1:length(valores)
    cad{ii}=valores(ii).String;
end
consumo=str2double(cad);
end

```

Con la orden `contour` se dibuja el mapa de consumo, así mismo con `clabel` se indica el valor de las líneas que aparecen en el mapa que se corresponden con el consumo específico de combustible. Más tarde con la propiedad `manual` de `clabel`, se le permite al usuario señalar puntos en el mapa de consumo. Estos puntos se irán guardando en formato cadena en la variable `cad`, para más tarde convertirlos a formato numérico en la variable `consumo`.

6.6.3.3 solucion.m

Esta función auxiliar obtiene el dato de la densidad de combustible de la ventana consumo de combustible, los datos de la tabla de par motor y velocidad de giro obtenido en la función `tabla_par_w`, así como los valores de consumo obtenidos en la función auxiliar `selecciona_consumo`.

Juntando todo esto va calculando el consumo de cada marcha (este consumo irá expresado ya en L/100 Km y se guardará en la variable `consumo_min`) y más tarde obtiene el mínimo, que será el consumo óptimo. Esto se

ve en las siguientes líneas de código:

```
function solucion(objeto,~)
data=guidata(objeto);
ro_comb=str2double(get(data.ro_comb,'String'));
global consumo
global tabla
global v_cons
consumo_min=consumo.*(tabla(:,1))'.*(tabla(:,2))'.*((2*pi/60)*0.001*(
1/ro_comb)*v_cons*0.01);
consumo_real=min(consumo_min);
marcha_opt=find(consumo_min==consumo_real);
if length(marcha_opt)>1
    marcha_opt=marcha_opt(1);
end
cad1=num2str(marcha_opt);
cad2=num2str(consumo_real);
```

6.6.4 Funciones auxiliares comunes

En este subcapítulo se mostrarán las dos funciones auxiliares que son comunes a las ventanas de pendiente, velocidad y consumo de combustible. Estas funciones auxiliares tienen que ver con la manera en la que se muestra la solución en cada una de estas pantallas, y son funciones realmente simples

6.6.4.1 muestra_sol.m

Esta función auxiliar se encarga de mostrar la solución por pantalla, acompañada del correspondiente mensaje que se le quiere dar al usuario. En lugar de esta función se podría haber utilizado la orden `msgbox`, pero el programador decidió diseñar esta función auxiliar con el objetivo de que la solución quedara lo más clara posible, es por ello que el color de fondo que le ha dado a esta ventana ha sido el amarillo.

Esta función puede ser llamada por las siguientes funciones auxiliares anteriormente vistas: `resolv_alpha.m`, `calc_fuerza_resist.m` y `solucion.m`.

Cuando se les llame desde dichas funciones también se le deberá pasar el mensaje que se desea que aparezca por pantalla. Además de dicho mensaje la función incorporará un botón de OK para hacer desaparecer la ventana creada con dicha función auxiliar.

6.6.4.2 salir.m

Con esta función auxiliar se consigue hacer invisible a la ventana creada con la función `muestra_sol`. Por lo tanto es llamada al pulsar sobre el botón OK presente en dicha ventana. Es por ello que se deberán pasar los datos de esa ventana mediante `guihandles` y `guidata` para que esta función pueda cambiar la propiedad Visible a off.

7 EJEMPLO DE LA INTERFAZ GRÁFICA

En este capítulo se va a llevar a cabo un ejemplo del funcionamiento de la interfaz gráfica de usuario para optimizar la subida y bajada, de la cual se mostró su diseño en el capítulo anterior. Para ello lo primero que se hace es obtener los datos necesarios de la ficha técnica de un automóvil, lo cual se ha llevado a cabo a través de la página web de coches.net. Se ha elegido esta página porque es una de las pocas páginas que dispone de datos relativos a la caja de cambios del automóvil, más concretamente las relaciones de transmisión.

Sin embargo, esta página no ofrece los valores máximos y mínimos de la velocidad de giro del motor, pero como sí que ofrece la velocidad máxima a la que puede llegar a circular el vehículo, por tanto estas velocidades de giro se calcularán mediante la ecuación 3-9 ya que se conoce la relación de transmisión de la última marcha, así como el radio de las ruedas del automóvil. Con esa relación (suponiendo que no hay pérdidas en el sistema de transmisión y que la relación del diferencial es de valor unidad) se calculará la velocidad de giro máxima y la mínima se obtendrá en función de la máxima y tendrá un valor de un poco menos de la mitad del valor de la velocidad de giro máxima del motor.

El automóvil elegido para este ejemplo es el Mini Cooper 5 puertas 136 CV 2020 que se muestra en la siguiente imagen.



Figura 7-1. Mini Cooper 5 puertas 136 CV 2020

Fuente: coches.net

En la siguiente tabla aparecen los datos necesarios para que sea posible calcular la pendiente máxima superable, la velocidad máxima y el consumo óptimo de combustible, a través de las distintas opciones con las que cuenta la ventana menú de la interfaz. En ella además el valor de cada uno de los parámetros va acompañado de la unidad de medida correspondiente.

Masa	1235 Kg	Par máximo	230 N·m
Distancia entre los ejes	2.567 m	Máxima velocidad	207 Km/h
Relación de carga	0.5	Velocidad de giro mínima del motor	80 rad/s
Altura	0.7125	Velocidad de giro máxima del motor	210 rad/s

Radio de las ruedas	0.1905 m	Relaciones de transmisión	1ª marcha	3.615
Área frontal	2.2 m		2ª marcha	1.952
Tipo de tracción	Delantera		3ª marcha	1.241
Tipo de combustible	Gasolina		4ª marcha	0.969
Densidad del combustible	680 Kg/m ³		5ª marcha	0.806
			6ª marcha	0.683

Tabla 7-1. Especificaciones técnicas Mini Cooper 5 puertas 136 CV 2020

Señalar que algunos datos como la altura, la distancia entre los ejes, el radio de las ruedas o el área frontal no aparecen de forma explícita en las fichas técnicas de coches.net. Por ejemplo la altura que el usuario debe introducir es la altura del centro de masas del automóvil, esta se ha supuesto que está en el punto medio de la altura real del coche (que es la que aparece en la ficha técnica), por ello la altura de la ficha técnica habrá que dividirla entre dos. La distancia entre los ejes aparece en la ficha técnica como “batalla” en el apartado Dimensiones. En cuanto a las ruedas aparece su diámetro, y este aparece en pulgadas, por lo tanto habrá que pasar este dato a m y dividirlo entre dos para obtener el radio de la rueda. Por último para obtener el área frontal del vehículo se ha multiplicado el ancho de este por la altura de este menos la altura libre que tiene el automóvil sobre el suelo.

Por otra parte parámetros como la relación de carga, el coeficiente aerodinámico, la eficiencia de la transmisión o la relación de transmisión en el diferencial se van a suponer ya que dichos datos no están reflejados en la ficha técnica que aparece en coches.net. Estos valores serán por tanto: 0.5 para la relación de carga, 0.35 para el coeficiente aerodinámico, 0.9 para la eficiencia de la transmisión y 1.9 para la relación de transmisión del diferencial.

Se recuerda que para inicializar la interfaz de usuario para optimizar la subida y bajada únicamente habría que escribir en la ventana de comandos `optim_subida_bajada` que es el nombre que recibe la función que da lugar a la interfaz gráfica. Una vez dicho esto, a continuación se verá un ejemplo de funcionamiento con cada una de las opciones de las que consta el menú de la interfaz (figura 6-2).

7.1 Cálculo de la pendiente máxima superable

Para calcular la pendiente máxima que el automóvil puede superar tan solo harían falta los datos relativos a la altura del automóvil, la distancia entre los ejes delantero y trasero, la relación de cargas, la adherencia máxima y el coeficiente de resistencia por rodadura. Además habrá que saber el tipo de tracción con el que está dotado el automóvil y hacia dónde se dirige, ya que se va a calcular la pendiente máxima tanto en pendiente ascendente como descendente.

Mirando la tabla 7-1 de especificaciones técnicas del automóvil en cuestión se observa que no aparece ni el valor de adherencia máxima ni el valor del coeficiente de resistencia por rodadura. Esto es debido a que estos dependen de las condiciones del terreno por el que circule el vehículo, como se vio en capítulos anteriores. Es por ello que estos datos se cogerán de las tablas 2-3 y 2-1, respectivamente, dependiendo del tipo del terreno en el que se encuentre el automóvil. Además se compararán los resultados de pendiente máxima superable obtenidos en diferentes superficies para su posterior análisis en el capítulo siguiente.

Teniendo todo esto claro se procede a realizar el ejemplo del cálculo de pendiente máxima superable para el caso del Mini Cooper. Para ello se pulsa sobre el botón pendiente en la ventana de menú de la interfaz y una vez hecho se abrirá la ventana pendiente y el usuario deberá completar los distintos campos, que son necesarios para calcular la pendiente máxima. Datos como la altura o la longitud se pueden obtener directamente de las especificaciones técnicas del Mini Cooper (Tabla 7-1), mientras que otras, como la

relación de carga se va a suponer que tiene un valor de 0.5 tal y como se indicó antes. Además tanto la adherencia como el coeficiente de resistencia por rodadura van a depender del terreno y el valor de estos parámetros se va a ver reflejado en la tabla de resultados que se va a obtener más tarde. En esta primera toma de resultados, se ha supuesto que los neumáticos del automóvil son de baja resistencia y que este circula sobre asfalto u hormigón seco. Sabiendo ya todo esto se introducen los datos necesarios y se elige el tipo de tracción que tiene el vehículo (también viene reflejado en las especificaciones técnicas) y si está en pendiente ascendente o descendente. Por tanto la ventana pendiente lucirá con el siguiente aspecto:

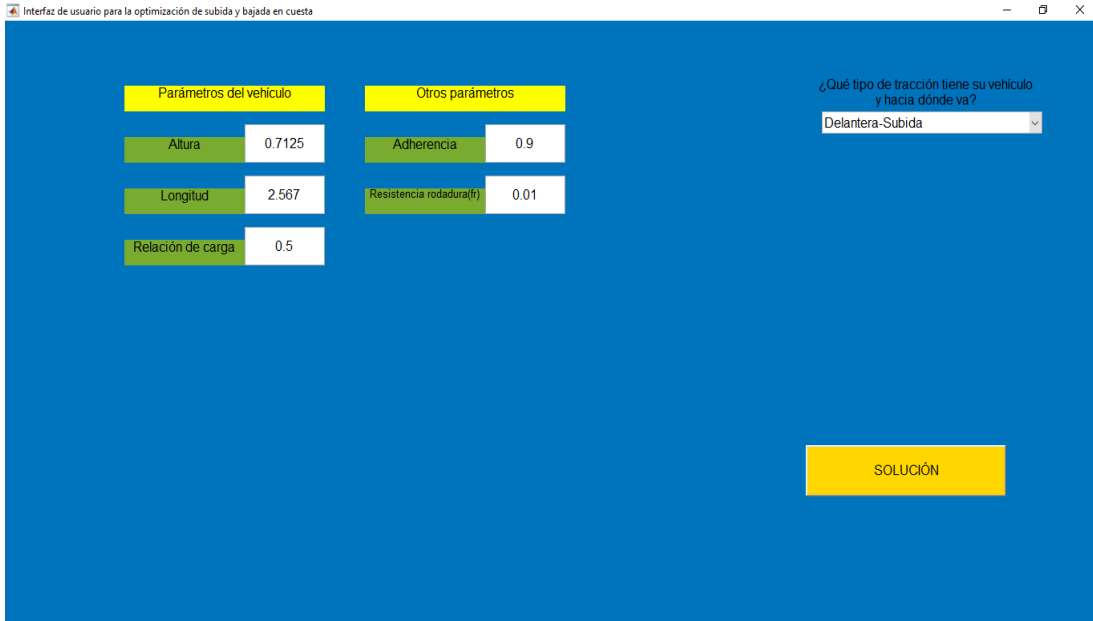


Figura 7-2. Ejemplo de introducción de datos en la ventana pendiente

Una vez hecho esto, se pulsa sobre el botón solución, y para el caso de que la pendiente sea ascendente aparecerá la siguiente ventana de texto con el valor de la pendiente máxima superable.

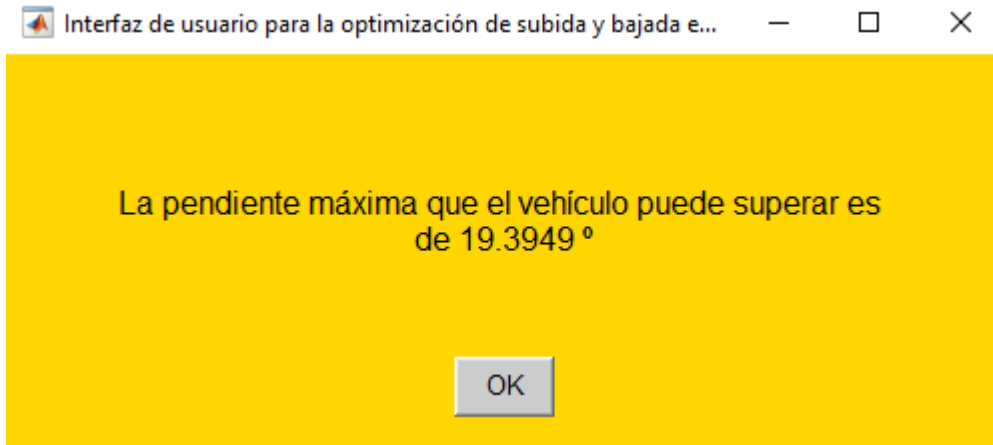


Figura 7-3. Solución de la pendiente máxima superable para el caso de pendiente ascendente

Análogamente, una vez se pulse sobre el botón OK, en el pop-up menú de la ventana pendiente se cambia el sentido de circulación del coche seleccionando la opción “Delantera-Bajada” y aparece la ventana de texto con la siguiente solución para el caso de que la pendiente sea descendente.

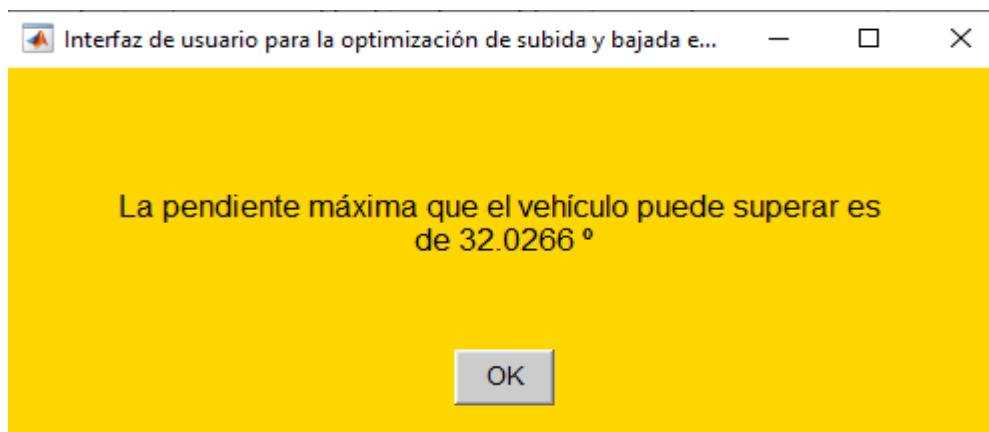


Figura 7-4. Solución de la pendiente máxima superable en el caso de pendiente descendente

Cuando ya se ha realizado el primer ejemplo para calcular la pendiente máxima superable, se va cambiando el valor de la adherencia y del coeficiente de resistencia de rodadura dependiendo de las condiciones del terreno. Con ello, se obtiene la siguiente tabla de resultados.

Tipo de asfalto y de neumático	Adherencia máxima	Coefficiente de resistencia por rodadura	Pendiente máxima superable (ascendente)	Pendiente máxima superable (descendente)
Neumáticos de baja resistencia sobre carretera lisa	0.9	0.01	19.3949°	32.0266°
Neumáticos ordinarios sobre asfalto seco	0.9	0.03	18.575°	34.0923°
Neumáticos ordinarios sobre asfalto húmedo	0.7	0.03	14.9996°	27.0857°
Neumáticos ordinarios sobre hormigón	0.9	0.02	18.986°	33.0715°
Neumáticos ordinarios sobre grava	0.6	0.055	11.8611°	26.6168°

Tabla 7-2. Tabla de resultados del cálculo de la pendiente máxima superable para el caso del Mini Cooper. Los resultados obtenidos en la tabla anterior serán analizados en el siguiente capítulo.

7.2 Cálculo de la velocidad máxima admisible

Para el cálculo de la velocidad máxima admisible son necesarios más datos que en el caso del cálculo de la pendiente máxima superable. Estos datos están agrupados en tres bloques, tal y como se vio en el diseño de la interfaz. Por una parte están los datos relativos al automóvil como son su masa, su altura, la distancia entre sus ejes delantero y trasero y su relación de carga. Por otra parte tenemos los parámetros aerodinámicos como son el coeficiente de resistencia aerodinámica (el cual va a tener un valor de 0.35 como ya se ha dicho), el área frontal del vehículo y la densidad del aire que se sabe que tiene un valor de 1.02 Kg/m^3 . Además existen otro tipo de parámetros como el coeficiente de resistencia por rodadura. En principio los datos obtenidos de

velocidad serán para la situación en la que el vehículo circule con neumáticos ordinarios sobre un asfalto seco (con este dato ya se sabe el valor que tendrá la adherencia y el coeficiente de resistencia por rodadura).

En otro bloque nos encontramos con los datos relativos a la caja de cambios y al motor del vehículo, así como el radio de las ruedas del vehículo. Estos el usuario deberá introducirlos pulsando el botón destinado para ello que se muestra claramente una vez este pulsa el botón correspondiente a velocidad en el menú de la interfaz.

El usuario también deberá indicar el tipo de tracción y la dirección del automóvil en el menú desplegable de la ventana velocidad.

A continuación se muestra como quedará la ventana velocidad una vez el usuario haya introducido una parte de los datos necesarios para el cálculo de la velocidad máxima admisible.

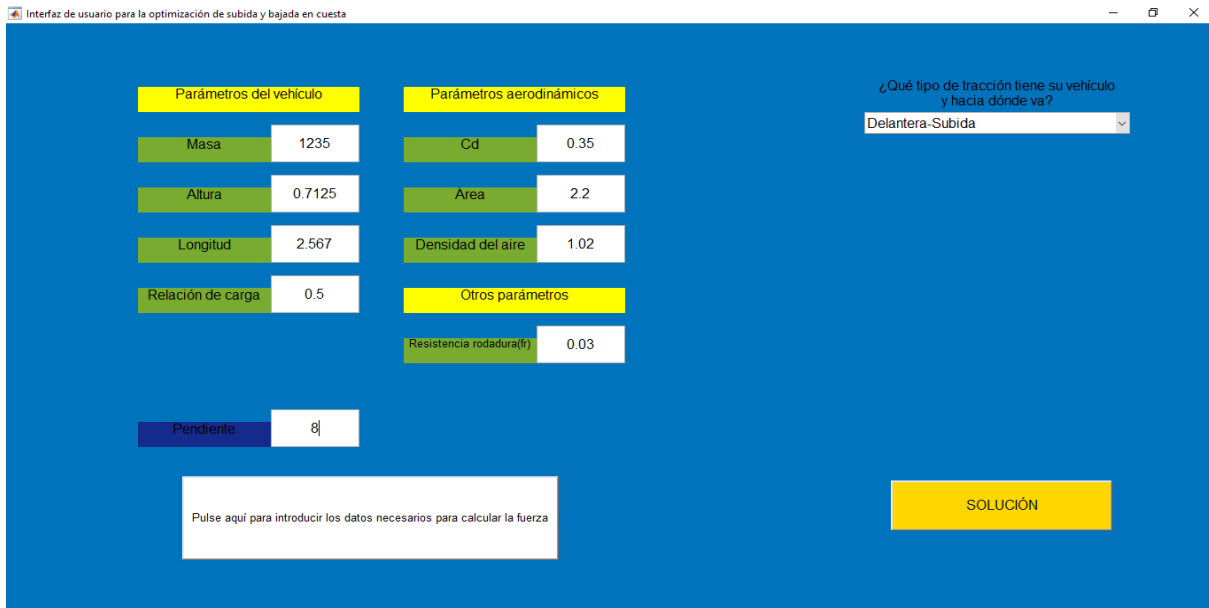


Figura 7-5. Ejemplo de introducción de datos en la ventana velocidad

A continuación, como bien se indica en la ventana el usuario deberá pulsar sobre el botón indicado para ello para introducir los datos necesarios para calcular la fuerza. Pulsando e introduciendo los datos deberá mostrarse una ventana con un aspecto semejante al de la siguiente figura.

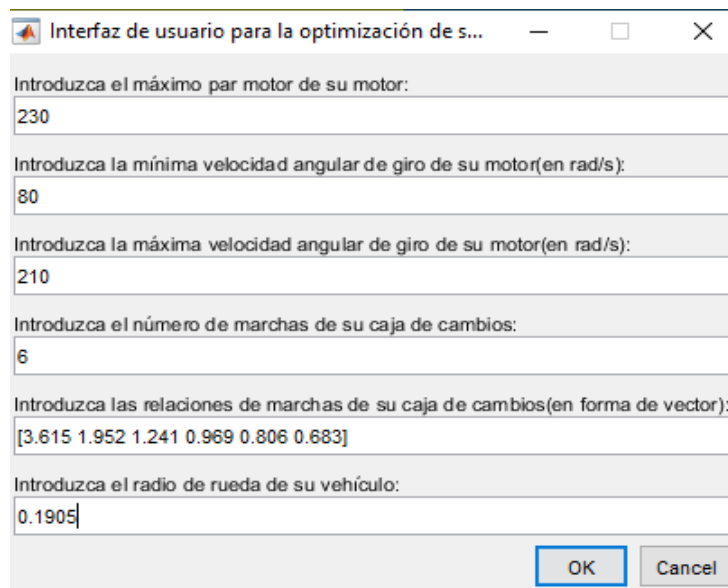


Figura 7-6. Ventana que se abre al pulsar sobre el botón destinado a introducir los datos del sistema de transmisión y el motor

Dándole al boton de OK deberá aparecer la siguiente gráfica y cuadro de diálogo.

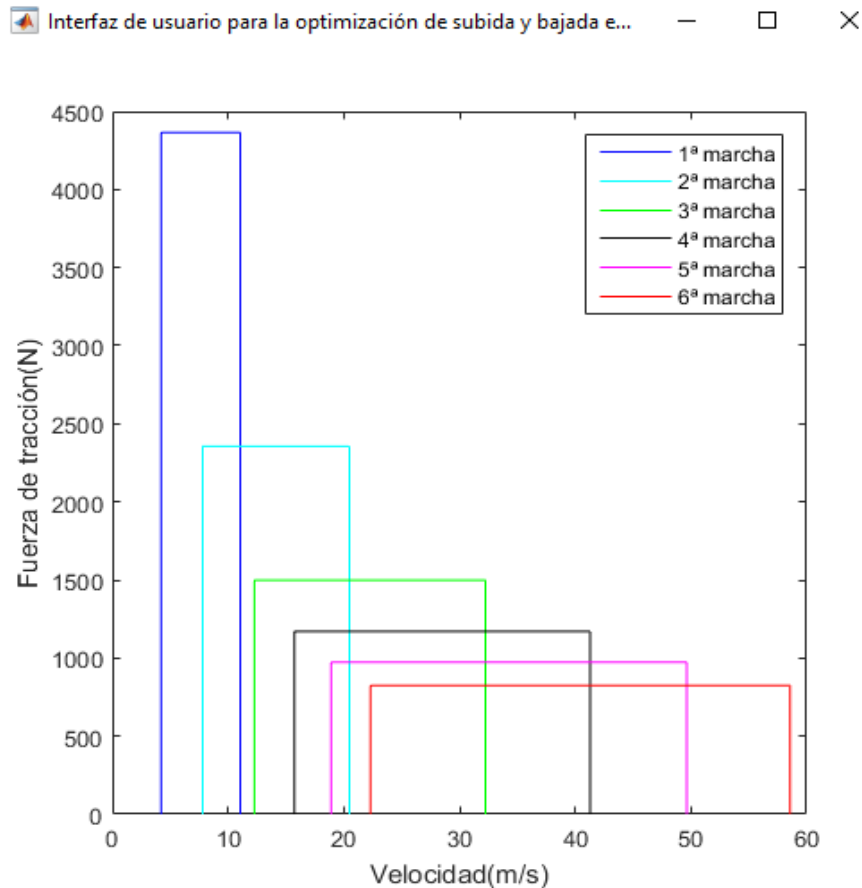


Figura 7-7. Gráfica que relaciona la fuerza de tracción con la velocidad del vehículo en función de las marchas

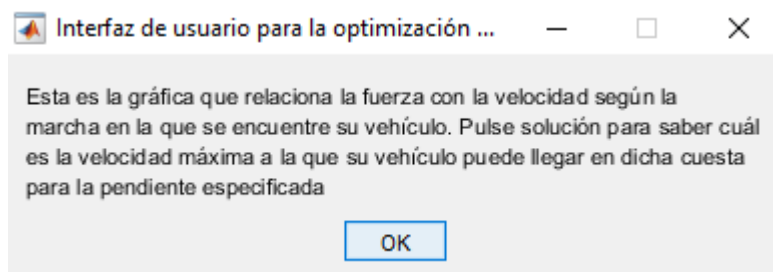


Figura 7-8. Cuadro de diálogo explicativo ventana velocidad

Como bien indica el cuadro de diálogo el usuario deberá pulsar sobre el botón solución si quiere obtener la velocidad máxima a la que puede ascender el Mini Cooper la pendiente de 8°. Realizando esto se obtiene la siguiente solución.

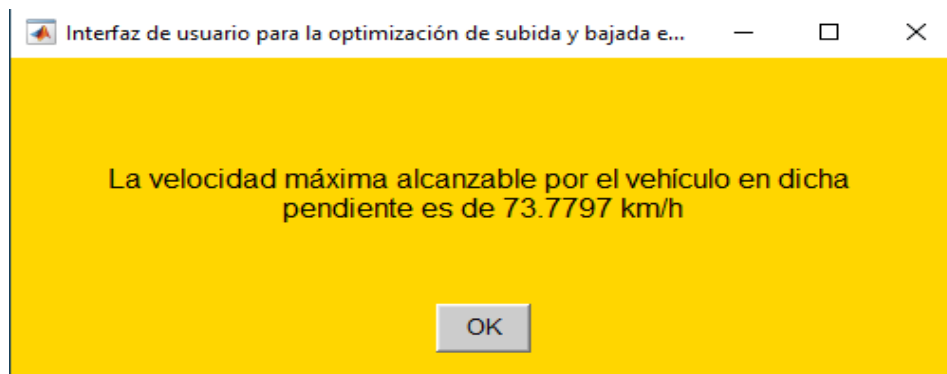


Figura 7-9. Solución de la velocidad máxima admisible para el caso de pendiente ascendente

Análogamente, una vez se pulse sobre el botón OK, en el pop-up menú de la ventana velocidad se cambia el sentido de circulación del coche seleccionando la opción “Delantera-Bajada” y aparece la ventana de texto con la siguiente solución para el caso de que la pendiente sea descendente.

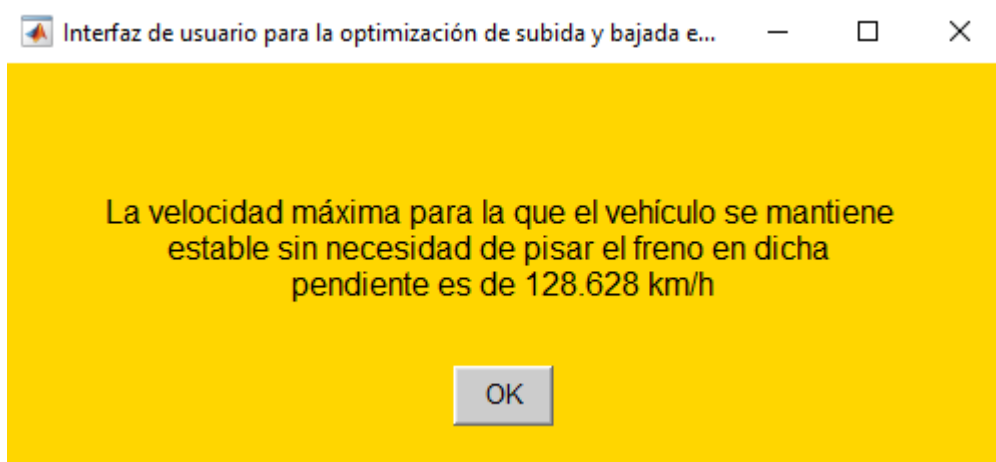


Figura 7-10. Solución de la velocidad máxima admisible en el caso de pendiente descendente

Una vez el usuario haya realizado el primer ejemplo para calcular la velocidad máxima admisible, se va cambiando el valor de la pendiente, teniendo en cuenta la pendiente máxima calculada en el sub-capítulo anterior para el caso de un automóvil con neumáticos ordinarios sobre un asfalto seco. Con ello, se obtiene la siguiente tabla de resultados.

Pendiente	Velocidad máxima alcanzable (pendiente ascendente)	Velocidad máxima alcanzable (pendiente descendente)
8°	73.7797 Km/h	128.628 Km/h
10°	39.839 Km/h	174.348 Km/h
15°	39.839 Km/h	73.7797 Km/h

Tabla 7-3. Tabla de resultados del cálculo de la velocidad máxima admisible para el caso del Mini Cooper

7.3 Cálculo del consumo óptimo de combustible

Para el caso del consumo óptimo de combustible serán necesarios los mismos datos que en el caso del cálculo de la velocidad máxima admisible, además se deberá conocer los datos relativos a la densidad de combustible. En esta ocasión la velocidad a la que circula el vehículo también deberá ser definida por el usuario, además de la relación de transmisión del diferencial y la eficiencia de la transmisión, que tendrán un valor de 1.9 y 0.9 respectivamente tal y como se supuso antes.

La ventana consumo de combustible, una vez el usuario haya introducido los datos, quedará como aparece la ventana velocidad en la figura 7-5 con la única diferencia de que además también estará presente la densidad del combustible. A continuación el usuario deberá pulsar sobre el botón destinado a calcular la tabla que contendrá la información relativa a las parejas de par motor y velocidad de giro del motor para cada marcha, una vez hecho esto aparecerá la siguiente ventana:

Interfaz de usuario para la optimizació...

Introduzca la velocidad a la que circula su vehículo (en km/h)
35

Introduzca el número de marchas de su caja de cambios:
6

Introduzca las relaciones de marchas de su caja de cambios (en forma de vector):
[3.615 1.952 1.241 0.969 0.806 0.683]

Introduzca la relación diferencial final:
1.9

Introduzca el rendimiento de la transmisión
0.9

Introduzca el radio de rueda de su vehículo:
0.1905

OK Cancel

Figura 7-11. Ventana que se abre al pulsar sobre el botón destinado a calcular la tabla de par motor y velocidad de giro

Al introducir los datos que aparecen en dicha figura y pulsar sobre el botón OK aparecen las dos siguientes ventanas.

Interfaz de usuario para la optimización de subida y bajada e...

	M (Nm)	w(rpm)
1ª marcha	64.1972	3.3474e+03
2ª marcha	118.8897	1.8075e+03
3ª marcha	187.0046	1.1491e+03
4ª marcha	239.4972	897.2621
5ª marcha	287.9315	746.3295
6ª marcha	339.7844	632.4355

Continuar

Figura 7-12 Ventana que contiene la tabla de par motor y velocidad de giro

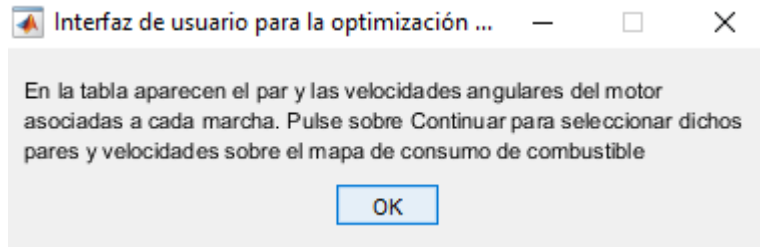


Figura 7-13 Cuadro de diálogo explicativo ventana consumo de combustible

Como bien indica el cuadro de diálogo explicativo, habrá que pulsar sobre el botón continuar en la ventana que contiene la tabla de pares motores y velocidades de giro. Al hacer esto nos aparece el mapa de consumo de combustible que tiene el siguiente aspecto.

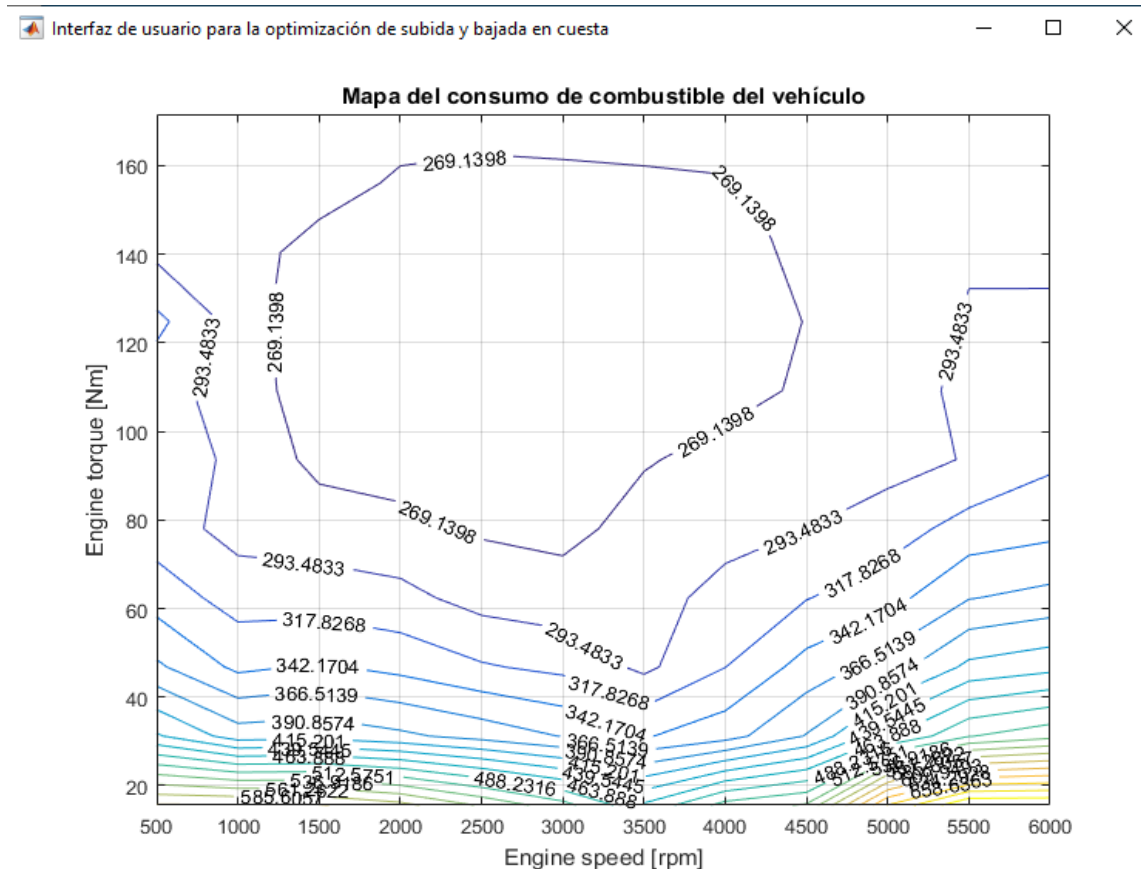


Figura 7-14 Mapa de consumo de combustible

Cuando ya se hayan indicado los puntos correspondientes a las parejas de la tabla, habrá que presionar el botón enter del teclado para que estos queden guardados. Después de esto habrá que pulsar sobre el botón solución de la ventana consumo de combustible para obtener el consumo óptimo y la marcha en la que se alcanza. Al pulsarlo se muestra la siguiente ventana (para el caso de la pendiente ascendente).

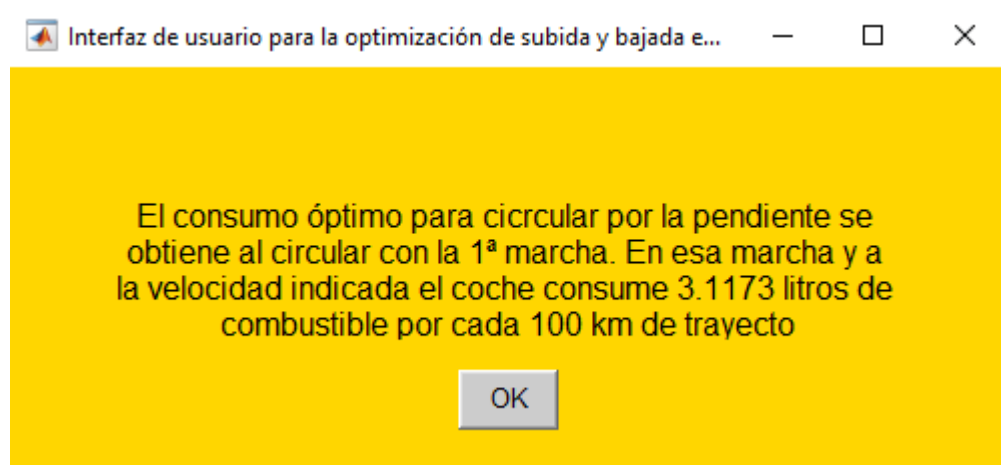


Figura 7-15. Solución del consumo óptimo de combustible para el caso de pendiente ascendente

Análogamente, una vez se pulse sobre el botón OK, en el pop-up menú de la ventana pendiente se cambia el sentido de circulación del coche seleccionando la opción “Delantera-Bajada” y aparece la ventana de texto con la siguiente solución para el caso de que la pendiente sea descendente.

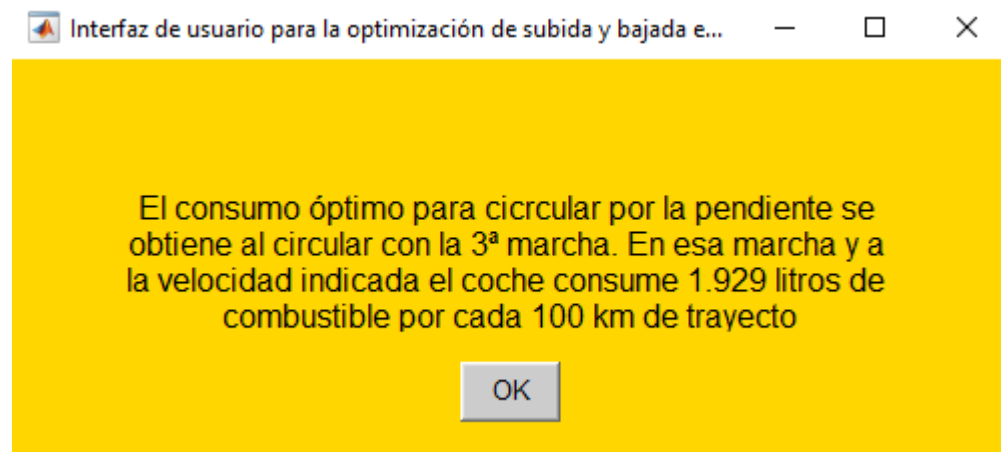


Figura 7-16. Solución del consumo óptimo de combustible para el caso de pendiente descendente

Este ejemplo ha sido calculado para una velocidad de circulación de 35 Km/h (ya que para una cierta pendiente la velocidad máxima que el vehículo podía alcanzar tan solo era de 39 Km/h) y para una pendiente de 8°. En la siguiente tabla aparecen los datos del consumo óptimo para distintos valores de las pendientes.

Pendiente	Consumo óptimo (pendiente ascendente)	Marcha en la que se alcanza	Consumo óptimo (pendiente descendente)	Marcha en la que se alcanza
8°	3.1173 L/100 Km	1ª	1.929 L/100 Km	3ª
10°	3.7394 L/100 Km	1ª	2.557 L/100 Km	5ª
15°	5.2733 L/100 Km	3ª	4.115 L/100 Km	1ª

Tabla 7-4. Tabla de resultados del cálculo del consumo óptimo de combustible para el caso del Mini Cooper

7.4 Obtención de resultados para otros vehículos

A continuación se va a obtener los resultados anteriormente obtenidos para el Mini Cooper, pero en esta ocasión para otros dos modelos distintos de automóviles. Se ha querido estudiar vehículos que contaran por

una parte con tracción trasera y por otra parte con tracción en los dos ejes, ya que en el caso del Mini Cooper este contaba con tracción delantera.

En primer lugar se ha analizado el SSANYONG Rexton, un automóvil que consta con tracción trasera y que luce tal y como aparece en la siguiente figura.



Figura 7-17. SSANGYONG REXTON D22DTR 4x2 Premium 181CV de 2018

Este vehículo cuenta con las siguientes especificaciones técnicas.

Masa	1995 Kg	Par máximo	400 N·m	
Distancia entre los ejes	2.865 m	Máxima velocidad	185 Km/h	
Relación de carga	0.5	Velocidad de giro mínima del motor	80 rad/s	
Altura	0.9125	Velocidad de giro máxima del motor	165 rad/s	
Radio de las ruedas	0.2159 m	Relaciones de transmisión	1ª marcha	4.489
Área frontal	3.13796 m		2ª marcha	2.337
Tipo de tracción	Trasera		3ª marcha	1.350
Tipo de combustible	Diesel		4ª marcha	1
Densidad del combustible	850 Kg/m ³		5ª marcha	0.784
			6ª marcha	0.679

Tabla 7-5. Especificaciones técnicas SSANGYONG REXTON D22DTR 4x2 Premium 181CV de 2018

En segundo lugar se ha estudiado el Audi A7, el cual cuenta con tracción en sus dos ejes. El vehículo en cuestión aparece más abajo junto a sus correspondientes especificaciones técnicas.



Figura 7-18. AUDI A7 Sportback 50 TDI 210kW quattro triptron. 286CV de 2020

Masa	1955 Kg	Par máximo	620 N·m	
Distancia entre los ejes	2.926 m	Máxima velocidad	250 Km/h	
Relación de carga	0.5	Velocidad de giro mínima del motor	80 rad/s	
Altura	0.711	Velocidad de giro máxima del motor	195 rad/s	
Radio de las ruedas	0.2286 m	Relaciones de transmisión	1ª marcha	5
Área frontal	2.71 m		2ª marcha	3.2
Tipo de tracción	En los dos ejes		3ª marcha	2.143
Tipo de combustible	Diesel		4ª marcha	1.720
Densidad del combustible	850 Kg/m ³		5ª marcha	1
			6ª marcha	0.823
			7ª marcha	0.64

Tabla 7-6. Especificaciones técnicas AUDI A7 Sportback 50 TDI 210kW quattro triptron. 286CV de 2020

7.4.1 Pendiente máxima superable

En este apartado se mostrarán los resultados de pendiente máxima superable para el Ssanyong Rexton y para el Audi A7, respectivamente. Como ya sucedió en el ejemplo del funcionamiento de la interfaz gráfica con el Mini Cooper, en las siguientes tablas de resultados se muestran las pendientes máximas variando el tipo de terreno y el tipo de neumático.

Tipo de asfalto y de neumático	Adherencia máxima	Coefficiente de resistencia por rodadura	Pendiente máxima superable (ascendente)	Pendiente máxima superable (descendente)
Neumáticos de baja resistencia sobre carretera lisa	0.9	0.01	31.6666°	19.6729°
Neumáticos ordinarios sobre asfalto seco	0.9	0.03	30.4883°	20.4586°
Neumáticos ordinarios sobre asfalto húmedo	0.7	0.03	22.3825°	17.2613°
Neumáticos ordinarios sobre hormigón	0.9	0.02	31.0811°	20.0667°
Neumáticos ordinarios sobre grava	0.6	0.055	16.8505°	16.5964°

Tabla 7-7. Tabla de resultados del cálculo de la pendiente máxima superable para el caso del Ssanyong Rexton

Tipo de asfalto y de neumático	Adherencia máxima	Coefficiente de resistencia por rodadura	Pendiente máxima superable (ascendente)	Pendiente máxima superable (descendente)
Neumáticos de baja resistencia sobre carretera lisa	0.9	0.01	31.7649°	32.3376°
Neumáticos ordinarios sobre asfalto seco	0.9	0.03	31.185°	32.9031°
Neumáticos ordinarios sobre asfalto húmedo	0.7	0.03	26.5618°	28.5771°
Neumáticos ordinarios sobre hormigón	0.9	0.02	31.4759°	32.6213°
Neumáticos ordinarios sobre grava	0.6	0.055	22.8778°	26.8907°

Tabla 7-8. Tabla de resultados del cálculo de la pendiente máxima superable para el caso del Audi A7

7.4.2 Velocidad máxima admisible

En este apartado se mostrarán los resultados de velocidad máxima admisible para el Ssanyong Rexton y para

el Audi A7, respectivamente. Como ya sucedió en el ejemplo del funcionamiento de la interfaz gráfica con el Mini Cooper, en las siguientes tablas de resultados se muestran las velocidades máximas para el caso en el que el vehículo tiene neumáticos ordinarios y va circulando sobre un asfalto seco. Además irá variando el valor de la pendiente para ver qué ocurre.

Pendiente	Velocidad máxima alcanzable (pendiente ascendente)	Velocidad máxima alcanzable (pendiente descendente)
8°	54.8757 Km/h	143.028 Km/h
10°	54.8757 Km/h	94.996 Km/h
15°	28.5686 Km/h	54.8757 Km/h

Tabla 7-9. Tabla de resultados del cálculo de la velocidad máxima admisible para el caso del Ssanyong Rexton

Pendiente	Velocidad máxima alcanzable (pendiente ascendente)	Velocidad máxima alcanzable (pendiente descendente)
8°	93.3007 Km/h	98.676 Km/h
10°	93.3007 Km/h	165.924 Km/h
15°	74.8844 Km/h	194.9904 Km/h

Tabla 7-10. Tabla de resultados del cálculo de la velocidad máxima admisible para el caso del Audi A7

7.4.3 Consumo óptimo de combustible

En este apartado se mostrarán los resultados del consumo óptimo de combustible para el Ssanyong Rexton y para el Audi A7, respectivamente. Como ya sucedió en el ejemplo del funcionamiento de la interfaz gráfica con el Mini Cooper, en las siguientes tablas de resultados se muestran los consumos óptimos y las marchas en las que se alcanzan estos para el caso en el que el vehículo circula a 35 Km/h (aunque esta velocidad exceda un poco la velocidad máxima admisible por el Ssanyong en una pendiente ascendente de 15°). Además irá variando el valor de la pendiente para ver qué ocurre.

Pendiente	Consumo óptimo (pendiente ascendente)	Marcha en la que se alcanza	Consumo óptimo (pendiente descendente)	Marcha en la que se alcanza
8°	4.0202 L/100 Km	1ª	2.5013 L/100 Km	2ª
10°	4.8241 L/100 Km	2ª	3.3129 L/100 Km	2ª
15°	6.8063 L/100 Km	2ª	5.3217 L/100 Km	2ª

Tabla 7-11. Tabla de resultados del cálculo del consumo óptimo de combustible para el caso del Ssanyong Rexton

Pendiente	Consumo óptimo (pendiente ascendente)	Marcha en la que se alcanza	Consumo óptimo (pendiente descendente)	Marcha en la que se alcanza
8°	3.9322 L/100 Km	1ª	2.4585 L/100 Km	2ª
10°	4.72 L/100 Km	1ª	3.2538 L/100 Km	5ª
15°	6.6625 L/100 Km	1ª	5.2223 L/100 Km	1ª

Tabla 7-12. Tabla de resultados del cálculo del consumo óptimo de combustible para el caso del Audi A7

8 CONCLUSIONES Y LÍNEAS DE MEJORA

En este último capítulo se van a analizar los resultados obtenidos para los distintos modelos de vehículos vistos en el anterior capítulo, además de mostrar las conclusiones globales que el autor de este trabajo de fin de grado ha sacado de él. Por último se sugerirán algunas de las mejoras que el autor cree que se pueden implementar en la interfaz gráfica diseñada.

Se puede decir que la interfaz para optimizar la subida y bajada en cuesta consigue uno de los objetivos iniciales de su diseño, que era el de ayudar a que el usuario estableciera un primer contacto con la simulación dinámica vehicular. Esta simulación se realiza de una forma muy simple, ya que la interfaz lo que hace es un cálculo simple sin cargar la pantalla de demasiadas gráficas y esquemas, lo que podría confundir en un primer momento al usuario principiante.

Además para indicar al usuario lo que tiene que hacer en todo momento y que este no se pierda entre tantos botones (como pasa en ciertos programas) se ha llevado a cabo un diseño minimalista que consta de un par de botones, los cuales si los pulsas en algunos casos le pedirán al usuario más datos y una vez estén estos introducidos se explica lo que ha hecho con un cuadro de diálogo, en el que también se refleja lo que tiene que hacer para seguir adelante con los cálculos.

En líneas generales la interfaz diseñada ofrece al usuario una gran facilidad para calcular parámetros como la pendiente máxima, la velocidad máxima o el consumo óptimo de combustible, a través de las especificaciones técnicas del automóvil. Además se da la oportunidad para que el usuario experimente que ocurre si cambia alguno de los datos introducidos.

Para ver cómo varían estos parámetros en el capítulo anterior se estudiaron hasta tres modelos, de los cuales se obtuvieron sus correspondientes tablas de resultados, que se van a analizar en el sub-capítulo que empieza a continuación.

8.1 Análisis de resultados

En este sub-capítulo se van a analizar los resultados obtenidos en el capítulo anterior. Estos serán analizados en el siguiente orden: pendiente máxima superable, velocidad máxima admisible y consumo óptimo de combustible.

8.1.1 Pendiente máxima

En vista de los resultados obtenidos en las tablas 7-2, 7-7 y 7-8, las cuales muestran los resultados del cálculo de la pendiente máxima superable, se puede decir que dentro de los vehículos estudiados el que mayor capacidad de subida de pendientes y bajada de pendientes presenta es el vehículo con tracción en los dos ejes, en este caso el Audi A7. También resulta curioso que el automóvil dotado con tracción trasera (Ssanyong Rexton) tiene una mejor capacidad de subida de pendientes que el automóvil con tracción delantera (Mini Cooper), pero sin embargo el de tracción trasera tiene una mejor capacidad de bajada de pendientes que el que tiene tracción trasera.

Por otra parte se observa que la mejor condición, en los tres modelos de automóviles, para ascender una pendiente es hacerlo con neumáticos de baja resistencia sobre una carretera lisa y para descenderla es hacerlo con neumáticos ordinarios sobre asfalto seco.

En general tanto el Ssanyong Rexton como el Audi A7 tienen una buena capacidad de subida de pendientes

mientras que el Mini Cooper tiene una capacidad de subida de pendientes un tanto pobre.

8.1.2 Velocidad máxima

Analizando los resultados obtenidos en las tablas 7-3, 7-9 y 7-10, las cuales muestran los resultados del cálculo de la velocidad máxima admisible, se observa como el vehículo con tracción en los dos ejes (Audi A7) es el que mayor velocidad puede llegar a conseguir en el ascenso de pendientes, mientras que depende del valor de la pendiente la el vehículo con tracción delantera (Mini Cooper) puede llegar a conseguir mayor o menor velocidad que el vehículo con tracción trasera (Ssanyong Rexton). Estos valores de velocidad máxima admisible en el ascenso de pendientes son en mayor o menor medida aceptables y coherentes con la realidad.

Por otra parte está el caso del descenso de pendientes donde las velocidades máximas obtenidas, en los tres modelos de automóviles estudiados, son excesivamente altas, lo que puede deberse a que tan solo se ha tenido en cuenta el uso del freno motor como fuerza resistente, y se ha despreciado el uso de las fuerzas de frenado, por lo que estos datos distan un poco de los resultados que se obtendrían en la realidad.

8.1.3 Consumo óptimo de combustible

Mirando los resultados obtenidos en las tablas 7-4, 7-11 y 7-12, las cuales muestran los resultados del cálculo del consumo óptimo de combustible, se observa como el vehículo que más optimiza el consumo de combustible es el Mini Cooper, el cual cuenta con tracción delantera. Seguido de este, se encuentra el Audi A7 (tracción en los dos ejes) y por último está el Ssanyong Rexton (tracción trasera).

Además se puede ver como en los tres modelos de automóviles estudiados, el consumo en pendiente descendente es menor que en pendiente ascendente, esto es lógico ya que si en una pendiente descendente se deja una marcha en concreto y se levanta el pie del acelerador, no se está inyectando combustible en ese momento y el vehículo seguiría descendiendo debido a la inercia. Sin embargo en una pendiente ascendente es necesario pisar el acelerador en mayor o menor medida para conseguir superarla.

También se observan algunas anomalías en la marcha en la que se consigue el consumo óptimo de combustible ya que en algunos casos sale una marcha demasiado alta para la velocidad de 35 Km/h que se ha seleccionado, lo que puede deberse a la limitación que supone usar el mapa de consumo con el que se ha diseñado la interfaz.

Por último, el consumo de combustible parece un tanto bajo ya que se ha estudiado para una velocidad un tanto baja (35 Km/h). Si se estudiara a una velocidad más alta el consumo quizás subiría y puede que fuese poco realista con el consumo que estos automóviles tienen en la realidad.

8.2 Líneas futuras de mejora

Dado que el diseño de la interfaz para optimizar la subida y bajada en cuesta es un tanto simplista, y que se han realizado varias hipótesis en los conceptos teóricos para no dificultar su implementación en MATLAB, es obvio que dicha interfaz puede incluir mejoras en un futuro.

Muchas de estas mejoras tienen que ver con los fundamentos teóricos sobre los que tiene su base dicha interfaz, y es que hay una serie de hipótesis simplificadoras que se podrían eliminar para representar más fielmente la realidad. La primera mejora que se podría plantear es la de incluir las fuerzas de inercia en la ecuación de las fuerzas que se oponen al movimiento (vista en el capítulo del modelo dinámico). Con ello se deberían estudiar también por tanto las aceleraciones que podrían ocurrir en el vehículo tanto en la subida como en la bajada, algo que no se ha tenido en cuenta y que daría un mayor juego a la interfaz.

Otro de los aspectos que se podrían mejorar es el de incluir fuerzas de frenado en la bajada del automóvil, ya que solo se ha tenido en cuenta la fuerza resistente que ejerce el freno motor al cambiar de marcha, algo que no es muy realista ya que el freno motor nunca puede sustituir la fuerza provocada por el pedal de freno.

Por otra parte se podría implementar la interfaz gráfica diseñada de manera que se le permitiera al usuario introducir la forma real que tiene la gráfica que relaciona el par motor con la velocidad de giro del motor, para

de esta forma representar más fielmente la gráfica que relaciona la fuerza de tracción o resistente con la velocidad del vehículo, ofreciendo así una visión más realista del problema.

Uno de los aspectos de la interfaz que habría que mejorar con casi total seguridad es la de la obtención de los datos de flujo másico de combustible, ya que este varía de un motor a otro. Por tanto lo ideal sería hacer un ensayo al automóvil, del cual se quiera obtener los resultados, con un dinamómetro. El objetivo de realizar esta acción no sería otro que el de dotar de un mayor realismo al cálculo del consumo óptimo de combustible, ya que este cálculo es probablemente el que más difiera con la realidad de todos los realizados.

Además, se podrían incluir varios objetos tipo slider en cada una de las ventanas a las que se puede acceder desde la ventana menú. Con estos objetos se podría variar el valor de los parámetros del vehículo que el usuario desee, mientras en otra ventana de solución aparecería cómo cambia el valor de la variable que el usuario esté calculando en ese momento.

Por último, con el objetivo de ampliar el rango de automóviles cuyos parámetros se podrían estudiar en esta interfaz, se intentaría introducir en un nuevo diseño la posibilidad de obtener dichos parámetros en vehículos eléctricos o híbridos.

REFERENCIAS

- [1] Font Mezquita, José, Dols Ruiz, Juan Fco. Tratado sobre automóviles: tomo IV, la dinámica del automóvil. Valencia: UPV, 2006. ISBN9788483630204.
- [2] Pintado Sanjuán, Publio. Un curso de automoción. Universidad de Sevilla: Escuela de Ingenieros Industriales, 1994. ISBN9788488783066.
- [3] Pintado Sanjuán, Publio. Transmisión. Universidad de Castilla La Mancha, 2000. ISBN9788484270751.
- [4] S.T. Smith, MATLAB®: Advanced GUI Development. Dog Ear Publishing, Indianapolis USA, 2006. ISBN1598581813
- [5] Wideberg, Johan. Dinámica longitudinal (Automóviles). 2019-2020. Universidad de Sevilla. Escuela Técnica Superior de Ingeniería. Departamento de Ingeniería y Ciencia de los Materiales y del Transporte. 2019
- [6] Díaz Madrigal, Santiago y Fernández Sánchez Fernando. Apuntes de Matemática Computacional. Universidad de Sevilla. Escuela Técnica Superior de Ingeniería. Departamento de Matemática Aplicada II. 2020
- [7] La Potencia y El Par Motor, Estudio De La Cadena Cinemtica, Estabilidad En Los Vehículos. ingemecanica.com [consulta: 26 de junio de 2020] Disponible en: ingemecanica.com/tutorialsemanal/tutorialn63.html.
- [8] Brake Specific Fuel Consumption (BSFC). x-engineer.com [consulta: 25 de junio de 2020] Disponible en: <https://x-engineer.org/automotive-engineering/internal-combustion-engines/performance/brake-specific-fuel-consumption-bsfc/>
- [9] Friccion por rodadura. Apuntes ingeniería mecánica [consulta: 26 de junio 2020] Disponible en: <https://apuntes-ing-mecanica.blogspot.com/2015/05/titulo-friccion-subtitulo-friccion-por.html?m=0>
- [10] R. Di Rado, Gustavo, Presta García, Daniel S., H. Devincenzi, Gustavo. ANALISIS DE LAS FUERZAS QUE ACTÚAN EN LA INTERFACE NEUMÁTICO – CARRETERA. MODELOS DE SIMULACIÓN DE ACELERACIÓN. Asociación Argentina de Mecánica Computacional. 2013. Disponible en: <https://pdfs.semanticscholar.org/e2e3/651fdafede1085d3f96a377fe1ff17148be7.pdf>
- [11] Ficha técnica Mini Cooper 5 puertas 136 CV 2020. coches.net [consulta: 25 de junio de 2020] Disponible en: https://www.coches.net/fichas_tecnicas/mini/mini/berlina/5-puertas/cooper_5_puertas_136cv_gasolina/60309/737591520200301/
- [12] Ficha técnica Ssangyong Rexton D22DTR 4x2 Premium 181 CV 2018. coches.net [consulta: 25 de junio de 2020] Disponible en: https://www.coches.net/fichas_tecnicas/ssangyong/rexton/4x4/5-puertas/d22dtr_4x2_premium_181cv_diesel/84133/761228420181005/
- [13] Ficha técnica Audi A7 Sportback 50 TDI 210kW quattro triptron. 286CV de 2020. coches.net [consulta: 25 de junio de 2020] Disponible en: https://www.coches.net/fichas_tecnicas/audi/a7/berlina/5-puertas/sportback_50_tdi_210kw_quattro_triptron_286cv_diesel/84432/794953120200210/

ANEXOS

Anexo A. Código de MATLAB para el diseño de la interfaz para optimizar la subida y bajada en cuesta

```

function optim_subida_bajada
close('all')
global tam_screen
tam_screen=get(0,'ScreenSize');
hfig=figure('Position',[tam_screen(1) tam_screen(2) tam_screen(3)
tam_screen(4)],...
'Color',[0 0.45 0.74],...
'MenuBar','none',...
'Name','Interfaz de usuario para la optimización de subida y bajada
en cuesta',...
'NumberTitle','off',...
'Tag','portada',...
'Units','pixels',...
'Visible','on');
uicontrol('Style','text',...
'Position',[155 625 1100 50],...
'BackgroundColor',[0 0.45 0.74],...
'String','Interfaz de usuario para la optimización de subida y bajada
en cuesta',...
'FontName','Arial',...
'FontSize',24,...
'FontWeight','bold');
axes('Units','pixels',...
'Position',[200 450 1000 150],...
'XTick',[],...
'YTick',[],...
'XColor','white',...
'YColor','white');
haxes=imread('tituloLogoGrande.jpg');
image(haxes);
axis off
uicontrol('Style','text',...
'Position',[325 300 750 100],...
'BackgroundColor',[0 0.45 0.74],...
'String','Autor: Jose Antonio Rodríguez Rodríguez',...
'FontName','Arial',...
'FontSize',24);
uicontrol('Style','text',...
'Position',[325 250 750 100],...
'BackgroundColor',[0 0.45 0.74],...
'String','Tutor: Johan Wideberg',...
'FontName','Arial',...
'FontSize',24);
uicontrol('Style','pushbutton',...

```

```

'Position',[1100 60 150 30],...
'BackgroundColor',[0.91 0.91 0.91],...
'String','Comenzar',...
'FontName','Arial',...
'FontSize',12,...
'Units','points',...
'Tag','comienzo',...
'Visible','On',...
'Callback',@menu);
global data_portada
data_portada=guihandles(hfig);
guidata(hfig,data_portada)
end

function menu(~,~)
hfig2=figure('Position',[500 200 350 450],...
'Color',[0.8 0.8 0.8],...
'MenuBar','none',...
'Name','Interfaz de usuario para la optimización de subida y bajada
en cuesta',...
'NumberTitle','off',...
'Tag','menu',...
'Units','pixels',...
'Visible','off');
movegui(hfig2,'center');
uicontrol('Style','text',...
'Position',[0 400 350 30],...
'BackgroundColor',[0.8 0.8 0.8],...
'String','¿Qué parámetro deseas que sea la incógnita?',...
'FontName','Arial',...
'FontSize',12,...
'Units','points',...
'Tag','pregunta',...
'Visible','On');
uicontrol('Style','pushbutton',...
'Position',[75 300 200 50],...
'BackgroundColor','white',...
'String','Pendiente',...
'FontName','Arial',...
'FontSize',12,...
'Units','points',...
'Tag','pendiente_menu',...
'Visible','On',...
'Callback',@pendiente);
uicontrol('Style','pushbutton',...
'Position',[75 200 200 50],...
'BackgroundColor','white',...
'String','Velocidad',...
'FontName','Arial',...
'FontSize',12,...
'Units','points',...
'Tag','vel_boton',...
'Visible','On',...
'Callback',@velocidad);
uicontrol('Style','pushbutton',...
'Position',[75 100 200 50],...

```

```

'BackgroundColor','white',...
'String','Consumo de combustible',...
'FontName','Arial',...
'FontSize',12,...
'Units','points',...
'Tag','fuerza_boton',...
'Visible','On',...
'Callback',@consumo_combustible);
set(hfig2,'Visible','on');
data_menu=guihandles(hfig2);
guidata(hfig2,data_menu)
end

function pendiente(objeto,~)
global data_portada
global tam_screen
data=guidata(objeto);
set(data.menu,'Visible','off');
set(data_portada.portada,'Visible','off');
hfig=figure('Position',[tam_screen(1) tam_screen(2) tam_screen(3)
tam_screen(4)],...
'Color',[0 0.45 0.74],...
'MenuBar','none',...
'Name','Interfaz de usuario para la optimización de subida y bajada
en cuesta',...
'NumberTitle','off',...
'Units','pixels',...
'Visible','on');
axes('Units','pixels',...
'Position',[950 250 400 300],...
'XTick',[],...
'YTick',[],...
'XColor','white',...
'YColor','white',...
'Visible','off',...
'tag','haxes');
uicontrol('Style','text',...
'Position',[1000 560 300 80],...
'BackgroundColor',[0 0.45 0.74],...
'String',[{'¿Qué tipo de tracción tiene su vehículo'},{'y hacia dónde
va?'}],...
'FontName','Arial',...
'FontSize',12,...
'Tag','pregunta');
uicontrol('Style','text',...
'Position',[150 600 250 30],...
'BackgroundColor','yellow',...
'String','Parámetros del vehículo',...
'FontName','Arial',...
'FontSize',12,...
'Tag','param_vehic');
uicontrol('Style','text',...
'Position',[450 600 250 30],...
'BackgroundColor','yellow',...
'String','Otros parámetros',...
'FontName','Arial',...

```

```

'FontSize',12,...
'Tag','param_otros');
uicontrol('Style','text',...
'Position',[150 540 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Altura',...
'FontName','Arial',...
'FontSize',12,...
'Tag','h_text');
uicontrol('Style','text',...
'Position',[150 480 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Longitud',...
'FontName','Arial',...
'FontSize',12,...
'Tag','L_text');
uicontrol('Style','text',...
'Position',[150 420 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Relación de carga',...
'FontName','Arial',...
'FontSize',12,...
'Tag','landa_text');
uicontrol('Style','text',...
'Position',[450 540 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Adherencia',...
'FontName','Arial',...
'FontSize',12,...
'Tag','nu_text');
uicontrol('Style','text',...
'Position',[450 480 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Resistencia rodadura(fr)',...
'FontName','Arial',...
'FontSize',10,...
'Tag','fr_text');
uicontrol('Style','edit',...
'Position',[300 540 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','h');
uicontrol('Style','edit',...
'Position',[300 480 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','L');
uicontrol('Style','edit',...
'Position',[300 420 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','landa');
uicontrol('Style','edit',...
'Position',[600 540 100 45],...

```



```

'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','nu');
uicontrol('Style','edit',...
'Position',[600 480 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','fr');
traccion=[{'Trasera-Subida'};{'Trasera-Bajada'};{'Delantera-
Subida'};{'Delantera-Bajada'};{'En los dos ejes-Subida'};{'En los dos
ejes-Bajada'}];
uicontrol('Style','popupmenu',...
'Position',[1020 500 275 100],...
'BackgroundColor','white',...
'String',traccion,...
'FontName','Arial',...
'FontSize',12,...
'Tag','trac_elec');
uicontrol('Style','pushbutton',...
'Position',[1000 150 250 60],...
'BackgroundColor',[1 0.84 0],...
'String','SOLUCIÓN',...
'FontName','Arial',...
'FontSize',12,...
'Tag','sol_boton',...
'Callback',@resolv_alpha);
data_alpha=guihandles(hfig);
guidata(hfig,data_alpha)
end

function velocidad(objeto,~)
global data_portada
global tam_screen
data=guidata(objeto);
set(data.menu,'Visible','off');
set(data_portada.portada,'Visible','off');
hfig3=figure('Position',[tam_screen(1) tam_screen(2) tam_screen(3)
tam_screen(4)],...
'Color',[0 0.45 0.74],...
'MenuBar','none',...
'Name','Interfaz de usuario para la optimización de subida y bajada
en cuesta',...
'NumberTitle','off',...
'Units','pixels',...
'Visible','on');
axes('Units','pixels',...
'Position',[950 250 400 300],...
'XTick',[],...
'YTick',[],...
'XColor','white',...
'YColor','white',...
'Visible','off',...
'tag','haxes');
uicontrol('Style','text',...

```

```

'Position',[150 600 250 30],...
'BackgroundColor','yellow',...
'String','Parámetros del vehículo',...
'FontName','Arial',...
'FontSize',12,...
'Tag','param_vehic');
uicontrol('Style','text',...
'Position',[450 600 250 30],...
'BackgroundColor','yellow',...
'String','Parámetros aerodinámicos',...
'FontName','Arial',...
'FontSize',12,...
'Tag','param_aero');
uicontrol('Style','text',...
'Position',[450 360 250 30],...
'BackgroundColor','yellow',...
'String','Otros parámetros',...
'FontName','Arial',...
'FontSize',12,...
'Tag','param_otros');
uicontrol('Style','text',...
'Position',[970 560 300 80],...
'BackgroundColor',[0 0.45 0.74],...
'String',[{'¿Qué tipo de tracción tiene su vehículo'},{'y hacia dónde
va?'}],...
'FontName','Arial',...
'FontSize',12,...
'Tag','pregunta');
uicontrol('Style','text',...
'Position',[150 540 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Masa',...
'FontName','Arial',...
'FontSize',12,...
'Tag','m_text');
uicontrol('Style','text',...
'Position',[150 480 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Altura',...
'FontName','Arial',...
'FontSize',12,...
'Tag','h_text');
uicontrol('Style','text',...
'Position',[150 420 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Longitud',...
'FontName','Arial',...
'FontSize',12,...
'Tag','L_text');
uicontrol('Style','text',...
'Position',[150 360 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Relación de carga',...
'FontName','Arial',...
'FontSize',12,...
'Tag','landa_text');
uicontrol('Style','text',...

```

```

'Position',[450 540 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Cd',...
'FontName','Arial',...
'FontSize',12,...
'Tag','cd_text');
uicontrol('Style','text',...
'Position',[450 480 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Área',...
'FontName','Arial',...
'FontSize',12,...
'Tag','A_text');
uicontrol('Style','text',...
'Position',[450 420 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Densidad del aire',...
'FontName','Arial',...
'FontSize',12,...
'Tag','ro_text');
uicontrol('Style','text',...
'Position',[450 300 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Resistencia rodadura(fr)',...
'FontName','Arial',...
'FontSize',10,...
'Tag','fr_text');
uicontrol('Style','edit',...
'Position',[300 540 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','m');
uicontrol('Style','edit',...
'Position',[300 480 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','h');
uicontrol('Style','edit',...
'Position',[300 420 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','L');
uicontrol('Style','edit',...
'Position',[300 360 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','landa');
uicontrol('Style','edit',...
'Position',[600 540 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...

```

```

    'Tag','cd');
uicontrol('Style','edit',...
'Position',[600 480 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','A');
uicontrol('Style','edit',...
'Position',[600 420 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','ro');
uicontrol('Style','edit',...
'Position',[600 300 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','fr');
uicontrol('Style','text',...
'Position',[150 200 150 30],...
'BackgroundColor',[0.08 0.17 0.55],...
'String','Pendiente',...
'FontName','Arial',...
'FontSize',12,...
'Tag','alpha_text');
uicontrol('Style','edit',...
'Position',[300 200 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','alpha');
pregunta=[{'Trasera-Subida'};{'Trasera-Bajada'};{'Delantera-Subida'};{'Delantera-Bajada'};{'En los dos ejes-Subida'};{'En los dos ejes-Bajada'}}];
uicontrol('Style','popupmenu',...
'Position',[970 500 300 100],...
'BackgroundColor','white',...
'String',pregunta,...
'FontName','Arial',...
'FontSize',12,...
'Tag','trac_elec');
uicontrol('Style','pushbutton',...
'Position',[200 65 425 100],...
'BackgroundColor','white',...
'String','Pulse aquí para introducir los datos necesarios para calcular la fuerza',...
'FontName','Arial',...
'FontSize',10,...
'Tag','fuerza_boton',...
'Callback',@graf_vel_f);
uicontrol('Style','pushbutton',...
'Position',[1000 100 250 60],...
'BackgroundColor',[1 0.84 0],...
'String','SOLUCIÓN',...
'FontName','Arial',...
'FontSize',12,...

```

```

    'Tag','sol_boton',...
    'Callback',@calc_fuerza_resist);
data_vel=guihandles(hfig3);
guidata(hfig3,data_vel)
end

function consumo_combustible(objeto,~)
global data_portada
global tam_screen
data=guidata(objeto);
set(data.menu,'Visible','off');
set(data_portada.portada,'Visible','off');
hfig=figure('Position',[tam_screen(1) tam_screen(2) tam_screen(3)
tam_screen(4)],...
    'Color',[0 0.45 0.74],...
    'MenuBar','none',...
    'Name','Interfaz de usuario para la optimización de subida y bajada
en cuesta',...
    'NumberTitle','off',...
    'Units','pixels',...
    'Visible','on');
axes('Units','pixels',...
    'Position',[1000 250 350 300],...
    'XTick',[],...
    'YTick',[],...
    'XColor','white',...
    'YColor','white',...
    'Visible','off',...
    'tag','haxes');
uicontrol('Style','text',...
    'Position',[150 600 250 30],...
    'BackgroundColor','yellow',...
    'String','Parámetros del vehículo',...
    'FontName','Arial',...
    'FontSize',12,...
    'Tag','param_vehic');
uicontrol('Style','text',...
    'Position',[450 600 250 30],...
    'BackgroundColor','yellow',...
    'String','Parámetros aerodinámicos',...
    'FontName','Arial',...
    'FontSize',12,...
    'Tag','param_aero');
uicontrol('Style','text',...
    'Position',[450 360 250 30],...
    'BackgroundColor','yellow',...
    'String','Otros parámetros',...
    'FontName','Arial',...
    'FontSize',12,...
    'Tag','param_otros');
uicontrol('Style','text',...
    'Position',[1000 560 300 80],...
    'BackgroundColor',[0 0.45 0.74],...
    'String',[{'¿Qué tipo de tracción tiene su vehículo'},{'y hacia dónde
va?'}],...
    'FontName','Arial',...

```

```
'FontSize',12,...
'Tag','pregunta');
uicontrol('Style','text',...
'Position',[150 540 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Masa',...
'FontName','Arial',...
'FontSize',12,...
'Tag','m_text');
uicontrol('Style','text',...
'Position',[150 480 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Altura',...
'FontName','Arial',...
'FontSize',12,...
'Tag','h_text');
uicontrol('Style','text',...
'Position',[150 420 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Longitud',...
'FontName','Arial',...
'FontSize',12,...
'Tag','L_text');
uicontrol('Style','text',...
'Position',[150 360 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Relación de carga',...
'FontName','Arial',...
'FontSize',12,...
'Tag','landa_text');
uicontrol('Style','text',...
'Position',[450 540 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Cd',...
'FontName','Arial',...
'FontSize',12,...
'Tag','cd_text');
uicontrol('Style','text',...
'Position',[450 480 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Área',...
'FontName','Arial',...
'FontSize',12,...
'Tag','A_text');
uicontrol('Style','text',...
'Position',[450 420 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Densidad del aire',...
'FontName','Arial',...
'FontSize',12,...
'Tag','ro_text');
uicontrol('Style','text',...
'Position',[450 300 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Resistencia rodadura(fr)',...
'FontName','Arial',...
'FontSize',10,...
```

```

'Tag','fr_text');
uicontrol('Style','text',...
'Position',[725 300 150 30],...
'BackgroundColor',[0.47 0.67 0.19],...
'String','Densidad del combustible',...
'FontName','Arial',...
'FontSize',10,...
'Tag','ro_comb_text');
uicontrol('Style','edit',...
'Position',[300 540 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','m');
uicontrol('Style','edit',...
'Position',[300 480 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','h');
uicontrol('Style','edit',...
'Position',[300 420 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','L');
uicontrol('Style','edit',...
'Position',[300 360 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','landa');
uicontrol('Style','edit',...
'Position',[600 540 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','cd');
uicontrol('Style','edit',...
'Position',[600 480 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','A');
uicontrol('Style','edit',...
'Position',[600 420 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','ro');
uicontrol('Style','edit',...
'Position',[600 300 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','fr');

```

```

uicontrol('Style','edit',...
'Position',[875 300 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','ro_comb');
uicontrol('Style','text',...
'Position',[150 200 150 30],...
'BackgroundColor',[0.08 0.17 0.55],...
'String','Pendiente',...
'FontName','Arial',...
'FontSize',12,...
'Tag','alpha_text');
uicontrol('Style','edit',...
'Position',[300 200 100 45],...
'BackgroundColor','white',...
'FontName','Arial',...
'FontSize',12,...
'Tag','alpha');
uicontrol('Style','pushbutton',...
'Position',[200 65 425 100],...
'BackgroundColor','white',...
'String','Pulse aquí para introducir los datos relativos a la
velocidad y la caja de cambios',...
'FontName','Arial',...
'FontSize',8,...
'Tag','fuerza_boton',...
'Callback',@tabla_par_w);
uicontrol('Style','pushbutton',...
'Position',[1000 100 250 60],...
'BackgroundColor',[1 0.84 0],...
'String','SOLUCIÓN',...
'FontName','Arial',...
'FontSize',12,...
'Tag','sol_boton',...
'Callback',@solucion);
pregunta=[{'Trasera-Subida'};{'Trasera-Bajada'};{'Delantera-
Subida'};{'Delantera-Bajada'};{'En los dos ejes-Subida'};{'En los dos
ejes-Bajada'}];
uicontrol('Style','popupmenu',...
'Position',[1000 500 300 100],...
'BackgroundColor','white',...
'String',pregunta,...
'FontName','Arial',...
'FontSize',12,...
'Tag','trac_elec');
data_vel=guihandles(hfig);
guidata(hfig,data_vel)
end

```


Anexo B. Código de MATLAB de las funciones auxiliares para el funcionamiento de la interfaz

```

function resolv_alpha(objeto,~)
data=guidata(objeto);
h=str2double(get(data.h,'String'));
L=str2double(get(data.L,'String'));
landa=str2double(get(data.landa,'String'));
nu=str2double(get(data.nu,'String'));
fr=str2double(get(data.fr,'String'));
traccion=get(data.trac_elec,'Value');
switch traccion
    case 1
        alpha_rad=atan((fr*L-nu*landa*L)/(nu*h-L));
    case 2
        alpha_rad=atan((nu*landa*L+fr*L)/(nu*h+L));
    case 3
        alpha_rad=atan((nu*(1-landa)*L-fr*L)/(L+nu*h));
    case 4
        alpha_rad=atan(fr*L+(nu*(1-landa)*L)/(L-nu*h));
    case 5
        alpha_d_rad=atan((2*nu*(1-landa)*L-fr*L)/(L+2*nu*h));
        alpha_t_rad=atan((fr*L-2*nu*landa*L)/(2*nu*h-L));
        alpha_rad=min(alpha_d_rad,alpha_t_rad);
    case 6
        alpha_d_rad=atan(fr*L+(2*nu*(1-landa)*L)/(L-2*nu*h));
        alpha_t_rad=atan((2*nu*landa*L+fr*L)/(2*nu*h+L));
        alpha_rad=min(alpha_d_rad,alpha_t_rad);
end
alpha=alpha_rad*180/pi;
cad=num2str(alpha);
mensaje=['La pendiente máxima que el vehículo puede superar es de
',cad,' °'];
muestra_sol(objeto,' ',mensaje)
end

function graf_vel_f(objeto,~)
data=guidata(objeto);
traccion=get(data.trac_elec,'Value');
f=inputdlg({'Introduzca el máximo par motor de su motor: ','...
'Introduzca la mínima velocidad angular de giro de su
motor(en rad/s): ','...
'Introduzca la máxima velocidad angular de giro de su
motor(en rad/s): ','...
'Introduzca el número de marchas de su caja de cambios:
','...
'Introduzca las relaciones de marchas de su caja de
cambios(en forma de vector): ','...
'Introduzca el radio de rueda de su vehículo: ','...
'Interfaz de usuario para la optimización de subida y
bajada en cuesta',1});
M=str2double(f{1,1});
% if traccion==2 || traccion==4 || traccion==6
%     M=M/2;
% end

```

```

w_min=str2double(f{2,1});
w_max=str2double(f{3,1});
n=str2double(f{4,1});
marchas=str2num(f{5,1});
R=str2double(f{6,1});
global F
global v_max
F=zeros(1,n);
v_min=zeros(1,n);
v_max=zeros(1,n);
for ii=1:n
    F(ii)=M*marchas(ii)/R;
    v_min(ii)=w_min*R/marchas(ii);
    v_max(ii)=w_max*R/marchas(ii);
end
color=[98 1 4 4 2 5 5 2];
c=0;
hfig4=figure('Position',[300 200 500 500],...
'Color','white',...
'MenuBar','none',...
'Name','Interfaz de usuario para la optimización de subida y bajada
en cuesta',...
'NumberTitle','off',...
'Units','pixels',...
'Visible','off');
texto=cell(1,n);
for ii=1:n
    x=[v_min(ii) v_min(ii) v_max(ii) v_max(ii)];
    y=[0 F(ii) F(ii) 0];
    c=c+color(ii);
    texto(ii)=strcat({num2str(ii)},{ 'a marcha'});
    plot(x,y,char(c));
    hold on
end
legend(texto)
xlabel('Velocidad(m/s)')
if traccion==1 || traccion==3 || traccion==5
    ylabel('Fuerza de tracción(N)')
    set(hfig4,'Visible','On')
    msgbox('Esta es la gráfica que relaciona la fuerza con la
velocidad según la marcha en la que se encuentre su vehículo. Pulse
solución para saber cuál es la velocidad máxima a la que su vehículo
puede llegar en dicha cuesta para la pendiente especificada',...
'Interfaz de usuario para la optimización de subida y bajada en
cuesta')
else
    ylabel('Fuerza debida al freno motor(N)')
    set(hfig4,'Visible','On')
    msgbox('Esta es la gráfica que relaciona la fuerza opuesta al
movimiento, ejercida por el freno motor, con la velocidad según la
marcha en la que se encuentre su vehículo. Pulse solución para saber
cuál es la velocidad a la que podría circular de forma estable su
vehículo, sin necesidad de pisar el pedal de freno,para la pendiente
especificada',...
'Interfaz de usuario para la optimización de subida y bajada en
cuesta')
end

```

```

end
function calc_fuerza_resist(objeto,~)
data=guidata(objeto);
global F
global v_max
g=9.81;
traccion=get(data.trac_elec,'Value');
m=str2double(get(data.m,'String'));
cd=str2double(get(data.cd,'String'));
A=str2double(get(data.A,'String'));
ro=str2double(get(data.ro,'String'));
fr=str2double(get(data.fr,'String'));
alpha=str2double(get(data.alpha,'String'));
n=length(v_max);
v=0:0.01:v_max(n);
requis=zeros(1,n);
if traccion==1 || traccion==3 || traccion==5
F_resist=m*g*(sin(alpha*pi/180)+fr*cos(alpha*pi/180))+0.5*ro*A*cd*v.^2
;
    for ii=1:n
        requis(ii)=length(F_resist(F_resist<=F(ii)));
    end
    indice_vel=requis(requis~=0);
    if length(indice_vel)>1
        indice_vel=min(indice_vel);
    end
    posi=find(requis==indice_vel);
    sol=v(indice_vel);
    if sol>v_max(posi)
        sol=v_max(posi);
    elseif sol<v_max(posi-1)
        sol=v_max(posi-1);
    end
    sol_km=sol*3600/1000;
    cad=num2str(sol_km);
    mensaje=['La velocidad máxima alcanzable por el vehículo en dicha
pendiente es de ',cad,' km/h'];
    else
        F_descend=m*g*(sin(alpha*pi/180)-fr*cos(alpha*pi/180))-
0.5*ro*A*cd*v.^2;
        bandera=0;
        r=0;
        for ii=1:n
            if ((any(F_descend>F(ii))==1) && (any(F_descend<F(ii))==1))
                requis(ii)=length(find(F_descend>=F(ii)));
            elseif (any(F_descend>F(ii))==0)
                requis(ii)=0;
                r=r+1;
            else
                bandera=1;
                requis(ii)=(m*g*(sin(alpha*pi/180)-fr*cos(alpha*pi/180))-
0.5*ro*A*cd*v_max(ii)^2)-F(ii);
            end
        end
        requis=requis(requis~=0);

```

```

    if bandera==0
        sol=zeros(1,length(requis));
        for ii=1:length(requis)
            sol(ii)=v(requis(ii));
        end
        sol=max(sol);
    else
        indice_vel=min(requis);
        indice_vel=(find(requis==indice_vel))+r;
        sol=v_max(indice_vel);
    end
    sol_km=sol*3600/1000;
    cad=num2str(sol_km);
    mensaje=['La velocidad máxima para la que el vehículo se mantiene
estable sin necesidad de pisar el freno en dicha pendiente es de
',cad,' km/h'];
end
muestra_sol(objeto,' ',mensaje)
end

function tabla_par_w(objeto,~)
data=guidata(objeto);
f=inputdlg({'Introduzca la velocidad a la que circula su vehículo (en
km/h)',...
'Introduzca el número de marchas de su caja de cambios: ',...
'Introduzca las relaciones de marchas de su caja de cambios (en forma
de vector):',...
'Introduzca la relación diferencial final:',...
'Introduzca el rendimiento de la transmisión',...
'Introduzca el radio de rueda de su vehículo:'},...
'Interfaz de usuario para la optimización de subida y bajada en
cuesta',1);
global v_cons
v_cons=str2double(f{1,1});
marchas=str2num(f{3,1});
rr=str2double(f{4,1});
nu_trans=str2double(f{5,1});
R=str2double(f{6,1});
n=length(marchas);
hfig2=figure('Position',[100 250 500 500],...
'Color','white',...
'MenuBar','none',...
'Name','Interfaz de usuario para la optimización de subida y bajada
en cuesta',...
'NumberTitle','off',...
'Units','pixels',...
'Visible','off');
g=9.81;
traccion=get(data.trac_elec,'Value');
m=str2double(get(data.m,'String'));
% h=str2double(get(data.h,'String'));
% L=str2double(get(data.L,'String'));
% landa=str2double(get(data.landa,'String'));
cd=str2double(get(data.cd,'String'));
A=str2double(get(data.A,'String'));
ro=str2double(get(data.ro,'String'));
fr=str2double(get(data.fr,'String'));

```

```

alpha=str2double(get(data.alpha,'String'));
if traccion==1 || traccion==3 || traccion==5

F_calc=m*g*(sin(alpha*pi/180)+fr*cos(alpha*pi/180))+0.5*ro*A*cd*(v_cons*1000/3600)^2;
else
    F_calc=m*g*(sin(alpha*pi/180)-fr*cos(alpha*pi/180))-
0.5*ro*A*cd*(v_cons*1000/3600)^2;
end
M=F_calc*R;
w_rueda=((v_cons/3.6)/R)*(60/(2*pi));
global tabla
tabla=zeros(n,2);
filas=cell(1,n);
format long
for ii=1:n
    tabla(ii,1)=M/(marchas(ii)*rr*nu_trans);
    tabla(ii,2)=w_rueda*marchas(ii)*rr;
    filas(ii)=strcat({num2str(ii)},{ 'a marcha'});
end
t=uitable(hfig2,'Data',tabla,...
    'Position',[50 150 400 200],...
    'ColumnWidth',{125,125},...
    'FontName','Arial',...
    'FontSize',16);
columnas={'M (Nm)', 'w(rpm)'};
set(t,'ColumnName',columnas);
set(t,'RowName',filas);
uicontrol('Style','pushbutton',...
    'Position',[400 50 75 60],...
    'BackgroundColor',[1 0.84 0],...
    'String','Continuar',...
    'FontName','Arial',...
    'FontSize',12,...
    'Tag','continua_boton',...
    'Callback',@selecciona_consumo);
set(hfig2,'Visible','On');
msgbox('En la tabla aparecen el par y las velocidades angulares del
motor asociadas a cada marcha. Pulse sobre Continuar para seleccionar
dichos pares y velocidades sobre el mapa de consumo de
combustible',...
'Interfaz de usuario para la optimización de subida y bajada en
cuesta');
data_fuerza=guihandles(hfig2);
guidata(hfig2,data_fuerza)
end

function selecciona_consumo(~,~)
w_motor=(500:500:6000)';
par_motor=[15.6 31.2 46.8 62.4 78 93.6 109.2 124.8 140.4 156 171.6];
flujo_mas=load('datos_flujo_masico.dat');
w_rad_s=w_motor*pi/30;
Pot_motor=w_rad_s*par_motor/1000;
BSFC=(flujo_mas./Pot_motor)*3600;
hfig3=figure('Position',[700 200 700 600],...
'Color','white',...

```

```

'MenuBar','none',...
'Name','Interfaz de usuario para la optimización de subida y bajada
en cuesta',...
'NumberTitle','off',...
'Units','pixels',...
'Visible','off');
[c,h]=contour(w_motor',par_motor',BSFC',20);
clabel(c,h)
grid on
xlabel('Engine speed [rpm]')
ylabel('Engine torque [Nm]')
title('Mapa del consumo de combustible del vehículo')
set(hfig3,'Visible','on');
valores=clabel(c,h,'manual');
for ii=1:length(valores)
    cad{ii}=valores(ii).String;
end
global consumo
consumo=str2double(cad);
end

function solucion(objeto,~)
data=guidata(objeto);
ro_comb=str2double(get(data.ro_comb,'String'));
global consumo
global tabla
global v_cons
consumo_min=consumo.*(tabla(:,1)).*(tabla(:,2)).*((2*pi/60)*0.001*(
1/ro_comb)*v_cons*0.01);
consumo_real=min(consumo_min);
marcha_opt=find(consumo_min==consumo_real);
if length(marcha_opt)>1
    marcha_opt=marcha_opt(1);
end
cad1=num2str(marcha_opt);
cad2=num2str(consumo_real);
mensaje=['El consumo óptimo para circular por la pendiente se
obtiene al circular con la ',cad1,'a marcha. En esa marcha y a la
velocidad indicada el coche consume ',cad2,' litros de combustible por
cada 100 km de trayecto'];
muestra_sol(objeto,'',mensaje)
end

function muestra_sol(~,~,mensaje)
hfig=figure('Position',[425 300 500 200],...
'Color',[1 0.84 0],...
'MenuBar','none',...
'Name','Interfaz de usuario para la optimización de subida y bajada
en cuesta',...
'NumberTitle','off',...
'Units','pixels',...
'Tag','solucion',...
'Visible','on');
uicontrol('Style','text',...
'Position',[50 65 400 70],...
'BackgroundColor',[1 0.84 0],...

```

```
'String',mensaje,...
'FontName','Arial',...
'FontSize',12,...
'Tag','param_vehic');
uicontrol('Style','pushbutton',...
'Position',[225 20 50 30],...
'BackgroundColor',[0.8 0.8 0.8],...
'String','OK',...
'FontName','Arial',...
'FontSize',10,...
'Tag','fuerza_boton',...
'Callback',@salir);
data_sol=guihandles(hfig);
guidata(hfig,data_sol)
end

function salir(objeto,~)
data=guidata(objeto);
set(data.solucion,'Visible','off')
end
```

