



Universidad
Zaragoza

Proyecto Fin de Carrera

DESARROLLO DE UN MODELO ACC PARA UN SOFTWARE DE SIMULACIÓN DE DINÁMICA DE VEHÍCULOS

Autor

HÉCTOR BENEDÍ BOZALONGO

Director y ponente

ARNO EICHBERGER
JAVIER CIVERA

ESCUELA DE INGENIERÍA Y ARQUITECTURA

2014

Resumen

El principal objetivo de este proyecto es el desarrollo de un modelo ACC (Automated Cruise Control) dentro de un programa de simulación de la conducción de un productor comercial de software. Este software será utilizado en un simulador de conducción cuyo objetivo es investigar la interacción del conductor con el sistema ACC para diferentes situaciones de conducción.

El objetivo de este PFC es implementar el modelo ACC, que consiste en un sensor radar, un control de velocidad y un control de distancia. Además un modelo de sistema de frenada de emergencia ha sido implementado para actuar en situaciones críticas, es el denominado AEB (Autonomous Emergency Braking).

Cuando no hay ningún objeto delante, el ACC mantiene una velocidad constante que es establecida por el conductor (del mismo modo que lo hace un cruise control estandar). Sin embargo, cuando se detecta un vehículo más lento circulando en el mismo carril, el sistema frena automáticamente y mantiene una cierta distancia. Si este vehículo deja de estar presente, por ejemplo porque cambia de carril, el sistema vuelve a acelerar otra vez hasta que alcanza la velocidad de crucero establecida.

El objetivo del sensor radar es detectar vehículos y su velocidad relativa basándose en el rango geométrico del radar, así como los objetos relevantes, por ejemplo el vehículo más cercano de aquellos que circulan en el mismo carril.

Por otro lado el AEB es un sistema de ayuda al conductor en situaciones de emergencia. Dicho sistema también hace uso del sensor radar para detectar e identificar situaciones de peligro, y en caso necesario aplicar los frenos para evitar una colisión.

Agradecimientos

En primer lugar quiero expresar mi más sincero agradecimiento a Arno Eichberger por aceptarme para realizar este proyecto, y por el aporte incalculable que esta experiencia ha supuesto en mi formación.

Me gustaría agradecer de manera especial y sincera a Cornelia Lex, por el apoyo, tiempo y dedicación que me ha brindado, por su capacidad de guiar las ideas en la dirección correcta y por sus atentas y rápidas respuestas.

También quisiera dar las gracias a Javier Civera, por el apoyo a la hora de elegir un proyecto y el posterior seguimiento en sus fases de desarrollo.

A Daniel Hammer, por la ayuda facilitada en las fases de diseño y ajuste de los modelos realizados y por su inagotable paciencia.

También quiero agradecer a todas las personas más cercanas que me han aguantado en los momentos difíciles, por darme siempre los mejores consejos y por su gran habilidad de hacerme siempre pensar en positivo.

Y por supuesto, el agradecimiento más profundo y sentido es para mi familia. A mi madre y a mi padre, por su apoyo incondicional y plena confianza que en mí han depositado. A mi hermano, por su entusiasmo y preocupación por mis progresos.

Por ellos, y para ellos.

Índice general

Resumen	i
Agradecimientos	iii
Índice general	vi
Símbolos	vii
1. Introducción	1
1.1. Descripción general	1
1.2. Simulador de conducción	2
1.3. Software utilizado: VSM	3
1.4. Trabajo realizado	4
2. Sensor radar	7
2.1. Rango geométrico	7
2.2. Identificación de objetos en rango	7
2.2.1. Rotación de vehículos	9
2.2.2. Matriz Cars	9
2.2.3. Matriz Shadows	11
2.3. Cálculo del punto superficie	11
2.4. Identificación del vehículo Target	13
3. Sistema ACC	15
3.1. Control de velocidad	16
3.2. Control de distancia	17
3.2.1. Contribuciones para el control de distancia	17
3.2.2. Bloque Function	19
3.2.3. Límites de aceleración	21
3.2.4. Control de deceleración	21
3.2.5. Cálculo del offset	26
3.3. Control de aceleración	30
3.4. Resultados ACC	31
4. Sistema AEB	33
4.1. Cómo funciona	33
4.2. Time to collision	34

4.3. Modo Ciudad	35
4.4. Modo Inter-Urbano	35
4.5. Resultados AEB	36
5. Conclusiones	39
Índice de figuras	I
Índice de cuadros	V
Bibliografía	VII
A. Apéndice	IX
B. Apéndice	XI
C. Apéndice	XIII
D. Apéndice	XIX
E. Apéndice	XXI
F. Apéndice	XXVII
G. Apéndice	XXXI
H. Apéndice	XXXV
I. Apéndice	XXXVII
J. Apéndice	XLVII

Símbolos

Variables

$Ident$	Número de identificación de cada vehículo
$line$	Carril por el que el vehículo circula
xx	Coordenada x del centro de gravedad de cada vehículo en relación a un sistema de coordenadas globales
yy	Coordenada y del centro de gravedad de cada vehículo en relación a un sistema de coordenadas globales
v	Módulo de la velocidad del vehículo en relación al sistema de coordenadas globales
h	Ángulo en el que está orientado el vehículo
d	Distancia entre la posición del radar y el vehículo
φ	Ángulo en el que se encuentra el vehículo
v_m	Módulo de la velocidad relativa del vehículo en relación con el vehículo Ego
γ_{vel}	Ángulo de la velocidad relativa
φ_{min}	Mínimo ángulo que crea una sombra
φ_{max}	Máximo ángulo que crea una sombra
d_s	Distancia donde empieza la sombra
d_1	Distancia a reducir con el vehículo Target
d_{soll}	Distancia a mantener con el vehículo Target
x_{rel}	Distancia actual al vehículo Target
K_g	Término proporcional para la velocidad relativa
τ_d	Término proporcional para la distancia d_1
Δv	Suma de las dos contribuciones para el control de distancia
τ	Time gap seleccionado por el conductor
d_{offset}	offset a añadir en el controlador de distancia
$mode$	modo del sistema ACC (estándar, dinámico o comfortable)
V_{target}	Velocidad del vehículo target
V_{Ego}	Velocidad del vehículo Ego
V_{rel}	Velocidad relativa velocity del Target en relación con el Ego
V_{EgoSim}	Velocidad del Ego simulada
$V_{EgoReal}$	Velocidad del Ego real
$doubleSteep$	Define si un modo tiene una o dos pendientes

$sw2$	Valor que define si el controlador de deceleración está en on o en off
k_{down}	Pendiente de deceleración
$k_{down.Aprox}$	Pendiente inicial de deceleración
k_{dist}	Distancia a la que se produce el cambio en la pendiente de deceleración
k_{inner}	Parámetro utilizado para definir la pendiente de las aceleraciones
k_{outer}	Parámetro utilizado para definir la pendiente de las aceleraciones fuera de los límites
a_{max}	Maximum positive acceleration
a_{Dmax}	Maximum negative acceleration
ttc	Time to collision
$diffV$	Diferencia entre las velocidades del Ego real y del simulado
$X_{relReal}$	Distancia relativa entre el Target y el Ego real
X_{relSim}	Distancia relativa entre el Target y el Ego simulado
$diffX$	Diferencia entre las distancias X_{relSim} and $X_{relReal}$
C_{13}	Parámetro 1 para la siguiente deceleración
C_{14}	Parámetro 2 para la siguiente deceleración
C_{df}	Término proporcional para la siguiente deceleración
k_p	Término proporcional para el control de aceleración
k_i	Término integral para el control de aceleración
k_{windup}	Parámetro para el diagrama del anti-windup
$mode_{AEB}$	Modo de operación del sistema AEB

Vectors

c	Matriz para la geometría del vehículo
$Cars$	Matriz que guarda la información de todos los vehículos
$Shadows$	Matriz que guarda la información de las sombras

1. Introducción

1.1. Descripción general

El proceso de automatización de los vehículos no es algo novedoso, desde hace tiempo se está invirtiendo mucho esfuerzo en sistemas que ayudan a la conducción, ya sea en momentos de emergencia o para situaciones normales.

El primer cruise control fue diseñado en 1948 por Ralph Teetor [8]. Este sistema regula la velocidad del vehículo para mantenerla constante en un valor establecido por el conductor. En 1958 Chrysler comercializó dicho invento bajo el nombre de Auto-Pilot, En la imagen 1.1 puede verse este primer modelo de cruise control. Numerosos sistemas han sido o se están desarrollando para mejorar la eficiencia, facilitar la conducción, o incrementar la seguridad de nuestras carreteras. Sistemas como el ESC (electronic stability control), el ABS (anti-lock braking system) o el AEB (advanced emergency braking) tienen como principal objetivo mejorar la seguridad y ayudar al conductor en momentos de emergencia. Sin embargo, el objetivo de otros sistemas es sustituir al conductor en algunas de sus funciones durante la conducción normal, y no en situaciones de emergencia. Algunos de estos sistemas son el CC (Cruise Control) mencionado anteriormente, o el ACC (Automated Cruise Control), el cual utiliza un sensor radar para detectar vehículos en la carretera y disminuir la velocidad en caso necesario.

De todos los sistemas descritos con anterioridad, los más novedosos son aquellos sobre los que menos estudios y tests se han realizado. La respuesta y comportamiento que las personas tienen ante estos sistemas es de interés para la industria automovilística a la hora de optimizar los parámetros de los nuevos modelos.

Este proyecto fin de carrera tiene como objetivo realizar un modelo de sensor radar, el ACC y el sistema de frenada de emergencia AEB. Estos sistemas serán instalados en un simulador de conducción, el cual está siendo construido por el Instituto de Ingeniería de Automoción de la universidad TU Graz en Austria, en colaboración con algunas compañías de automóviles como AVL.

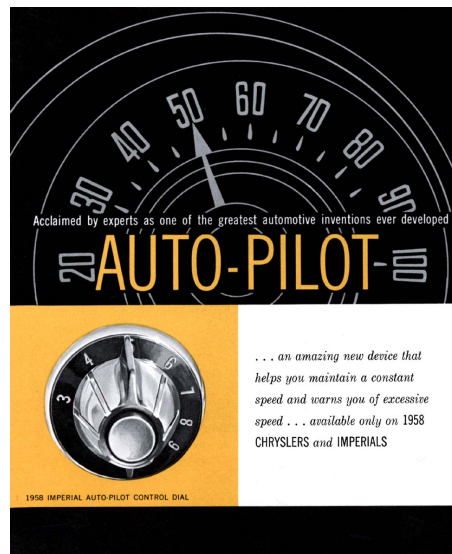


Figura 1.1.: Primer cruise control desarrollado por la compañía Chrysler en 1958, basado en el invento anterior de Ralph Teetor.

1.2. Simulador de conducción

Según estudios realizados [5], más del 90 por ciento de todos los accidentes de tráfico son causados por error humano. Investigaciones detalladas de esos casos muestran una diferencia significativa con respecto a sexo y edad.

Por lo tanto es crucial adaptar los sistemas de asistencia avanzada al conductor (ADAS: Advanced Driver Assistance System) a toda esa variedad de conductores que pueda haber. Diferentes estudios muestran que por ejemplo en los giros a la izquierda en intersecciones con señal de stop, las mujeres tienden a realizar el giro cuando el hueco es mayor que en el caso de los hombres, los cuales tienden a correr mayores riesgos, [9]. El posible error humano tiene que tenerse en cuenta a la hora de diseñar la estrategia de intervención del ADAS. De esta forma se evitan reacciones adversas en situaciones de peligro, lo que contribuye a una mayor aceptación de este sistema por parte del conductor.

El proyecto del simulador pretende realizar investigaciones sistemáticas del comportamiento humano en situaciones normales y críticas. Aspectos como género, edad y variedad de escenarios de conducción se estudiarán en un entorno controlado que a la vez permite test repetitivos bajo las mismas condiciones. Para una mayor realidad y fiabilidad de los resultados se instalarán unas novedosas pantallas de visualización 3D sin gafas. La reacción de los voluntarios será registrada con apropiados sistemas de medida que, combinados con cuestionarios psicológicos que serán rellenados por los voluntarios, proporcionarán resultados lo más fiables posible para su posterior análisis.

Para verificar que el simulador se aproxima suficientemente a la realidad se realizarán

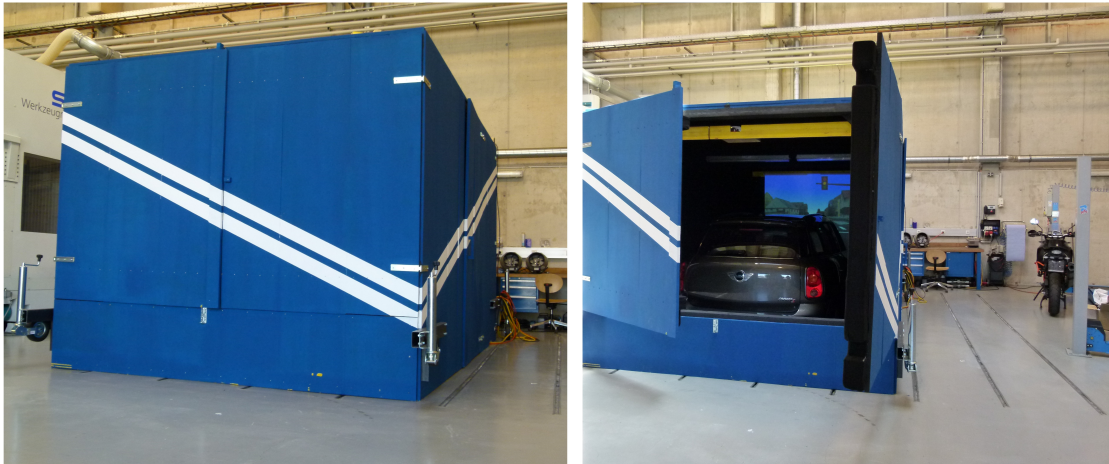


Figura 1.2.: Simulador de conducción en fase avanzada de construcción.

unas pruebas con conductores. De esta forma podrá determinarse que los datos obtenidos son fiables y así, será posible realizar experimentos variando la estrategia de intervención del ADAS. Se evaluará la interacción del conductor con los sistemas de ayuda en la conducción y determinar qué estrategia es la óptima.

En la imagen 1.2 se muestra el conjunto del proyecto del simulador. Consiste en un coche modelo Mini CountryMan que está instalado dentro de una estructura que lo aísla del ruido y la luz, ya que un sistema acústico se incluirá en el simulador para hacerlo más realista. Por otro lado en la imagen 1.3 pueden verse las pantallas que se van a instalar delante del vehículo, las frontales son pantallas normales de televisión de 55 pulgadas cuya única condición es la de tener un marco pequeño, y las instaladas en las ventanillas son monitores de ordenador de 23 pulgadas.

1.3. Software utilizado: VSM

El software utilizado se denomina VSM (Vehicle Simulation Model [4]). Es un simulador de la dinámica del vehículo en tiempo real que pertenece a la empresa AVL. Este software tiene un modelo muy completo del vehículo, tanto que es utilizado por equipos de Fórmula 1 para estudiar y optimizar los parámetros del coche [3].

Fue desarrollado por Ferrari en colaboración con AVL, y hoy en día, se ha convertido en una herramienta indispensable a la hora de la puesta a punto del vehículo. El programa incluye la simulación de factores tan variados como la resiliencia de los neumáticos o el consumo de combustible. Cuenta también con una amplia variedad de superficies con diferentes agarres dependiendo de la calidad del asfalto (pudiéndose incluir irregularidades de la carretera, como por ejemplo baches). Además, permite la simulación para



Figura 1.3.: Ensayo con un proyector de cómo y dónde se formará la imagen (izquierda), estructura y pantallas 3D que se montarán enfrente del simulador y en ambas ventanillas para crear el efecto 3D y efecto de velocidad (derecha).

condiciones de seco y mojado.

VSM ha sido utilizado en este proyecto fin de carrera para ajustar los parámetros de los sistemas ACC y AEB. En lugar de realizar los ajustes mediante técnicas más comunes, tales como el lugar de las raíces o análisis frecuencial, se ha optado por un ajuste paramétrico mediante simulación. Las capacidades y precisión del software utilizado justifican esta opción, ya que los resultados son, de esta manera, más cercanos a la realidad.

En la figura 1.4 puede verse una imagen del programa VSM, donde se aprecian valores como las revoluciones del motor, velocidad a la que se circula, marcha, o las fuerzas que se generan sobre los neumáticos.

1.4. Trabajo realizado

El presente proyecto fin de carrera tiene como objetivo diseñar e implementar los sistemas incorporados en el vehículo tales como el ACC, el AEB y el sensor radar.

El ACC (automated cruise control) es un sistema que mantiene la velocidad del vehículo constante e igual a un valor previamente establecido por el conductor, al mismo tiempo que reduce dicha velocidad en caso de que otro vehículo más lento circule delante. Este sistema tiene unos parámetros que el conductor puede modificar dependiendo de la respuesta que se espera por parte del sistema ACC. Por ejemplo, se puede variar la distancia de seguridad que se mantiene con el vehículo que circula delante, así como establecer unas aceleraciones y deceleraciones más o menos bruscas, variando así el confort en la

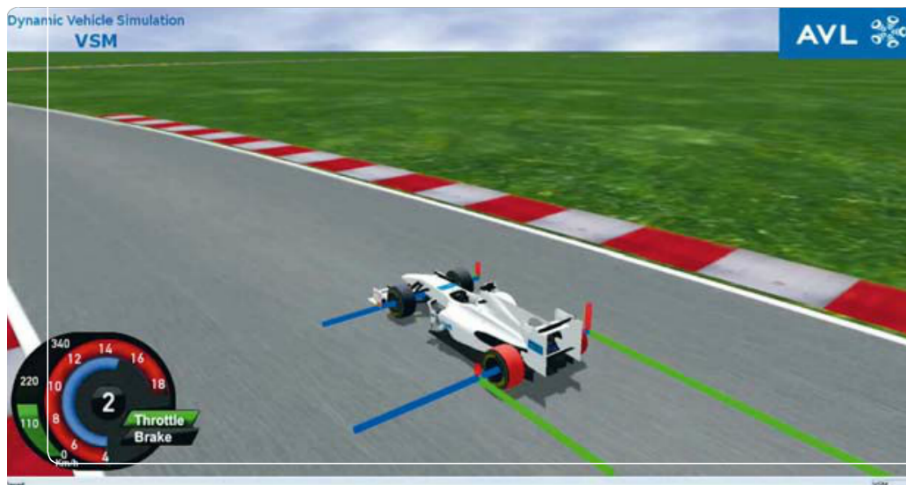


Figura 1.4.: Imagen extraída de la revista de AVL [4] que muestra una imagen el software utilizado: VSM Dynamic Real Time Vehicle Simulation Model.

conducción.

Dicho sistema necesita de un sensor radar, el cual le proporciona la información necesaria sobre otros vehículos en la carretera (en caso de que los haya), acerca de su velocidad y la distancia a la que se encuentran. De esta manera el ACC puede ajustar la velocidad en carretera para mantener una distancia de seguridad.

En el caso de que un objeto aparezca de repente delante del vehículo, como un peatón, una bici, u otro vehículo en general, el radar detecta la velocidad y la distancia a la que se encuentra. El sistema AEB (advanced emergency braking) calcula una variable que indica si la colisión es inminente o no, y en caso de que sí lo sea el sistema se activa. Una vez activado, primero produce una señal acústica y luminosa para avisar al conductor del peligro, y en el caso de que no responda se aplicarán los frenos para evitar la colisión.

El conjunto del proyecto fin de carrera se ha realizado utilizando Matlab y Simulink. Se han desarrollado dos modelos, uno exclusivo para el radar y otro en el que se incluyen el ACC y el AEB.

2. Sensor radar

El sensor radar es uno de los sistemas más importantes de un vehículo equipado con ACC, ya que proporciona información sobre la distancia y la velocidad de cada vehículo dentro de su rango de acción, de esta manera el sistema reducirá la velocidad al acercarse a otro vehículo y acelerará de nuevo a la velocidad deseada cuando el tráfico lo permita.

Dicho sensor está instalado en el denominado vehículo Ego. De todos aquellos vehículos que detecte el radar sólo aquel que se encuentra más cerca y circulando en el mismo carril es de interés para el sistema, este vehículo se denomina vehículo Target. El código completo utilizado en el sensor radar se muestra en el apéndice I.

2.1. Rango geométrico

El rango geométrico es la combinación del alcance y la abertura del radar a la que es capaz de detectar vehículos, este rango geométrico se muestra en la figura 2.1. El radar simulado tiene un rango de longitud de unos 100 metros y un ángulo de 30 grados.

2.2. Identificación de objetos en rango

Para la identificación de objetos en el rango el input que tenemos es una matriz de vehículos. El programa necesita toda la información guardada en dicha matriz, ya que

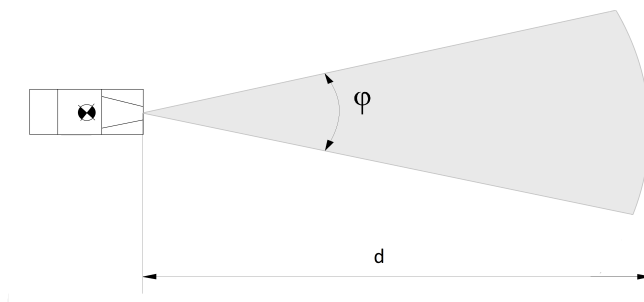


Figura 2.1.: Rango geométrico del radar

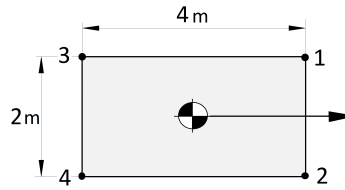


Figura 2.2.: Esquinas del vehículo

es un modelo simplificado.

Uno de los inputs es la velocidad del vehículo, la cual podría ser calculada a partir de los datos de la posición para dos instantes de tiempo diferentes: $V = (x_2 - x_1)/(tiempo)$. Lo mismo ocurre con el ángulo en el que cada vehículo está orientado, o incluso el carril en el que el vehículo se encuentra, ya que podrían ser calculados de la misma manera. Sin embargo para esta aplicación no es necesario, y de esta manera el modelo es menos complicado. Por lo tanto la matriz input de vehículos recoge toda esta información:

- *Ident*: número de identificación del vehículo. Este número puede variar en función del tipo de vehículo que se detecte, coche, camión, peatón, etc.
- *Line*: representa el carril de la carretera donde cada vehículo se encuentra. Sólo los vehículos en el mismo carril que el coche Ego serán de interés para el sistema.
- *xx, yy*: posición del centro de gravedad del vehículo en el espacio, con relación a un sistema de coordenadas global.
- *h*: el heading del vehículo es el ángulo en el que está orientado referido a un sistema de coordenadas global.
- *v*: velocidad del vehículo también en relación a un sistema de coordenadas global.
- *c*: la geometría del vehículo se expresa por las coordenadas locales de cada esquina (en relación con el centro de gravedad de ese vehículo). Por ejemplo un coche de 4 metros de largo y 2 metros de ancho (Figura: 2.2) tendrá la matriz *c* que define su contorno (eq 2.1):

$$\mathbf{c} = \begin{bmatrix} 2 & 2 & -2 & -2 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad (2.1)$$

En nuestro experimento tenemos un máximo de 10 vehículos diferentes que rodean el vehículo Ego, todos ellos con los valores explicados anteriormente. El input para la función de Simulink se muestra en la figura 2.3 donde podemos ver todos estos valores en la segunda parte de la ecuación "función [output] = FCN (input)".

Con el fin de determinar qué vehículos son detectados el algoritmo realiza dos bucles. El objetivo es determinar cuáles de estos vehículos están dentro del rango geométrico exclu-

```
function [Ident1a,d1,a1,s1,as1,T1, Ident2a,d2,a2,s2,as2,T2, Ident3a,d3,a3,s3,as3,T3,...
Ident4a,d4,a4,s4,as4,T4, Ident5a,d5,a5,s5,as5,T5, Ident6a,d6,a6,s6,as6,T6,...
Ident7a,d7,a7,s7,as7,T7, Ident8a,d8,a8,s8,as8,T8, Ident9a,d9,a9,s9,as9,T9,...
Ident10a,d10,a10,s10,as10,T10] =...
fcn(num,frontCar,Ident,line,xx,yy,h,ap,l,v,Ident1,line1, x1,y1,h1,v1,c1,...
Ident2,line2,x2,y2,h2,v2,c2, Ident3,line3,x3,y3,h3,v3,c3,Ident4,line4,x4,y4,h4,v4,c4,...
Ident5,line5,x5,y5,h5,v5,c5,Ident6,line6,x6,y6,h6,v6,c6,Ident7,line7,x7,y7,h7,v7,c7,...
Ident8,line8,x8,y8,h8,v8,c8,Ident9,line9,x9,y9,h9,v9,c9,Ident10,line10,x10,y10,h10,v10,c10)
```

Figura 2.3.: Input para la función de simulink

yendo aquellos que se encuentren en la sombra de otro vehículo. Para ello se necesitan dos matrices: Cars y Shadows (explicadas en las secciones 2.2.2 y 2.2.3). Sin embargo, en primer lugar tenemos que hacer una transformación de coordenadas para referir cada vehículo a un sistema de coordenadas solidario al vehículo Ego.

2.2.1. Rotación de vehículos

El input que el radar recibe es la matriz descrita en el capítulo 2.2, en la cual todos los datos están referidos a un sistema de coordenadas global, por eso es necesario rotar todos los vehículos para transformar esa matriz a coordenadas referidas al vehículo Ego.

Esto se consigue mediante dos rotaciones, la primera consiste en rotar únicamente los centros de gravedad de cada vehículo y en la segunda se rotan todas y cada una de las esquinas de los vehículos. De esta forma al final todos los vehículos quedan orientados según se ven desde la posición y el ángulo que tiene el vehículo Ego.

En la imagen 2.4 puede verse un ejemplo gráfico de cómo se realiza la rotación de los vehículos, el coche número 1 es el original situado en el sistema de coordenadas global (ejes negros) y el coche 2 es cómo queda cuando los ejes relativos al vehículo Ego (ejes azules) se toman como referencia absoluta. Para ver un ejemplo de cálculo ver el apéndice F.

2.2.2. Matriz Cars

Esta matriz se utiliza para guardar la información de todos los vehículos. Como se explica en la sección 2.2 el input está referenciado a un sistema de coordenadas global, cada valor se transforma a un sistema de coordenadas relativo y la matriz Cars guarda esta información relativa al vehículo Ego.

Es importante tener en cuenta que el programa no detecta un vehículo con la geometría que se muestra en la figura 2.2, sino un punto determinado de la superficie (figura 2.8). Este punto se denomina punto de la superficie y su cálculo se explica en la sección 2.3.

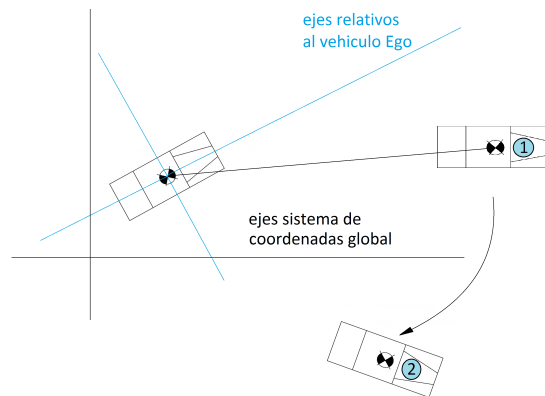


Figura 2.4.: Rotación de vehículos

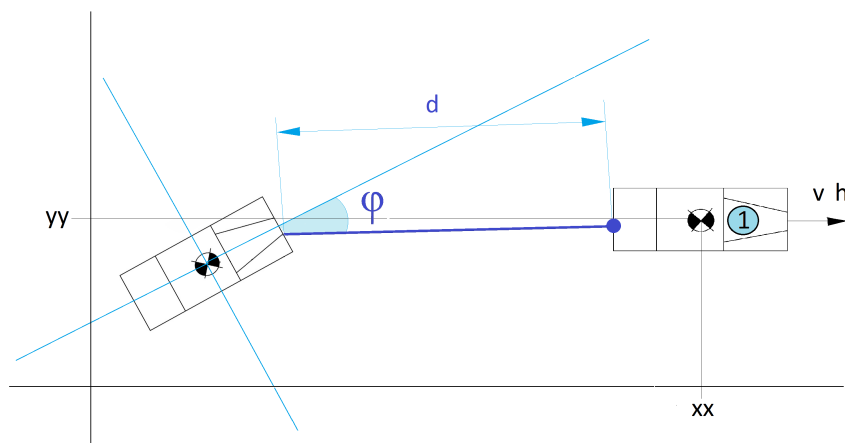


Figura 2.5.: Ejemplo con un solo coche

En la figura 2.5 vemos en letras negras los valores de input (coordenadas absolutas) y en letras azules los valores en coordenadas relativas. Dichos valores son los que se guardan en la matriz Cars, ya que cuando el radar detecta un vehículo los únicos datos relevantes son los referidos al punto de la superficie: la distancia a la que se encuentra y el ángulo, además de la velocidad relativa del vehículo Target con respecto al Ego.

Para ver más información sobre la manera en que esta matriz es creada, y ver en detalle cómo se han realizado los cálculos, véase el apéndice G.

2.2.3. Matriz Shadows

En esta sección se explica cómo se implementa el efecto sombra, lo que representa el hecho de que los vehículos no son transparentes para el radar. Si por ejemplo tenemos un vehículo situado a 20 metros en un ángulo de 10 grados y a otro vehículo a 30 metros en el mismo ángulo, el radar sólo detecta el vehículo más cercano, y el otro "desaparece" detrás del primero.

La matriz Shadows se crea con el fin de eliminar aquellos vehículos que, estando dentro del rango geométrico, el radar no los detecta por estar en la sombra de otro vehículo. Se crea en el primer bucle que el programa realiza y consta de dos ángulos y una distancia a partir de la que se crea la sombra. Por lo tanto esta matriz tendrá tres columnas que representan d_s , φ_{min} y φ_{max} .

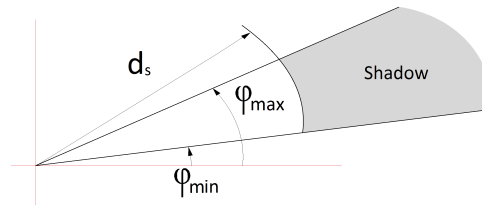


Figura 2.6.: Parámetros para definir una sombra

Para crear una sombra a partir de los datos del vehículo, primero se calculan los ángulos a los que se encuentran las cuatro esquinas del vehículo y se seleccionan el mayor y el menor de ellos. Estos dos ángulos son φ_{max} y φ_{min} respectivamente. Por último se selecciona la distancia al segundo corner más cercano como la variable d_s , y de esta forma queda creada la sombra.

Hay que decir que en este punto estamos considerando una aproximación, ya que la distancia a partir de la que se crea la sombra no es exactamente la del segundo corner más cercano. En la figura 2.7 se muestra la sombra real que cada vehículo crea (gris claro) y la sombra adicional creada en nuestro programa (gris oscuro).

Para más detalles sobre la matriz Shadows véase el apéndice H.

2.3. Cálculo del punto superficie

Un radar real no detecta una de las esquinas de un vehículo o el centro de gravedad del mismo sino una superficie sólida, por lo tanto, la simulación del radar tratará de imitar lo que el radar real hace, detectando el punto superficie tal y como se muestra en la figura 2.8 donde el punto rojo representa el punto detectado por el radar.

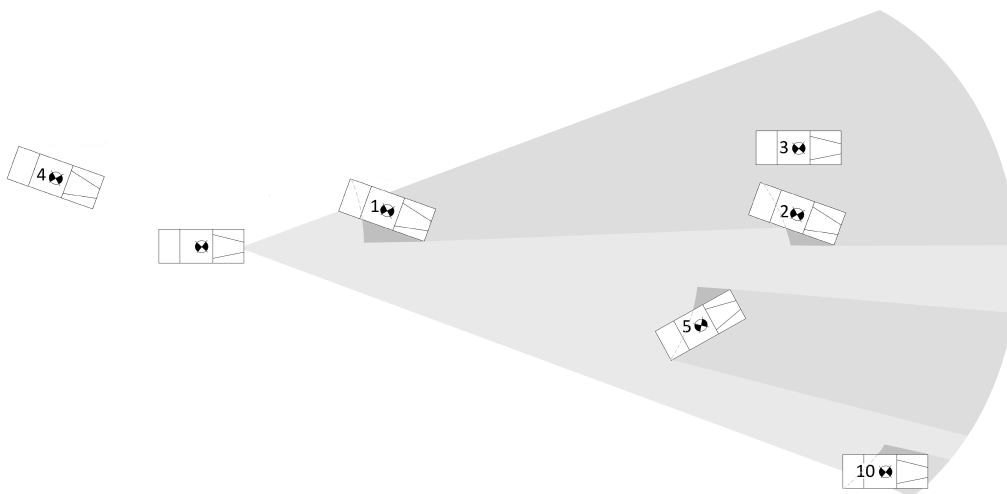


Figura 2.7.: Sombras reales creadas por cada vehículo (gris claro), y sombras adicionales que el simulador del radar crea (gris oscuro).

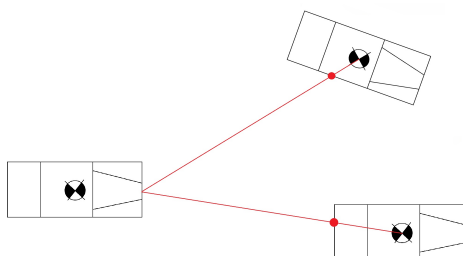


Figura 2.8.: Punto de la superficie detectado

El punto superficie es aquel punto en la superficie del vehículo que resulta de la intersección de dos líneas, la línea desde el radar del vehículo Ego hasta centro de gravedad del vehículo Target y la línea que define la superficie en sí del vehículo. Puesto que hay cuatro superficies diferentes en cada vehículo habrá cuatro puntos posibles, pero sólo uno de ellos será el correcto (figura 2.9).

Las cuatro líneas que definen la superficie del vehículo son las que conectan las esquinas 1-2, 2-4, 3-4 y 1-3 (esquinas del vehículo en la figura 2.2).

Con el fin de seleccionar cuál de los 4 puntos es el correcto, en primer lugar el programa de seleccionar sólo aquellos dos puntos cuya coordenada x se encuentra entre las coorde-

nadas x de las esquinas que se utilizan para crear la línea. En realidad significa aquellos dos puntos que realmente están en la superficie del vehículo (puntos b y c).

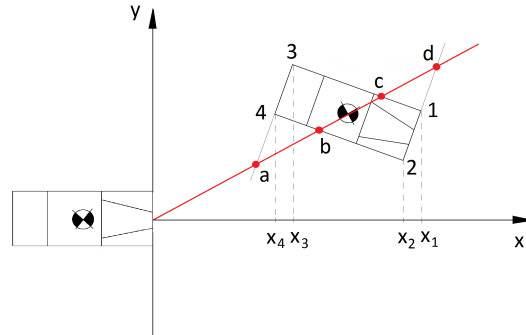


Figura 2.9.: Cuatro posibles soluciones a las ecuaciones de intersección de líneas, sólo uno de ellos es la solución correcta, en este caso el punto b .

Por ejemplo la línea creada por las esquinas 1 y 2 cruzan la línea roja en el punto d , la coordenada x de ese punto no se encuentra entre las coordenadas x de los puntos 1 y 2 sino que es mayor que ambas (eq 2.2). Lo mismo ocurre para el punto a , donde la coordenada x de a no está entre las coordenadas x de 3 y 4 sino que es más pequeño que ambos (eq 2.3).

Por el contrario, b y c tienen coordenadas x que están entre las coordenadas x de los puntos utilizados para crear sus líneas, 2-4 y 1-3 respectivamente (eq 2.4 y 2.5). Así que estos dos puntos son los posibles resultados para el punto superficie. En el último paso del programa sólo debe seleccionar el punto más cercano de estos dos puntos, que en este caso es el punto superficie b .

$$x_d > x_1 > x_2 \quad (2.2)$$

$$x_a < x_4 < x_3 \quad (2.3)$$

$$x_4 < x_b < x_2 \quad (2.4)$$

$$x_3 < x_c < x_1 \quad (2.5)$$

2.4. Identificación del vehículo Target

El vehículo Target es aquel que es de interés para el controlador a distancia del sistema ACC.

El radar detectará todos los vehículos dentro del rango geométrico y no en la sombra, sin embargo, el controlador de distancia sólo toma en consideración un coche, y los datos de

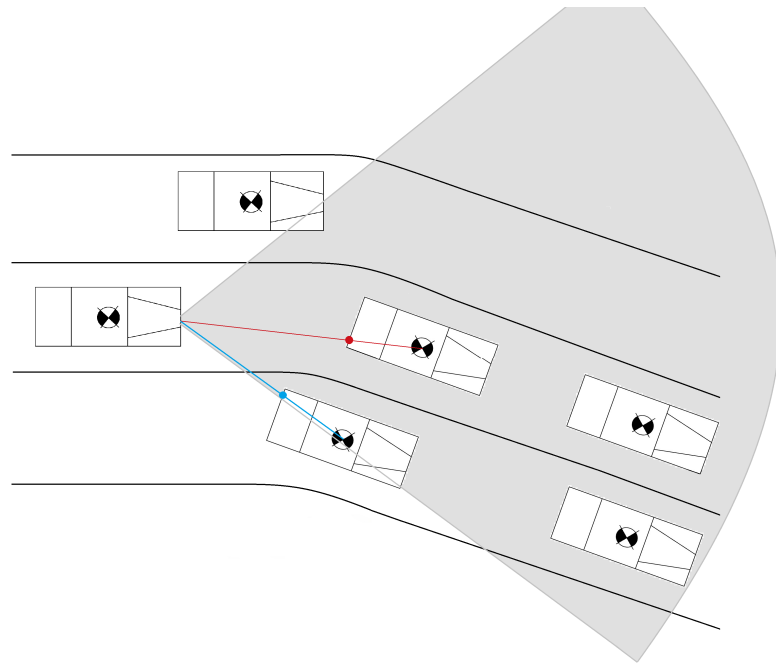


Figura 2.10.: Coches circulando en diferentes carriles de la carretera.

ese coche serán utilizados para controlar la distancia y la velocidad con el fin de mantener una distancia de seguridad. Por eso, la identificación adecuada del coche Target es clave para el correcto comportamiento del controlador de distancia.

La identificación del vehículo Target se hace eligiendo el vehículo más cercano de entre todos aquellos que circulen en el mismo carril. Cada vehículo tiene un input llamado *line* que guarda el número del carril donde este vehículo está circulando. Por lo tanto lo único que el programa debe hacer es elegir cuál de esos vehículos está más cerca (Fig. 2.10).

3. Sistema ACC

El sistema ACC que se va a implementar es un modelo que simula el Automated Cruise Control instalado en el Audi A6. Consiste en dos controladores diferentes, el controlador de distancia y el de velocidad (que se explica en las secciones 3.1 y 3.2).

El controlador de velocidad recibe un valor establecido por el conductor y lo alcanza sin importar qué otros obstáculos puedan estar en la carretera. Por otro lado el controlador de distancia toma en consideración la presencia de otros vehículos más lentos y mantiene una cierta distancia de seguridad.

Cada control demandará una aceleración diferente para lograr sus propios requisitos. Por ejemplo, en el caso mostrado en la figura 3.1 el vehículo Ego tiene una velocidad de cruceo demandada de 120 km/h, sin embargo está conduciendo a 80 km/h, por lo que el control de velocidad requerirá una aceleración positiva con el fin de llegar al valor de 120 km/h. Por otro lado, el control de distancia detecta un vehículo dentro del rango del radar por lo que la aceleración que demanda el control de distancia es menor, ya que su objetivo es mantener la distancia de seguridad.

El sistema ACC elige el valor mínimo de estas dos aceleraciones y lo pasa al control de aceleración (sección 3.3) que producirá una aceleración entre los posibles valores del vehículo. El modelo de Simulink del sistema ACC se puede ver en la figura 3.2 donde el controlador de distancia, velocidad y aceleración están pintados en verde, amarillo y azul respectivamente. El apéndice J muestra imágenes mas concretas y detalladas de los modelos implementados en Simulink.

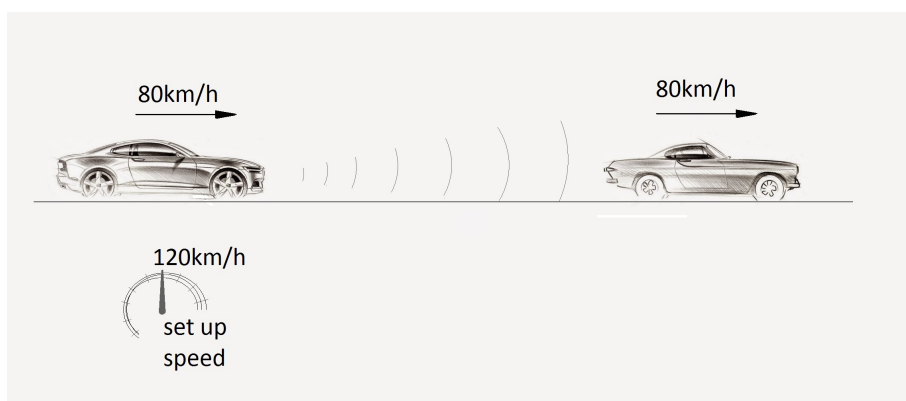


Figura 3.1.: Control de velocidad y distancia actuando al mismo tiempo.

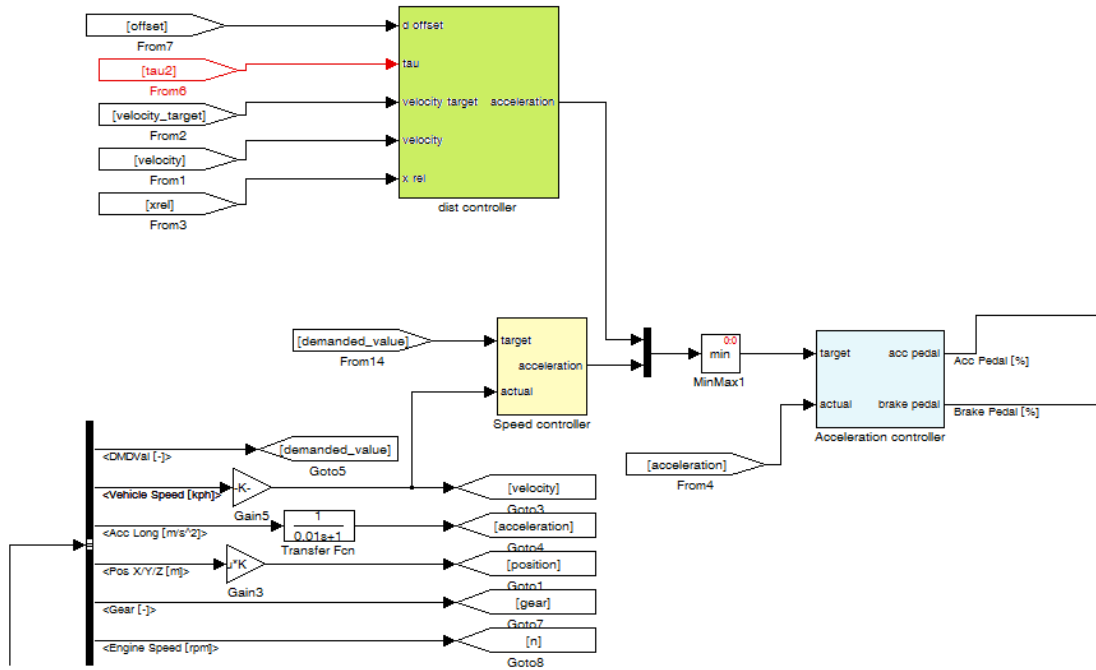


Figura 3.2.: Modelo de simulink con el control de distancia (verde), control de velocidad (amarillo) y control de aceleración (azul)

3.1. Control de velocidad

El control de velocidad tiene dos inputs diferentes (velocidad target y velocidad actual). Se calcula el error entre ambos y el output es la aceleración que este control demanda para alcanzar el valor target de velocidad.

El control es simplemente proporcional, un tipo de control más complicado no es necesario puesto que el error en estado estacionario es igual a cero. Esto es debido a que el software de dinámica del vehículo ya tiene un comportamiento integral, ya que la salida del controlador es una aceleración y ésta se convierte en una velocidad por integración. Por lo tanto, sólo necesitamos un término proporcional para controlar el sistema.

Con el fin de seleccionar un valor para la constante P se realizaron numerosas simulaciones. Como condición se exige una sobreoscilación igual a cero, además de un tiempo de respuesta alrededor de los 12 segundos para un escalón de 30 km/h. El experimento realizado establece una velocidad inicial de 66 km/h, en ese momento el conductor demanda una velocidad de cruce de 96 km/h. Hay que añadir que las aceleraciones producidas deben estar entre los límites de comfort del conductor, por lo que se establece una aceleración máxima de $2,5 \text{ m/s}^2$.

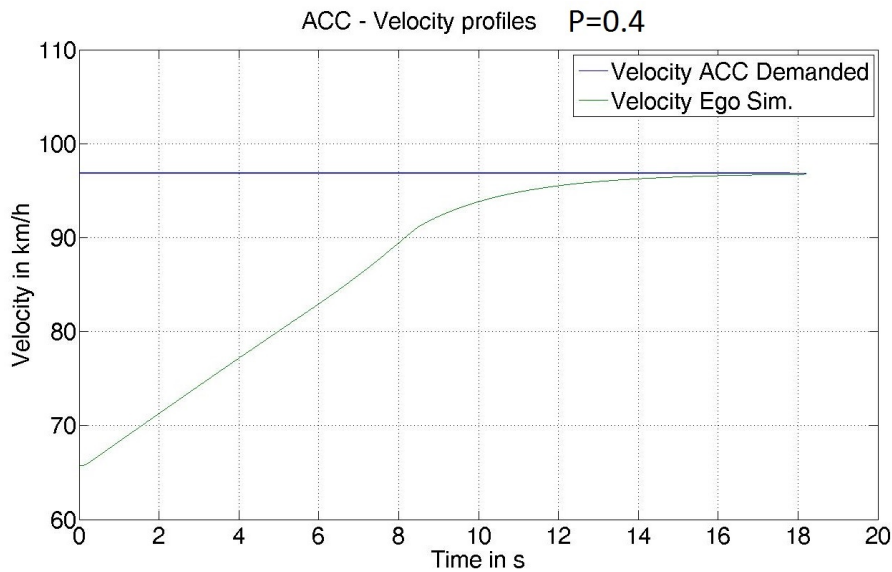


Figura 3.3.: Respuesta del sistema para un escalon de 30km/h.

Tras realizar las simulaciones se determinó que el valor óptimo para el controlador es $P=0,4$. El tiempo de respuesta que se obtiene es 11.7 segundos con una sobreoscilación nula. Para obtener más información sobre el comportamiento del sistema para diferentes valores de P se puede ver el apéndice A

3.2. Control de distancia

El controlador de distancia puede ser dividido en 3 partes diferentes. La primera de ellas se refiere a las distintas contribuciones que se tienen en consideración para el cálculo de la distancia (sección 3.2.1). La segunda parte es el "function blok" que se utiliza para calcular la aceleración resultante (sección 3.2.2), y la tercera son los límites de aceleración establecidos en función del modo que esté seleccionado (sección 3.2.3).

3.2.1. Contribuciones para el control de distancia

El controlador de distancia tiene dos partes diferentes. La primera se refiere a la distancia que queremos reducir entre el Target y el vehículo Ego (d_1), y la segunda se refiere a la velocidad relativa del vehículo Target. El esquema del modelo de Simulink se muestra en la figura 3.4.

La importancia de estas dos contribuciones se rige por dos parámetros K_g y τ_d . Mediante la variación de estos dos parámetros se puede hacer que el sistema reaccione de maneras diferentes. Por ejemplo, si el vehículo Ego está conduciendo a 100 km/h, y hay un

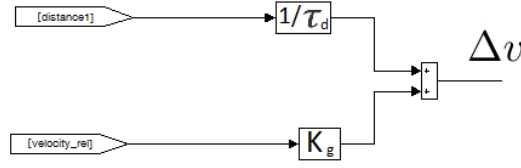


Figura 3.4.: Esquema de las dos contribuciones que intervienen en el control de distancia.

vehículo Target conduciendo a 60km/h a una distancia de 150 metros, en un primer momento el valor Δv es positivo, esto significa que el sistema demanda un valor positivo de aceleración para mantener la velocidad actual. El vehículo comenzará a frenar cuando este valor se convierta en negativo. Por ejemplo si suponemos que queremos mantener una distancia de 15 metros con el coche de delante, y solo queremos empezar a frenar cuando $x_{rel} = 100$ metros, en ese momento $d_1 = 85$ metros. Por lo tanto la relación entre $K_{ganancia}$ y τ_d tiene que ser tal que haga que Δv cambie a negativo en ese preciso momento (ecuación 3.1). Un valor típico para K_{gain} es 0.9, por lo que el valor de τ_d que cumple lo anterior es 8.5, de esta forma la frenada empezará justo cuando la distancia al coche de delante sea 100m (ecuación 3.3).

$$\Delta v = d_1(1/\tau_d) + V_{rel}(K_g) = 0 \quad (3.1)$$

$$85(1/\tau_d) + (60 - 100)/3,6(K_g) = 0 \quad (3.2)$$

$$85(1/8,5) - 11,1(0,9) = 0 \quad (3.3)$$

Mediante el uso de estos dos valores podemos interferir directamente en algunos aspectos del sistema como la distancia a la que el vehículo comienza a frenar.

La distancia d_1 a la que se ha hecho referencia en el párrafo anterior, representa la diferencia entre la distancia que tenemos con el siguiente vehículo (x_{rel}) y el valor de referencia d_{soll} . Este valor representa la distancia que queremos mantener en el estado estacionario con el coche de delante (en el ejemplo anterior eran 15 metros). Esta relación se puede ver en la ecuación 3.4. Por lo tanto d_{soll} representa el valor de referencia de la distancia y este valor tiene dos contribuciones, la relativa al time gap $\tau \cdot V_{target}$ más la debida al offset (ecuación 3.5).

El **time gap** es una variable seleccionada por el conductor, el cual puede elegir entre cuatro modos diferentes, dependiendo de lo grande o pequeña que quiere que sea la distancia hasta el siguiente vehículo (tabla 3.1). Las unidades son segundos y este valor se convierte en una distancia al multiplicarlo por la velocidad del vehículo Target. El **offset** es un valor que no es constante, sino que cambia en función de la velocidad del vehículo y el modo seleccionado por el conductor. El cálculo se explica a continuación en la sección 3.2.5.

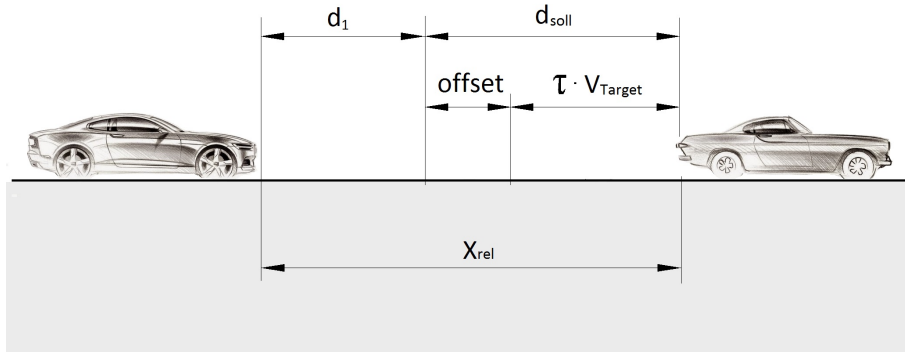


Figura 3.5.: Distancias existentes entre vehículos.

Además una descripción gráfica se muestra en la figura 3.5. Como ejemplo, si el vehículo Target circula a 30m/s , y el conductor en el vehículo Ego selecciona el modo 2, la distancia que la ACC mantendrá será : $30 \cdot 1,3 = 39$ metros , más la distancia debida para el offset.

Cuadro 3.1.: Valores de time gap obtenidos de la guía del Audi A6 [2]

τ Mode	Time [s]
1	1
2	1.3
3	1.8
4	2.3

$$d_1 = x_{rel} - d_{soll} \quad (3.4)$$

$$d_{soll} = \tau \cdot V_{target} + d_{offset} \quad (3.5)$$

Una vez que tenemos la distancia d_1 y la velocidad V_{target} , dos controladores proporcionales se utilizan para dar más importancia a un valor u a otro tal y como se explica en la sección 3.2.1. Ambos valores se suman dando lugar a Δ_v , que es el input para el bloque Function, explicado en la siguiente sección (3.2.2).

3.2.2. Bloque Function

Esta función se utiliza para determinar la aceleración que se va a demandar al controlador de aceleración que se encuentra aguas abajo de este bloque. Como anteriormente se ha descrito en la introducción del capítulo 3, este controlador recibe dos aceleraciones y elige la menor de ambas.

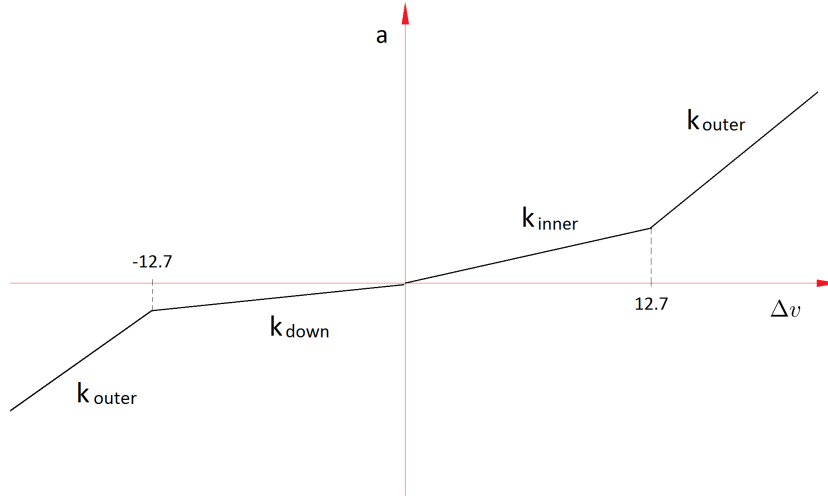


Figura 3.6.: Relación entre Δv y la aceleración que se produce, también pueden verse las diferentes k utilizadas.

El input utilizado para calcular la aceleración es el valor previamente calculado Δv (ecuación 3.1), esta variable se multiplica por una constante k que determina la aceleración resultante que es demandada. En un principio hay cuatro valores distintos de k dependiendo del input que se reciba. Si este valor es negativo o positivo, la k tiene un valor u otro, al igual que si es mayor que un límite establecido, el valor también cambia.

En las ecuaciones 3.6 hasta 3.9 se muestra el comportamiento y valores de k en función del input recibido Δv .

$$\Delta v < -12,7 \Rightarrow k = k_{outer} \quad (3.6)$$

$$-12,7 \leq \Delta v < 0 \Rightarrow k = k_{down} \quad (3.7)$$

$$0 \leq \Delta v \leq 12,7 \Rightarrow k = k_{inner} \quad (3.8)$$

$$\Delta v > 12,7 \Rightarrow k = k_{outer} \quad (3.9)$$

Todo esto se traduce en que el sistema presenta diferente comportamiento y diferentes valores de aceleración en caso de que se esté frenando o acelerando, y en qué medida se esté haciendo. En la imagen 3.6 se muestra cómo es la relación cualitativa entre Δv y la aceleración correspondiente.

Como hemos dicho con anterioridad, en principio hay cuatro valores diferentes de k aunque esto no es del todo cierto, ya que uno de los controles utilizados se encarga de variar el valor de k_{down} (la constante utilizada para las deceleraciones) para tratar

de ajustarse a las medidas reales lo más posible. Este comportamiento se explica en la sección 3.2.4.

3.2.3. Límites de aceleración

Aparte de todos los cálculos realizados hasta el momento, se establecen unos límites para la aceleración positiva y negativa con la intención de que el comportamiento del sistema no sea muy brusco.

Para ello se añade un bloque de saturación cuyos valores se introducen desde el fichero `init` de Matlab. El máximo de aceleración que se utiliza para todas las simulaciones es $a_{max} = 2,5m/s^2$, mientras que para las deceleraciones se introduce un nuevo concepto, el llamado *ttc* (time to colision) que se explica más adelante en la sección 4.2.

Este valor es básicamente el tiempo que tenemos antes de que se produzca una colisión, por lo tanto en el caso $ttc > 5.5$ segundos, el sistema se encuentra en una situación normal, de ahí que el valor límite de deceleración es $a_{max} = -1,3m/s^2$. En caso contrario, si el *ttc* es menor que 5.5 significa que el sistema está cerca de entrar en una situación de emergencia. Por lo tanto antes de activar dicha situación, la deceleración se hace de manera más agresiva: $a_{max} = -3,5m/s^2$. Si aun así no se consigue frenar lo suficiente entonces entra en juego el sistema de emergencia de frenada AEB explicado en el capítulo 4.

3.2.4. Control de deceleración

El motivo por el que se añade este controlador es debido a que las mediciones realizadas sobre vehículos reales sugieren que la k utilizada para las deceleraciones no es constante sino que varía dependiendo de otros parámetros.

Se han identificado dos posibles situaciones de deceleración, la primera es la deceleración de aproximación que se produce cuando nos aproximamos a un vehículo Target que acaba de aparecer en la carretera, y a la segunda se le llama deceleración de seguimiento, que es aquella que se produce cuando el vehículo Ego va siguiendo al Target y en un momento determinado dicho vehículo frena.

Deceleración de aproximación:

Esta característica del sistema se ve de manera muy clara sólo para algunas de las medidas. Por ejemplo para un caso concreto en el que el conductor selecciona el modo dinámico y `time gap` igual a 1, se puede apreciar claramente un cambio brusco en la pendiente de bajada de la velocidad.

En el gráfico 3.7 se muestra el vehículo target conduciendo a una velocidad constante de 57km/h y el vehículo Ego, que reduce su velocidad para amoldarse a esta nueva situación del tráfico. Como puede verse para tiempo aproximado de 13 segundos, la

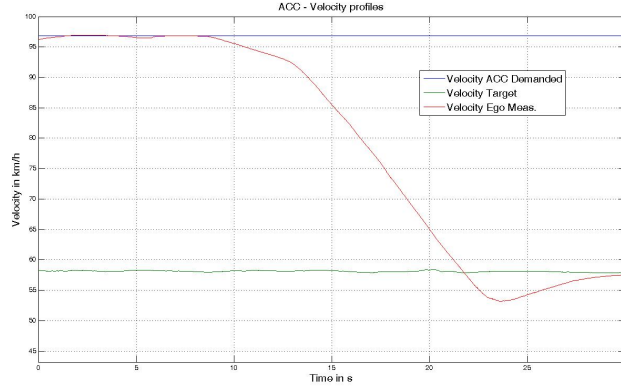


Figura 3.7.: Velocidades del vehículo Ego y Target tomadas de las medidas reales, donde pueden apreciarse dos pendientes diferentes de deceleración. El cambio se encuentra alrededor de $t = 13$ s .

deceleración cambia bruscamente de valor. Para ajustar el comportamiento del modelo a esta situación se introducen dos pendientes diferentes (dos valores de k) y un valor de distancia a partir de la cual se produce el cambio. Al primer valor de la pendiente se le denomina $k_{downAprox}$ y se entiende como la pendiente de aproximación inicial. El segundo valor se le llama k_{down} y representa la pendiente general de frenada, y por último el valor a partir del cual se produce el cambio se denomina k_{dist} .

Para este caso concreto los valores se muestran en las ecuaciones de 3.10 hasta 3.12.

$$k_{downAprox} = 0,08s^{-1} \quad (3.10)$$

$$k_{dist} = 60m \quad (3.11)$$

$$k_{down} = 0,25s^{-1} \quad (3.12)$$

Con estos valores el sistema se ajusta de manera bastante precisa. Los resultados para esta pendiente de bajada se muestran en figura 3.8. En dicha imagen se observan dos gráficas, la primera representa las velocidades de ambos vehículos en km/h mientras que la segunda compara el comportamiento de los dos vehículos Ego, el real y el simulado, para analizar y determinar el correcto funcionamiento del modelo. La línea verde representa la diferencia de posición y la línea negra es la diferencia en velocidades de dichos vehículos.

Puede verse que esta diferencia en las velocidades de los Ego real y simulado nunca excede el límite de ± 2 km/h, al igual que la diferencia entre las posiciones nunca sobrepasa los 2 metros. Esto quiere decir que para cualquier momento la velocidad del Ego simulado presenta una diferencia máxima de 2 km/h con respecto a la velocidad del Ego real, al igual que la posición relativa del Ego simulado con respecto al target (x_{rel}) nunca será

mayor ni menor de 2 metros en comparación con la distancia entre el Ego real y el Target. En la figura 3.9 puede verse gráficamente el caso en el que $X_{relReal} > X_{relSim}$ y en las ecuaciones 3.13 y 3.14 se muestra los valores de $diffX$ y $diffV$.

$$diffX = X_{relReal} - X_{relSim} \quad (3.13)$$

$$diffV = V_{EgoSim} - V_{EgoReal} \quad (3.14)$$

El ajuste de este controlador se ha llevado a cabo realizando numerosas simulaciones, variando los parámetros descritos con anterioridad hasta que los valores de diferencia de velocidades y de posición quedaban entre los márgenes establecidos de ± 2 km/h y metros. En algunos casos como el de la figura 3.8 se ve muy claro y el ajuste es más sencillo, pero en otros casos no se ve tan claramente en qué momento la pendiente cambia de valor y el número de simulaciones necesarias puede aumentar.

Por último se muestra el resultado obtenido antes incluir el controlador de la pendiente. Se observa en la figura 3.10 que la diferencia entre el comportamiento del Ego simulado y el real es considerablemente mayor. En algún momento puede verse que la diferencia entre las x_{rel} de ambos vehículos alcanza los 3 metros.

Deceleración de seguimiento:

Esta situación solo se produce cuando el vehículo Ego está siguiendo al Target en una situación estacionaria y por la razón que sea el Target reduce su velocidad.

La velocidad relativa se define como la velocidad del Target menos la del Ego: $V_{rel} = V_{Target} - V_{Ego}$, por lo tanto cuando el Target empieza a decelerar esta velocidad se hace negativa. El controlador se activa en el momento en el que V_{rel} alcanza un valor negativo determinado. En ese momento el Ego empieza a frenar de una manera más brusca hasta que el controlador se vuelve a desactivar, lo que ocurre cuando la velocidad relativa alcanza otro valor determinado. Estos dos valores corresponden a las variables C_{13} y C_{14} respectivamente. Con estos dos límites de operación combinados con el grado de intervención medido por una constante llamada Cdf se corrige la deceleración de seguimiento.

Unos valores típicos para este controlador serían: $C_{13} = -2km/h$, $C_{14} = 1km/h$ y $Cdf = 2$; en la figura 3.11 se muestra en qué momento actúa el controlador que para este caso es desde que $V_{rel} < C_{13}$ hasta $V_{rel} > C_{14}$. En la figura 3.12 se muestra el comportamiento del sistema dinámico con time gap 1, para el caso sin controlador (izquierda) y el caso con controlador (derecha). Puede verse que la diferencia para este caso es sustancial.

Mantener el mínimo:

El último control local se utiliza en combinación con los otros dos para terminar de ajustar las deceleraciones. Este control detecta un mínimo en la deceleración y lo mantiene hasta que se cumple una determinada condición.

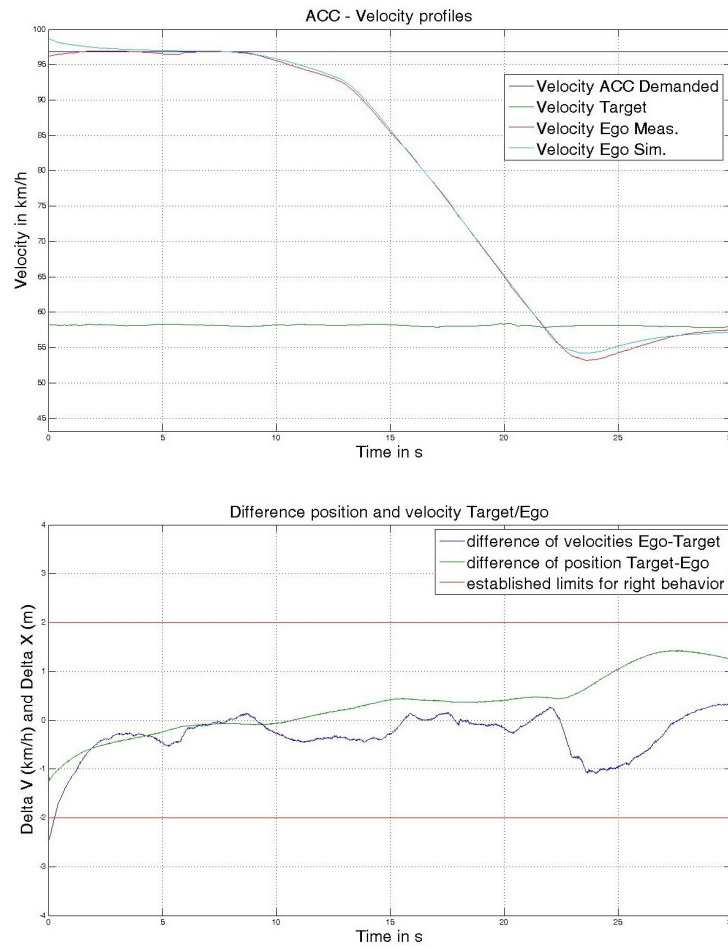


Figura 3.8.: Perfiles de velocidad de los vehículos (arriba). Diferencia entre las velocidades del Ego real y simulado ($difV$) y diferencia entre la x_{rel} del Ego real y simulado (abajo).

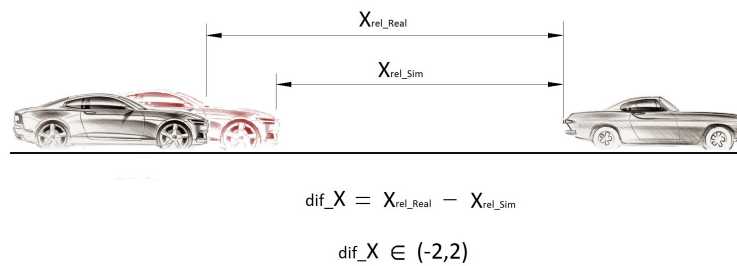


Figura 3.9.: Vehículo Ego simulado (rojo) y real (negro) con un Target delante, para comparar las distancias relativas de ambos vehículos Ego.

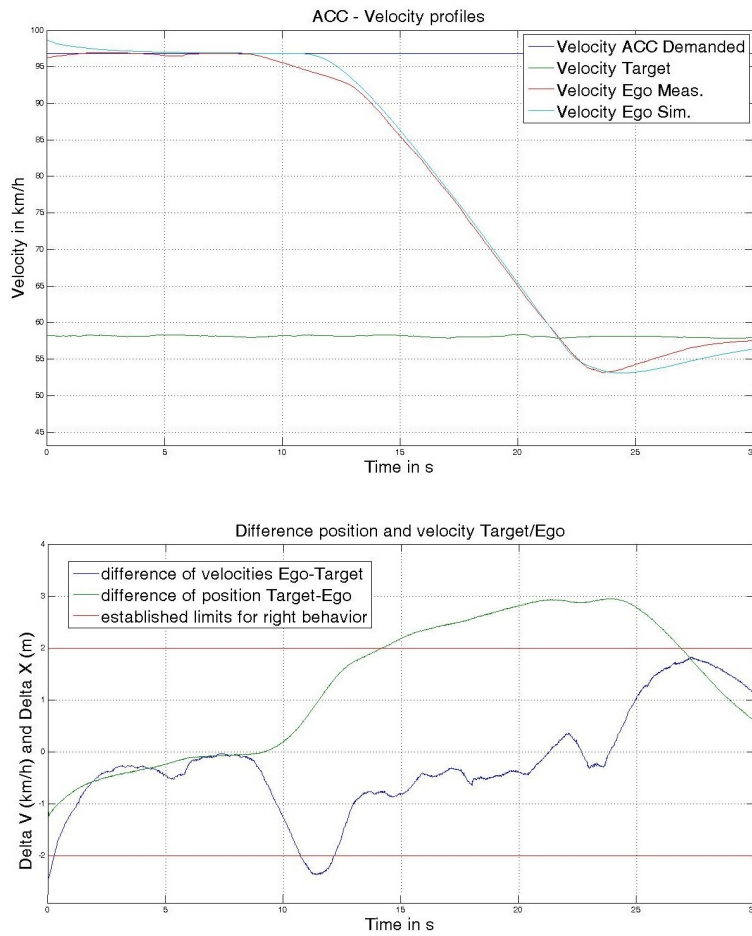


Figura 3.10.: Perfil de velocidades de todos los vehículos (arriba). Diferencia en velocidades $diffV$ y posición $diffX$ (abajo) para el caso en el que la doble pendiente de bajada no está implementada.

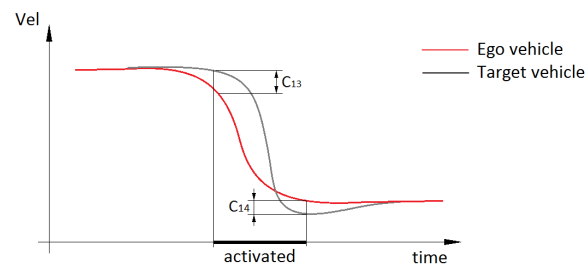


Figura 3.11.: Esquema simplificado de los valores que rigen la deceleración de seguimiento.

3. Sistema ACC

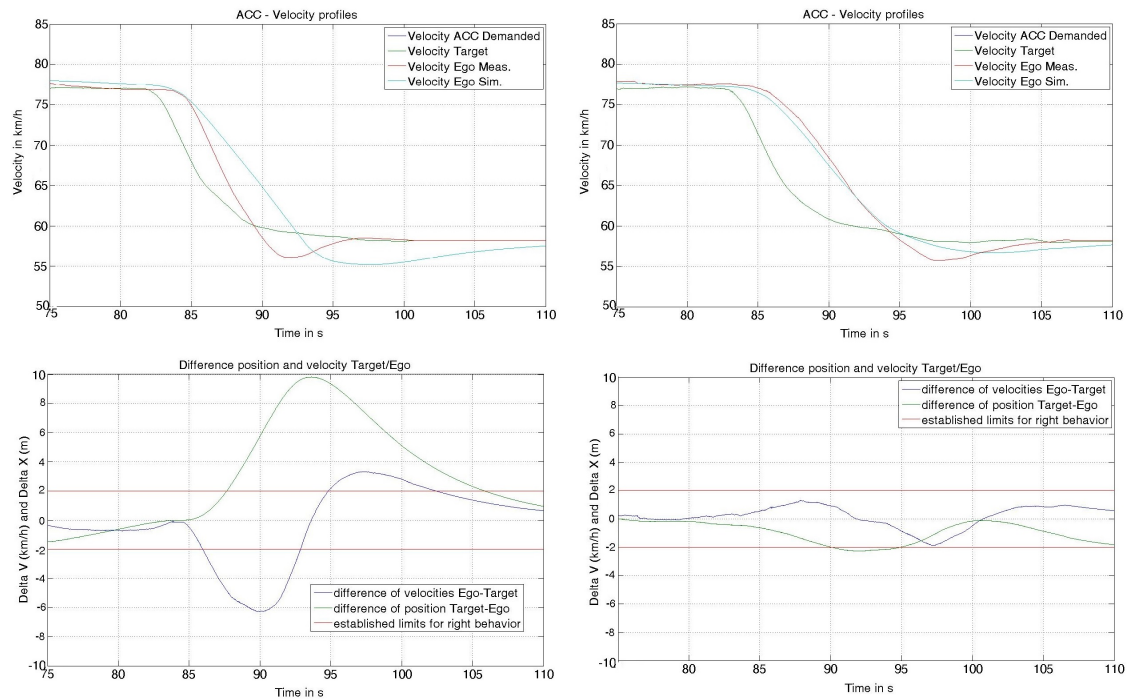


Figura 3.12.: Perfil de velocidades (arriba) y diferencias $diffX$ y $diffV$ (abajo) para dos casos concretos. El control de deceleración de seguimiento apagado (izquierda) y encendido (derecha)

Este controlador actúa hasta que la velocidad relativa entre el vehículo Target y el Ego es mayor que un determinado valor. Este valor queda determinado por la variable vel_{UP} , la cual se encuentra definida en el archivo init de matlab.

La figura 3.13 muestra el comportamiento de este controlador. El segundo gráfico (abajo) muestra ambas aceleraciones, la línea verde es la resultante al utilizar este controlador, mientras que la azul representa la aceleración original. Puede verse que la línea verde es siempre igual a la azul excepto cuando se presenta un mínimo en la aceleración, en ese momento el controlador mantiene dicho mínimo hasta que se cumple $V_{rel} > vel_{UP}$.

3.2.5. Cálculo del offset

El offset es una variable que se suma a la distancia que se guarda con el vehículo de delante, con el fin de aumentar el margen de seguridad.

Dicho parámetro no es constante, sino que cambia en función del time gap seleccionado por el conductor y de la velocidad del vehículo Ego. De acuerdo con los resultados, la ecuación que gobierna el offset cambia significativamente cuando se elige un time gap u otro, y ligeramente dependiendo del modo seleccionado por el conductor (confort,

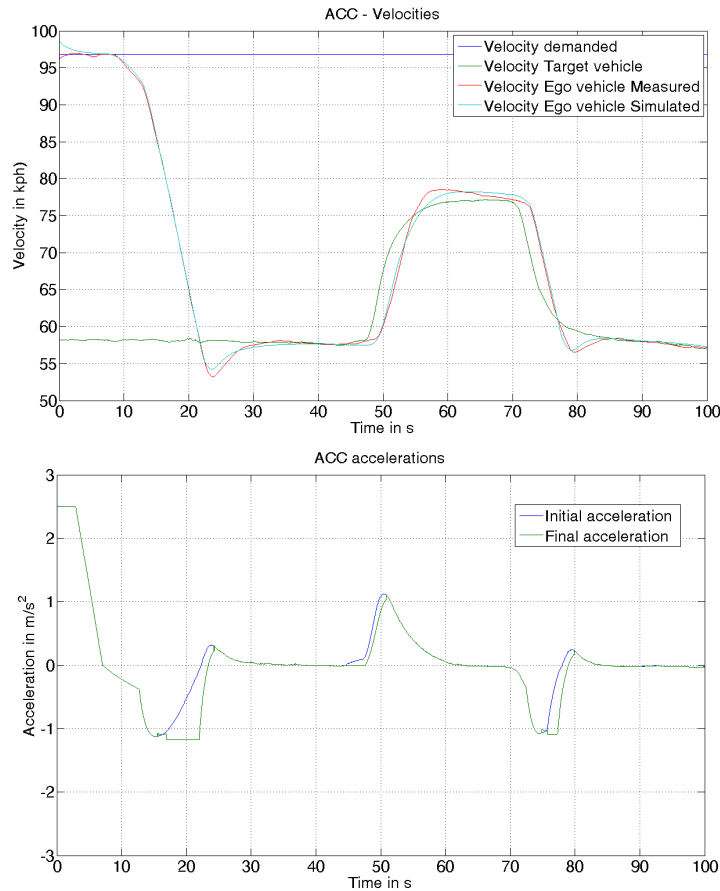


Figura 3.13.: Velocidades (arriba), y aceleración inicial y final para el controlador "Mantener el mínimo".

dinámico o estándar).

Las mediciones realizadas en el coche real Audi A6 comparan el comportamiento para time gap 1 y 4 y para cada modo seleccionado por el conductor.

Como se sabe, la distancia que el vehículo Ego mantiene con el coche de delante se expresa en la ecuación 3.5. En un estado estacionario, es decir, cuando ambos vehículos conduzcan a la misma velocidad y guarden una distancia constante, se mide la distancia entre ambos vehículos, que es d_{soll} , y la velocidad del Target V_{Target} . Como se sabe qué modo había sido seleccionado por el conductor en cada situación sólo tenemos que calcular el offset despejando la variable (ec 3.15).

$$d_{offset} = d_{soll} - \tau \cdot V_{target} \quad (3.15)$$

Time Gap 1:

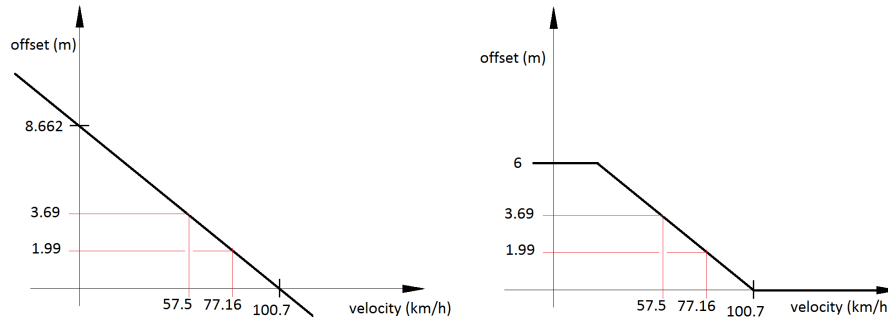


Figura 3.14.: Correlación entre el offset y la velocidad del Ego para $\tau=1$

Como ejemplo, se muestra el caso de **modo estándar** y **time gap** $\tau = 1$, las medidas reales se muestran en la figura 3.15 y podemos ver que la situación estacionaria tiene lugar alrededor del *tiempo* = 46s y *tiempo* = 79s (puntos 1 y 2 en dicha gráfica).

Las medidas tomadas del gráfico 3.15 se muestran en la tabla 3.2 y la ecuación 3.15 para el primer caso (*tiempo* = 46s) se resuelve de la siguiente manera: $offset = 19,66 - (57,5/3,6) \cdot 1 = 3,69m$.

Cuadro 3.2.: Datos obtenidos de las medidas reales para $\tau = 1$

Time (s)	V_{target} (km/h)	d_{soll} (m)	offset (m)
46	57.5	19.66	3.69
79	77.16	23.42	1.99

Como puede verse el offset se reduce al aumentar la velocidad. Esto puede deberse a que cuando la velocidad es grande, no necesitamos ningún offset ya que la distancia $\tau \cdot V_{target}$ es lo suficientemente grande como para conducir de una manera segura, por otro lado cuando la velocidad es cero sigue siendo necesario guardar una cierta distancia con el coche de delante. De acuerdo con las mediciones para este caso en particular (el modo estándar y time gap 1) el valor del offset se rige por la ecuación 3.16 donde la velocidad se expresa en km/h, y el offset en metros.

$$d_{offset} = 8,662 - 0,086 \cdot V_{Ego} \quad (3.16)$$

Límites del offset: ya que sólo hay medidas para dos situaciones la ecuación puede no ser válida para cada velocidad, por lo tanto, hay algunas correcciones que se deben hacer. Por ejemplo, un offset negativo puede no tener sentido, así como uno muy grande podría no ser necesario. En la figura 3.14 se muestra una descripción gráfica del comportamiento del offset en función de la velocidad del vehículo Ego (V_{Ego}), donde se pueden ver su comportamiento antes y después de establecer los límites.

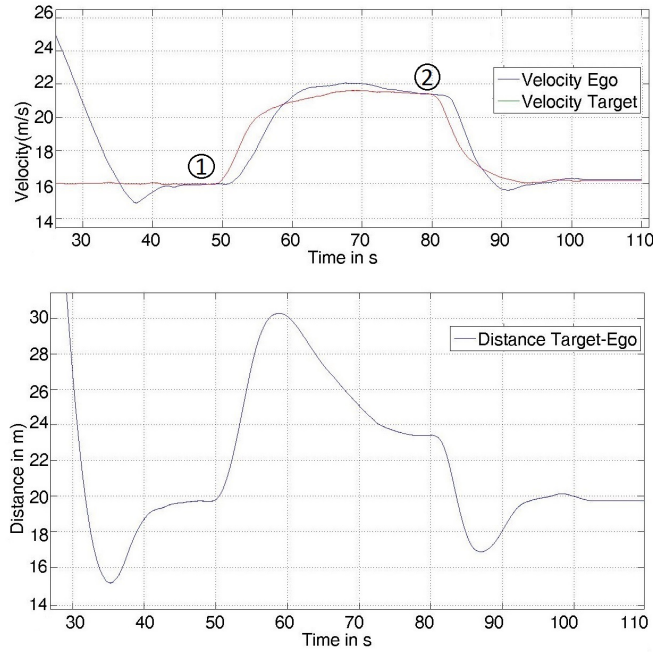


Figura 3.15.: Gráficas utilizadas para medir la distancia d_{soll} para modelo estándar y $\tau = 1$ en el estado estacionario.

Time Gap 4:

Como hemos visto hasta ahora, a mayor velocidad menor offset, sin embargo, el comportamiento para $\tau = 4$ es completamente diferente. De acuerdo a las mediciones para este caso a mayor velocidad mayor offset se necesita.

Como en el caso anterior, se hace necesario establecer unos límites para el offset. De lo contrario tendremos un desplazamiento negativo cuando la velocidad es cero. La correlación del offset con la velocidad se puede ver en la figura 3.16 y los datos se muestran en la siguiente tabla(3.3). Se puede leer más información sobre el offset y consultar sus gráficas de comportamiento en el apéndice E.

Cuadro 3.3.: Datos obtenidos de las medidas reales para $\tau = 4$

Time (s)	V_{target} (km/h)	d_{soll} (m)	offset (m)
46	57.6	39.05	2.21
79	77.1	53.94	4.66

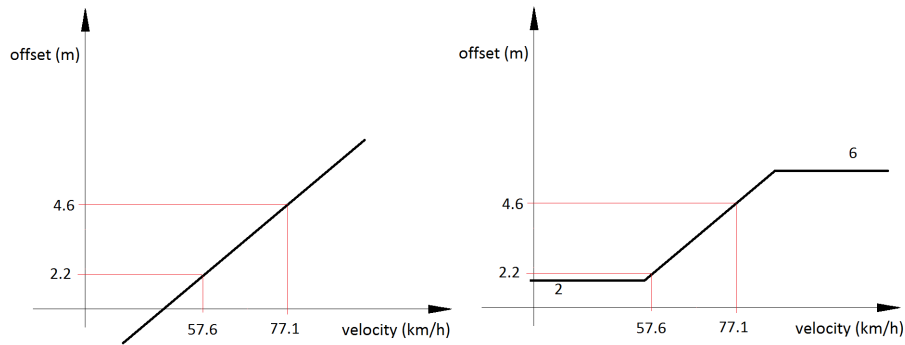


Figura 3.16.: Correlación entre el offset y la velocidad del Ego para $\tau=4$

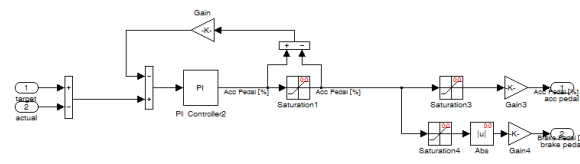


Figura 3.17.: Control de la aceleración con un PI + anti-windup

3.3. Control de aceleración

El control de aceleración es el bloque azul en la figura 3.2. Como se puede ver en la imagen, hay dos inputs diferentes para este controlador y se selecciona el mínimo de ambos (el porqué se explica en la introducción del capítulo 3).

El controlador está compuesto de un término P más un término I con un diagrama de anti-windup. El primer controlador que se probó era un control P, sin embargo este diseño presenta un error en el estado estacionario, ya que nunca se alcanza el valor deseado de aceleración. Al añadir un operador integrar dicho error se elimina, sin embargo esta solución produce una gran sobreoscilación debido al error acumulado que se produce durante la subida (windup). Con el fin de evitar este problema se añade el diagrama de anti-windup. El controlador resultante se muestra en la figura 3.17.

Para determinar los valores de este controlador se realizaron numerosas simulaciones. Se exige una sobreoscilación nula además de una respuesta considerablemente rápida por debajo del medio segundo. Tras llevar a cabo las simulaciones se determinó que los valores óptimos de la constante proporcional, integral y el valor para el anti-windup del controlador son: $k_p = 0,12$, $k_i = 3$ y $k_{windup} = 100$.

En la figura 3.18 podemos ver el comportamiento del sistema ante un escalón de $0,7m/s^2$, se puede ver que no hay sobreoscilación y que el error en estado estacionario es cero. Se muestran más gráficos y resultados en el apéndice B.

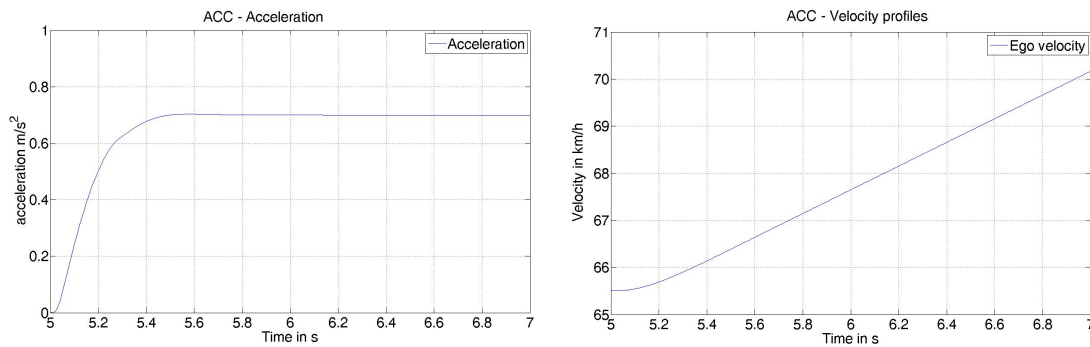


Figura 3.18.: Comportamiento del control de aceleración ante un escalón de $0,7m/s^2$

3.4. Resultados ACC

Los resultados para el modelo ACC se muestran en este apartado. Tanto el vehículo Ego real como el simulado tienen el mismo vehículo Target. El objetivo es que la diferencia entre las velocidades de los dos vehículos Ego (real y simulado) no exceda el valor $\pm 2\text{km/h}$. Por otro lado la distancia relativa entre el Target y el Ego simulado y entre el Target y el Ego real no debe exceder los 2 metros.

Los resultados obtenidos para el caso de seleccionar el modo dinámico y un time gap igual a 4 se muestran en la figura 3.19. Puede verse como estos límites previamente establecidos no se sobrepasan en ningún momento. Para más resultados de otros modos y time gap véase el apéndice C.

Para el resto de posibles valores de τ (2 y 3) no se realizaron medidas con coches reales porque una vez se obtienen los parámetros del modelo para $\tau = 1$ y $\tau = 4$ se pueden interpolar los resultados. Por lo tanto estos ajustes quedan pendientes de realizarse al igual que el correspondiente al modelo confortable.

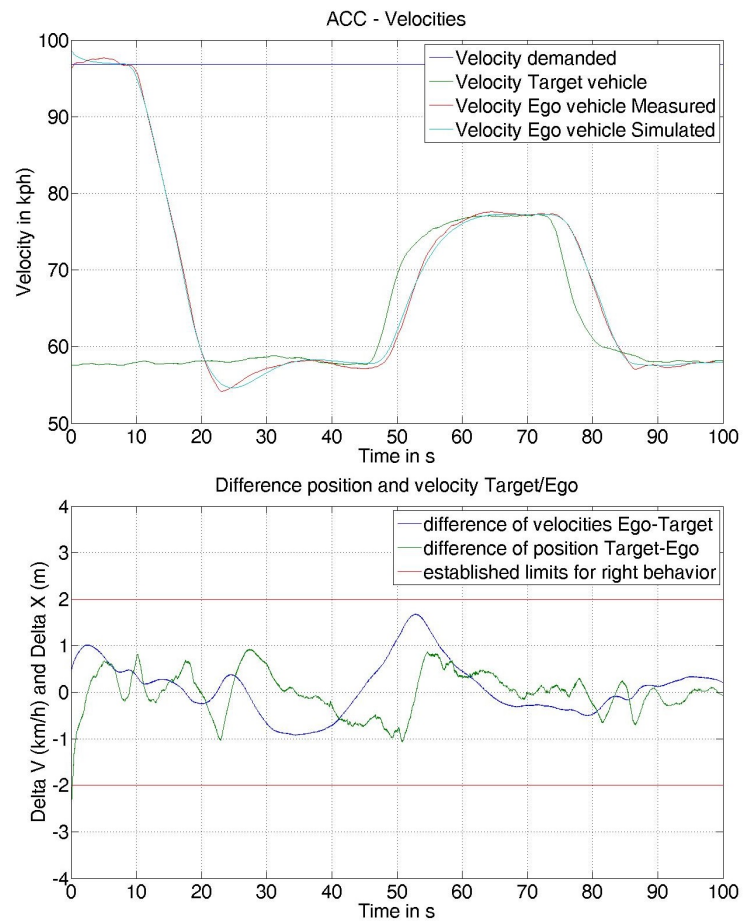


Figura 3.19.: Perfil de velocidades (arriba) y diferencias $diffX$ y $diffV$ (abajo) para el caso dinámico y time gap igual a 4.

4. Sistema AEB

El AEB es sistema de frenada de emergencia automático desarrollado para mejorar la seguridad en la carretera. Este sistema utiliza sensores para calcular la distancia y velocidad relativa al vehículo de delante y detecta cuando una colisión está a punto de producirse. En este momento, el sistema activa automáticamente los frenos con el fin de evitar la colisión o al menos reducir sus efectos.

Se ha demostrado que el AEB tiene un efecto directo sobre el número de accidentes de tráfico, y será obligatorio para recibir la calificación de cinco estrellas en las pruebas de la Euro NCAP. Algunos estudios reflejan que si todos los coches tuviesen este sistema, los accidentes se reducirían hasta en un 27% evitando alrededor de 1.000 muertos y 4.000 heridos graves al año, [7]. Sin embargo, este sistema está lejos de estar completamente extendido en Europa. De hecho, no está disponible en el 79% de los modelos que están a la venta actualmente [1].

El diagrama de bloques desarrollado para la implementación del sistema de frenada de emergencia (AEB) se muestra en el apéndice J.

4.1. Cómo funciona

El sistema AEB tiene 2 modos de funcionamiento diferentes dependiendo de en qué situación nos encontramos, ya que no es lo mismo las posibles situaciones que encontramos en ciudad que las que encontramos en carreteras interurbanas, por eso el sistema de frenado tendrá diferentes comportamientos.

La variable que define en qué modo estamos es la velocidad. De acuerdo con el sistema implementado en el Mercedes S el valor límite es 30km/h [6] (sin embargo, de acuerdo a las medidas realizadas por la universidad TU Graz dicho valor es ligeramente superior, por lo que el valor límite se establece en 35 km/h). Cuando conducimos por debajo de esta velocidad el sistema AEB está en el modo ciudad y cada vez que se aumenta la velocidad por encima de ese valor el sistema cambia al modo inter-urbano. La variable más importante de este sistema es el tiempo para la colisión (*ttc*: time to collision) el cual se explica en la sección 4.2.

4.2. Time to collision

El *ttc* (time to collision) es el tiempo que tenemos antes de que una colisión se produzca. Si por ejemplo $ttc = 1$, quiere decir que en un segundo se producirá un impacto con el coche de delante a menos que se apliquen los frenos.

Esta variable es la que gobierna el comportamiento del AEB, ya que para diferentes valores de *ttc* el sistema de freno tendrá una respuesta u otra (estos diferentes comportamientos se explican en la sección 4.3 y 4.4). La relación que define esta variable se muestra en la ecuación 4.1 donde x_{rel} y v_{rel} son la posición y velocidad relativas entre los coches Ego y Target.

$$ttc = \frac{x_{rel}}{|V_{rel}|} \quad (4.1)$$

El sistema de frenada de emergencia se activa cuando el *ttc* se reduce por debajo de un determinado valor, pero hay que ser especialmente cuidadoso con la condición de desactivación. Esto es debido a que cuando la velocidad relativa se aproxima a cero, el *ttc* tiende a infinito, lo que no significa que no haya peligro ni que se pueda desactivar el AEB.

Teniendo cuenta que la velocidad relativa es $V_{rel} = V_{Target} - V_{Ego}$, en el momento en que un Target se acerca a nosotros la velocidad relativa es negativa. Cuando se aplican los frenos esa velocidad aumenta hasta que se hace cero (ambos vehículos a la misma velocidad), para luego hacerse positiva (el vehículo Target se aleja otra vez). El momento en el que la velocidad relativa se hace cero el *ttc* tiende a infinito, pero puede ser que en ese momento la distancia entre ambos coches sea 1 metro y por lo tanto sea necesario seguir aplicando los frenos.

Debido al valor negativo en la velocidad relativa el *ttc* también será negativo, por eso se utiliza el valor absoluto de V_{rel} como se muestra en la imagen 4.1.

Por lo tanto la condición para que el AEB se active es que el *ttc* disminuya por debajo de un determinado valor, pero una vez activado no se desactiva cuando el *ttc* vuelve a aumentar sino que la frenada se prolonga hasta que se da otra condición diferente (imagen 4.1 derecha). A este respecto se presentan dos modos diferentes de operación:

$modo_{AEB} = 0$: la frenada se prolonga hasta que la velocidad del vehículo Ego es cero, es decir, hasta una parada total.

$modo_{AEB} = 1$: la frenada se prolonga hasta que la distancia entre ambos vehículos (Ego y Target) aumenta por encima de un valor establecido. Este valor depende de la situación en la que nos encontramos: si se está conduciendo en ciudad es 2 metros, y si se circula por carretera es 8 metros.

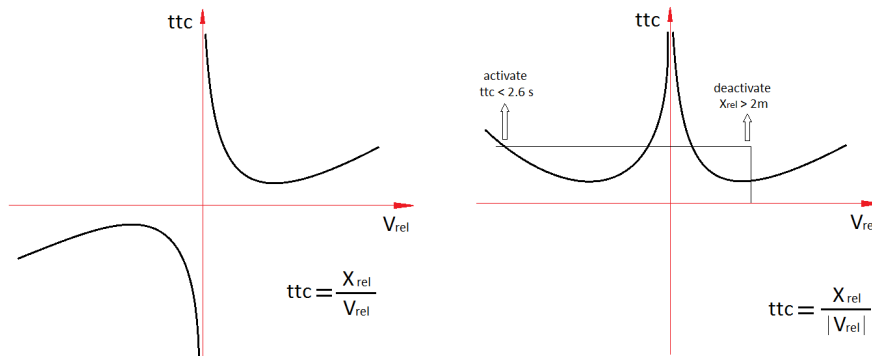


Figura 4.1.: Relación del ttc con la velocidad relativa entre vehículo Ego y Target.

4.3. Modo Ciudad

Para el modo ciudad el sistema se activa cuando la velocidad del vehículo Ego es menor o igual a 35 km/h.

En este escenario, el sistema tiene 2 fases diferentes y dependiendo del ttc se pasa de la primera fase a la segunda con el fin de evitar la colisión.

- $ttc \leq 1,5s$: **Señal óptica y acústica (3 veces)**: cuando el ttc cae por debajo del tiempo establecido de 1,5 segundos, la primera fase del sistema AEB se activa. El objetivo de este paso es advertir al conductor para que pise el freno, ya que una colisión es inminente. En caso de no recibir respuesta por parte del conductor y si el ttc sigue disminuyendo el sistema pasa a la segunda fase.
- $ttc \leq 1s$: **Frenada total automática**: si no hay reacción por parte del conductor, el sistema aplica los frenos a su máxima capacidad con el fin de evitar la colisión. El valor de la deceleración es $-10m/s^2$.

4.4. Modo Inter-Urbano

La velocidad a la que estamos conduciendo en este caso es mayor de 35km/h y las distancias se suponen más grandes que en el modo ciudad, por lo tanto, este modo tiene 3 fases diferentes. La variable a considerar es de nuevo el time to collision ttc .

- $ttc \leq 2,6s$: **Señal óptica y acústica (3 veces)**: cuando el ttc cae por debajo del tiempo establecido de 2,6 segundos, el primer paso del sistema AEB se activa. El objetivo de este primer paso es, al igual que en el modo ciudad, advertir al conductor. Si no hay respuesta del conductor se pasa a la siguiente fase.
- $ttc \leq 1,6s$: **Frenada parcial automática**: este fase es el paso intermedio que el modo inter-urbano presenta. El objetivo de esta frenada parcial es ganar tiempo

esperando que el conductor tome el control del vehículo. Si no lo hace el sistema salta al paso 3. El valor de esta frenada es una deceleración de $-4m/s^2$.

- $ttc \leq 0,6$: **Frenada total automática**: cuando el ttc es menor o igual a 0.6 segundos y no ha habido respuesta del conductor el sistema realiza la frenada total para finalmente evitar la colisión, la cual significa una deceleración de $-10m/s^2$.

Todos los datos relativos a la velocidad que define modo ciudad o urbano, así como el tiempo en el que los sistemas pasan de warning a frenada parcial y frenada total, han sido obtenidos del modelo Mercedes S [6].

4.5. Resultados AEB

El AEB es un sistema que actúa en situaciones de emergencia, por lo tanto para analizar su comportamiento realizamos una simulación en la que el vehículo Ego circula a aproximadamente 100km/h cuando de repente otro vehículo aparece enfrente a una distancia de 60 metros y con una velocidad de 23km/h.

En esta situación el AEB opera en modo Inter-Urbano, lo que significa que primero realiza la señal acústica y luminosa para avisar al conductor. Al no haber ninguna respuesta por parte del conductor, el sistema AEB aplica los frenos a mitad de su capacidad (produciendo una deceleración de $-4m/s^2$), y más tarde a la máxima deceleración ($-10m/s^2$). Todo esto puede verse en la figura 4.2 en la parte de abajo donde AEB mode igual a 1 significa warning, 2 es frenada parcial, y 3 es frenada total.

En la figura 4.2 puede verse el comportamiento del sistema. En la primera gráfica se ven los perfiles de velocidades de ambos coches, y la distancia que hay entre ellos (expresada en dm para una mejor visualización). La primera parte de la deceleración (entre 0 y 1.5 segundos) es llevada a cabo por el ACC y tiene un valor de $-3,5m/s^2$, el porqué de este valor de deceleración se explica en la sección 3.2.3. Al mismo tiempo que el ACC está decelerando el sistema AEB avisa al conductor, de ahí que en la segunda gráfica el valor del AEB mode pasa a ser 1. La segunda parte de la deceleración (entre 1.5 y 3.2 segundos) equivale a la frenada parcial. En la gráfica de arriba puede verse que la pendiente de deceleración del Ego cambia y en la segunda gráfica, se ve que el AEB mode cambia a 2. Por último la tercera pendiente de deceleración se lleva a cabo a partir de los 3.2 segundos, en ese momento en la segunda gráfica el valor de AEB mode cambia a 3, lo que significa frenada total.

Una vez alcanzado el valor máximo de deceleración, se mantiene hasta que la distancia es mayor que el valor establecido. En este caso dicho valor es 8 metros ya que la velocidad inicial es 100km/h (modo inter-urbano), pero durante la frenada la velocidad desciende por debajo de los 35km/h por lo que el valor establecido cambia a 2 metros (modo ciudad). Debido a esto el sistema aguanta la frenada hasta que la distancia aumenta de nuevo a 2 metros (o 20dm) lo que ocurre para un tiempo aproximadamente igual a 4.8 segundos. Una vez que el AEB se desactiva, el ACC no se vuelve a activar hasta que

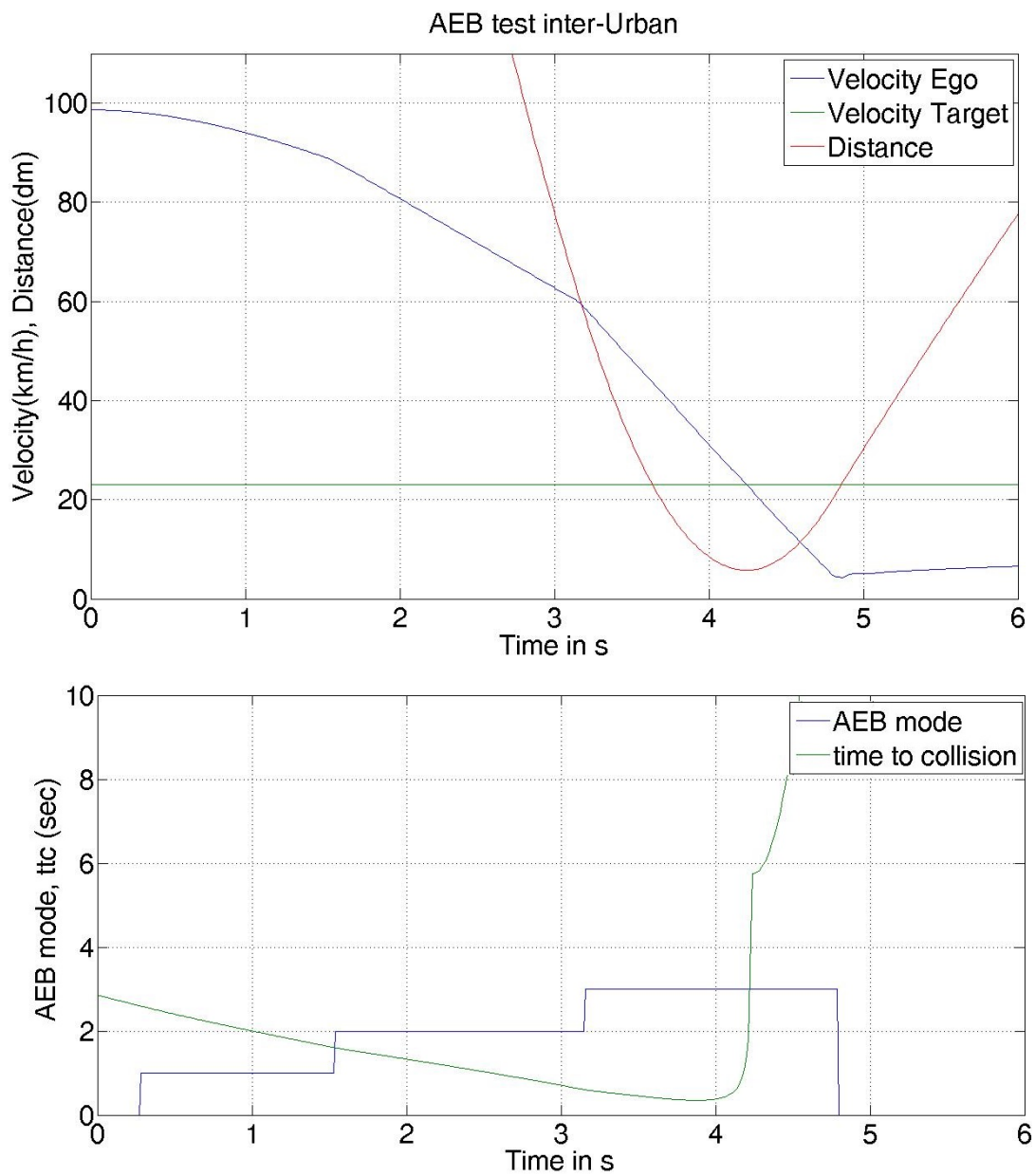


Figura 4.2.: El sistema AEB evita una colisión. La velocidad inicial del vehículo Ego es 100km/h cuando aparece un Target a 60 metros circulando a 23km/h. AEB mode cambia de valor 1(warning) a 2 (frenada parcial) y 3 (frenada total) (abajo). Perfil de velocidades (arriba).

4. Sistema AEB

el conductor no lo hace manualmente, es por eso que la velocidad del vehículo Ego no aumenta hasta los 23km/h que es lo que haría el ACC.

La mínima distancia que encontramos es alrededor de 0.8 metros, por lo tanto el AEB ha actuado de manera satisfactoria. En el apéndice D se recogen más resultados, incluida una simulación para examinar el modo city.

5. Conclusiones

En este proyecto se ha realizado e implementado un modelo simplificado de los sistemas reales ACC, AEB y sensor radar, instalados en vehículos comerciales. Dichos modelos serán instalados en un simulador de la conducción que el Instituto de Automoción de la Universidad TU Graz está desarrollando. El objetivo es instalar dichos modelos en un simulador de conducción, para extraer medidas y resultados que puedan servir para mejorar los sistemas que se desarrollen en el futuro. Por ello es de crucial importancia que el modelo simplificado refleje un comportamiento lo más similar posible a la realidad.

El sensor radar es un modelo simplificado que simula el comportamiento de un radar real. Su función es detectar objetos en la carretera así como su velocidad relativa, para recopilar la información necesaria para el correcto funcionamiento de los sistemas ACC y AEB. Se ha implementado un efecto sombra, así como la detección de un punto en el contorno exterior del vehículo. Como limitaciones, queda decir que todos aquellos vehículos cuyo centro de masas se encuentra por detrás del sensor radar, no producen ninguna sombra.

El modelo ACC desarrollado simula el sistema real instalado en el Audi A6. Dicho modelo presenta tres modos diferentes de operación: estándar, dinámico y confortable. En este proyecto se han implementado el modo estándar y dinámico, haciendo especial mención al comportamiento del dinámico, por sus estrechos márgenes de error.

Los ajustes que en este proyecto se han realizado, en relación al sistema ACC, son para un time gap de 1 y 4, con lo que un futuro desarrollo será posible interpolando los resultados a partir de estos valores para time gap de 2 y 3.

Por último queda mencionar el modelo AEB, el cual simula el sistema real instalado en el Mercedes S. El comportamiento de este modelo se ajusta de manera satisfactoria y diferencia entre los tipos de intervención disponibles en el sistema real.

5. Conclusiones

Índice de figuras

1.1. Primer cruce control desarrollado por la compañía Chrysler en 1958, basado en el invento anterior de Ralph Teetor.	2
1.2. Simulador de conducción en fase avanzada de construcción.	3
1.3. Ensayo con un proyector de cómo y dónde se formará la imagen (izquierda), estructura y pantallas 3D que se montarán enfrente del simulador y en ambas ventanillas para crear el efecto 3D y efecto de velocidad (derecha).	4
1.4. Imagen extraída de la revista de AVL [4] que muestra una imagen el software utilizado: VSM Dynamic Real Time Vehicle Simulation Model.	5
2.1. Rango geométrico del radar	7
2.2. Esquinas del vehículo	8
2.3. Input para la función de simulink	9
2.4. Rotación de vehículos	10
2.5. Ejemplo con un solo coche	10
2.6. Parámetros para definir una sombra	11
2.7. Sombras reales creadas por cada vehículo (gris claro), y sombras adicionales que el simulador del radar crea (gris oscuro).	12
2.8. Punto de la superficie detectado	12
2.9. Cuatro posibles soluciones a las ecuaciones de intersección de líneas, sólo uno de ellos es la solución correcta, en este caso el punto b.	13
2.10. Coches circulando en diferentes carriles de la carretera.	14
3.1. Control de velocidad y distancia actuando al mismo tiempo.	15
3.2. Modelo de simulink con el control de distancia (verde), control de velocidad (amarillo) y control de aceleración (azul)	16
3.3. Respuesta del sistema para un escalon de 30km/h.	17
3.4. Esquema de las dos contribuciones que intervienen en el control de distancia.	18
3.5. Distancias existentes entre vehículos.	19
3.6. Relación entre Δv y la aceleración que se produce, también pueden verse las diferentes k utilizadas.	20
3.7. Velocidades del vehículo Ego y Target tomadas de las medidas reales, donde pueden apreciarse dos pendientes diferentes de deceleración. El cambio se encuentra alrededor de $t = 13$ s	22
3.8. Perfiles de velocidad de los vehículos (arriba). Diferencia entre las velocidades del Ego real y simulado ($diffV$) y diferencia entre la x_{rel} del Ego real y simulado (abajo).	24

3.9. Vehículo Ego simulado (rojo) y real (negro) con un Target delante, para comparar las distancias relativas de ambos vehículos Ego.	24
3.10. Perfil de velocidades de todos los vehículos (arriba). Diferencia en velocidades $difV$ y posición $difX$ (abajo) para el caso en el que la doble pendiente de bajada no está implementada.	25
3.11. Esquema simplificado de los valores que rigen la deceleración de seguimiento.	25
3.12. Perfil de velocidades (arriba) y diferencias $difX$ y $difV$ (abajo) para dos casos concretos. El control de deceleración de seguimiento apagado (izquierda) y encendido (derecha)	26
3.13. Velocidades (arriba), y aceleración inicial y final para el controlador "Mantener el mínimo".	27
3.14. Correlación entre el offset y la velocidad del Ego para $\tau=1$	28
3.15. Gráficas utilizadas para medir la distancia d_{soll} para modelo estándar y $\tau = 1$ en el estado estacionario.	29
3.16. Correlación entre el offset y la velocidad del Ego para $\tau=4$	30
3.17. Control de la aceleración con un PI + anti-windup	30
3.18. Comportamiento del control de aceleración ante un escalón de $0,7m/s^2$	31
3.19. Perfil de velocidades (arriba) y diferencias $difX$ y $difV$ (abajo) para el caso dinámico y time gap igual a 4.	32
4.1. Relación del ttc con la velocidad relativa entre vehículo Ego y Target.	35
4.2. El sistema AEB evita una colisión. La velocidad inicial del vehículo Ego es 100km/h cuando aparece un Target a 60 metros circulando a 23km/h. AEB mode cambia de valor 1(warning) a 2 (frenada parcial) y 3 (frenada total) (abajo). Perfil de velocidades (arriba).	37
A.1. Response of the system fot a step of 30km/h for P=0.1 (left) and P=0.3 (right)	X
A.2. Response of the system fot a step of 30km/h for P=0.4 (left) and P=10 (right)	X
B.1. Response of the system fot a step of 30km/h for P=0.4 (left) and P=10 (right)	XII
C.1. Velocities of Target vehicle, Ego real and simulated for Dynamic mode and time gap equal to 1.	XIV
C.2. Velocities of Target vehicle, Ego real and simulated for Standard mode and time gap equal to 4.	XV
C.3. Velocities of Target vehicle, Ego real and simulated for Standard mode and time gap equal to 1.	XVI
C.4. Test for the behavior of the ACC controller when the Target speed up over the set-up speed (left), and when the driver press the gas over the set-up speed (right).	XVII

C.5. Test fo the behavior of the ACC when the driver switch off the system (left) and when the driver brakes (right). controller	XVII
D.1. Behavior of the AEB city,Ego vehicle drives at 32km/h, Target vehicle driving at 5km/h at a distance of 5 meters)	XIX
D.2. Response of the AEB system when the ACC is switched off	XX
E.1. Measures for different modes using $\tau = 1$	XXII
E.2. Measures for different modes using $\tau = 4$	XXII
E.3. Graphics used to measure the distance d_{soll} for standard model and time gap $\tau = 1$ in the steady state.	XXIII
E.4. Correlation between offset and the velocity of the Ego vehicle for diferent modes keeping the same $\tau = 1$	XXIV
E.5. Correlation between offset and the velocity of the Ego vehicle for diferent modes keeping the same $\tau = 4$	XXIV
F.1. Rotation of the gravity center of the vehicle	XXVIII
F.2. Rotation of the corners of the vehicle (left) and one only corner (right).	XXVIII
G.1. One car example	XXXI
G.2. Relative velocity	XXXII
G.3. Example of situation	XXXII
G.4. data input	XXXIII
H.1. 4 different angles in a car.	XXXV
I.1. Código Radar 1.	XXXVIII
I.2. Código Radar 2.	XXXIX
I.3. Código Radar 3.	XL
I.4. Código Radar 4.	XLI
I.5. Código Radar 5.	XLII
I.6. Código Radar 6.	XLIII
I.7. Código Radar 7.	XLIV
I.8. Código Radar 8.	XLV
I.9. Código Radar 9.	XLVI
J.1. Modelo en Simulink del conjunto del sistema ACC. El control de distancia se encuentra en el número 1.	XLVIII
J.2. Controlador de distancia. Pueden verse los diferentes controles: 1 control de deceleración de seguimiento, 2 control de deceleración de aproximación y 3 mantener el mínimo. Además el número 4 corresponde al sistema AEB.XLIX	XLIX
J.3. Sistema de frenada de emergencia AEB con sus diferentes modos ciudad e inter-urbano.	L

Índice de cuadros

3.1. Valores de time gap obtenidos de la guía del Audi A6 [2]	19
3.2. Datos obtenidos de las medidas reales para $\tau = 1$	28
3.3. Datos obtenidos de las medidas reales para $\tau = 4$	29
A.1. Results for different values of P for the velocity controller.	IX

Bibliografía

- [1] ARTRAC. EuroNCAP to drive availability of Autonomous Emergency Braking systems for safer cars in Europe. Available at <http://artrac.org/index.php/page/EuroNCAP-to-drive-availability-of-Autonomous-Emergenc-2012-08-08l>, 2011. Accessed on 14 Jun 2014.
- [2] Audi AG. Audi A 6 | S 6. Betriebsanleitung. Available at http://www.audi.de/content/dam/nemo/models/misc/pdf/my-2015/kurzanleitungen/kurzanleitung_a6_s6_rs-6-avant.pdf.
- [3] Automotive News. AVL simulates races to tune F1 engines. Available at <http://www.autonews.com/article/20030616/SUB/306160996/avl-simulates-races-to-tune-f1-engines>, 2003. Accessed on 20 August 2014.
- [4] AVL. AVL Powertrain World App. Available at https://www.avl.com/html/static/emag/focus_2013_2_E/index.html#22/, 2013. Accessed on 20 August 2014.
- [5] Marc Green and John Senders. Human Error in Road Accidents. Available at <http://www.visualexpert.com/Resources/roadaccidents.html>. Accessed on 12 May 2014.
- [6] Mercedes S. Available at http://techcenter.mercedes-benz.com/de_DE/pre_safe_brake/detail.html.
- [7] UNECE. UNECE works on new standards to increase the safety of trucks and coaches. Available at http://www.unece.org/press/pr2011/11trans_p10e.html, 2011. Accessed on 14 Jun 2014.
- [8] wikipedia. Ralph Teetor. Available at http://en.wikipedia.org/wiki/Ralph_Teetor. Accessed on 12 Jun 2014.
- [9] YAN, X., RADWANN, E. und GUO, D. Effects of major-road vehicle speed and driver age and gender on left-turn gap acceptance, 2007. Accessed on 12 May 2014.

A. Apéndice

Control de velocidad:

El ajuste del regulador de velocidad se hace seleccionando el valor que hace que el sistema sea lo suficientemente rápido, pero teniendo en cuenta la confortabilidad de la conducción.

Algunas simulaciones se realizaron utilizando un escalón de 30 kmh (de 66km/h a 96km/h) y los resultados obtenidos del tiempo de respuesta se muestran en la tabla A.1, es importante decir que la aceleración máxima del vehículo es de $2,5m/s^2$ de manera que el sistema nunca será más rápido de 9,6 segundos para este caso particular.

Cuadro A.1.: Results for different values of P for the velocity controller.

P	t_r [s]	SO [%]
0.1	29	0
0.3	15	0
0.35	12.1	0
0.4	11.7	0
0.45	11.05	0
0.5	10.9	0
10	9.6	1

Para el valor de P que ha sido seleccionado en el control(0.4) el sistema acelera con el valor máximo durante aproximadamente 8 segundos, para entonces alcanzar lentamente la velocidad deseada de 96 km/h. Por el contrario para P=10 el sistema acelera todo el tiempo a la máxima aceleración $2,5m/s^2$ y cuando se alcanza el valor 96 kmh reduce bruscamente la aceleración.

El gráfico para estos casos se muestran en la página siguiente en las figuras A.1 y A.2.

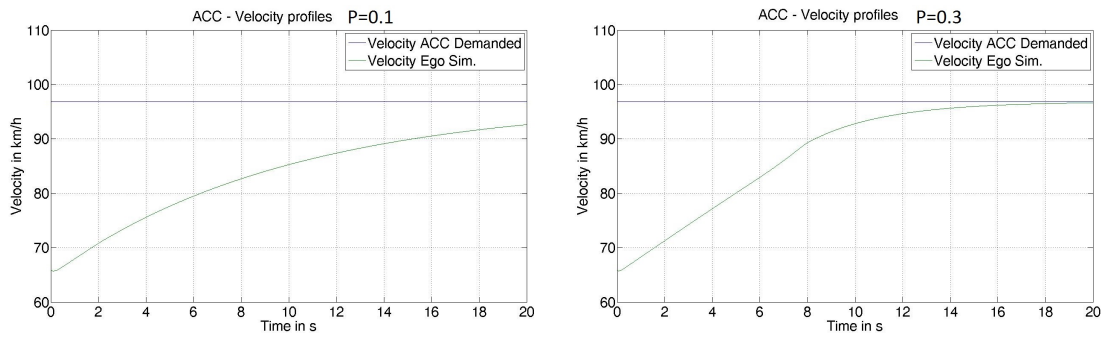


Figura A.1.: Response of the system for a step of 30km/h for P=0.1 (left) and P=0.3 (right)

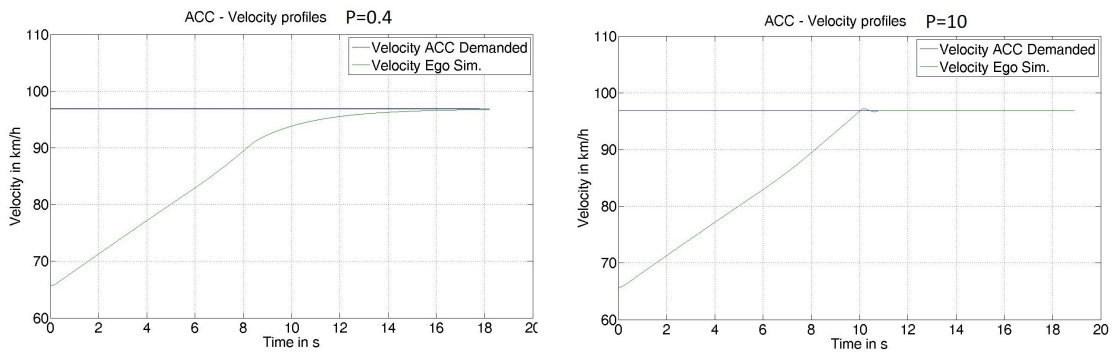


Figura A.2.: Response of the system for a step of 30km/h for P=0.4 (left) and P=10 (right)

B. Apéndice

Control de aceleración:

En la figura B.1 podemos ver el comportamiento del sistema en función del tipo de controlador que elegimos para controlar la aceleración. Cuando se utiliza el controlador P la velocidad nunca alcanza el valor deseado, que es alrededor de 97 kmh, por otro lado, cuando el término integrador se añade el error en el estado estacionario es cero pero hay una sobreoscilación importante de aproximadamente de un 50%.

El último gráfico de la imagen muestra el comportamiento del modelo cuando el diagrama PI + anti-windup es implementado, los valores seleccionados son: $k_p = 0,12$, $k_i = 3$ y $k_{windup} = 100$. Con estos valores, el tiempo de respuesta es de unos 10 segundos y no presenta sobreoscilación.

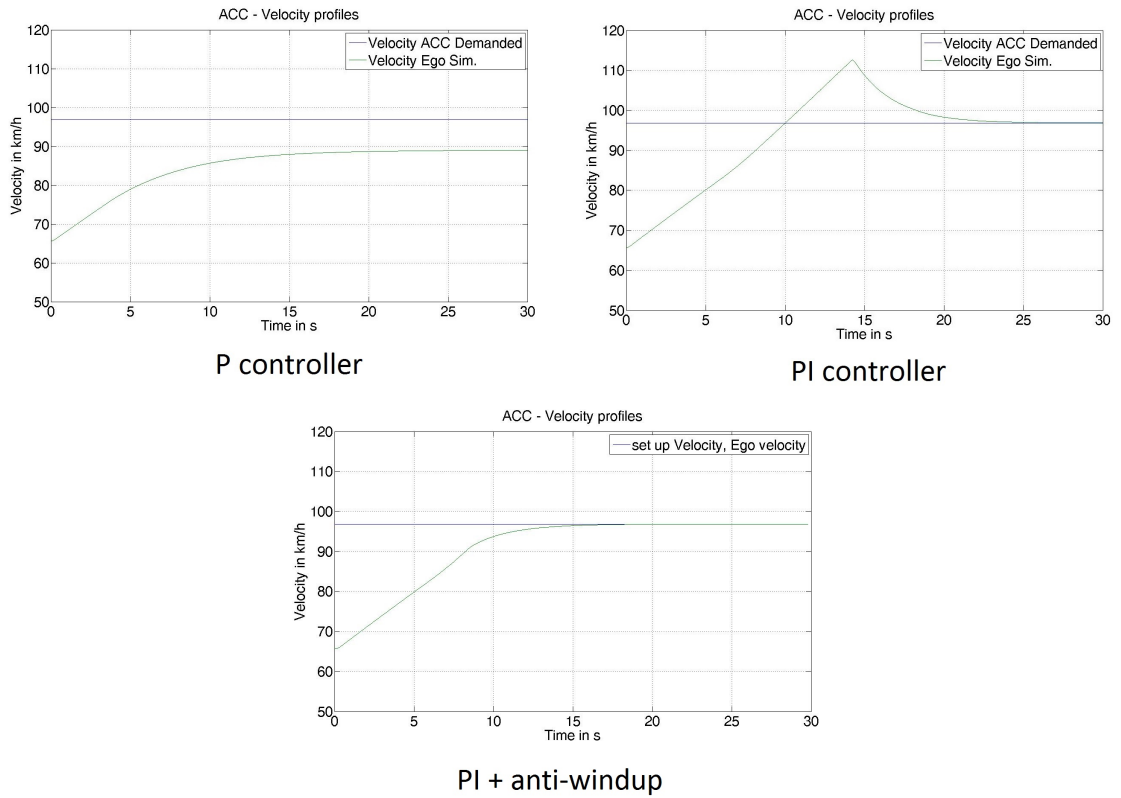


Figura B.1.: Response of the system for a step of 30km/h for $P=0.4$ (left) and $P=10$ (right)

C. Apéndice

Resultados ACC:

Los resultados obtenidos para el modo dinámico son los más satisfactorios, como se ha visto en la sección 3.4 el comportamiento se ajusta bastante al modelo real. Cuando el modo es dinámico pero seleccionamos el time gap igual a 1 el sistema también presenta un comportamiento dentro de los límites establecidos. Esto puede verse en la imagen C.1.

El caso estandar con time gap 4 se muestra en la figura C.2, puede verse que el sistema también responde acorde a los límites, solo sobrepasa el límite de posición para el segundo 90, cuando la diferencia de posiciones relativas aumenta hasta -2.2 metros.

Por último para el modo estandar con time gap igual a 1 se sobrepasa el límite de posición llegando casi a los 3 metros durante la pendiente de subida. Es posible que este caso pueda corregirse estableciendo un controlador adicional para la pendiente de subida tal y como se hizo en el apartado 3.2.4 para las deceleraciones.

Por último se muestran algunos casos en los que se ve cómo se comporta el ACC para diversas situaciones.

-**Caso 1:** el vehículo Ego se encuentra siguiendo a un Target a una velocidad de 90km/h cuando en $time = 20s$ el Target aumenta su velocidad por encima de la velocidad demandada por el conductor (97km/h), vajo esta circunstancia el Ego deja de seguir al Target y establece una velocidad de cruceo de 97km/h (figura C.4 izquierda).

-**Caso 2:** el Ego esta circulando a la velocidad demandada de 97km/h cuando el conductor pisa el acelerador ($time = 10s$), en ese momento el ACC pasa al motor la aceleracion que demanda el conductor con el pedal, por lo que empieza a acelerar. En $time = 15$ el conductor deja de pisar el pedal de gas por lo que el ACC vuelve a la velocidad establecida de 97km/h (figura C.4 derecha).

- **Caso 3:** el ACC sigue al vehiculo target a 57km/h cuando el conductor apaga el sistema, la velocidad desciende porque no hay ningún input en el pedal de gas, por lo tanto se va frenando debido al rozamiento (figura C.5 izquierda).

- **Caso 4:** misma situación que la anterior solo que aquí al diferencia es que el ACC se apaga porque el conductor pisa el freno en $time = 30$, 3 segundos después suelta el freno pero el sistema ACC no se vuelve a conectar, solo se enciende de nuevo si el conductor lo demanda expresamente pretando el botón (figura C.5 derecha).

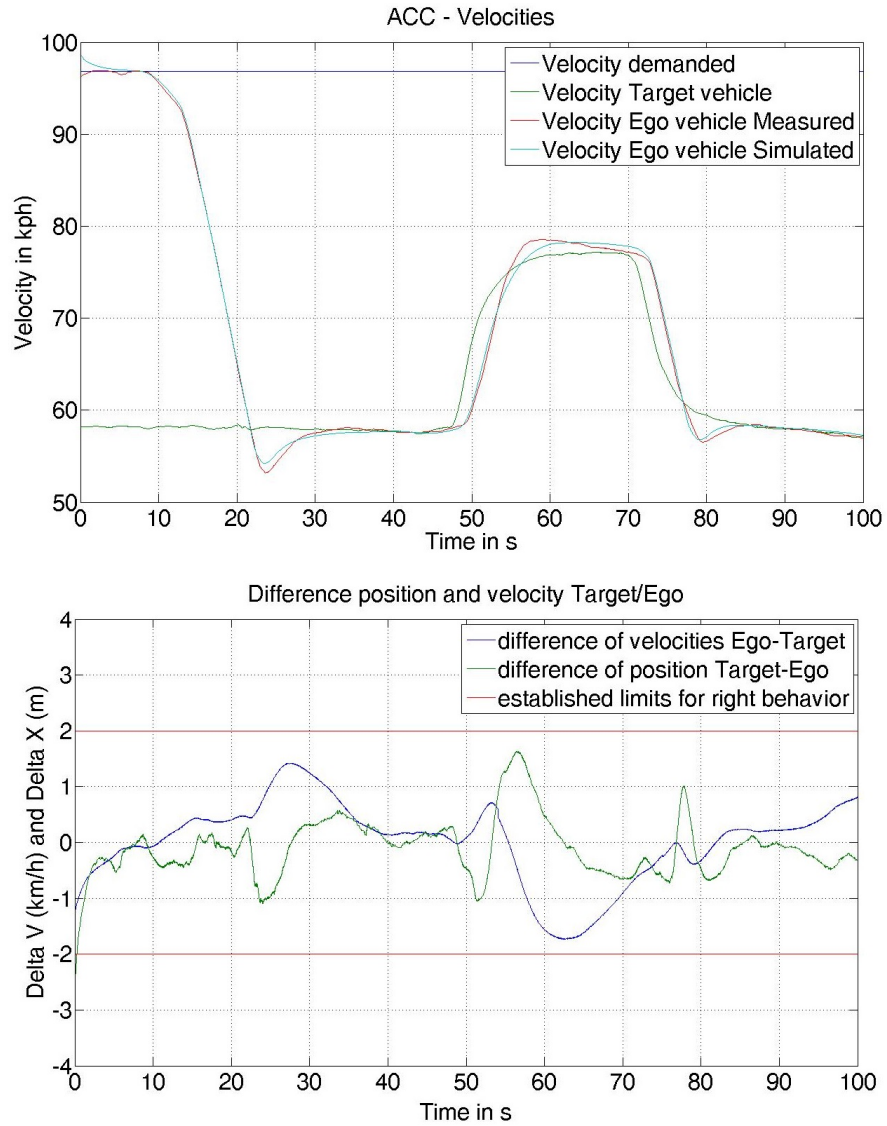


Figura C.1.: Velocities of Target vehicle, Ego real and simulated for Dynamic mode and time gap equal to 1.

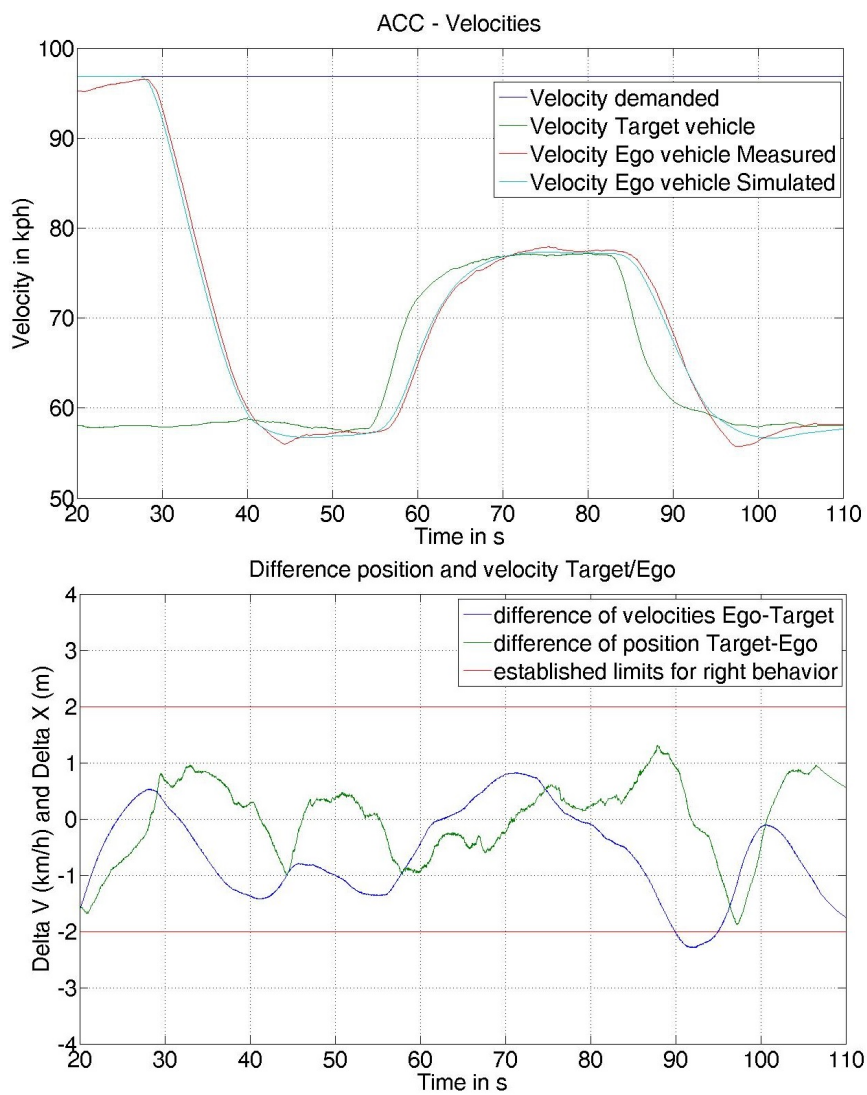


Figura C.2.: Velocities of Target vehicle, Ego real and simulated for Standard mode and time gap equal to 4.

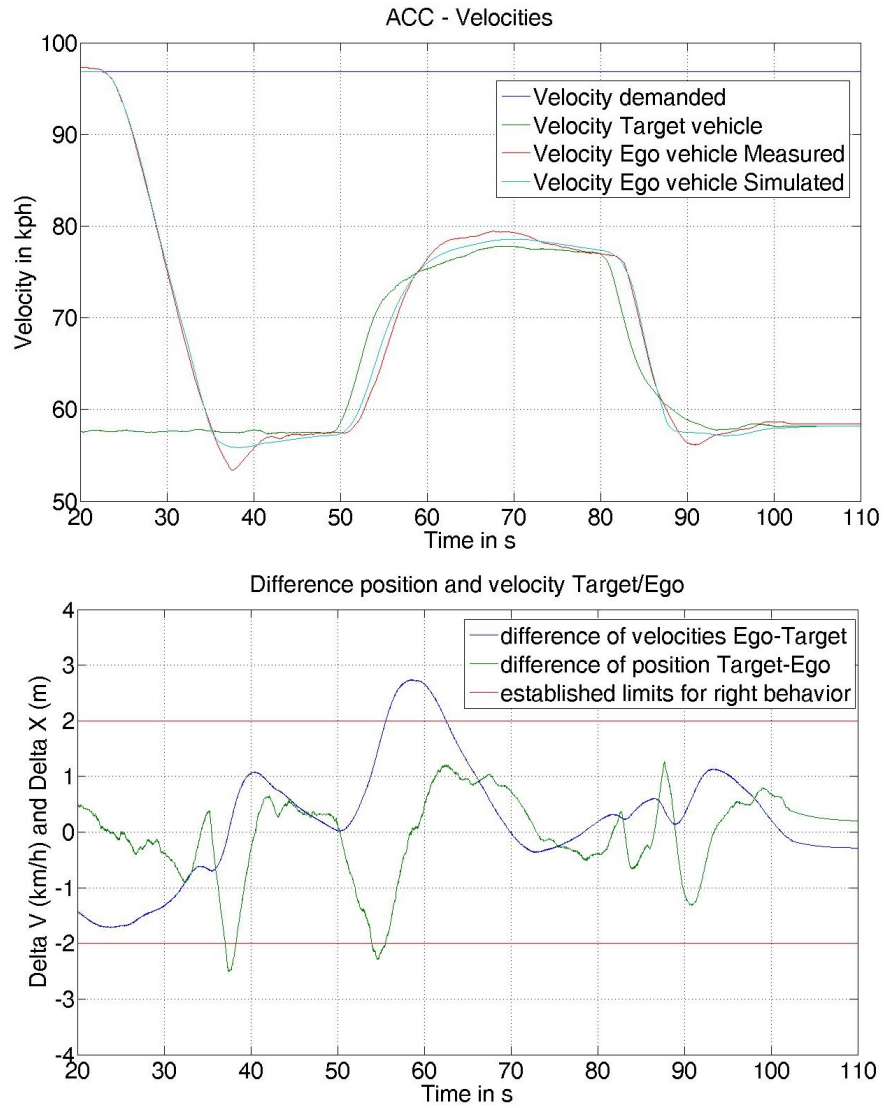


Figura C.3.: Velocities of Target vehicle, Ego real and simulated for Standard mode and time gap equal to 1.

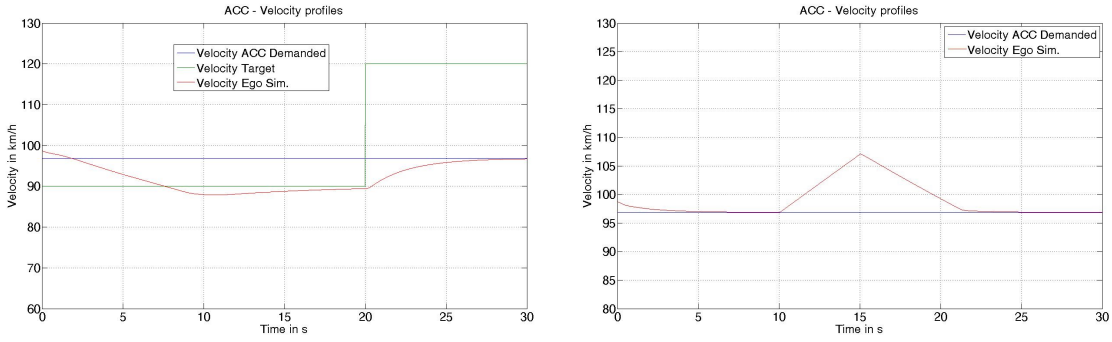


Figura C.4.: Test for the behavior of the ACC controller when the Target speed up over the set-up speed (left), and when the driver press the gas over the set-up speed (right).

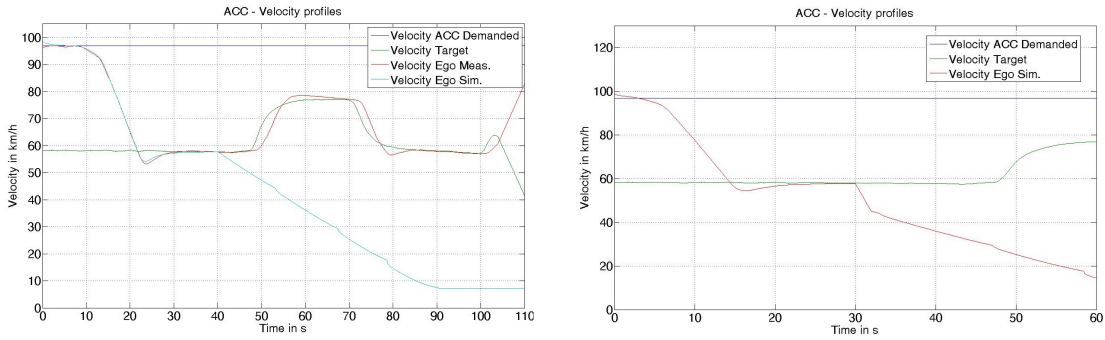


Figura C.5.: Test fo the behavior of the ACC when the driver switch off the system (left) and when the driver brakes (right). controller

D. Apéndice

Resultados AEB:

Otras pruebas para el AEB se muestran en este anexo. El caso del AEB Inter-Urban se ve en la sección 4.5, por lo tanto en este apéndice se muestra el comportamiento para AEB city.

Como fue descrito en dicho capítulo el comportamiento para el modo ciudad solo presenta dos situaciones, warning y frenada total. Por lo tanto en la figura D.1 puede observarse que solamente hay una pendiente de frenada, la cual se hace con toda la capacidad ($-10m/s^2$) y se mantiene hasta que la distancia (línea roja) aumenta hasta los 2 metros, entonces el AEB se desactiva.

Lo que ocurre después es simplemente que el coche empieza a moverse debido a que el software utilizado (vehicle dynamics) mantiene la primera marcha, por lo tanto el coche se mueve a la velocidad correspondiente al mínimo de revoluciones (ralentí).

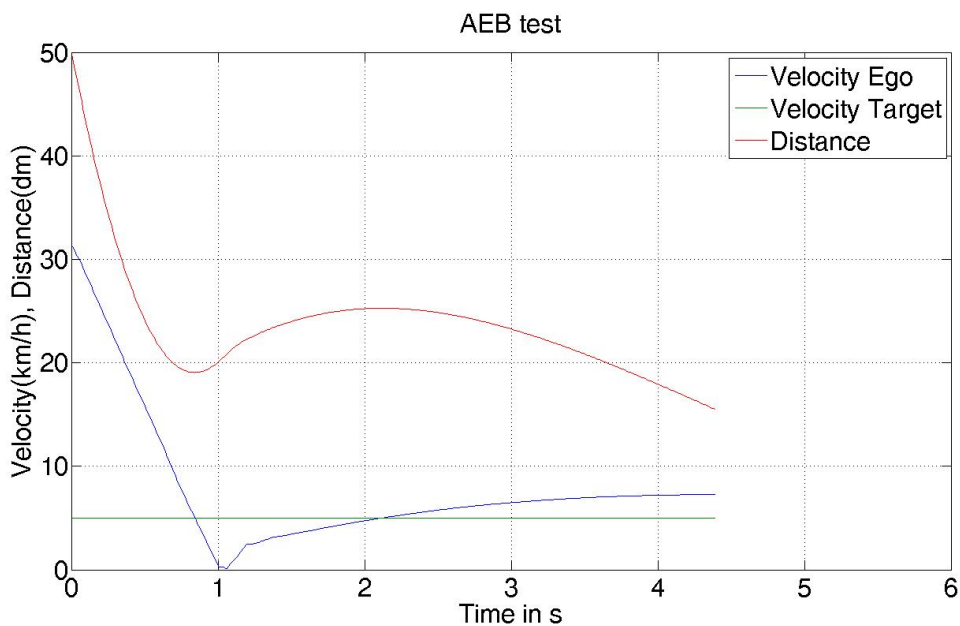


Figura D.1.: Behavior of the AEB city, Ego vehicle drives at 32km/h, Target vehicle driving at 5km/h at a distance of 5 meters)

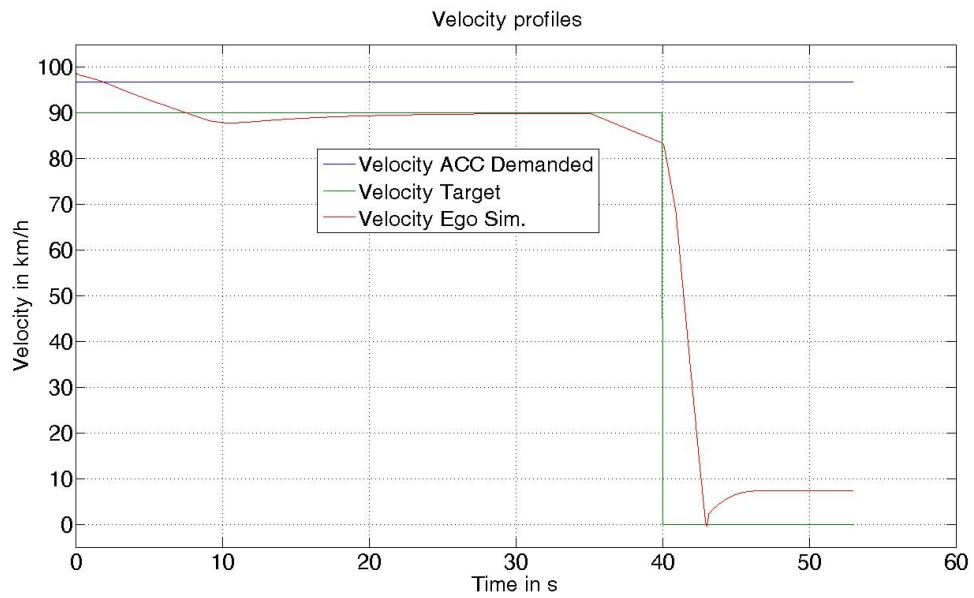


Figura D.2.: Response of the AEB system when the ACC is switched off

En la figura D.2 se ilustra el caso en el que el AEB actúa aun cuando el sistema ACC está desactivado. En esta situación el vehículo Ego se mueve a una velocidad inicial de 100km/h con el ACC activado, otro vehículo está circulando enfrente por lo que se ve obligado a frenar y amoldar su velocidad a la del Target. Alrededor del segundo 35 el conductor apaga el ACC (puede verse que la velocidad disminuye ya que el conductor tampoco pisa el acelerador). En el segundo 40 el vehículo Target se detiene por completo por el motivo que sea, obligando al AEB a actuar.

Puede verse que la frenada se realiza con éxito incluso cuando el ACC está apagado.

E. Apéndice

Cálculo del offset:

En la sección 3.2.5 resolvimos dos casos concretos para la ecuación del offset, sin embargo en este anexo analizamos más situaciones posibles.

En las imágenes E.1 y E.2 se pueden ver todos los datos tomados para los diferentes casos. N representa el número del experimento, por lo que se puede ver que tenemos 2 medidas para cada situación concreta. En todo cálculo el procedimiento es el mismo:

- 1- Se identifica dónde la situación es estacionaria (ambos coches conducen a la misma velocidad y guardando una distancia constante).
- 2- En ese momento se mide la distancia entre ambos coches y la velocidad del Target.
- 3- Una vez tenemos esto se calcula el offset con la ecuación: $d_{offset} = d_{soll} - \tau \cdot V_{target}$
- 4- Se busca el siguiente punto con situación estacionaria y se procede de igual manera.
- 5- Una vez tenemos ambos valores de offset para ambas velocidades se establece una correlación entre el offset y la velocidad del vehículo Ego.

Se ha supuesto que el offset depende de la velocidad del Ego y no de la del Target porque en la ecuación E.1 ya hay una contribución debida a la velocidad del Target, por lo tanto el offset debe ser un valor relacionado con el Ego.

$$d_{soll} = offset + \tau \cdot V_{target} \quad (E.1)$$

Para $\tau = 1$ la correlación resultante del offset en función de la velocidad del coche Ego (V_{Ego}) para los distintos modos se muestra en la ecuación E.2 para modo dinámico, la ecuación E.3 para confortable y la ecuación E.4 para el estándar. Una descripción gráfica de estas tres ecuaciones se muestra en la figura E.4.

$$offset_{dy} = -0,07 \cdot V_{Ego} + 7,69 \quad (E.2)$$

$$offset_{cm} = -0,074 \cdot V_{Ego} + 8,12 \quad (E.3)$$

$$offset_{st} = -0,088 \cdot V_{Ego} + 8,97 \quad (E.4)$$

Por otro lado, para $\tau = 4$ las correlaciones se muestran en las ecuaciones E.5 para modo dinámico, ecuación E.6 para comfortable y la ecuación E.7 para estandare.

STANDARD								
N	tauMode	tau_real	time	V	DIST	OFF		
11	1	1	60	57.65	20.1	4.09	tau1 ST	
	1	1	91	77.29	23.78	2.31		
12	1	1	46	57.5	19.66	3.69		
	1	1	79	77.16	23.42	1.99		
DYNAMIC								
N	tauMode	time gap	time	V	DIST	OFF		
00	1	1	38	57.76	19.58	3.54	tau 1 DY	
	1	1	70	76.82	23.65	2.31		
02	1	1	52	57.73	19.75	3.71		
	1	1	80	77	23.63	2.24		
COMFORT								
N	tauMode	tau_real	time	V	DIST	OFF		
04	1	1	40	57.7	19.87	3.84		
	1	1	78	76.79	23.7	2.37		
05	1	1	42	57.52	20.02	4.04	tau1 CM	
	1	1	73	77.34	24.06	2.58		
06	1	1	48	57.85	19.72	3.65		
	1	1	84	77.25	23.73	2.27		

Figura E.1.: Measures for different modes using $\tau = 1$

STANDARD								
N	tauMode	tau_real	time	V	DIST	OFF		
13	4	2.3	42	57.67	39.05	2.21	tau4 ST	
	4	2.3	73	77.13	53.94	4.66		
14	4	2.3	40	57.6	37.93	1.13		
	4	2.3	72	77.09	53.6	4.35		
DYNAMIC								
N	tauMode	time gap	time	V	DIST	OFF		
01	4	2.3	39	57.87	38.42	1.45	tau 4 DY	
	4	2.3	71	77.16	52.93	3.63		
03	4	2.3	41	57.6	39.23	2.43		
	4	2.3	62	77.02	53.21	4.00		
COMFORT								
N	tauMode	tau_real	time	V	DIST	OFF		
09	4	2.3	64	57.71	39.27	2.40	tau4 CM	
	4	2.3	83	77.04	55	5.78		
010	4	2.3	54	57.4	38.85	2.18		
	4	2.3	78	76.9	55	5.87		

Figura E.2.: Measures for different modes using $\tau = 4$

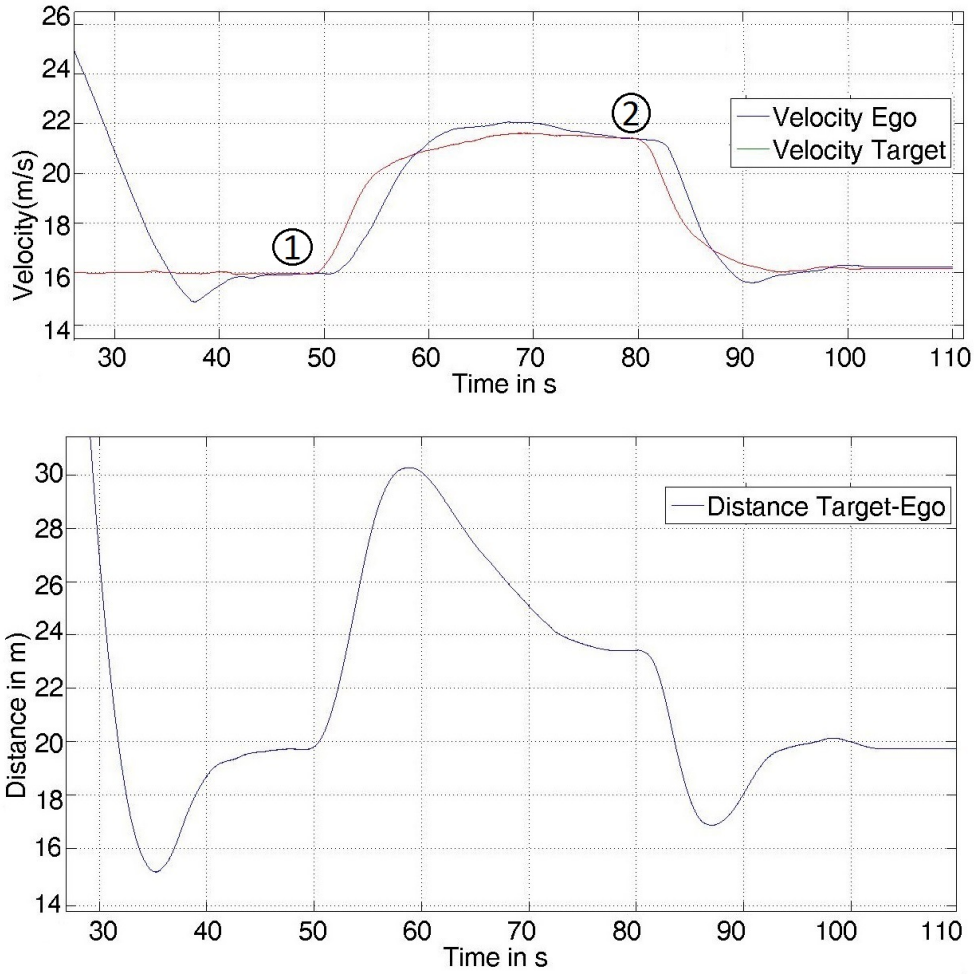


Figura E.3.: Graphics used to measure the distance d_{soll} for standard model and time gap $\tau = 1$ in the steady state.

$$offset_{dy} = 0,09 \cdot V_{Ego} - 3,64 \quad (E.5)$$

$$offset_{cm} = 0,18 \cdot V_{Ego} - 8,19 \quad (E.6)$$

$$offset_{st} = 0,15 \cdot V_{Ego} - 6,74 \quad (E.7)$$

Es importante decir que para el caso $\tau = 4$ y el modo comfortable las mediciones no son exactas. Las dos medidas de la distancia que están marcados en la figura E.2 no son 100 % fiables porque cuando el vehículo Target comenzó a frenar de 80 km/h a 60 km/h la situación había alcanzado el equilibrio (punto número 2 en el gráfico).

El comportamiento final del offset se muestra en las figuras E.4 and E.5. Como se explicó

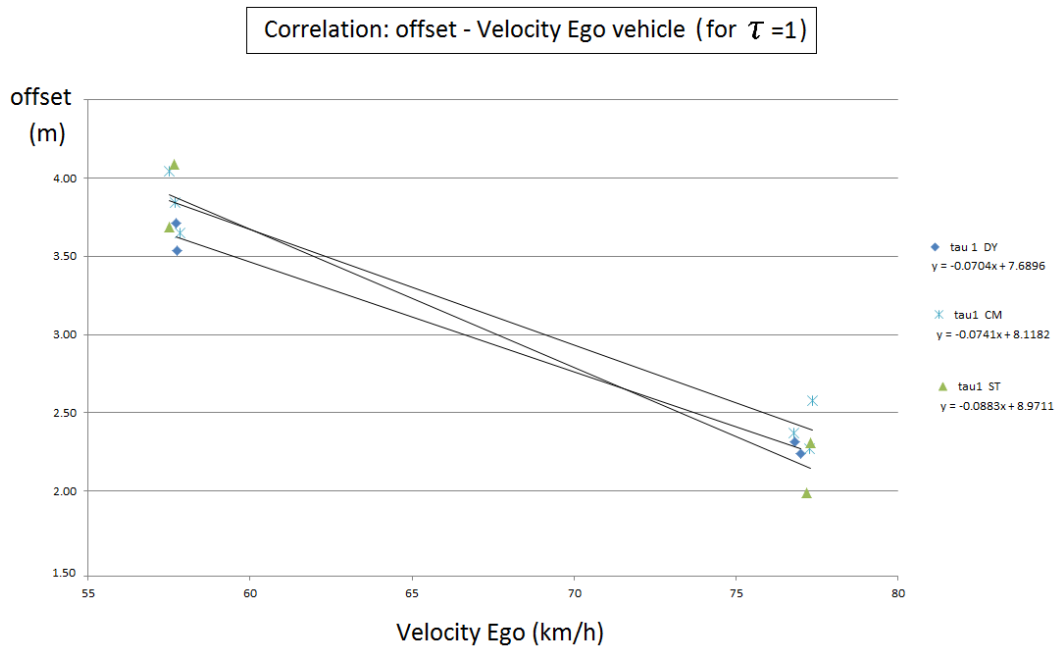


Figura E.4.: Correlation between offset and the velocity of the Ego vehicle for diferent modes keeping the same $\tau = 1$.

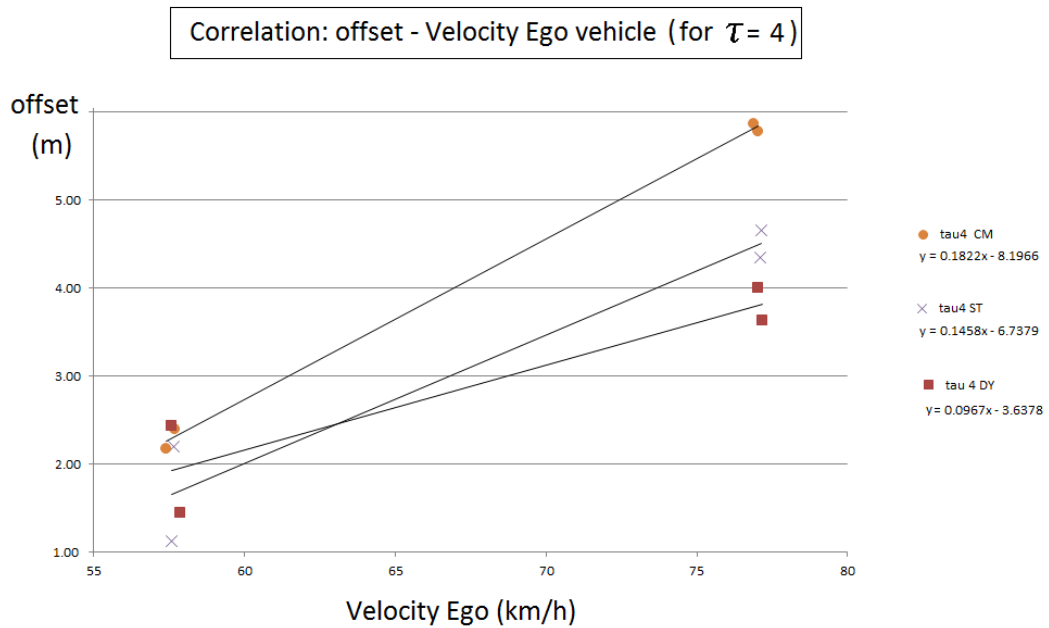


Figura E.5.: Correlation between offset and the velocity of the Ego vehicle for diferent modes keeping the same $\tau = 4$.

en la sección 3.2.5 dos límites, superior e inferior deben ser establecidos, aunque en esta figura no se muestren.

F. Apéndice

Rotación de vehículos:

En este apéndice se resuelve un caso particular de la rotación de vehículos explicada en la sección 2.2.1. Para este ejemplo tenemos el vehículo Ego situado en (2,2.5) con un ángulo θ_{Ego} de 40 grados, y un vehículo Target en (8,3) con un ángulo de 5 grados (ψ). Estos valores son los input que el radar recibe y están referidos a un sistema de coordenadas global por lo que la situación es la descrita en la figura F.1 (izquierda).

El vector a se define en la ecuación F.1

$$a = [8 - 2, 3 - 2,5] = [6, 0,5] \quad (\text{F.1})$$

De aquí se calcula el módulo de a es $|a| = 6,02$ y el ángulo con respecto al sistema global $\varphi = 4,7$. Por lo tanto el ángulo con respecto al sistema relativo al Ego se define en la ecuación F.2.

$$\phi = \varphi - \theta_{Ego} = 4,7 - 40 = -35,3 \quad (\text{F.2})$$

Una vez tenemos estos valores ya podemos situar el centro de gravedad del Target referido a las coordenadas locales del Ego (figura F.1 derecha) que para este caso es:

$$x_c = |a| \cdot \cos(\phi) = 4,9 \quad (\text{F.3})$$

$$y_c = |a| \cdot \sin(\phi) = -3,48 \quad (\text{F.4})$$

Por último solo queda rotar los corners del coche. Si antes hemos dicho que $\psi = 5$ ahora necesitamos calcular el ángulo relativo. Este ángulo le llamamos β_1 y responde la ecuación F.6

$$\beta_1 = \psi - \theta_{Ego} = 5 - 40 = -35 \quad (\text{F.5})$$

Por lo tanto tras la rotación del centro de gravedad el coche queda como en la figura F.2 (vehículo negro), y para completar el giro tenemos que rotar cada corner un ángulo igual a -35 grados, y de esta forma el coche queda completamente girado, (vehículo rojo).

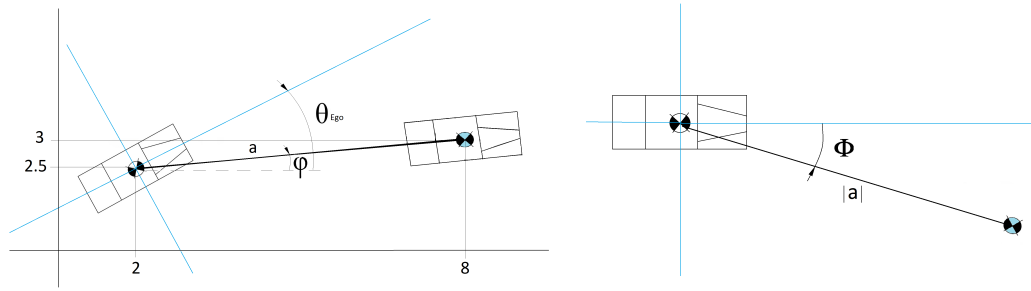


Figura F.1.: Rotation of the gravity center of the vehicle

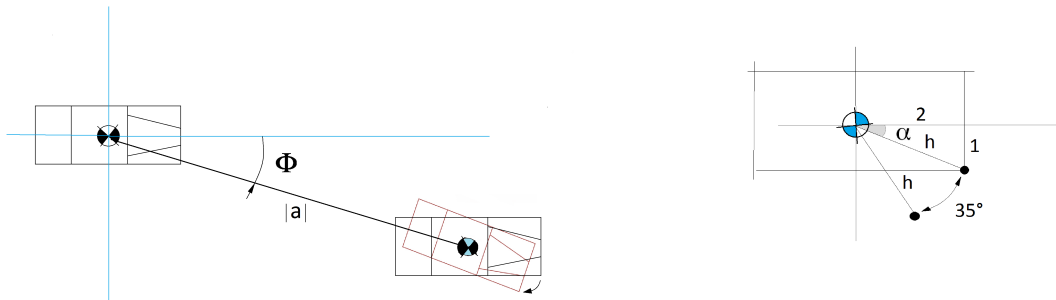


Figura F.2.: Rotation of the corners of the vehicle (left) and one only corner (right).

En la figura F.2 (derecha) podemos ver cuáles son los valores que necesitamos para calcular las nuevas coordenadas de cada punto. Para el punto señalado en el dibujo (corner número 2 del coche) se calcula primero el valor:

$$\alpha = \arctg(1/2) = 26,5 \quad (\text{F.6})$$

Este valor se le suma al ángulo al que está orientado el vehículo Target con respecto al Ego (35 grados). Por lo tanto el nuevo ángulo que nos interesa es $\beta = -26,5 - 35 = -61,5$.

De aquí se calcula la posición del corner ya girado con respecto al centro de gravedad del vehículo Target, tal y como se muestra a continuación:

$$x_2 = h \cdot \cos(\beta) = 1,07 \quad (\text{F.7})$$

$$y_2 = h \cdot \sen(\beta) = -1,96 \quad (\text{F.8})$$

$$h = (2^2 + 1)^{1/2} = 2,23 \quad (\text{F.9})$$

Estos valores de x_2 e y_2 son referidos al CG del vehículo target, por lo que si queremos saber la coordenada x e y con respecto al centro del vehículo Ego solo hay que sumar los valores anteriores.

$$X_2 = 1,07 + 4,9 = 5,97 \quad (\text{F.10})$$

$$Y_2 = -1,96 - 3,48 = -5,44 \quad (\text{F.11})$$

$$(\text{F.12})$$

G. Apéndice

Matriz Cars:

Como se ha descrito en el capítulo 2.2.2 los únicos datos relevantes son los referidos al punto de la superficie. Por lo tanto la matriz Cars guarda la información de ese punto para cada vehículo. Las filas de la matriz representan los diferentes vehículos y las columnas se explican a continuación:

- **distancia** (d): medida desde donde se encuentra el radar (en la parte frontal del vehículo Ego) hasta el susodicho punto superficie.
- **ángulo** (φ): el ángulo donde el punto superficie se encuentra, en coordenadas relativas al vehículo Ego.
- **velocidad** (v_m): módulo de la velocidad del vehículo Target (relativa al vehículo Ego).
- **ángulo de la velocidad** (γ_{vel}): ángulo del vector velocidad relativa al vehículo Ego.

La fila del vector Cars respectiva al vehículo de la figura G.1 sería: $d=20\text{m}$, $\varphi=15^\circ$, $v_m=1.17$ y $\gamma_{vel}=0.21$. Los cálculos realizados para la velocidad relativa se muestran en la figura G.2.

Un ejemplo del conjunto de la matriz Cars se muestra en la figura G.3, donde tenemos una distribución de los vehículos que rodean el coche Ego. Los coches de la matriz que se crean a partir de esa situación no tienen en cuenta si son detectados por el radar o

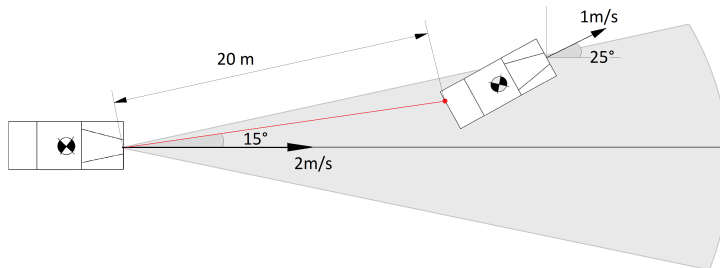


Figura G.1.: One car example

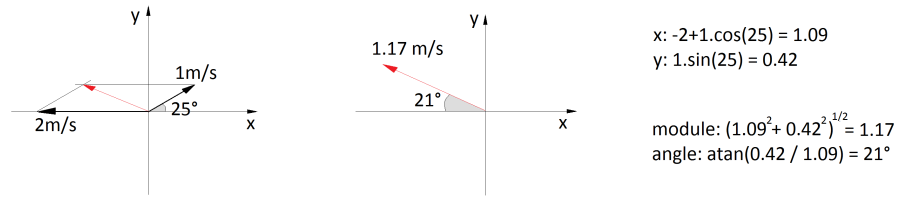


Figura G.2.: Relative velocity

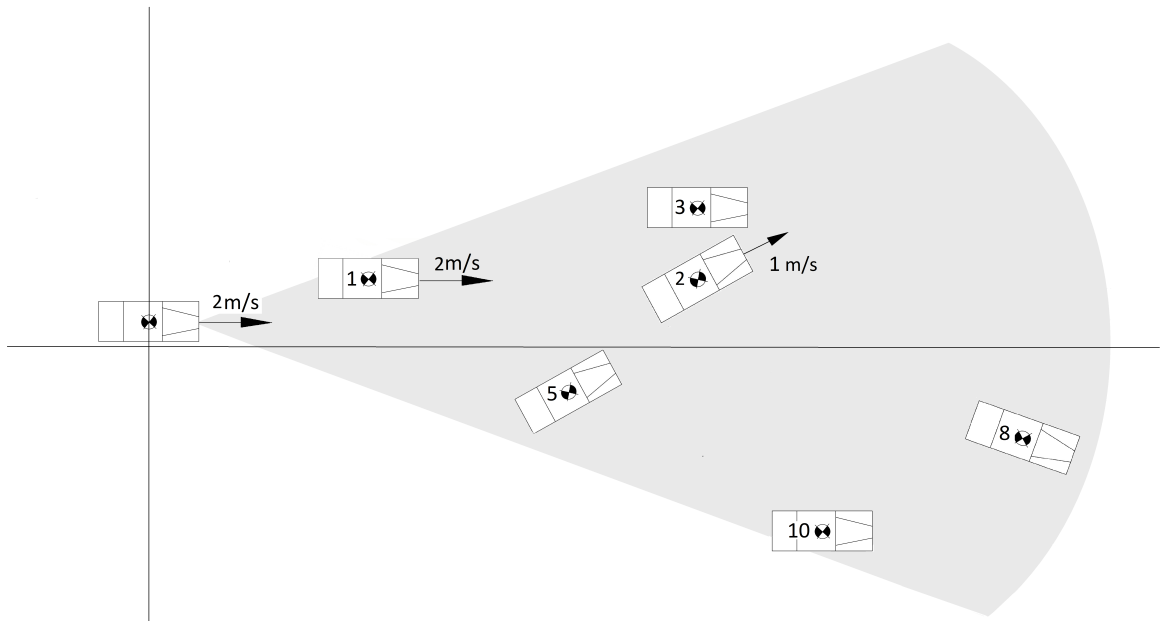


Figura G.3.: Example of situation

no, esta matriz sólo se utiliza para guardar la información de cada vehículo en relación con la distancia relativa, el ángulo y la velocidad.

El input para esta situación se añade en la figura G.4, donde X e Y representan la posición del centro de gravedad del vehículo, h es el ángulo en el que el vehículo está orientado y v la velocidad (todos ellos con relación a un sistema de coordenadas global). Como puede verse en la figura todos los vehículos tienen una velocidad igual a cero excepto el primero, que tiene una velocidad igual a dos como el vehículo Ego.

La matriz Cars resultante se muestra en la ecuación G.1. Si nos fijamos en la tercera columna de la primera fila, la cual representa la velocidad relativa del vehículo 1, podemos ver que la velocidad es cero, esto se debe a que tanto el vehículo 1 como el vehículo Ego tienen la misma velocidad y orientación. Por otro lado si analizamos la segunda fila de

```

%      Obstacle(X Y h v )
obs1 = Obstacle(8, 4, 0, 2);
obs2 = Obstacle(24, 8, 0.436, 1);
obs3 = Obstacle(21, 15, 0, 0);
obs4 = Obstacle(-10, 10, -0.463, 0);
obs5 = Obstacle(16, -5, 0.463, 0);
obs6 = Obstacle(-30, 35, pi/8, 0);
obs7 = Obstacle(-20, -10, 0, 0);
obs8 = Obstacle(90, -10, -0.463, 0);
obs9 = Obstacle(-40, 20, -0.436, 0);
obs10 = Obstacle(40, -30, 0, 0);

%Radar      X Y h ap length v
radar = Radar(0, 2, 0, pi/2, 70, 2);

```

Figura G.4.: data input

esta matriz, encontramos el vehículo anteriormente representado en la figura G.1. Como se ha calculado anteriormente, el punto superficie de este vehículo se encuentra a una distancia de 20 metros, un ángulo de 15 grados y la velocidad relativa es 1,17 con un ángulo de 2,77 rad (160 grados).

Por el contrario, el resto de vehículos tienen una $v_m = 2$, debido a que su velocidad absoluta es igual a cero y la velocidad del vehículo ego es 2. Por lo tanto todos ellos se están moviendo hacia el vehículo Ego en un sistema relativo de coordenadas.

$$\text{Cars} = \begin{bmatrix} 3,1623 & 18,4349 & 0 & 0 \\ 20,7743 & 15,2577 & 1,1723 & 2,775 \\ 20,5984 & 34,3803 & 2,00 & 3,1416 \\ 12,4065 & 150,2538 & 2,00 & 3,1416 \\ 13,6525 & -26,5651 & 2,00 & 3,1416 \\ 44,8917 & -45,8814 & 2,00 & 3,1416 \\ 22,7818 & 28,6105 & 2,00 & 3,1416 \\ 86,79590 & -7,7652 & 2,00 & 3,1416 \\ 43,6936 & -23,1986 & 2,00 & 3,1416 \\ 47,0643 & -40,1009 & 2,00 & 3,1416 \end{bmatrix} \quad (\text{G.1})$$

H. Apéndice

Array Shadows:

En este apéndice se explica con más detalle como se calculan los ángulos a los que se encuentran las esquinas de cada vehículo.

Debido a que sabemos las coordenadas de todas las esquinas con respecto al vehículo Ego: x_1, y_1, x_2, y_2 , etc (explicado en la sección 2.2.1) el cálculo que se necesita es simple y puede verse en las ecuaciones H.1 a H.4 para las esquinas del 1 al 4, respectivamente.

$$\varphi_1 = \text{arctg}(y_1/x_1) \tag{H.1}$$

$$\varphi_2 = \text{arctg}(y_2/x_2) \tag{H.2}$$

$$\varphi_3 = \text{arctg}(y_3/x_3) \tag{H.3}$$

$$\varphi_4 = \text{arctg}(y_4/x_4) \tag{H.4}$$

Una vez que tenemos estos cuatro ángulos, la sombra será creada desde el ángulo mínimo hasta el ángulo máximo de todos ellos. Una descripción gráfica se muestra en la figura H.1 donde $\varphi_{min} = \varphi_2$ y $\varphi_{max} = \varphi_3$.

Para el ejemplo mostrado en la figura H.1 los datos relevantes que se utilizan para crear la sombra son φ_2, φ_3 y la distancia a la segunda esquina más cercana, en este caso el

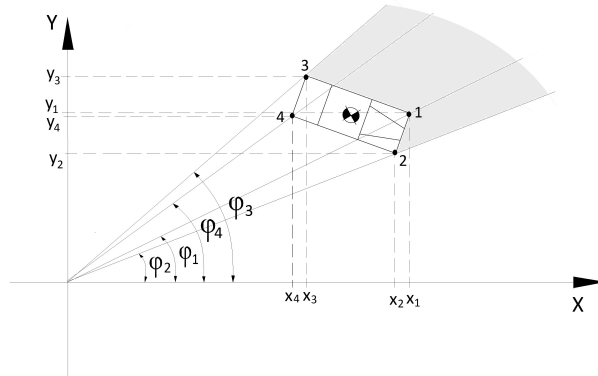


Figura H.1.: 4 different angles in a car.

número 3: $D_3 = (x_3^2 + y_3^2)^{1/2}$ por lo que estos tres datos se guardan en las tres columnas de la matriz Shadows.

I. Apéndice

Código del sensor radar:

Este apéndice muestra el código que se ha utilizado en el diseño del sensor radar. Está dividido en dos bucles diferentes.

El primer bucle lee la información de los coches que están rodeando al vehículo Ego, a partir de la cual crea las dos matrices *Cars* Y *Shadows*. El segundo bucle utiliza ambas matrices para determinar qué vehículos son detectados por el radar y cuáles no. El código se muestra en las imágenes de I.1 hasta I.9.

```

function [Ident1a,d1,a1,s1,as1,T1, Ident2a,d2,a2,s2,as2,T2,
Ident3a,d3,a3,s3,as3,T3,Ident4a,d4,a4,s4,as4,T4,Ident5a,d5,a5,s5,as5,T5,Ident6
a,d6,a6,s6,as6,T6,Ident7a,d7,a7,s7,as7,T7,Ident8a,d8,a8,s8,as8,T8,
Ident9a,d9,a9,s9,as9,T9,Ident10a,d10,a10,s10,as10,T10] =
fcn(num,frontCar,Ident,line,xx,yy,h,ap,l,v,Ident1,line1, x1,y1,h1,v1,c1,
Ident2,line2,x2,y2,h2,v2,c2,Ident3,line3,x3,y3,h3,v3,c3,Ident4,line4,x4,y4,h4,
v4,c4,Ident5,line5,x5,y5,h5,v5,c5,Ident6,line6,x6,y6,h6,v6,c6,Ident7,
line7,x7,y7,h7,v7,c7,Ident8,line8,x8,y8,h8,v8,c8,Ident9,line9,x9,y9,h9,v9,c9,
Ident10,line10,x10,y10,h10,v10,c10)

Ident1a=1; Ident2a=Ident2; Ident3a=Ident3; Ident4a=Ident4;
Ident5a=Ident5; Ident6a=Ident6; Ident7a=Ident7; Ident8a=Ident8;
Ident9a=Ident9; Ident10a=Ident10;
ID=zeros(1,10);

d1=0;a1=0;s1=0;as1=0; d2=0;a2=0;s2=0;as2=0; d3=0;a3=0;s3=0;as3=0;
d4=0;a4=0;s4=0;as4=0;
d5=0;a5=0;s5=0;as5=0; d6=0;a6=0;s6=0;as6=0; d7=0;a7=0;s7=0;as7=0;
d8=0;a8=0;s8=0;as8=0;
d9=0;a9=0;s9=0;as9=0; d10=0;a10=0;s10=0;as10=0;

T1=0;T2=0;T3=0;T4=0;T5=0;T6=0;T7=0;T8=0;T9=0;T10=0;
lineCar=0;
Targets=zeros(2,10);
ka=0;

cars=zeros(10,4);
shadow=zeros(10,3);
front=frontCar;

if (num > 0)

    phi = h;
    Rz = [cos(phi) sin(phi); -sin(phi) cos(phi)];

    % LOOP NUMBER 1: different vehicles -----
    -----
    for counter = 1:1:num
        if counter==1
            xcar=x1;
            ycar=y1;
            hcar=h1;
            vcar=v1;
            ccar=c1;
        elseif counter==2
            xcar=x2;
            ycar=y2;
            hcar=h2;
            vcar=v2;
            ccar=c2;
        elseif counter==3
            xcar=x3;
            ycar=y3;
            hcar=h3;
            vcar=v3;
            ccar=c3;
        elseif counter==4

```

Figura I.1.: Código Radar 1.

```

        xcar=x4;
        ycar=y4;
        hcar=h4;
        vcar=v4;
        ccar=c4;
elseif counter==5
        xcar=x5;
        ycar=y5;
        hcar=h5;
        vcar=v5;
        ccar=c5;
elseif counter==6
        xcar=x6;
        ycar=y6;
        hcar=h6;
        vcar=v6;
        ccar=c6;
elseif counter==7
        xcar=x7;
        ycar=y7;
        hcar=h7;
        vcar=v7;
        ccar=c7;
elseif counter==8
        xcar=x8;
        ycar=y8;
        hcar=h8;
        vcar=v8;
        ccar=c8;
elseif counter==9
        xcar=x9;
        ycar=y9;
        hcar=h9;
        vcar=v9;
        ccar=c9;
elseif counter==10
        xcar=x10;
        ycar=y10;
        hcar=h10;
        vcar=v10;
        ccar=c10;
else
        xcar=0;
        ycar=0;
        hcar=0;
        vcar=0;
        ccar=[0 0; 0 0;0 0; 0 0];
end

delta_p_v = [xcar - xx; ...
            ycar - yy];

phi_t = phi + hcar;

%relative velocity of the Target car referred to Ego car
vr=[0 0];
v_x=v*cos(h); %velocity of EGO
v_y=v*sin(h);

```

Figura I.2.: Código Radar 2.

```
vcar_x=vcar*cos(hcar); %velocity of EGO
vcar_y=vcar*sin(hcar);
vr=[vcar_x-v_x vcar_y-v_y];

% relative position of target, distance between the center of the
% Ecar (Ego car) to the center of the Tcar
p_rel = Rz*delta_p_v;

% relative angle between cars delta_phi
delta_phi = hcar - h;

Rz_delta_phi = [cos(delta_phi) -sin(delta_phi); ...
                sin(delta_phi) cos(delta_phi)];

%rotation of corners in the car
%p_p1 is a 2x4 matrix to save new coord of every corner of the car
p_p1 = Rz_delta_phi*[ccar(1,1) ccar(2,1) ccar(3,1) ccar(4,1);...
                    ccar(1,2) ccar(2,2) ccar(3,2) ccar(4,2)];

% vector of every corner refered to the EGO car
% vector saves the distance of every corner, from the front of the car
% (where the radar is located) to every corner.
vector=[0 0 0 0;0 0 0 0];

%minus 'front' in x coordinate because the radar is in the front of
the car
%so that the x coordinate is 2 meters less than the original array
p_rel+p_p1

%due to the position of the radar in the FRONT of the Ecar
k=1;
for j=1:1:4
    vector(k,j)= p_rel(k)+p_p1(k,j)-front;
end

k=2;
for j=1:1:4
    vector(k,j)= p_rel(k)+p_p1(k,j);
end

% vector distance to every corner
distCorners=[0 0 0 0];

for j=1:1:4
    distCorners(j)= sqrt((vector(1,j))^2+(vector(2,j))^2);
end
distance=min(distCorners);

% angle calculation (we create the angle array) angle from front of
%the ego car to every corner in the Tcar
angle=[0 0 0 0];
```

Figura I.3.: Código Radar 3.

```

for c=1:1:4

    if ( vector(1,c)>=0 )
        %condition of TARGET car is not behind EGO car
        angle(c) = atan(vector(2,c) / vector(1,c));
    elseif ( vector(2,c)>=0 )
        angle(c) = pi + atan(vector(2,c) / vector(1,c));
    else angle(c) = -pi + atan(vector(2,c) / vector(1,c));
    end
end %the angles goes from 0 to 180 (up) and from 1 to -180 (down)

% create of array orden (this array define the order of the distance
% to every corner, for example if distance=[20 25 15 10] the orden
array
% will be orden=[4 3 1 2] the closest is the corner in the position 4,
then 3, 1, and 2.

orden=[0 0 0 0];
d=distCorners;

n=1;
k=1;
c=min(d);

while k<5
    for i=1:1:4
        if c==d(i)
            orden(k)=i;
            d(i)=1000;
            k=k+1;
        end
    end
    c=min(d);
end

%we define the array CG, from the front of the Ecar to the center of
the Tcar
%CG is refered to the Ecar!!

CG= [p_rel(1)-front p_rel(2)];

%we are gonna calculate the point on the surface of the car which is
%detected by the radar, the radar does not detect a corner of the car
or the
%central point but a point in the suface.

%It is calculated by the
%intersection of two lines, line 1:from front Ecar to center of Tcar
% line 2: from corner to corner (possibilities: 1..2 2..4 3..4
1..3)
% the point where these two lines intersect will be the detected point
by the radar
% we'll have 4 points (2 of them are in the surface of the car) we
choose the closest one.

%the first equation is always from 0,0 to the GC of the TARGET car

```

Figura I.4.: Código Radar 4.

```
xx1=0;
yy1=0;
xx2=CG(1);
yy2=CG(2);
A1=(yy1-yy2)/(xx1-xx2);
B1=yy1-xx1*A1;

%the second equation is made in this loop
dx=[1000,1000,1000,1000];
k=1;
for i=1:3:4      %1-2  1-3  4-2  4-3

    for j=2:1:3
        a1=vector(1,i);
        b1=vector(2,i);

        a2=vector(1,j);
        b2=vector(2,j);

        A2=(b1-b2)/(a1-a2);
        B2=b1-a1*A2;

        if a1==a2 %case when the Tcar is 0° or 90° refered to the Ecar
            x=a1;
        else x=(B2-B1)/(A1-A2);
        end

        if (a2>=a1)
            if (x<=a2) && (x>=a1)
                dx(k)=x;
                k=k+1;
            end
        else
            if (x>a2) && (x<a1)
                dx(k)=x;
                k=k+1;
            end
        end
    end
end

if CG(1)>0
rho = atan(CG(2)/CG(1));
else rho = pi + atan(CG(2)/CG(1));
end

% create array CARS, where every car is saved: distance(1), angle(2)
and
% relative velocity: module(3) and angle(4)

cars(counter,1)=min(abs(dx))/cos(rho);
cars(counter,2)=rho;
```

```

cars(counter,3)=sqrt(vr(1)^2+vr(2)^2);

alpha=0;
if vr(1)>0 %velocidad relativa
    alpha=atan(vr(2)/vr(1));
else alpha=pi+atan(vr(2)/vr(1));
end

annngle1=(alpha)*180/pi;
anggggle=(alpha-h)*180/pi;
cars(counter,4)=alpha-h;

% if vr(1)>0 %velocidad relativa
%     cars(counter,4)=atan(vr(2)/vr(1))-h+pi;
% else cars(counter,4)=2*pi-atan(vr(2)/vr(1))-h;
% end
% crate array SHADOW, every shadow is saved as: 2 angles and a
% distance
% the shadow is created as if starting from the second closest point:
orden(2)
% we only create a shadow if the car on positive values of X

if CG(1)>0
    shadow(counter,1)=min(angle);
    shadow(counter,2)=max(angle);
    shadow(counter,3)=distCorners(orden(2));
end

end
% END OF LOOP 1 -----
% -----
% -----

%-----LOOP NUMBER 2 ----loop to detect which cars are inside the range,
% including the shadow effect
for counter = 1:1:num
    shad=0;

    %we check if the car is in any of all shadows we have, from the shadow
    %created by car 5 to the one created by car10
    %if there is any shadow over the car (counter) the value shad won't be
    0

    for k = 1:1:num
        if k~=counter %we avoid car making shadow in itself (it happens)
            if ((cars(counter,2) > shadow(k,1))&&...
                (cars(counter,2) < shadow(k,2))&&...
                (cars(counter,1) > shadow(k,3)))
                shad=shad+1;
            end
        end
    end
end

end

```

Figura I.6.: Código Radar 6.

```
%we check if the car is inside the range, and not in shadow

if (cars(counter,1) <= 1) &&...
  ((abs(cars(counter,2))) <= ap/2) &&...
  (shad==0)

  ID(counter)=1;
  dd=cars(counter,1);
  aa=cars(counter,2);
  ss=cars(counter,3);
  as=cars(counter,4);

  if counter==1
    d1=dd;
    a1=aa;
    s1=ss;
    as1=as;
  elseif counter==2
    d2=dd;
    a2=aa;
    s2=ss;
    as2=as;
  elseif counter==3
    d3=dd;
    a3=aa;
    s3=ss;
    as3=as;
  elseif counter==4
    d4=dd;
    a4=aa;
    s4=ss;
    as4=as;
  elseif counter==5
    d5=dd;
    a5=aa;
    s5=ss;
    as5=as;
  elseif counter==6
    d6=dd;
    a6=aa;
    s6=ss;
    as6=as;
  elseif counter==7
    d7=dd;
    a7=aa;
    s7=ss;
    as7=as;
  elseif counter==8
    d8=dd;
    a8=aa;
    s8=ss;
    as8=as;
  elseif counter==9
    d9=dd;
    a9=aa;
```

Figura I.7.: Código Radar 7.

```

        s9=ss;
        as9=as;
    elseif counter==10
        d10=dd;
        a10=aa;
        s10=ss;
        as10=as;
    end

    %the car is outside the range: we make ident=0
else
    if counter==1
        Ident1a=0;
    elseif counter==2
        Ident2a=0;
    elseif counter==3
        Ident3a=0;
    elseif counter==4
        Ident4a=0;
    elseif counter==5
        Ident5a=0;
    elseif counter==6
        Ident6a=0;
    elseif counter==7
        Ident7a=0;
    elseif counter==8
        Ident8a=0;
    elseif counter==9
        Ident9a=0;
    elseif counter==10
        Ident10a=0;

    end

end

end

end

for counter=1:1:num %we go through every car to see which ones are inside
our line

    if counter==1
        lineCar=line1;
    elseif counter==2
        lineCar=line2;
    elseif counter==3
        lineCar=line3;
    elseif counter==4
        lineCar=line4;
    elseif counter==5
        lineCar=line5;
    elseif counter==6
        lineCar=line6;
    elseif counter==7
        lineCar=line7;
    elseif counter==8
        lineCar=line8;

```

Figura I.8.: Código Radar 8.

```
elseif counter==9
    lineCar=line9;
elseif counter==10
    lineCar=line10;
end

if (lineCar==line)&&(ID(counter)==1)    %INSIDE & same LINE
    Targets(1,counter)=1;
    Targets(2,counter)=cars(counter,1);
end

end

distC=1000;
for counter=1:1:num
    if (Targets(1,counter)==1)&&(Targets(2,counter)<distC)
        distC=Targets(2,counter);
        ka=counter;
    end

end    % at the end of this loop we have ka: number of the closest car
      % inside the range and in the same line

% OUTPUT
if ka==0
    T1=0; T2=0; T3=0; T4=0; T5=0; T6=0; T7=0; T8=0; T9=0; T10=0;
elseif ka==1
    T1=1;
elseif ka==2
    T2=1;
elseif ka==3
    T3=1;
elseif ka==4
    T4=1;
elseif ka==5
    T5=1;
elseif ka==6
    T6=1;
elseif ka==7
    T7=1;
elseif ka==8
    T8=1;
elseif ka==9
    T9=1;
elseif ka==10
    T10=1;
end

end
end
```

Figura I.9.: Código Radar 9.

J. Apéndice

Modelos de simulink (ACC y AEB):

Este apéndice muestra imágenes de los diagramas de bloques que han sido implementados. En la imagen J.1 se muestra el conjunto del modelo, la parte importante de esta figura es la parte central, que coincide con lo descrito en la figura 3.2. En esta parte se encuentran el controlador de distancia, aceleración y velocidad.

El control de distancia es el más complejo de los tres, por lo que se muestra en la figura J.2 para una mejor comprensión. Los controladores descritos en la sección 3.2.4 se señalan con números, siendo 1: deceleración de seguimiento, 2: deceleración de aproximación, y 3: mantener el mínimo. Por último el número 4 corresponde al sistema AEB.

El sistema AEB está incluido dentro del controlador de distancia. La imagen correspondiente es J.3, donde puede diferenciarse claramente el bloque correspondiente al modo ciudad y al modo inter-urbano.

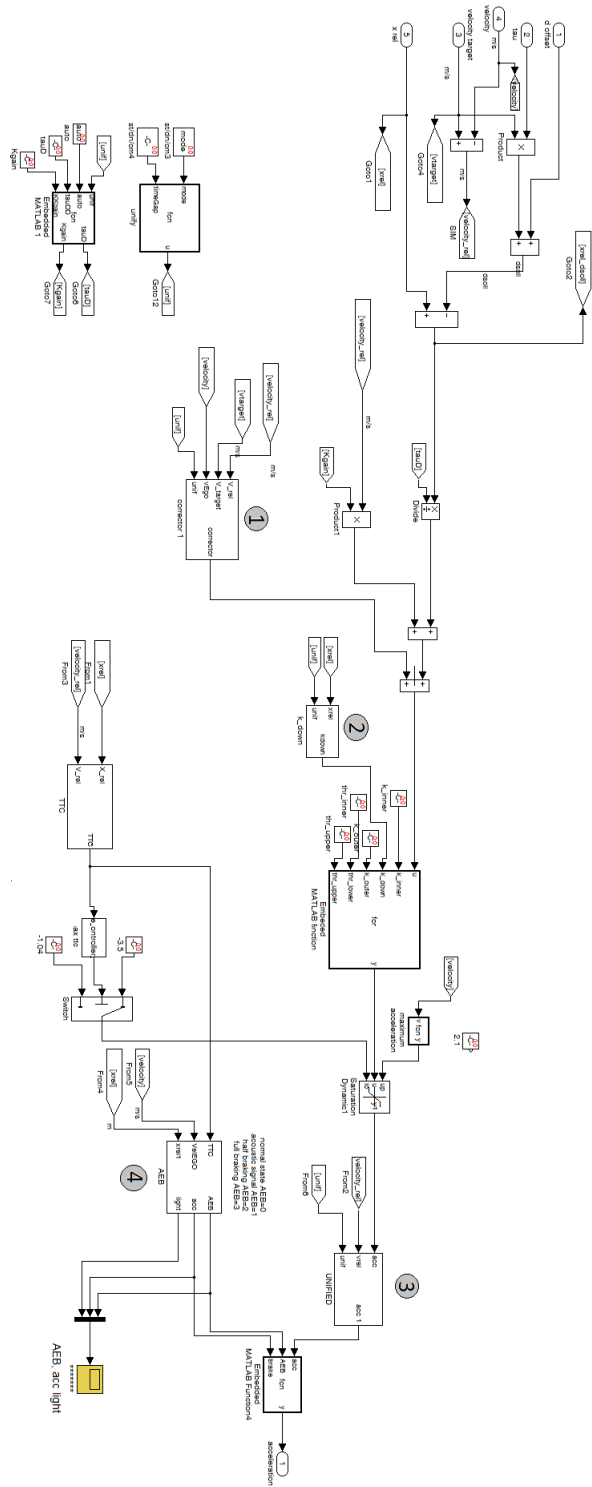


Figura J.2.: Controlador de distancia. Pueden verse los diferentes controles: 1 control de deceleración de seguimiento, 2 control de deceleración de aproximación y 3 mantener el mínimo. Además el número 4 corresponde al sistema AEB.

