

Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

Diseño de Estrategias de Control de Tracción para
Vehículos Eléctricos con Motores en Rueda

Autor: Ángel del Ojo Jiménez

Tutor: Carlos Bordons Alba

Dep. de Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2016



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías Industriales

Diseño de Estrategias de Control de Tracción para Vehículos Eléctricos con Motores en Rueda

Autor:
Ángel del Ojo Jiménez

Tutor:
Carlos Bordons Alba
Catedrático

Dep. de Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2016

Agradecimientos

Me gustaría agradecer a mi familia por su apoyo durante los años que he estudiado para conseguir terminar el grado. Por supuesto, también a mis amigos, y en especial a mi novia, pues siempre han estado ahí cuando los he necesitado.

A los miembros del *Fuel Cell Lab Control* me gustaría agradecerles su ayuda con todos los problemas que he ido encontrando. En especial, a David Marcos y a Gonzalo Hernández, con quienes he tenido el placer de trabajar más, desde el primer día me han ayudado a conocer este proyecto. Siempre han tenido tiempo para ayudarme con cualquier problema que me surgiera, y sus consejos han servido para dar forma a este trabajo y para aclararme muchos conceptos que desconocía y no entendía.

Por último, me gustaría agradecer a Carlos Bordons todo el tiempo que me ha dedicado y su orientación siempre que la he necesitado, no sólo en este proyecto, sino en todo lo relacionado con mi formación.

Gracias a todos.

Ángel del Ojo Jiménez

Sevilla, 2016

Resumen

Actualmente existe una gran preocupación a nivel global sobre las emisiones de CO₂ a la atmósfera. Uno de los principales contribuyentes a estas emisiones son los coches con motores de combustión interna alternativos, a los cuales se les impone cada vez más restricciones para poder ser comercializados. Esta evidente problemática abre el camino a nuevos diseños de motores y, por tanto, de vehículos. Una de estas alternativas a los vehículos tradicionales son los vehículos eléctricos.

Los vehículos eléctricos ofrecen muchas facilidades a la hora de ser controlados, pues se les puede imponer directamente el voltaje necesario para que den el par que demanda el conductor. En los vehículos tradicionales, sin embargo, existe una mayor cantidad de limitaciones, por lo que se suele actuar sobre el par de frenado que se aplica a las ruedas, lo cual hace que se gaste energía en el calentamiento de la rueda.

Dentro de este grupo de vehículos eléctricos, existen los vehículos con tracción total o tracción en las cuatro ruedas. Estos coches tienen cuatro motores eléctricos, uno por cada rueda, de manera que podemos controlar cada rueda de una manera más precisa que si se controlase únicamente un motor central. Esto aumenta la dificultad del sistema de control del vehículo, pero ofrece numerosas mejoras, como una mejora significativa de la estabilidad del vehículo, y un mejor aprovechamiento de la energía.

Este proyecto se va a basar en el modelo de un coche real, el FOX, que se encuentra en el laboratorio del departamento, lo que va a permitir diseñar controladores genéricos, pero caracterizados para este vehículo, de manera que luego pueda comprobarse su funcionamiento.

Los algoritmos de control de estabilidad que se van a desarrollar consisten en dos sistemas completos de control. La parte más compleja es un controlador de alto nivel, en el que se le dice el error respecto a algunas referencias, y actúa modificando el par (para lo cual se sirve del resto de partes del controlador) que debe dar a cada rueda mediante un controlador de bajo nivel, modificando el voltaje aplicado a las mismas. Para el cálculo de las referencias se usarán dos generadores de estas basados, el primero de ellos basado en datos de simulación del coche, y el segundo en las ecuaciones de la dinámica del modelo.

Abstract

Nowadays, there is a global concern about emissions to the environment. One of the main emissors are the vehicles motorized by internal combustion engines, which are being imposed more and more restrictions for being able to be sold in the markets. New solutions are being studied and created in order to solve this problem, one of which are the electric vehicles in substitution of the fuel-based vehicles.

These electric vehicles can be controlled in more ways than any other vehicle. It is possible to actuate directly on the voltage applied to the engine, which ends in a better control of stability and power consumption, whereas on a traditional car the control affects mainly wheel braking, which is less efficient, as it uses energy for braking, which results in tire heating.

There is a specific tipology of electric vehicles, the ones with four direct-driven in-wheel motors. These cars, as its own name says, have four motors, one per wheel, so we can make develop a more accurate control strategy compared to the control systems on cars with one central motor. Since there are four engines which need to be controlled, the difficulty is greatly increased, because there are more variables, which are related. The benefit of this kind of vehicles is that their stability is greatly improved, and the energy consumption is more efficient.

This project is going to be based on a real car model, named FOX, which is found on the laboratories of the department. This is going to let us design a generic control system, but specifically designed for this vehicle, so it can be tried in order to check its performance.

Each control algorithym that is going to be designed consists mainly on a high-level controller, which is fed with some errors compared to a reference, and acts (by using the other parts of the control system) varying the torque that each wheel needs to give, which is done by applying a calculated voltage to each wheel. For calculating references, two different generators are going to be used. The first one uses experimental data, and the second one uses vehicle dynamics equations.

Índice

Agradecimientos	v
Resumen	vi
Abstract	vii
Índice	ix
Índice de Figuras	xi
Índice de Tablas	xiv
Notación	xv
1 Introducción	1
1.1 <i>Motivación del Proyecto</i>	1
1.2 <i>Objetivos</i>	2
2 Estado del arte	3
2.1 <i>Tipología de Controladores</i>	3
2.1.1 Control Predictivo	3
2.1.2 Control Adaptativo	4
2.1.3 Control de Lógica Difusa	4
2.2 <i>Controladores específicos para Vehículos</i>	6
2.2.1 Control Antideslizamiento	6
2.2.2 Control de Estabilidad	7
3 Dinámica básica de vehículos	8
3.1 <i>Tracción</i>	8
3.2 <i>Ángulos</i>	9
3.3 <i>Momento</i>	9
3.4 <i>Deslizamiento</i>	10
3.5 <i>Viraje</i>	11
4 Vehículo FOX	12
4.1 <i>Características del vehículo</i>	12
4.1.1 Motores	13
4.1.2 Baterías	13
4.1.3 ECU	13
4.1.4 Sensores	13
4.2 <i>Modelo de Simulink</i>	14
5 Sistema de Control de Estabilidad Basado en Lógica Difusa	17
5.1 <i>Estructura del sistema de control</i>	17
5.2 <i>Generador de Referencias</i>	18
5.3 <i>Sistema de Control Superior</i>	18
5.3.1 <i>Conversión a Lógica Difusa (Fuzzification)</i>	18
5.3.2 <i>Proceso de Decisión Difusa (Fuzzy Decision Process)</i>	19

5.3.3	Conversión a Magnitudes Reales (Defuzzification)	21
5.3.4	Escalado (Output Scaling)	23
5.4	<i>Sistema de Control Inferior</i>	24
5.5	<i>Par Máximo Transmisible (MTTE)</i>	27
6	Sistema de Control de Estabilidad Basado en Controlador Proporcional	29
6.1	<i>Estructura del sistema de control</i>	29
6.2	<i>Sistema de Control Supervisor</i>	30
6.3	<i>Sistema de Control Superior</i>	31
6.3.1	Modo de maniobrabilidad ESC – γ	31
6.3.2	Modo de estabilidad ESC – β	32
6.4	<i>Sistema de Control Inferior</i>	33
6.4.1	Control de tracción	33
6.5	<i>Cálculo de Fuerzas en los Neumáticos</i>	33
7	Montaje de los Controladores en Simulink	35
7.1	<i>Diagrama de bloques del vehículo</i>	35
7.2	<i>Controlador de Lógica Difusa</i>	38
7.2.1	Diagrama de Bloques	38
7.2.2	Implementación Lógica Difusa en Simulink/Matlab	43
7.3	<i>Controlador Proporcional</i>	47
7.4	<i>Códigos auxiliares</i>	54
8	Resultados en Simulación	55
8.1	<i>Presentación de resultados</i>	55
8.2	<i>Giro brusco hacia la izquierda</i>	56
8.2.1	Descripción maniobra	56
8.2.2	Resultados Controlador lógica difusa	56
8.2.3	Resultados controlador proporcional	62
8.3	<i>Doble cambio de carril a 60 km/h</i>	67
8.3.1	Descripción maniobra	67
8.3.2	Resultados controlador lógica difusa	67
8.3.3	Resultados controlador proporcional	73
8.4	<i>Doble cambio de carril a 100 km/h</i>	74
8.4.1	Descripción maniobra	74
8.4.2	Resultados controlador lógica difusa	74
8.4.3	Resultados controlador proporcional	79
8.5	<i>Comparativa entre controladores</i>	83
8.5.1	Giro brusco hacia la izquierda	84
8.5.2	Doble cambio de carril a 100 km/h	88
9	Montaje del Controlador de Lógica Difusa en C (QNX)	94
9.1	<i>Código Matlab</i>	94
9.2	<i>Comparaciones de la simulación del modelo de bloques y el modelo en código</i>	99
9.3	<i>Código lenguaje C</i>	103
9.4	<i>Código C para QNX</i>	107
10	Conclusiones y Trabajo Futuro	108
10.1	<i>Conclusiones</i>	108
10.2	<i>Trabajo futuro</i>	109
	Referencias	110

ÍNDICE DE FIGURAS

Figura 2-1 Esquema de Control Adaptativo.	4
Figura 2-2 Intervalos de temperatura según lógica difusa.	5
Figura 2-3 Sistema ABS	6
Figura 3-1 Representación fuerzas en trayectoria recta	8
Figura 3-2 Ángulo de deslizamiento	10
Figura 3-3 Subviraje y Sobreviraje	11
Figura 4-1 Vehículo FOX	12
Figura 4-2 Bloques del vehículo	15
Figura 4-3 Ejemplos de algunos de los bloques empleados	15
Figura 4-4 Subsistema del FOX	16
Figura 5-1 Esquema del sistema de Control	17
Figura 5-2 Estructura controlador de lógica difusa	18
Figura 5-3 Posibilidad 1 de momento correctivo empleando frenos	20
Figura 5-4 Posibilidad 2 de momento correctivo empleando frenos	21
Figura 5-5 Funciones de pertenencia para el error del ángulo de deslizamiento	22
Figura 5-6 Funciones de pertenencia para el error en Yaw Rate	22
Figura 5-7 Funciones de pertenencia para la salida del sistema Mz	23
Figura 5-8 Resultado de Lógica Difusa	23
Figura 5-9 Esquema de fuerzas en par diferencial	25
Figura 5-10 Balance de fuerzas en una rueda	26
Figura 5-11 Variables de la dinámica longitudinal de un coche	28
Figura 5-12 Esquema de control de MTTE	28
Figura 6-1 Esquema de Control 2	29
Figura 6-2 Esquema de la activación de los modos de control	30
Figura 6-3 Modelo de la bicicleta	32
Figura 7-1 Bloques Goto, From y To Workspace	35
Figura 7-2 Bloque para el cálculo del par que pide el conductor	36
Figura 7-3 Interior del bloque “Conductor”	36
Figura 7-4 Modelo Coche sin control	36
Figura 7-5 Modelo final del simulador con todas las señales	37
Figura 7-6 Diagrama de bloques del controlador	38
Figura 7-7 Bloque Controlador	39
Figura 7-8 Bloque Generador de Referencia de Yaw Rate	39
Figura 7-9 Bloque “Fuzzy”	40

Figura 7-10 Bloque de Cálculo de las Fuerzas de Rozamiento	42
Figura 7-11 Bloque MTTE	42
Figura 7-12 Cálculo MTTE	43
Figura 7-13 Ventana para diseñar sistemas empleando Lógica Difusa	43
Figura 7-14 Ventana principal controlador Lógica Difusa	44
Figura 7-15 Funciones de pertenencia del error de deslizamiento	45
Figura 7-16 Funciones de pertenencia del error de yaw rate	45
Figura 7-17 Funciones de pertenencia de la salida	46
Figura 7-18 Implementación de reglas lingüísticas según el método Mamdani	46
Figura 7-19 Obtención de la salida de lógica difusa	47
Figura 7-20 Diagrama de bloques del segundo controlador en <i>Simulink</i>	48
Figura 7-21 Subsistema Inferior	49
Figura 7-22 Bloque Cálculo Momento, dentro del subsistema Inferior	50
Figura 7-23 Importancia Histéresis	50
Figura 7-24 Derivadas de una señal con ruido	52
Figura 7-25 Interior bloque Cálculo Pares, del subsistema Inferior	53
Figura 8-1 Entrada del volante primera simulación	56
Figura 8-2 Par de cada rueda Exp.1 Control 1	57
Figura 8-3 Aceleración transversal Exp.1 Control 1	57
Figura 8-4 Ángulo de deslizamiento del coche Exp.1 Control 1	58
Figura 8-5 Ángulo de deslizamiento de las ruedas Exp.1 Control 1	58
Figura 8-6 Deslizamiento porcentual ruedas Exp.1 Control 1	59
Figura 8-7 Yaw rate Exp.1 Control 1	59
Figura 8-8 Velocidad del vehículo Exp.1 Control 1	60
Figura 8-9 Trayectorias descritas Control 1	61
Figura 8-10 Par de cada rueda Exp.1 Control 2	62
Figura 8-11 Aceleración transversal Exp. 1 Control 2	63
Figura 8-12 Ángulo de deslizamiento del coche Exp. 1 Control 2	63
Figura 8-13 Ángulo de deslizamiento de las ruedas Exp. 1 Control 2	64
Figura 8-14 Deslizamiento porcentual ruedas Exp.1 Control 2	64
Figura 8-15 Yaw Rate Exp.1 Control 2	65
Figura 8-16 Velocidad del vehículo Exp. 1 Control 2	65
Figura 8-17 Trayectorias descritas control 2	66
Figura 8-18 Giro del volante en la maniobra de cambio de carril	67
Figura 8-19 Par de cada rueda Exp.2 Control 1	68
Figura 8-20 Aceleración transversal Exp.2 Control 1	69
Figura 8-21 Ángulo deslizamiento coche Exp.2 Control 1	69
Figura 8-22 Ángulo de deslizamiento de las ruedas Exp.2 Control 1	70
Figura 8-23 Deslizamiento porcentual ruedas Exp.2 Control 1	70

Figura 8-24 Yaw Rate Exp.2 Control 1	71
Figura 8-25 Yaw Rate aumentado Exp.2 Control 1	71
Figura 8-26 Velocidad del vehículo Exp.2 Control 1	72
Figura 8-27 Par de cada rueda Exp.2 Control 2	73
Figura 8-28 Yaw Rate Exp.2 Control 2	73
Figura 8-29 Ángulo de deslizamiento Exp.2 Control 2	74
Figura 8-30 Par de cada rueda Exp. 3 Control 1	75
Figura 8-31 Aceleración transversal Exp. 3 Control 1	75
Figura 8-32 Ángulo de deslizamiento del vehículo Exp. 3 Control 1	76
Figura 8-33 Ángulo de deslizamiento de cada rueda Exp. 3 Control 1	76
Figura 8-34 Deslizamiento porcentual de cada rueda Exp. 3 Control 1	77
Figura 8-35 Yaw Rate Exp. 3 Control 1	77
Figura 8-36 Velocidad del vehículo Exp. 3 Control 1	78
Figura 8-37 Par de cada rueda Exp. 3 Control 2	79
Figura 8-38 Aceleración transversal Exp. 3 Control 2	79
Figura 8-39 Ángulo de deslizamiento del vehículo Exp. 3 Control 2	80
Figura 8-40 Ángulo de deslizamiento de cada rueda Exp. 3 Control 2	81
Figura 8-41 Deslizamiento porcentual de cada rueda Exp. 3 Control 2	81
Figura 8-42 Yaw Rate Exp. 3 Control 2	82
Figura 8-43 Velocidad del vehículo Exp. 3 Control 2	82
Figura 8-44 Comparativa de par 1	84
Figura 8-45 Comparativa de aceleración transversal 1	85
Figura 8-46 Comparativa de ángulo de deslizamiento del vehículo 1	85
Figura 8-47 Comparativa de ángulo de deslizamiento de las ruedas 1	86
Figura 8-48 Comparativa de deslizamiento porcentual de las ruedas 1	86
Figura 8-49 Comparativa de Yaw Rate 1	87
Figura 8-50 Comparativa de velocidad del vehículo 1	87
Figura 8-51 Comparativa de trayectorias 1	88
Figura 8-52 Comparativa de par 2	89
Figura 8-53 Comparativa de aceleración transversal 2	90
Figura 8-54 Comparativa de ángulo de deslizamiento del vehículo 2	90
Figura 8-55 Comparativa de ángulo de deslizamiento de las ruedas 2	91
Figura 8-56 Comparativa de deslizamiento porcentual de las ruedas 2	91
Figura 8-57 Comparativa de Yaw Rate 2	92
Figura 8-58 Comparativa de Yaw Rate aumentada 2	92
Figura 8-59 Comparativa de velocidad del vehículo 2	93
Figura 9-1 Implementación buscada	94
Figura 9-2 Comparación entre implementaciones del par	100
Figura 9-3 Comparación entre implementaciones de la aceleración transversal	101

ÍNDICE DE TABLAS

Tabla 5-1 Términos Lingüísticos	19
Tabla 5-2 Reglas Lingüísticas	20
Tabla 9-1 Tabulación de Lógica Difusa	95

Notación

ECU	Electronic Control Unit
MPC	Model Predictive Control
ABS	Anti Blocking-System
ESP	Electronic Stability Program
MTTE	Maximum Transmissible Torque Estimation
ESC- γ	Electronic Stability Control - Manoeuvrability
ESC- β	Electronic Stability Control – Lateral Stability

1 INTRODUCCIÓN

1.1 Motivación del Proyecto

Uno de los grandes problemas de la sociedad actual son las emisiones de gases de efecto invernadero a la atmósfera. El *Protocolo de Kyoto*, un pacto firmado por los principales países industrializados en el marco de *United Nations Framework Convention on Climatic Change*, ahora adoptado por 125 países, es una clara prueba de esta preocupación global. En él, los países pertenecientes se comprometieron a reducir las emisiones de gases que provocan el efecto invernadero, en concreto de dióxido de Carbono (CO_2), gas Metano (CH_4), óxido Nitroso (N_2O) y otros gases del ámbito industrial principalmente, como el hexafluoruro de azufre, usado como aislante eléctrico.

Uno de los productores más numerosos de Dióxido de Carbono son los motores de combustión interna, presente en la mayoría de coches actualmente. En el año 2011, según datos del INE (Instituto Nacional de Estadística), el 12.1 % de las emisiones españolas de CO_2 fueron debidas a transporte, una cifra para nada insignificativa, comparada con el 27.2 % que genera la industria manufacturera, el mayor emisor según sus datos.

Estos motores no sólo emiten este gas, sino que también emiten Óxidos de Nitrógeno (NO_x), resultado de la combustión a alta temperatura en su interior. Este tipo de gases es especialmente nocivo para el ser humano, pues es un vasodilatador pulmonar, y se quedan allí retenidos, lo que empeora la salud de los ciudadanos.

En términos termodinámicos, son motores con una eficiencia muy baja, del orden de 25% para los que consumen gasolina (ciclo de Otto), y del orden del 35% para los que consumen diésel (ciclo de Diésel). Los motores eléctricos presentan una eficiencia del 90%, lo cual hace que, por cada unidad de energía aportada al motor, un motor eléctrico genere el triple de energía mecánica que un motor de combustión. Esto es capaz de reducir significativamente la cantidad de combustible que se necesita para hacer que los vehículos se desplacen, lo que acaba incidiendo en una reducción de emisiones nocivas y en un menor calentamiento del planeta.

Desde el punto de vista del usuario, los motores de combustión tienen un par variable, que depende principalmente de la velocidad de giro. Los vehículos eléctricos no presentan esta variabilidad del par, pues tienen un par aproximadamente plano en todo su rango de funcionamiento. Además, al poder ser actuables directamente aplicando un voltaje diferente al motor, son mucho más manipulables, por lo que podemos obtener mejores resultados en estabilidad y consumo energético si se implementan las estrategias de control pertinentes. Al actuar directamente sobre el motor, y no hacerlo en los frenos (como hacen la mayoría de coches actuales), se disipa menos energía, por lo que mejora la eficiencia energética del vehículo.

Por tanto, los motores eléctricos se están posicionando como candidatos a sustituir a los motores de combustión.

En este proyecto se va a desarrollar un controlador que mejore la estabilidad de los vehículos eléctricos, basándose en el vehículo experimental FOX. En una primera parte, se van a simular controladores, y luego se codificarán en el coche real, de manera que se podrá ver su funcionamiento y se podrán añadir mejoras, pues pasamos del problema ideal de la simulación al problema real de un vehículo físico.

1.2 Objetivos

El objetivo principal de este trabajo es diseñar dos algoritmos de control para vehículos eléctricos con tracción en las cuatro ruedas. Con ello queremos mejorar principalmente el control que tiene el conductor sobre el vehículo, pues el sistema de control va a ser capaz de ayudar al usuario en algunas maniobras, lo que hace que responda mejor el vehículo a lo que el conductor desee. Además, se pretende mejorar el consumo energético total del vehículo debido a la tracción del mismo, de manera que aumente su autonomía.

La primera parte consiste en la simulación e implementación de controladores de estabilidad en un modelo virtual del FOX en *Simulink*, *Matlab*, que se creó en la Tesis Doctoral de D. David Marcos Rodríguez [1]. En dicho modelo, se tienen unos bloques que simulan cada parte mecánica del coche y sus uniones, de manera que podemos acceder a cada motor y conectar la salida del controlador diseñado para ver su respuesta.

En el primer esquema de control propuesto, el controlador de alto nivel que calcula el par que necesita aportar cada rueda se basa en lógica difusa (Fuzzy Logic), del que más adelante se discutirá el motivo de su elección. A este controlador le entregamos unos errores de algunos parámetros del coche respecto de unas referencias calculadas por un controlador de alto nivel, y decide qué momento extra necesita generar el coche. Una vez tenemos ese momento adicional, se pasa ese dato a un controlador de bajo nivel, que decide en función del estado actual de cada rueda y del coche, qué par se debe añadir a cada rueda, y qué par restar a las ruedas necesarias, de manera que el coche consiga ese momento adicional para conseguir la trayectoria que el conductor pide.

Para el cálculo de las referencias antes nombradas, se usarán datos experimentales del coche, para obtener una referencia que sea propia del vehículo real en la que vamos a probarlo.

La segunda parte del proyecto relacionado con este sistema de control consiste en pasar del controlador en el entorno de *Simulink* a un código en lenguaje C siguiendo el código principal que gestiona la ECU (Electronic Control Unit) del coche, que emplea el sistema operativo en tiempo real QNX.

El segundo esquema de control será parecido en cuanto a forma al anterior, pero con varios niveles: un supervisor que escoge el modo de control necesario, un sistema superior que calcula las referencias para el controlador, uno inferior para calcular el par que se debe aplicar a cada rueda en función de la demanda del conductor y de los errores respecto a las referencias, y un sistema de control de tracción para mejorar la tracción del vehículo. La topología de este controlador será la de un controlador proporcional, y de la misma manera que con el primer controlador, luego se justificará su elección.

2 ESTADO DEL ARTE

2.1 Tipología de Controladores

Existe una amplia gama de controladores en la actualidad. A continuación se presentarán algunos que se han analizado en una primera etapa de selección de qué tipo de controlador se iba a emplear.

2.1.1 Control Predictivo

Los controladores predictivos (Model Predictive Control o MPC) gozan de un gran éxito tanto en el campo de la industria como en el de la comunidad investigadora. Esto es debido, principalmente, a su gran cantidad de ventajas [2].

Entre las ventajas del MPC se pueden destacar:

- Formulación en el dominio del tiempo, flexible, abierta y bastante intuitiva.
- Posibilidad de tratar con una gran variedad de sistemas, sean lineales o no lineales, monovariantes o multivariantes, usando para todos la misma formulación.
- La ley de control responde a criterios óptimos.
- Posibilidad de incorporación de restricciones en el diseño.

Se aprecia que son ventajas que hacen del MPC un tipo de control con muchas posibilidades, pero presenta algunas desventajas a tener en cuenta:

- Requiere el conocimiento de un modelo dinámico del sistema suficientemente preciso.
- Requiere un alto coste computacional, lo que puede hacer que no sea óptimo para sistemas donde se tiene poco tiempo para calcular la salida del controlador.

En principio se postula como un posible controlador para este proyecto, pues tiene una formulación que funciona bien para un sistema tan complejo como un vehículo con cuatro motores, que es un sistema sobreactuado. Sin embargo, las desventajas que presenta son muy acusadas en nuestro caso, pues el modelo de un vehículo es, hasta cierto punto, una aproximación, pues hay algunos componentes, como las ruedas, que son bastante complejas de modelar, y los modelos no representan con una precisión absoluta la realidad.

El problema del coste computacional actualmente no es un problema tan grave, pues cada vez la electrónica es más rápida, y si se hace un buen diseño del controlador, no debería dar problemas de especificaciones temporales. Aún así, no hay que olvidar que nuestro sistema se basa en un sistema operativo de tiempo real, donde el tiempo que tardan las acciones en ejecutarse es crucial.

No obstante, es importante tener en cuenta este tipo de control a la hora de elegir qué controlador se quiere emplear, pues existen numerosos sistemas de control de estabilidad para vehículos eléctricos basados en él.

2.1.2 Control Adaptativo

Muchas veces se presenta el problema de tener diseñado un controlador, pero aparecen cambios en la dinámica del sistema y perturbaciones que hacen que nuestro control no sea del todo eficiente. Ante este tipo de problemas surge el Control Adaptativo. Estos problemas son bastante frecuentes en un coche, pues en una conducción normal aparecen numerosas perturbaciones y cambios en la dinámica no planeados, como ráfagas de viento, variabilidad del coeficiente de deslizamiento del suelo y calentamiento de motores que hace que respondan diferente a cuando están más fríos, entre otros.

Este tipo de control crea una estructura en la que se adapta el controlador a la salida del sistema, de manera que si no es la que se calculó (debido a perturbaciones entre otras cosas), se cambian los parámetros del controlador para conseguir que el sistema responda como queríamos. El esquema [3] que se propone es el siguiente:

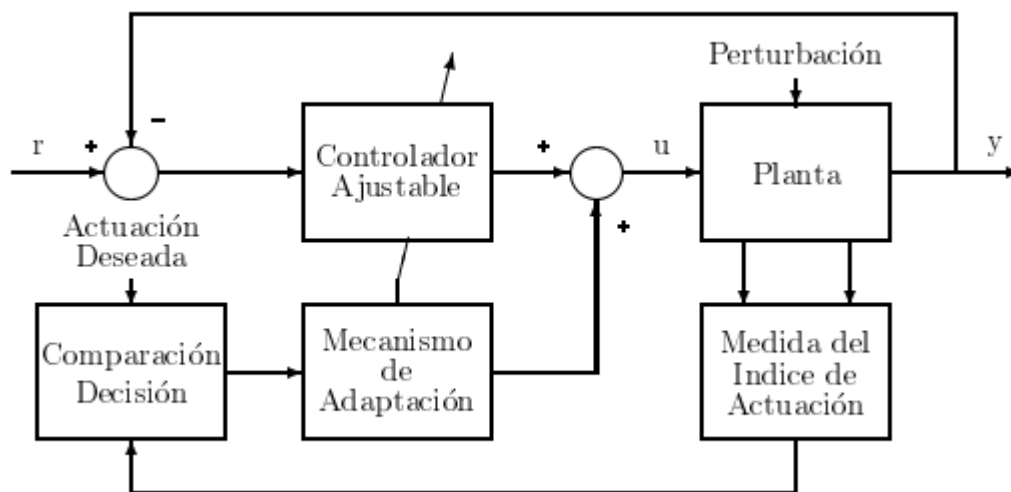


Figura 2-1 Esquema de Control Adaptativo.

Por consiguiente, podemos pensar en este tipo de controlador para nuestro proyecto, pues parece que arregla una gran cantidad de los problemas que podría presentar el control de un coche eléctrico. Sin embargo, presenta dos grandes desventajas:

- Necesitamos conocer el vector de estados del sistema, que no siempre es accesible.
- Los parámetros del conjunto planta-controlador deben ser modificables, cosa que no siempre es cierto.

Por tanto, es un posible controlador, pero la necesidad de modificar los parámetros puede hacerlo un poco complicado de implementar.

Actualmente se usa en algunos sistemas de ayuda a la conducción, como el crucero adaptativo, pero hay bastante discrepancia sobre su utilidad debido a las limitaciones del mismo, que no son objeto de este trabajo.

2.1.3 Control de Lógica Difusa

Finalmente, llegamos a un tipo de control basado no tanto en ecuaciones físicas, sino más en resultados experimentales y en aplicación de reglas basadas en la lógica [4].

La lógica difusa pretende servir como solución para diseñar controladores de sistemas complejos con una sencillez mayor que si se diseñaran con técnicas tradicionales. Cabe mencionar que surge en la década de 1960 (ideado por el ingeniero Lofti Zadeh), cuando la electrónica no era muy avanzada, y se necesitaba controlar sistemas complejos con pocos recursos computacionales. Para ello, actúa con reglas lógicas, de manera similar al pensamiento humano. En este ámbito del conocimiento, no existen las afirmaciones totalmente verdaderas o totalmente falsas, sólo existen grados de pertenencia a conjuntos definidos. Por ejemplo, la temperatura puede

ser medida y dar un dato, 29°C. La lógica difusa propone analizar ese dato como una pertenencia a varios conjuntos, definidos en la siguiente figura:

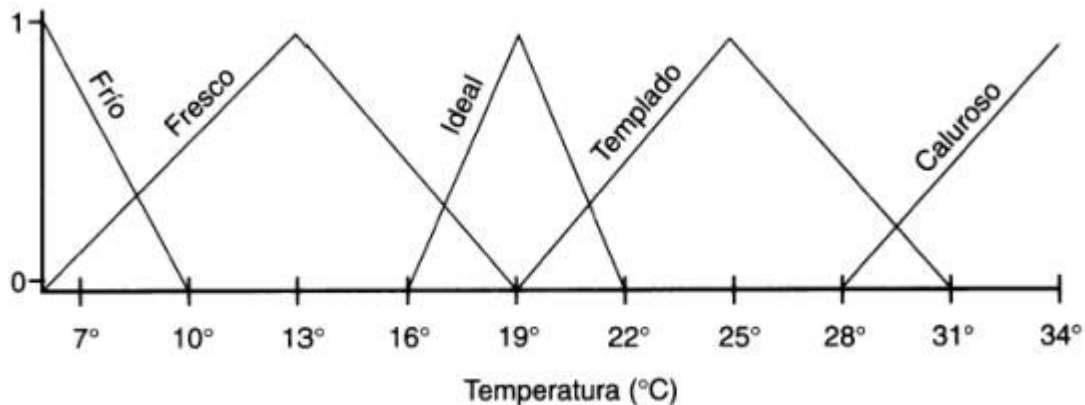


Figura 2-2 Intervalos de temperatura según lógica difusa¹.

Por tanto, una temperatura de 29°C se consideraría 30% templado y 20% caluroso. En función de estas afirmaciones, se pueden definir unas reglas, de manera que si la temperatura es templada, se actúa de cierta manera, y si es calurosa, se actúa de otra. Al pertenecer a ambos intervalos, se haría un promedio de la actuación a realizar, definido mediante una regla lógica.

Se observa que es un procedimiento eminentemente experimental, pues para diseñar un controlador, necesitamos ver la respuesta del sistema ante las diferentes entradas. Una vez se tienen los datos, se procede a crear unas reglas para que el sistema funcione como queramos.

Las principales ventajas de este tipo de control son:

- Fácil comprensión, pues usa reglas lógicas basadas en el lenguaje humano.
- Rapidez de cálculo. Las condiciones lógicas son fácilmente implementables y tardan poco en ser calculadas, pues son estructuras condicionales del tipo *switch case*. Además, la salida del controlador se hace sumando áreas y calculando los centros de gravedad, operaciones que no requieren mucho tiempo de cálculo debido a que las figuras que se obtienen suelen ser triángulos o trapecios.
- Es aplicable a numerosos tipos de sistemas, pues simplemente necesitamos ver las relaciones entrada-salida de los mismos. Permite, además, control monovariable como multivariable.
- Se basa en la experiencia. Otro tipo de controladores presentan más restricciones a la hora de trabajar con datos experimentales, la lógica difusa se basa directamente en ellos.
- Es compatible con otros sistemas de control, de manera que podemos usarlo como un control complementario.

Sin embargo, presenta también algunas desventajas:

- No se puede asegurar su funcionamiento en todos los casos. Si se hace un mal modelado del controlador, no se puede asegurar que no falle.
- Puede llegar a ser bastante largo de diseñar. Si se quieren controlar sistemas muy complejos, se necesitan crear muchas normas lógicas, y llega un momento en el que es más sencillo hacer un controlador de otro tipo.

Viendo todas las ventajas, se observa que la principal es la rapidez de cálculo y la simplicidad del control. Por tanto, para un sistema de tiempo real, puede ser una solución adecuada.

¹ <http://kirainet.com/lgica-difusa-fuzzy-logic/>, con fecha de Noviembre de 2015

2.2 Controladores específicos para Vehículos

Una vez vistos algunos tipos de controladores generales, vamos a concretar algunos tipos de control específicos para la estabilidad de vehículos.

2.2.1 Control Antideslizamiento

Este tipo de sistemas de control intenta reducir el deslizamiento del vehículo, que es una de las principales causas de pérdida de control por parte del conductor. Este conjunto de controladores fue de los primeros en diseñarse buscando mejorar la estabilidad de los vehículos, gracias al avance de la electrónica, que permitía realizar cálculos rápidos y actuar inmediatamente sobre el vehículo, principalmente actuando sobre los frenos de las ruedas. El sistema más famoso es el ABS (Anti Blocking System).

Este sistema se creó pensando en el freno de las ruedas de los aviones, para evitar que se bloquearan al frenar en la maniobra de aterrizaje, es decir, que deslizara y no permitiera que el piloto pudiera controlar la aeronave en tierra. En un principio, esta idea surge en la década de 1930 por *Bosch*, pero no era fácilmente implementable debido a la tecnología del momento, eminentemente analógica. Más adelante, en la década de 1970, debido al avance de la electrónica digital, este sistema se empezó a comercializar.

Básicamente, consiste en una bomba controlada electrónicamente que bombea el líquido de frenos de las ruedas, de manera que la ECU, que calcula cuánto deberían frenar las ruedas para que no se bloqueen, manda la señal pertinente a la bomba, lo que hace que las ruedas modifiquen su velocidad, medida con un conjunto de sensores. Este sistema, en el caso de tener que actuar, lo hace muy rápido y a una frecuencia alta (en el entorno de 100 veces por segundo), de manera que se perciben vibraciones en el pedal del freno.

En la siguiente imagen puede verse la estructura de este sistema:

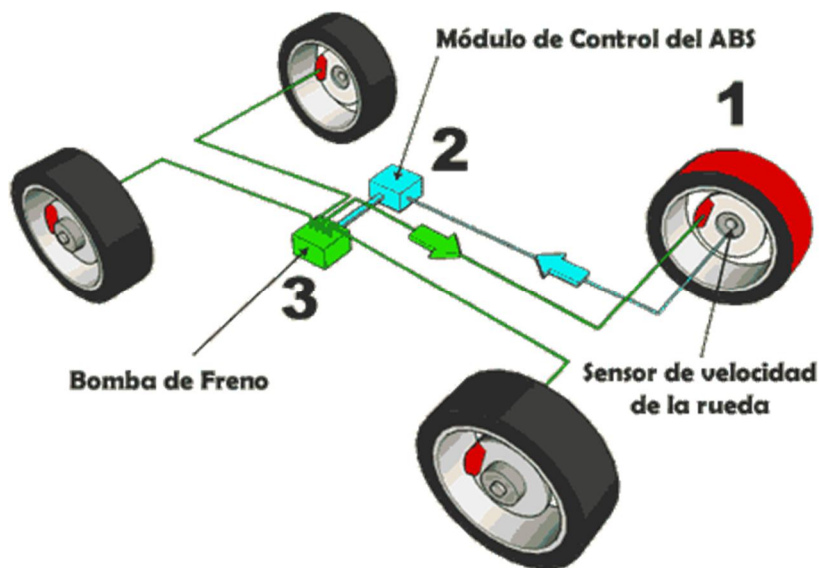


Figura 2-3 Sistema ABS ²

Los cálculos que realiza el módulo de control del ABS (normalmente los realiza la ECU), son bastante complejos, pues los problemas mecánicos en los que interviene el rozamiento no son sencillos de resolver. Este es el motivo de que se tuviera que esperar desde que se diseñó hasta que se pudo comercializar, pues la electrónica de la época era obsoleta para este tipo de control.

Como conclusión, el uso del ABS reduce la distancia de frenado, y mejora el control del vehículo. Actualmente, es un sistema muy importante en todo vehículo comercial.

² <http://www.racesimonline.com/articulos/ESP.php>, a fecha de Octubre de 2015

2.2.2 Control de Estabilidad

Los sistemas de control de estabilidad son elementos de seguridad activa de los automóviles. Actúan frenando individualmente cada rueda para evitar derrapes y pérdidas de control de los vehículos. El más famoso de este grupo de controladores es el ESP (Electronic Stability Program).

El ESP fue desarrollado por *Bosch* en colaboración con *Mercedes-Benz* en los años 1990. Es una mejora del sistema ABS, pues el primero servía para la frenada de las ruedas, pero este sirve para controlar la tracción de cada rueda. Su principio es el mismo, actuar sobre el freno de cada rueda para conseguir mejorar la dinámica del vehículo, pero su objetivo es diferente.

Este sistema consta de las bombas pertinentes en el sistema de frenado, y de varios sensores, como el sensor de ángulo de dirección, el de velocidad de giro de la rueda, y el de ángulo de giro y aceleración transversal del vehículo. Tras realizar los cálculos pertinentes, se actúa sobre el freno de cada rueda para así conseguir la tracción deseada en ellas, de manera que el vehículo pueda responder mejor a la voluntad del conductor.

En principio, es un sistema que siempre está activo, salvo que se pulse el botón que lo deshabilite, de manera que no debe entorpecer la conducción normal. Esto es importante a la hora de diseñar un controlador, pues de nada sirve tener un controlador que mejora la tracción del vehículo, si a cambio se empeoran el resto de parámetros de la conducción.

Como más adelante se verá, este tipo de sistemas de control que tienen en cuenta tantas variables del coche, y en función de ellas actúa para conseguir la tracción deseada, son bastante complejos. Por tanto, se entiende que esta medida de seguridad sea más reciente que el sistema de ABS, pues necesita una electrónica relativamente potente para poder realizar los cálculos.

Los sistemas de control que se plantean en este trabajo, se basan en parte en el funcionamiento del ESP, pero como se ha diseñado para un vehículo eléctrico con un motor por rueda, presenta bastantes diferencias, como la actuación, que en este caso no va a ser sobre el freno de cada rueda, sino sobre el motor de cada una. Por tanto, se verán las mejoras que esta configuración de vehículos ofrece.

3 DINÁMICA BÁSICA DE VEHÍCULOS

En este capítulo se van a presentar los conceptos más básicos necesarios para entender el funcionamiento del sistema de control. No se va a entrar al nivel de detalle de ecuaciones físicas, pues eso se hará en el apartado que sea necesario.

3.1 Tracción

Según la RAE (Real Academia Española), la tracción se define como “Acción y efecto de tirar de algo para moverlo o arrastrarlo, y especialmente los carruajes sobre la vía”. Por tanto, la tracción es la principal causa del movimiento del vehículo. Hay diferentes maneras de generar tracción en los vehículos, pero en nuestro caso la tracción se consigue mediante el movimiento de cuatro motores, uno por cada rueda.

Aplicando un par en cada rueda, conseguimos generar fuerzas en ellas, de manera que la resultante será la que provoque la tracción. En la siguiente figura se presentan algunas de las fuerzas presentes:

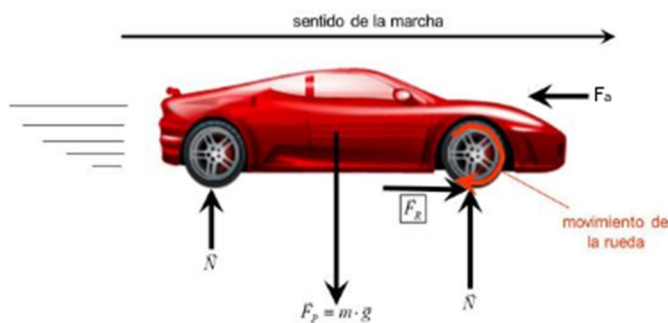


Figura 3-1 Representación fuerzas en trayectoria recta³

Donde **N** es la fuerza normal del vehículo, repartida entre las ruedas, **F_p** es la fuerza de gravedad, **F_r** es la fuerza debida al rozamiento y **F_a** es la fuerza aerodinámica. El perfil de esta última fuerza, que es de las más importantes, es bastante complejo, y depende, entre otros, de la geometría del vehículo. Al sumar las fuerzas en todas las ruedas, obtenemos una resultante que es la que determina la tracción del vehículo, es decir, la fuerza con la que se empuja al vehículo visto desde el suelo.

³ <https://fisica1spp.wikispaces.com> a fecha de Noviembre de 2015

3.2 Ángulos

Para una primera descripción del estado actual del vehículo, necesitamos el conocimiento de ciertas magnitudes, como posición, velocidad y fuerzas. Sin embargo, nuestro sistema de control necesita de más datos para poder calcular bien la actuación a realizar, así que hay que añadir ciertas magnitudes para conocer la orientación del vehículo. Estos ángulos se denominan Roll, Pitch y Yaw:

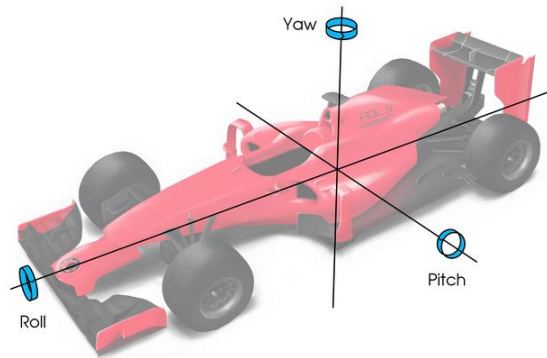


Figura 3-2 Ángulos Roll Pitch y Yaw⁴

El ángulo de Roll mide la inclinación según el eje longitudinal del vehículo, el Pitch mide la inclinación de ese eje, y el Yaw mide la orientación del vehículo. Por tanto, con estos tres ángulos y las magnitudes posición, velocidad y fuerzas se puede conocer con precisión el estado de un vehículo.

Para nuestro controlador, vamos a usar principalmente el Yaw o Ángulo de Guiñada, pues es el que nos va a indicar la orientación del vehículo sobre el asfalto.

3.3 Momento

El momento \mathbf{M} de una fuerza aplicada en un punto \mathbf{P} es el producto vectorial del vector \mathbf{r} y la fuerza aplicada \mathbf{F} , donde el vector \mathbf{r} es el vector que une el origen del sistema de coordenadas y el punto de aplicación \mathbf{P} .

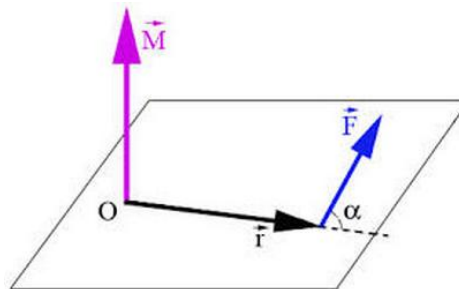


Figura 3-3 Definición de Momento de una fuerza

Hemos visto que las fuerzas aplicadas en cada rueda generan una resultante que determinan la tracción total del vehículo. Sin embargo, la mecánica establece que el momento resultante de un conjunto de fuerzas no concurrentes no es el momento producido por la fuerza resultante, sino la suma de momentos de cada fuerza. Dicho de otra manera, el momento aplicado al vehículo no es el momento de la fuerza resultante, sino la suma de los momentos de las fuerzas de cada rueda.

Teniendo esto en cuenta, podemos modificar fácilmente la orientación de nuestro vehículo actuando directamente en las fuerzas de cada rueda. Si calculamos el momento de guiñada extra necesario a aplicar al coche para que siga la trayectoria que queremos, es casi inmediato el reparto del par de cada rueda. De calcular el momento adicional necesario para seguir una trayectoria se ocupará el controlador de alto nivel, mientras que para calcular el par necesario a aplicar en cada rueda, se ocupará el controlador de bajo nivel.

⁴ www.laf1.es a fecha de Diciembre de 2015

3.4 Deslizamiento

El deslizamiento es uno de los principales problemas a solucionar con nuestro controlador, pues consiste en que la velocidad angular de las ruedas del vehículo no coincide con la que correspondería según la velocidad lineal del vehículo, es decir, que las ruedas no siguen la trayectoria exacta del coche a su velocidad. Esto puede pasar bien por aceleración, de manera que las ruedas giran más rápido, o bien por frenada, cuando se bloquean y no giran lo que debieran. En cualquier caso, implica pérdida parcial o total del control del vehículo, y pérdida de energía en el giro de las ruedas, cuando este giro no provoca en su plenitud movimiento del coche.

Es especialmente crítico en curvas, donde podemos perder el control del vehículo, pudiendo llegar hasta el punto de volcar, y en suelos con un bajo coeficiente de rozamiento.

Existen dos maneras de cuantificar el deslizamiento. La primera de ellas es el denominado Ángulo de Deslizamiento, o *Sideslip Angle*. Se define como el ángulo existente entre la dirección actual de la rueda y la dirección del vehículo, como se ve en la siguiente figura:

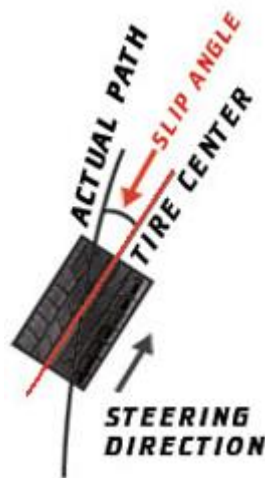


Figura 3-2 Ángulo de deslizamiento

Esta definición, a su vez, hace diferenciar entre dos tipos diferentes de ángulos. Por un lado, el ángulo de deslizamiento de la rueda, y por otro, el ángulo de deslizamiento del coche. Para calcularlo, se toma el arcotangente con signo del cociente de la velocidad en el eje y del elemento a medir y la velocidad en el eje x:

$$\beta = \text{atan}(v_y/v_x) \quad (3.1)$$

Si coinciden la velocidad transversal y longitudinal del coche, se ve que el ángulo es cero, de manera que el vehículo no está deslizando en ángulo. Si se estudian las velocidades de las ruedas, llegamos a la misma conclusión para cada rueda analizada.

Es de vital importancia mantener el ángulo de deslizamiento del coche inferior a 12° en valor absoluto, pues lo contrario provocaría que el vehículo volcase. En carreteras con un coeficiente de deslizamiento más bajo, como carreteras nevadas, este ángulo se debe limitar a 2° [5].

La segunda manera de cuantificar el deslizamiento consiste en comparar la velocidad de una rueda con la velocidad del coche, de manera que podemos saber si las ruedas están deslizando. En términos energéticos, es malo que la rueda gire a una mayor velocidad de la que avanza el coche, pues estamos dando energía a las ruedas, pero no se está empleando plenamente para provocar tracción del vehículo.

Existen varias maneras para calcular esta magnitud, pero vamos a usar la siguiente fórmula para calcular el *Slip Ratio* [6]:

$$\lambda = \frac{V_{\omega} - V}{\max(V_{\omega}, V)} \quad (3.2)$$

Donde V_{ω} es la velocidad angular de la rueda, y V es la velocidad del vehículo. Si ambas magnitudes coinciden, la rueda estudiada no deslizará, y mientras mayor sea este cociente, mayor deslizamiento habrá.

3.5 Viraje

Cuando el conductor de un vehículo quiere realizar un giro, el coche puede responder de tres maneras diferentes: girando lo que el conductor pide mediante el volante, que la parte delantera del vehículo gire más que la trasera, o que sea la trasera la que más gire. Que se dé un caso u otro depende principalmente de la tracción, del deslizamiento y de la distribución de masa del vehículo entre otros factores, pero estos van a ser los más importantes para este trabajo.

Se denomina sobreviraje a la condición de giro en la que el eje trasero gira más que el delantero, por lo que se realizan curvas más cerradas. Esto es peligroso pues puede hacer que el coche patine, con la consiguiente pérdida de control del vehículo.

Se denomina subviraje a la condición de giro en la que el eje delantero gira más que el trasero, por lo que los giros son más abiertos. Tampoco es recomendable que se dé esta condición en el vehículo, pues al no poder realizar el giro debidamente, estamos perdiendo control sobre el mismo.

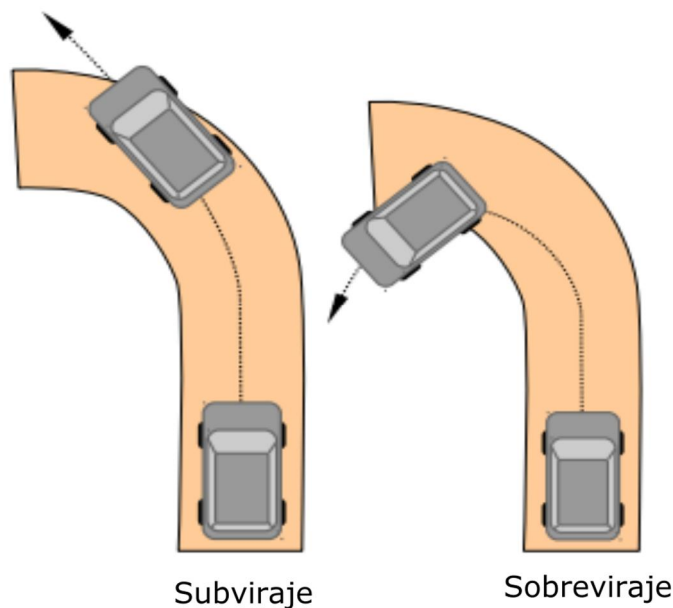


Figura 3-3 Subviraje y Sobreviraje

El subviraje tiende a darse en vehículos donde la distribución de peso sea mayor en el eje delantero que en el trasero, y el sobreviraje en los que la distribución de peso del eje trasero sea mayor que en el delantero. Esto es algo muy importante a la hora de diseñar un controlador, pues según la dinámica del vehículo, deberemos corregir el sobreviraje o el subviraje. Sin embargo, en vehículos orientados a competiciones deportivas, suele interesar una tendencia a sobrevirar, pues así los pilotos expertos pueden tomar las curvas mejor.

4 VEHÍCULO FOX

El estudio de controladores de reparto de par se va a llevar a cabo teniendo en cuenta el modelo del que disponemos, el *FOX*. Este vehículo es un prototipo de coche eléctrico con un motor por rueda desarrollado en la Universidad de Sevilla. Consta de dos fuentes de energía, pila de hidrógeno y baterías, con su respectivo banco de supercondensadores. Sin embargo, durante la realización de este trabajo sólo estaba disponible la alimentación debida a la batería. A la hora del diseño de controladores de estabilidad, esta modificación afecta únicamente a algunos parámetros del modelo con el que trabajamos, como masa y reparto de masa. Como controlamos el par de cada rueda, la fuente de la que proviene la energía no hace que los controladores diseñados varíen, por lo que si se introduce la pila de hidrógeno, no habría que hacer apenas cambios.

Este capítulo se va a desarrollar en dos partes bien diferenciadas. En la primera, se hará una descripción del vehículo, mientras que en la segunda parte, se hará lo mismo con el simulador del que consta el vehículo.

4.1 Características del vehículo

Para el diseño del vehículo FOX, se ha partido del chasis de competición S2 fabricado por SilverCars. Se ha modificado el interior para poder introducir los cuatro motores, las fuentes de alimentación, pantallas para poder modificar los algoritmos a probar y diversas modificaciones secundarias. La comunicación entre los diferentes componentes está basada en CAN.



Figura 4-1 Vehículo FOX

4.1.1 Motores

Como ya se ha comentado, se disponen de cuatro motores sin escobillas, de 7 kW de potencia, y de 78 Nm de par máximo. Al ser motores sin escobilla, ofrecen un menor rozamiento, de manera que se pierde menos energía, algo fundamental para este tipo de vehículo. Estos motores están introducidos dentro de cada rueda, sin ninguna conexión mecánica entre ellos, de manera que cada motor gira independientemente del resto, lo cual permite una mayor variedad de estrategias de control. Esto, a su vez, mejora la transferencia de energía del motor a la rueda, pues hay menos intermediarios.

Se alimentan mediante corriente alterna, proveniente de un conversor electrónico que transforma la corriente continua entregada por las baterías.

Al igual que los motores son de vital importancia, se dispone también de frenos, accionados mediante el pedal pertinente. En este trabajo se va a trabajar con la alimentación de los motores principalmente, pero es importante mencionar este elemento, pues el conductor siempre los tendrá disponible, sin ninguna modificación por parte del controlador.

4.1.2 Baterías

Como se adelantó al principio del capítulo, el vehículo tiene como fuente de energía actual un conjunto de baterías, en particular 6 paquetes de baterías de 4 celdas cada una. No se va a entrar en mucho detalle sobre sus características pues no es objeto de este trabajo, pero sí es importante comentar que, debido a su composición, obtenemos dos ventajas muy importantes para este estudio.

La primera es que son unas baterías que contaminan menos al ser recicladas que la mayoría de baterías existentes para esta aplicación, algo que es de vital importancia para nuestro vehículo, pues al principio se propuso este tipo de coches como una alternativa a los que tienen motores de combustión porque emiten menos residuos perjudiciales al medio ambiente. La segunda ventaja [10] radica en que son baterías con un peso inferior a la mayoría de las existentes en el mercado con esta capacidad, de manera que evitamos cargar demasiado peso en el vehículo, lo que hace que su autonomía, que ya de por sí es bastante alta, sea aún mayor.

Por tanto, podemos concluir que las baterías del vehículo son un factor muy importante a la hora de potenciar las características propias del FOX, pues es un vehículo que contamina poco, y queremos obtener de él la máxima eficiencia, que es uno de los puntos débiles de este tipo de vehículos.

Es importante destacar que estas baterías se encuentran en la parte trasera del vehículo, lo cual hace que la distribución de peso en el eje trasero del FOX sea mayor que en el eje delantero.

4.1.3 ECU

Este elemento es uno de los más importantes en el vehículo, pues permite controlar el funcionamiento del vehículo. Es, junto con el hardware de supervisión, el que permite que el vehículo funcione correctamente, pues son el centro de comunicación de todos los componentes. Los algoritmos que se van a mostrar en este trabajo irán luego cargados allí, de manera que es muy importante su existencia para poder optimizar la distribución de par y el consumo energético.

4.1.4 Sensores

Ya se vió que para implementar un sistema de control de estabilidad como el ESP, se necesitan muchas medidas de diferentes parámetros del vehículo. Existen dos maneras de estimar estos valores, o bien mediante sensores, o bien mediante cálculos relacionados con los datos disponibles. Durante el desarrollo del controlador en el siguiente capítulo, veremos todos los datos que necesitamos, pero primero voy a introducir las diferentes medidas que podemos obtener directamente del coche. Para ello, usamos diferentes sensores, entre los que cabe destacar los siguientes:

4.1.4.1 Sensor de giro del volante

Para poder medir el ángulo de giro del volante, se dispone de un potenciómetro rotatorio que permite detectar giros del mismo de hasta 360°. Está alimentado a 5V, como el resto de potenciómetros del vehículo.

4.1.4.2 Sensor de posición de los pedales

Estos sensores son potenciómetros lineales, con salida de 0 a 5V en función de la carrera de los pedales, de unos 12.5 mm.

4.1.4.3 Sensores de suspensión

Con estos sensores, que también son potenciómetros lineales, podemos conocer el estado de la suspensión del vehículo, de manera que en la ECU podemos inferir la distribución de peso si el coche está quieto y las fuerzas de reacción entre las ruedas y el suelo. Es un ejemplo de sensor empleado para obtener una medida indirecta.

4.1.4.4 IMU

Es uno de los sensores más importantes, y consta principalmente de acelerómetros en cada uno de los tres ejes. Como su propio nombre indica, Inertial Measurement Unit, este elemento da medidas inerciales, como la aceleración en cada uno de los tres ejes. Además, consta de un GPS, de manera que luego puede estudiarse el recorrido del vehículo en la prueba.

4.2 Modelo de Simulink

De cara a tener un modelo de simulación de un vehículo, hay dos opciones principalmente. La primera consiste en coger las ecuaciones características de la dinámica de vehículos, y montar un sistema en Simulink con ellas, de manera que podemos obtener, en función de las hipótesis adoptadas, un simulador más o menos real del vehículo. Esta opción es viable, pero requiere de demasiadas simplificaciones, pues un vehículo real como el FOX es extremadamente complejo de caracterizar mediante unas ecuaciones. La segunda opción consiste en usar un programa de modelado mecánico, que ya implementa de por sí las ecuaciones, y exportarlo al entorno de simulación que queramos. Esta es la opción que se ha tomado en este trabajo, pues ya existe un modelo de Simulink empleando Simmechanics creado por David Marcos Rodríguez [1]. A continuación, voy a explicar brevemente su modelo para enseñar el entorno en el que he estado probando los controladores.

El simulador se basa en bloques sensores, actuadores y de unión de los bloques que se muestran en la siguiente figura, de manera que las ecuaciones físicas ya están incorporadas dentro de ellas. Por tanto, si queremos una medida que en el coche real no se puede obtener directamente, podemos activarla como salida del bloque sensor pertinente. La otra opción es estimar esa medida, como se verá más adelante con las fuerzas aplicadas en las ruedas.

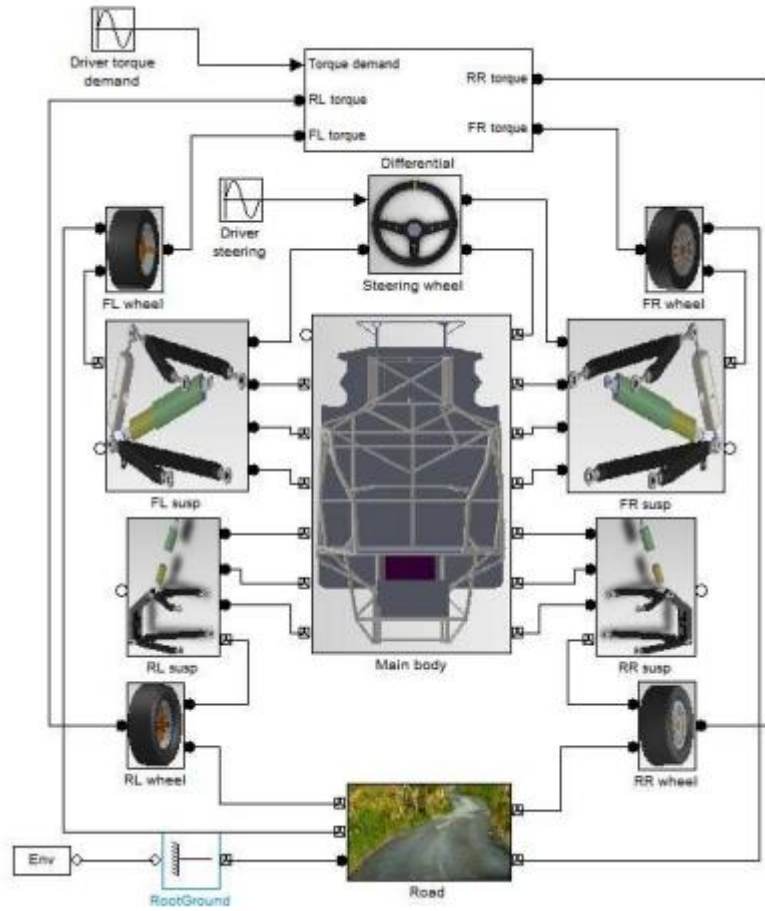


Figura 4-2 Bloques del vehículo

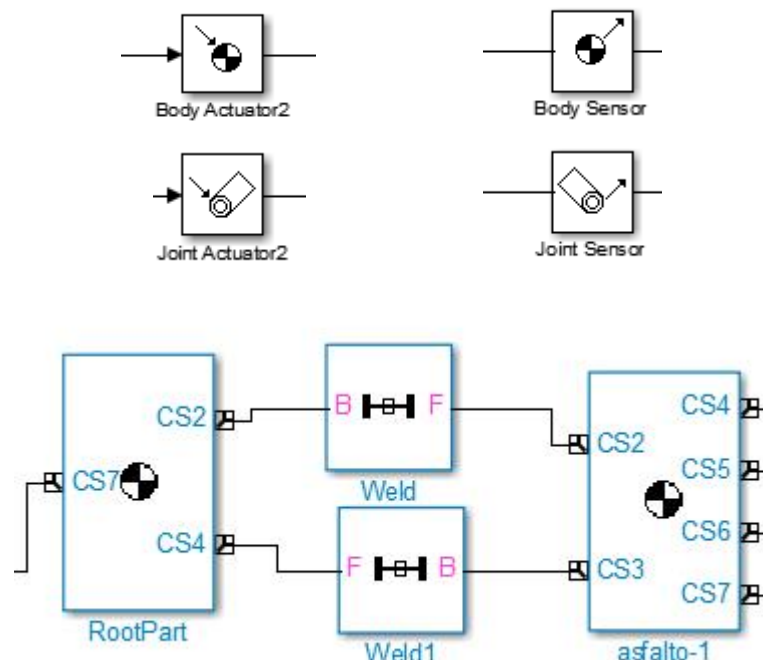


Figura 4-3 Ejemplos de algunos de los bloques empleados

Se han tomado las siguientes simplificaciones para obtener el modelo:

- Todas las partes del vehículo son sólidos rígidos, incluyendo los neumáticos.

- Los ocupantes del vehículo también son sólidos rígidos, unidos al chasis.
- La fuerza entre la rueda y la carretera se aplica en el punto más bajo de la circunferencia de la rueda.
- Se han modificado algunas suspensiones para poder montar el modelo con el programa Simmechanics.
- La fuerza de fricción del aire se considera sólo en el eje longitudinal del vehículo.
- Sustitución del contacto rueda-carretera por un carrito, por limitaciones del programa.

Con todas estas simplificaciones, se consigue un modelo que responde bastante bien al comportamiento real del vehículo.

Para ocupar menos espacio, dado que el modelo del FOX en Simulink es bastante grande, se ha optado por reducirlo todo a un subsistema, en el que dentro existen "Goto" para cada variable importante, de manera que luego son accesibles desde fuera del subsistema usando la etiqueta "From". Para simular el conductor, se mete en "Volante_in" la maniobra del volante que se quiera hacer, y en "Par" el par que se aplica a cada rueda como una señal con cuatro componentes, una por rueda. Se ha dejado la variable v_veh como salida directa del modelo por simplicidad a la hora de montar el esquema de control.

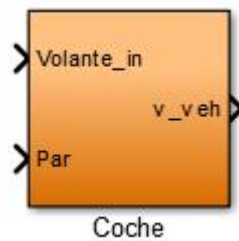


Figura 4-4 Subsistema del FOX

Para generar la entrada de volante, simplemente se mete el valor del ángulo del mismo en grados sexagesimales.

5 SISTEMA DE CONTROL DE ESTABILIDAD BASADO EN LÓGICA DIFUSA

En este capítulo se va a explicar el sistema de control de estabilidad adoptado para conseguir mejoras en el funcionamiento del vehículo. Aunque se presenten algunas ecuaciones de la dinámica de vehículos y métodos para calcular algunos parámetros, como tenemos el simulador donde se nos facilitan casi todas las variables medibles, no vamos a usarlas todas. Sin embargo, a la hora de codificar el algoritmo de control en el vehículo real, sí vamos a tener que usar muchas de estas ecuaciones, pues como vimos antes, muchas medidas no se obtienen directamente de los sensores, sino que hay que calcularlas indirectamente, como es el caso de las fuerzas presentes en cada vehículo.

5.1 Estructura del sistema de control

Ya se ha comentado que vamos a tener dos sistemas de control, uno superior para calcular la actuación adicional que es necesaria aplicar para que el vehículo mejore su comportamiento, y uno inferior en el que esta actuación calculada se transforme en actuación de cada rueda. A continuación, se añade un bloque que calcula el par máximo transmisible a cada rueda como parte final del sistema de control. Además, se dispone de un generador de referencias, de manera que el sistema completo queda como se muestra en la siguiente figura:

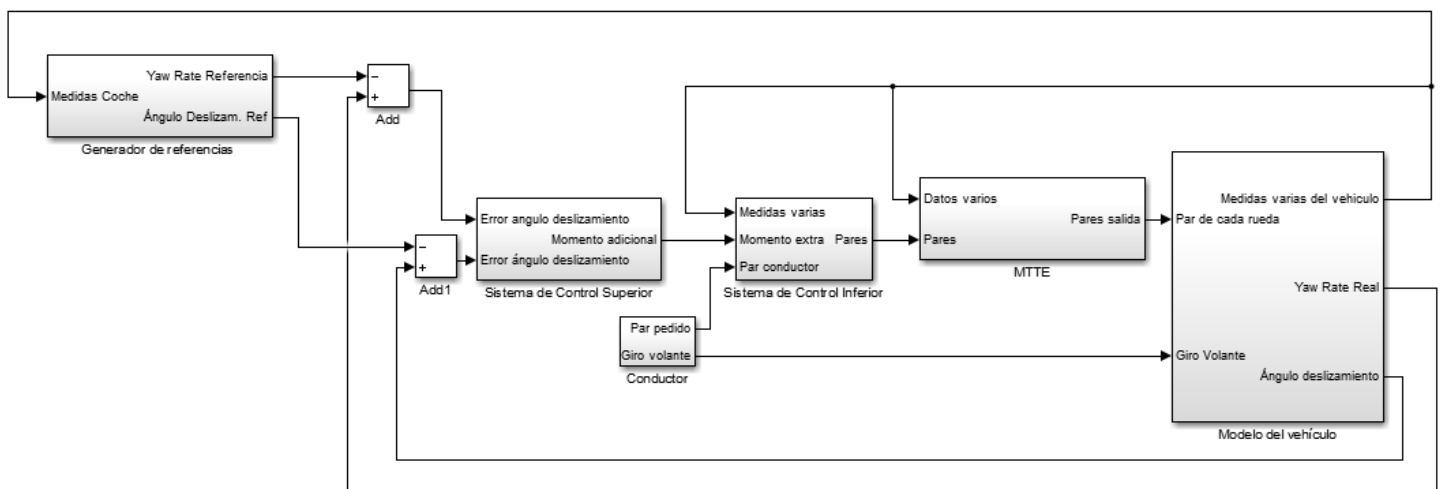


Figura 5-1 Esquema del sistema de Control

El modelo del vehículo aporta diferentes medidas en función del par de cada rueda y del ángulo de giro del volante, como fuerzas, velocidades de cada rueda, ángulo de deslizamiento y Yaw Rate (derivada del ángulo

Yaw), que alimentan al Generador de Referencias. Una vez calculados los errores de la dinámica actual y la deseada, se pasa esa información al Sistema de Control Superior, que calcula el Momento Adicional a aplicar al vehículo para llegar a las referencias, y el Sistema de Control Inferior se ocupa de que al vehículo le llegue el par necesario para corregir la dinámica. De esta manera se estructura el controlador que vamos a implementar.

5.2 Generador de Referencias

Para este controlador, necesitamos generar dos referencias. La primera referencia es la del ángulo de deslizamiento, que se tomará como 0 radianes, pues el mejor caso es en el que el coche no desliza. Sin embargo, esta referencia no va a ser alcanzable en muchos casos, pues si el vehículo está girando, tiene las ruedas orientadas hacia un lugar, de manera que siempre tendrá alguna componente de velocidad transversal al vehículo, haciendo que el ángulo de deslizamiento, como se veía en la ecuación 3.1, no sea cero.

Para generar la segunda referencia, la de Yaw Rate, se han tomado datos del modelo del vehículo girando a muy baja velocidad [1], pues en esos casos el comportamiento del vehículo es bastante más parecido a un comportamiento ideal (sin deslizamiento). Con estos datos, se ha creado una tabla en la que se entra por el ángulo del volante en el momento actual, y se tiene como salida el yaw rate que tenía el coche en el experimento a baja velocidad multiplicado por la velocidad actual, de manera que podemos usarlo como referencia para nuestro modelo, pues como dije antes, es un caso casi ideal para nuestro vehículo en cuanto a términos de giro.

5.3 Sistema de Control Superior

Para esta parte del controlador, se ha optado por emplear la lógica difusa, pues como se comentó con anterioridad, permite reducir bastante el tiempo de computación necesario para usarlo en un sistema de tiempo real, además de simplificar mucho los sistemas complejos, de manera que podemos obtener un controlador para nuestro vehículo de una manera relativamente más sencilla que con otros métodos.

El controlador que se va a emplear se propuso por B.L. Boada [7] para vehículos de combustión interna, pero se va a modificar para potenciar las mejoras que los vehículos eléctricos de cuatro motores pueden ofrecer, de manera que se va a conseguir un sistema de control completo (trayectoria y deslizamiento) similar al propuesto por Jalali Kiumars [8].

Esta parte del controlador se compone de cuatro subsistemas, mostrados en la siguiente figura, y que se explican a continuación:

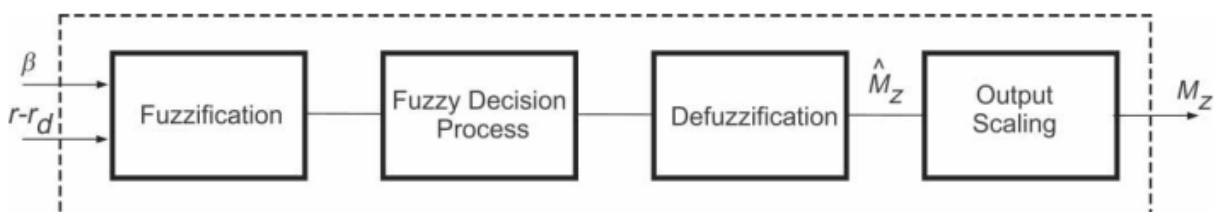


Figura 5-2 Estructura controlador de lógica difusa

Donde $r - r_d$ es el error en yaw rate (real menos referencia), β es el error en ángulo de deslizamiento del vehículo (se toma de referencia 0), \widehat{M}_z es la salida normalizada del controlador de lógica difusa, y M_z es el momento necesario para corregir la dinámica del vehículo.

5.3.1 Conversión a Lógica Difusa (Fuzzification)

Consiste en pasar de unas medidas de errores a valores entendibles por la lógica difusa, es decir, a variables lingüísticas. Por tanto, para cada entrada al sistema, se efectúa una conversión. Según J. Espinosa [9], el número de variables de decisión que un ser humano medio puede manejar en una decisión está en el intervalo [2, 7], de

manera que se definirán 7 sets para el controlador. De esta manera, se definen los siguientes términos lingüísticos para cada variable, que se explicarán a continuación:

$$\{e(\beta), e(r)\} = \{NB, NS, ZE, PS, PB\} \quad (5.1)$$

$$\{\widehat{M}_z\} = \{NB, NM, NS, ZE, PS, PM, PB\} \quad (5.2)$$

Tabla 5-1 Términos Lingüísticos

Símbolo	Significado
NB	Negativo Grande (Negative Big)
NB	Negativo Medio (Negative Medium)
NS	Negativo Pequeño (Negative Small)
ZE	Cero (Zero)
PS	Positivo Grande (Positive Small)
PM	Positivo Mediano (Positive Medium)
PB	Positivo Grande (Positive Big)

De manera que las entradas y la salida del controlador se estudiarán como si fueran grandes, medianas, pequeñas, nulas o cero, con su signo. Así se consigue la conversión a la lógica difusa de las entradas y salidas del controlador.

5.3.2 Proceso de Decisión Difusa (Fuzzy Decision Process)

Una vez tenemos las entradas del controlador en términos lingüísticos, definimos la actuación del controlador en función de las entradas. En principio, se va a seguir el método de Mandami. Este método se basa en usar reglas del tipo SI entrada 1 es e1 y entrada 2 es e2 ENTONCES salida es s1 (IF-THEN). Tanto el antecedente (entrada es e1 y entrada 2 es e2) como la conclusión (salida es s1) son reglas lingüísticas. Estas reglas se definen en la siguiente tabla:

Tabla 5-2 Reglas Lingüísticas

$e(\beta)$	$e(r)$	\hat{M}_z	Peso
NB	PB	NB	1
NS	PB	NB	1
ZE	PB	NM	1
PS	PB	NB	1
PB	PB	NB	1
NB	PS	NB	1
NS	PS	NM	1
ZE	PS	NS	1
PS	PS	NM	1
PB	PS	NS	1

$e(\beta)$	$e(r)$	\hat{M}_z	Peso
NB	NB	PB	1
NS	NB	PB	1
ZE	NB	PM	1
PS	NB	PB	1
PB	NB	PB	1
NB	NS	PB	1
NS	NS	PM	0.5
ZE	NS	PS	0.5
PS	NS	PM	0.5
PB	NS	PS	1

$e(\beta)$	$e(r)$	\hat{M}_z	Peso
NB	ZE	NS	1
NS	ZE	NS	1
ZE	ZE	ZE	1
PS	ZE	PS	1
PB	ZE	PS	1

Para el diseño de estas reglas, se ha tenido en cuenta que el momento adicional a generar depende principalmente del yaw rate. Por tanto, visualizando cada caso, se pueden definir estas reglas siguiendo el siguiente guión. Se presentarán figuras para comprender gráficamente dos de los siguientes cuatro casos, pues el resto se entienden por simetría.

- Si $e(r) = r - r_d > 0$

- $r > 0, r_d > 0$, con $r > r_d$. En este caso, el momento a aplicar debe generarse en la dirección contraria a r_d para disminuir r .

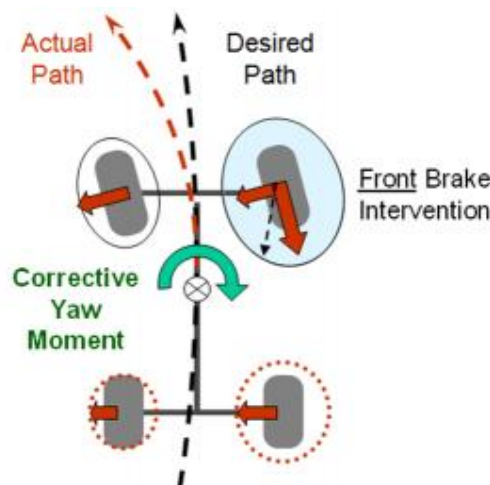


Figura 5-3 Posibilidad 1 de momento correctivo empleando frenos

- $r < 0, r_d < 0$, con $r < r_d$. En este caso, el momento a aplicar debe generarse en la dirección de r_d para aumentar r .

- Si $e(r) = r - r_d < 0$

- $r > 0, r_d > 0$, con $r < r_d$. En este caso, el momento a aplicar debe generarse en la dirección de r_d para aumentar r .

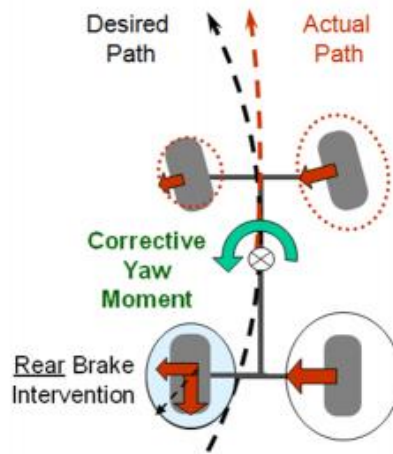


Figura 5-4 Posibilidad 2 de momento correctivo empleando frenos

- $r < 0, r_d < 0$, con $r > r_d$. En este caso, el momento a aplicar debe generarse en la dirección contraria a r_d para disminuir r .

- Mientras mayor sea el error en Yaw Rate, mayor será el momento a generar.

Para ajustar la salida del sistema en función del error del ángulo de deslizamiento, se van modificando los diferentes valores hasta que se obtiene la salida deseada. De ahí el motivo de que si el error en Yaw Rate es pequeño (NS), salvo que el error del slip angle sea muy grande (PB o NB), en ese caso la regla influirá la mitad que el resto de casos.

5.3.3 Conversión a Magnitudes Reales (Defuzzification)

En esta etapa, calculamos la salida del controlador, que estará normalizada entre -1 y 1 para ver rápidamente su grado de actuación: estará actuando mucho cuando dé valores cercanos a -1 o 1, y estará actuando poco cuando dé valores cercanos a 0.

Para calcular la salida, necesitamos definir las funciones de pertenencia y los rangos de valores de las dos entradas y de la salida. Para ello, y debido a que en parte es un método que se comprende muy bien gráficamente, definimos las funciones visualmente:

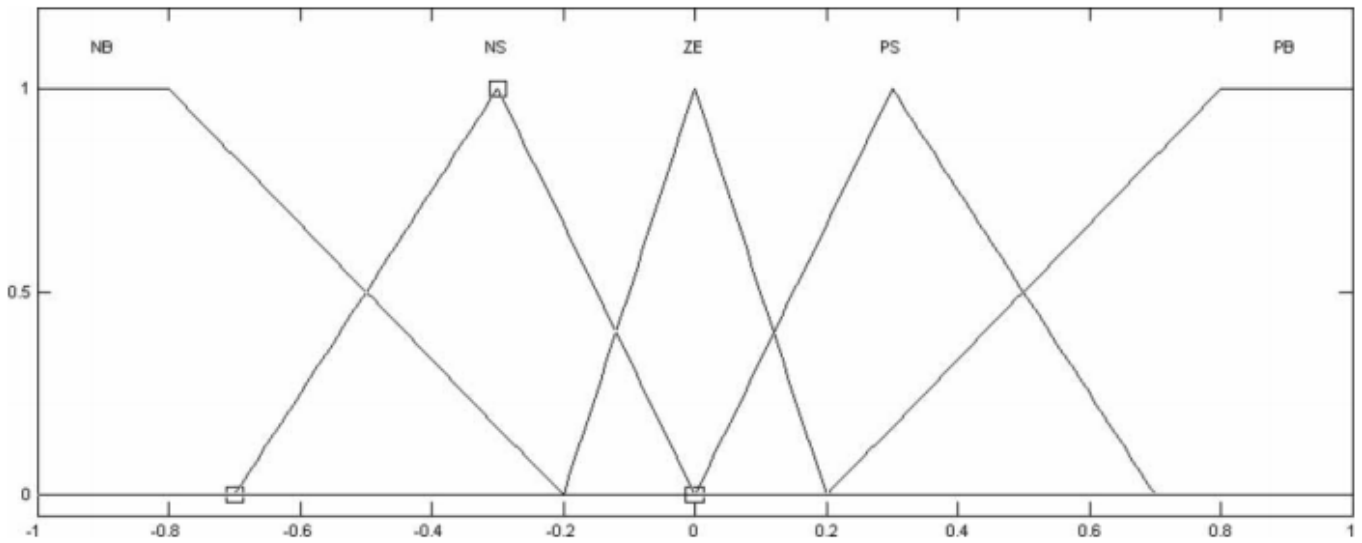


Figura 5-5 Funciones de pertenencia para el error del ángulo de deslizamiento

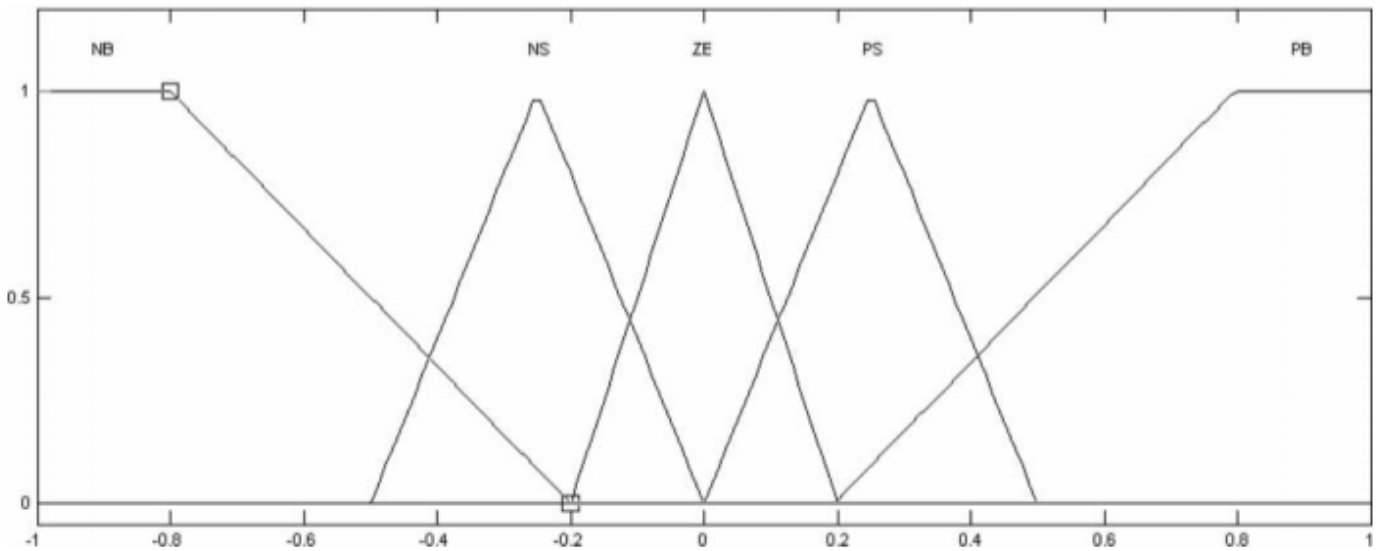


Figura 5-6 Funciones de pertenencia para el error en Yaw Rate

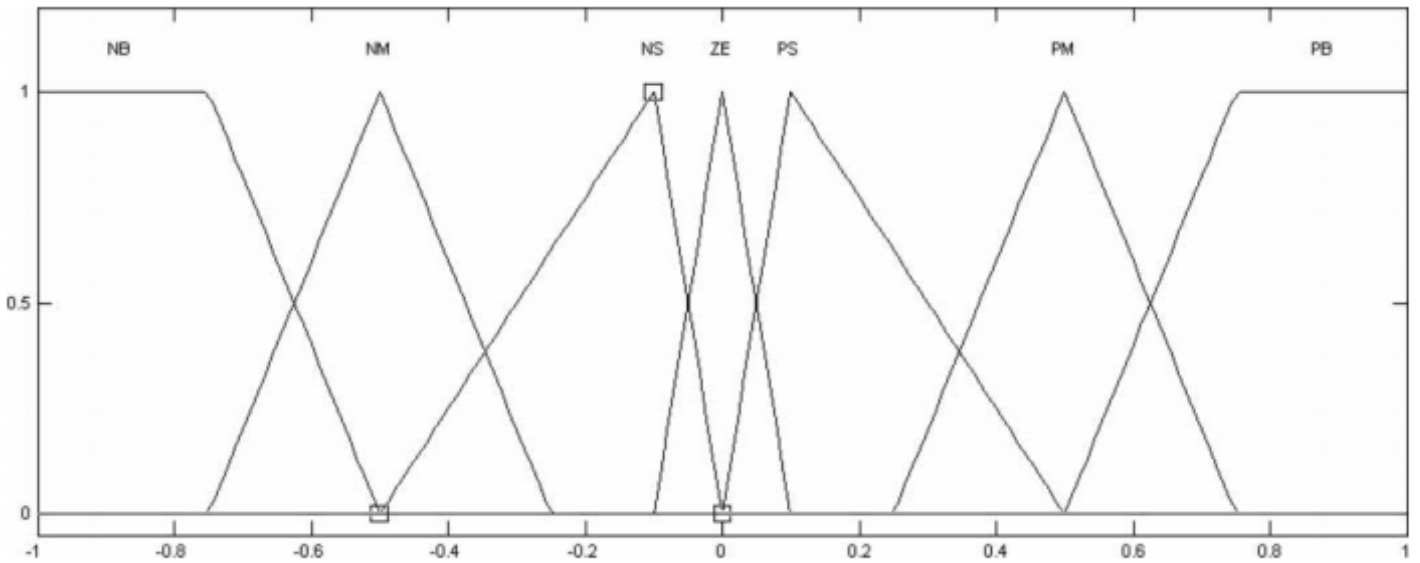


Figura 5-7 Funciones de pertenencia para la salida del sistema \hat{M}_z

Se observa que las entradas del controlador pertenecen al intervalo $[-1, 1]$. Esto es más que suficiente, pues el error de Yaw Rate no suele ser superior a 1 rad/s en conducción normal, ni el error en ángulo de deslizamiento mayor de 1 rad, como luego se verá en las diferentes simulaciones.

De esta manera, obtenemos la siguiente figura, donde se ve la actuación de la lógica difusa en función de las entradas:

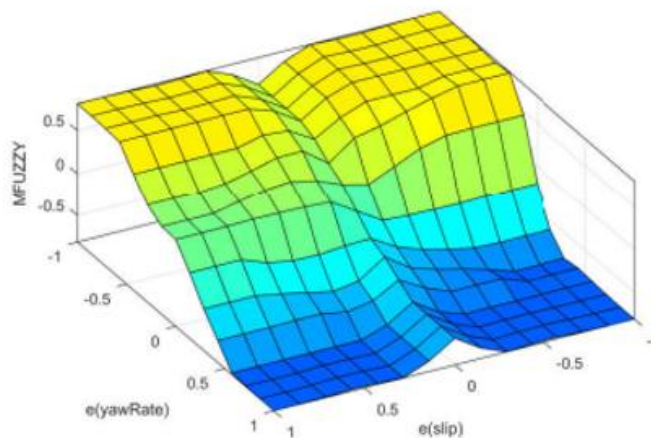


Figura 5-8 Resultado de Lógica Difusa

5.3.4 Escalado (Output Scaling)

Esta parte consiste en transformar la salida normalizada de -1 a 1 de la lógica difusa para que obtengamos un momento de guiñada suficientemente grande como para corregir la dinámica del vehículo. Más concretamente, según el esquema presentado en la Figura 5-2 "Estructura controlador de lógica difusa" consiste en pasar de \hat{M}_z a M_z , donde M_z es el momento correctivo necesario a aplicar en el vehículo.

Para conseguir esta transformación, se emplea una ganancia, que se escoge haciendo diferentes pruebas y viendo la que hace que el comportamiento del vehículo mejore más, sin perjudicar el resto de parámetros. Hay que tener en cuenta dos limitaciones principalmente, relacionadas con que el controlador se está diseñando para un vehículo real, no para uso exclusivo en simulación.

La primera de las limitaciones son las saturaciones de los diferentes elementos del vehículo, en particular la de los motores. No podemos pedir un momento demasiado grande si luego se traduce en que los motores den un par que se salga de los límites reales, en nuestro caso, de 78 Nm para cada motor. Por tanto, hay que escoger una ganancia que haga que el par máximo que se pidiera a cada rueda fuese de 78 Nm.

La segunda limitación es la falsa independencia a la hora de actuar sobre los motores. Es cierto que podemos calcular para cada rueda el par que sea, aunque sea muy diferente al del resto, pero en un vehículo real esto no puede ser así. Si estamos girando, el controlador de bajo nivel se encargará de que las ruedas de un lado tengan más par, y las del lado opuesto tengan menos par, de manera que se genere el momento extra calculado. Si se escoge una ganancia demasiado alta, podemos hacer que unas ruedas tengan un par negativo, y otras un par positivo, algo que no es aceptable en la realidad.

Tampoco podemos poner una ganancia muy pequeña, pues entonces el controlador casi no tendrá efecto. Con todo esto, hay que escoger cuidadosamente esta ganancia, pues tenemos que conseguir que el controlador mejore la dinámica del vehículo, pero teniendo en cuenta las limitaciones antes presentadas.

Como resultado de varios experimentos, se propone el valor de 400 para dicha ganancia. Más adelante, en las simulaciones, se verán las curvas del par de cada rueda y se entenderá por qué se ha escogido ese valor. Si no hubiesen existido estas limitaciones, podríamos aumentar bastante más la ganancia, haciendo que se llegue casi a cualquier referencia que se introduzca relativamente rápido, pero como se ha visto no es el objetivo de este trabajo.

5.4 Sistema de Control Inferior

Una vez se tiene el momento de guiñada a generar, es necesario transformar este valor a par a sumar o restar (según el caso) al par que pide el conductor [8].

Para calcular el par que va a recibir cada rueda, primero repartimos la tarea de generarlo en cada eje [10]:

$$M_z = M_{front} + M_{rear} \quad (5.3)$$

donde

$$M_{front} = M_z * \theta_{front} \quad (5.4)$$

$$M_{rear} = M_z * (1 - \theta_{front}) \quad (5.5)$$

Se va a escoger $\theta_{front} = 0.5$, de manera que cada eje trabaje por igual. Si la distribución de masa en el vehículo fuera muy desigual, habría que modificar este parámetro, pero en principio no está muy desproporcionada.

Si plantemos un balance de fuerzas en cada eje, obtenemos:

$$M_{front} = F_{x,FR} * \frac{t_f}{2} + F_{x,FL} * \frac{t_f}{2} \quad (5.6)$$

$$M_{rear} = F_{x,RR} * \frac{t_r}{2} + F_{x,RL} * \frac{t_r}{2} \quad (5.7)$$

Donde $F_{x,FR}, F_{x,FL}, F_{x,RR}, F_{x,RL}$ son las fuerzas en el eje x de las ruedas delantera derecha (FR), delantera izquierda (FL), trasera derecha (RR) y trasera izquierda (RL). t_f y t_r son el tamaño de los ejes delantero y trasero, respectivamente.

Como podemos actuar sobre cada rueda, vamos a aplicar un par diferencial para conseguir el momento. Para ello, seguimos el esquema presentado en la siguiente figura, en el que se representa que las ruedas del mismo eje tienen la misma fuerza, pero con diferente sentido:

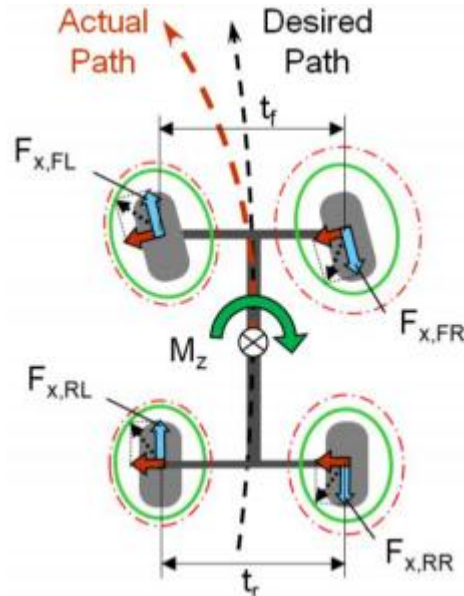


Figura 5-9 Esquema de fuerzas en par diferencial

A continuación, presentamos la siguiente igualdad, que representa el par diferencial a aplicar, es decir, la fuerza que cada rueda debe generar, en sentidos opuestos.

$$F_{x,F} = F_{x,FR} = F_{x,FL} \quad (5.8)$$

$$F_{x,R} = F_{x,RR} = F_{x,RL} \quad (5.9)$$

Reagrupamos:

$$F_{x,F} = \frac{M_{z,front}}{t_f} \quad (5.10)$$

$$F_{x,R} = \frac{M_{z,rear}}{t_r} \quad (5.11)$$

Y obtenemos la fuerza que ha de generarse en cada rueda. Como el cálculo de la fuerza se ha hecho en módulo exclusivamente, luego debemos hacer que a cada rueda se le sume o reste el par que debiera.

Sin embargo, este dato de por sí sólo no nos aporta mucha información, pues lo que podemos controlar más fácilmente en el vehículo es el par de cada rueda. Por tanto, se planteará un balance en una rueda, y a partir de él sacaremos las ecuaciones:

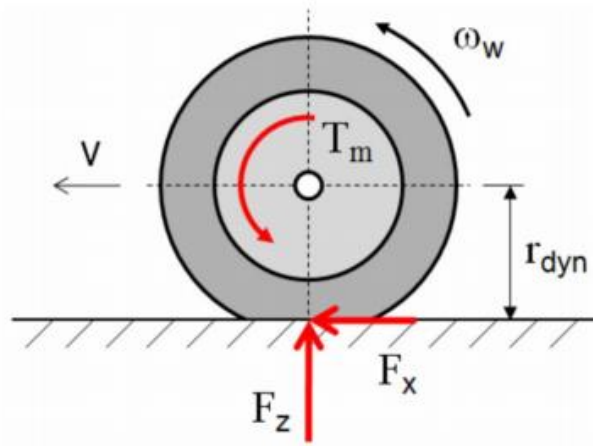


Figura 5-10 Balance de fuerzas en una rueda

Donde V es la velocidad longitudinal de la rueda, F_z la fuerza en el eje z de la rueda, F_x la fuerza en el eje x de la rueda (la que provoca tracción), ω_w la velocidad angular de la rueda en el eje perpendicular a la misma, T_m el par que aporta el motor de la rueda y r_{dyn} el radio dinámico de la rueda. Es bastante complicado conocer este valor con exactitud, pues el estudio de la deformación de una rueda en movimiento es complejo, de manera que simplificamos tomando como radio dinámico el radio de la rueda sin deformar r .

La ecuación obtenida de este balance es:

$$T_m - F_x * r = I_{yy} * \dot{\omega} \quad (5.12)$$

Donde I_{yy} es el momento de inercia de la rueda en su eje de giro.

Despejando la fuerza obtenemos:

$$F_x = \frac{T_m - I_{yy} * \dot{\omega}}{r} \quad (5.13)$$

De esta manera, obtenemos cuatro ecuaciones, una por rueda. Si igualamos a las expresiones (5.10) y (5.11) obtenemos:

$$T_{m,i} = \frac{r}{t_f} * M_{front} + I_{yy} * \dot{\omega}, \text{ donde } i = FR, FL \quad (5.14)$$

$$T_{m,i} = \frac{r}{t_r} * M_{rear} + I_{yy} * \dot{\omega}, \text{ donde } i = RR, RL \quad (5.15)$$

Por tanto, ya sabemos qué par hay que aplicarle a cada rueda para conseguir el momento deseado. Como todo este estudio se está haciendo para un momento adicional al que ya tiene el coche, este sistema de control inferior necesita sumar y restar estos pares anteriormente calculados al par que actualmente tiene cada rueda. Para saber en qué ruedas hay que sumar, y en qué ruedas hay que restar par, se propone un algoritmo basado en el yaw rate actual y el yaw rate de referencia. Lo que se hace es cambiarle el signo del par a añadir a cada rueda, de manera que luego se suman los pares.

Este algoritmo se basa en el método empleado para calcular las reglas de la lógica difusa, pero se hace énfasis no en el signo del momento, sino en los errores entre el yaw rate y la referencia o valor deseado. La nomenclatura empleada es yaw_r para yaw real, yaw_d para yaw deseado.

- Si $\text{yaw}_r > \text{yaw}_d$:
 - o Si ambos son mayores que cero se debe restar par a las ruedas de la izquierda, y sumar a las de la derecha.

- Si ambos son menores que cero se debe sumar par a las ruedas de la izquierda, y restar a las de la derecha.
- Si yaw_r es mayor que cero, y yaw_d es cero, se debe restar par a las ruedas de la izquierda, y sumar a las de la derecha.
- Si yaw_r < yaw_d
 - Si ambos son mayores que cero, se debe sumar par a las ruedas de la izquierda, y restar a las de la izquierda.
 - Si ambos son menores que cero, se debe restar par a las ruedas de la izquierda, y sumar a las de la derecha.
 - Si yaw_r es menor que cero, y yaw_d es cero, se debe sumar par a las ruedas de la izquierda, y restar a las de la derecha.

Por tanto, si al par que pide el conductor para cada rueda se le suman estos pares con signo, se obtendrá el momento de guiñada deseado. Para calcular el par demandado por el conductor, se hace un reparto equitativo de dicho par. Es decir, si pide T, se aplicará T/4 a cada rueda más el calculado por el controlador. En esto se basa que el controlador funcione bien, en que se varía el par que recibe de cada rueda, de manera que podemos conseguir más giro del que en un principio se conseguiría sin el control.

5.5 Par Máximo Transmisible (MTTE)

Una vez tenemos el par que se va a aplicar a cada rueda, tenemos que pasarlo por una última etapa en la que se calcula el par máximo que se le podría dar a cada rueda para evitar un deslizamiento elevado, de manera que si el sistema de control inferior pide un par mayor al calculado en esta etapa, se saturará a este valor, reduciendo el deslizamiento que podría producirse al dar un par mayor.

Para ello, se empleará el modelo propuesto por Dejun Yin, Sehoon Oh, y Yoichi Hori [11], llamado MTTE (Maximum Transmissible Torque Estimation), o Estimación del Par Máximo Transmisible. La elección de esta técnica se basa en la complejidad de estimar el deslizamiento de la rueda, lo cual no ocurre con este método. Como su propio nombre indica, es una estimación, de manera que no podemos usarlo como único elemento de control de tracción, pero sí puede ser útil como elemento final.

Primero, se proponen las siguientes ecuaciones que describen la dinámica longitudinal del vehículo:

$$I_{yy} * \dot{\omega} = T - r * F_d \quad (5.16)$$

$$M\dot{V} = F_d - F_{dr} \quad (5.17)$$

$$V_w = r * \omega \quad (5.18)$$

$$F_d = \mu N \quad (5.19)$$

Donde I_{yy} es el momento de inercia de la rueda respecto el eje perpendicular a la misma, $\dot{\omega}$ es la aceleración angular de la rueda, T es el par de la rueda, r su radio, V la velocidad del vehículo, V_w la velocidad lineal de la rueda, M la masa del vehículo, F_d es la fuerza en la rueda según el eje longitudinal del coche, y F_{dr} es la fuerza de empuje total del vehículo. En la siguiente figura se muestran estas variables:

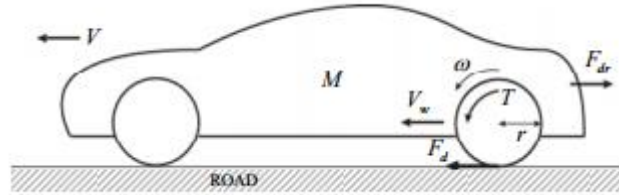


Figura 5-11 Variables de la dinámica longitudinal de un coche

Para conocer el valor de F_d tenemos dos caminos. En simulación, esta fuerza se obtiene del modelo directamente, pero en el vehículo no es una medida directa de ningún sensor, así que tenemos que calcularla con la siguiente ecuación:

$$F_d = \frac{T}{r} - \frac{I_{yy} \cdot \dot{V}_w}{r^2} \tag{5.20}$$

Una vez tenemos todos los valores, necesitamos definir el siguiente cociente:

$$\alpha = \frac{\dot{V}}{\dot{V}_w} = \frac{(F_d - F_{dr})/M}{(T_{max} - r \cdot F_d)r/I_{yy}} \tag{5.21}$$

Donde este coeficiente, conocido como factor de relajación, da una medida sobre el deslizamiento del vehículo. Según el autor, α debería ser mayor que la unidad para asegurar una buena estabilidad. Sin embargo, cuando es así, el par máximo estimado que se puede aplicar a la rueda T_{max} es inferior al que se podría aplicar en la realidad, de manera que el controlador no actuaría correctamente. Por tanto, se debe tomar un valor de α ligeramente inferior a 1 para mejorar el efecto del deslizamiento. Tras varias pruebas con nuestro modelo, se toma $\alpha = 0.9$.

Finalmente, se calcula el par máximo transmisible como:

$$T_{max} = \left(\frac{I_{yy}}{\alpha \cdot M \cdot r^2} + 1 \right) r \cdot F_d \tag{5.22}$$

Una vez tenemos las ecuaciones de este controlador, se plantea el siguiente esquema de control:

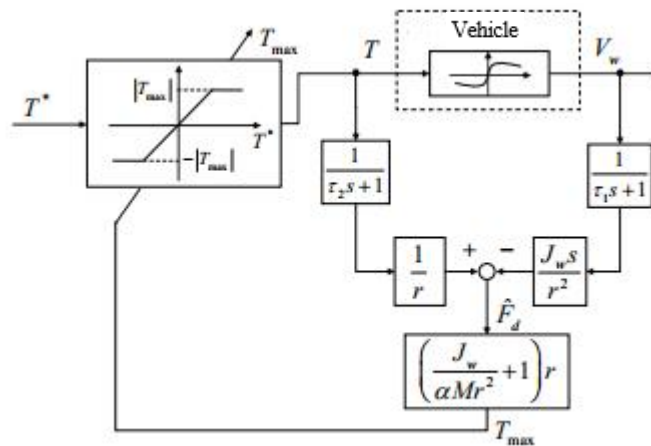


Figura 5-12 Esquema de control de MTTE

Por tanto, como se adelantó al principio, se calculará el par máximo que podría transmitirse al vehículo, y si el par que llega a esta etapa es mayor que el calculado, se dará a la rueda el par máximo estimado, y si es inferior, se dará el par completo sin modificar.

6 SISTEMA DE CONTROL DE ESTABILIDAD BASADO EN CONTROLADOR PROPORCIONAL

Este capítulo consiste en el desarrollo teórico de otro controlador de estabilidad, esta vez basado en un control proporcional. De la misma manera que en el anterior capítulo, se van a presentar ecuaciones para calcular algunos parámetros necesarios para que funcione el controlador, pero no se van a usar todas pues el modelo nos ofrece directamente algunas de estas variables. No obstante, de cara a futuras pruebas en el vehículo real, sí pueden llegar a ser necesarias muchas de estas ecuaciones pues no se dispone de todos los sensores necesarios.

6.1 Estructura del sistema de control

La estructura de este sistema de control consiste en cuatro bloques bien diferenciados. El primero de ellos, es el llamado Supervisor, que decide si se activa o no el sistema de control, y decide el modo de control cuando está activo, pues existen dos modos: de maniobrabilidad y de estabilidad lateral; el Superior, cuya función principal es calcular una referencia de yaw rate y de ángulo de deslizamiento en función del estado actual del vehículo para el siguiente bloque, el Inferior, cuya tarea consiste en calcular el par que se debe aplicar a cada rueda para modificar la dinámica del vehículo. Esta última parte incorpora un control de tracción, que de la misma manera que en el sistema de control basado en lógica difusa, mejora la adherencia de las ruedas. En la siguiente figura se puede observar este esquema, basado en el propuesto por J. Bravo Riera y P. Luque Rodríguez [12]:

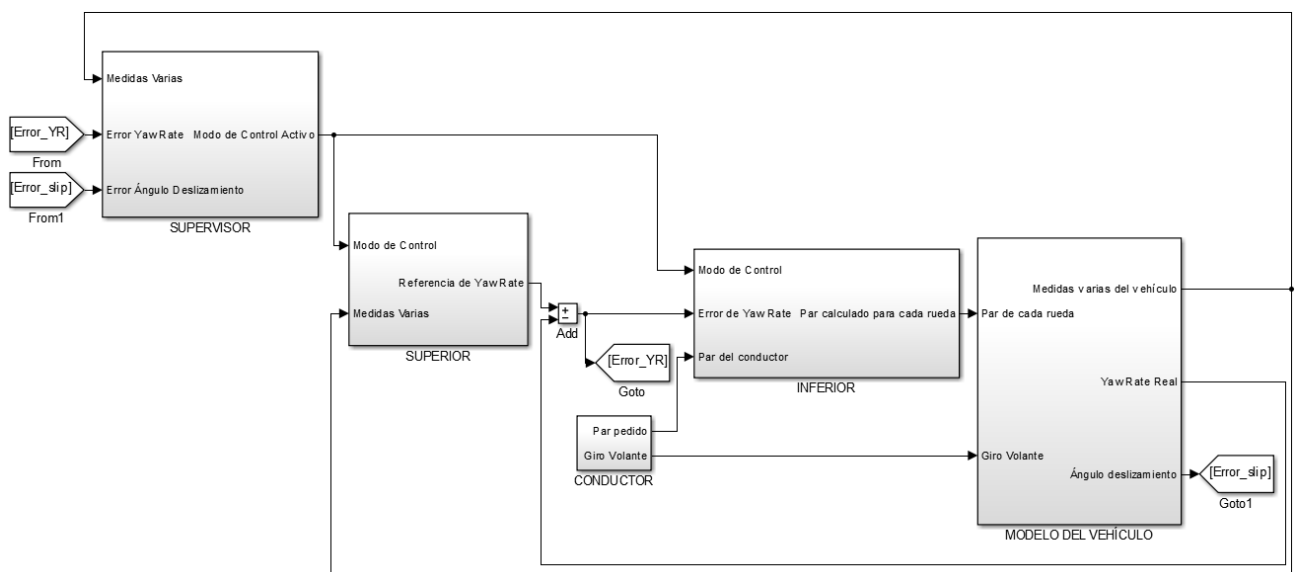


Figura 6-1 Esquema de Control 2

Es necesario hacer algunas aclaraciones. En primer lugar, este esquema de control no puede usarse con el primero propuesto en este trabajo, son dos controles diferentes cuyo único punto en común es su objetivo, mejorar la dinámica del vehículo. Por tanto, o bien se usa uno, o se usa el otro.

Por otra parte, el modo de control activo afecta principalmente al subsistema Superior, pues en función del modo activo genera una referencia u otra. Sin embargo, el subsistema de control Inferior también tiene como entrada el modo de control: en este caso, le concierne únicamente si está activo el controlador o no, para decidir si el par que se va a aplicar a cada rueda es el que pide el conductor, o por el contrario es el que se ha calculado.

6.2 Sistema de Control Supervisor

Esta primera parte del sistema de control consiste en la elección del modo de control a activar, si es necesario aplicar alguno. Para ello, nos basamos en los propuestos por Yoon, J., Yim, S., Cho, W., Koo, B., & Yi, K. [13], que son dos: el modo de maniobrabilidad, ESC- γ (*Electronic Stability Control*), y el modo de estabilidad lateral, ESC- β . El primer modo tiene como objetivo corregir la trayectoria de un vehículo circulando sobre una carretera de elevada adherencia, mientras que el segundo modo pretende mejorar el comportamiento sobre una carretera de baja adherencia. Debido a que en este trabajo no se van a considerar cambios de adherencia en la carretera, pues a priori es complejo determinar la adherencia de una carretera y sus coeficientes, vamos a usar el primer modo de control como corrección del yaw rate del vehículo cuando el ángulo de deslizamiento del vehículo sea bajo, y el segundo modo de control cuando este ángulo sobrepase un umbral. Por tanto, para cambiar de un modo de control a otro, se va a ver si las variables controladas sobrepasan ciertos límites.

El modo de control ESC- γ (maniobrabilidad), entra en juego cuando el error entre el yaw rate calculado como referencia y el real supere el umbral de 0.08 rad/s. Como se dijo antes, esto va a ayudar a corregir la trayectoria del vehículo, pues es la variable que se intenta controlar. En el segundo modo de control ESC- β (estabilidad lateral) se activa cuando el ángulo de deslizamiento del vehículo supera los 3° , y pretende reducirlo, por lo que será un sistema de control más agresivo, pues si el ángulo supera los $\pm 12^\circ$, se produce la inestabilidad del vehículo [5]. Por tanto, se entiende que este modo tiene prioridad sobre el de maniobrabilidad, pues está destinado a evitar que el vehículo se vuelva inestable, mientras que el otro pretende mejorar la conducción.

En la siguiente figura podemos ver una representación de la activación de los modos de control:

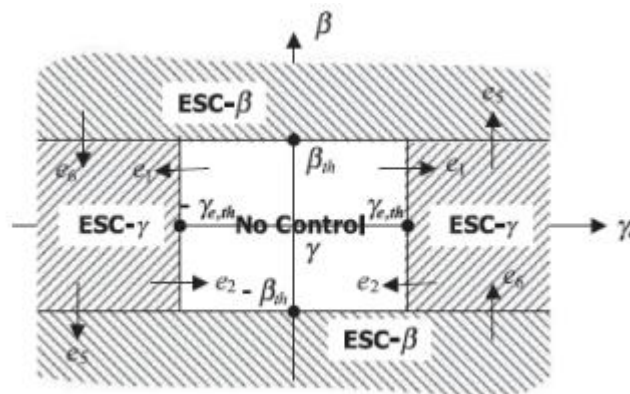


Figura 6-2 Esquema de la activación de los modos de control

Para entender mejor esta ilustración, hay que identificar que en el eje de abscisas se tiene el error de yaw rate, y en el eje de ordenadas el ángulo de deslizamiento, no su error, pues de manera análoga a lo que ocurría en el controlador de lógica difusa, se toma como referencia de este ángulo 0° , la cual no se va a poder alcanzar nunca en una curva pues al girar, el vehículo tiene las ruedas orientadas hacia un lado, por lo que es imposible que sea 0° ya que siempre va a haber una componente de velocidad transversal. Por otro lado, se ve la prioridad del modo de estabilidad lateral antes mencionada. Se observa que en la zona central no hay control activo, pues se entiende que el vehículo va dentro de los límites de error de las variables controladas. De cara al reparto de par, se hará igual que en el capítulo 5.

Una vez explicada la activación de cada modo de control (transiciones e1 y e5), merece mencionarse la desactivación de cada modo (transiciones e2 y e6). En un principio, para desactivarse cada modo de control podría valer con los mismos umbrales que para activarse. Sin embargo, podríamos hacer que el sistema se

inestabilice pues si una variable está oscilando en torno a los umbrales, estaríamos cambiando de referencia muy a menudo y, por tanto, de par aplicado a cada rueda, y para el conductor no es bueno que los pares varíen mucho de un instante a otro. Por tanto, se propone para la activación y desactivación de los modos de control una histéresis: para activarse el modo de control ESC – γ , se debe superar el umbral de 0.08 rad/s, y para desactivarse, el error debe ser de 0.01. De la misma manera, el modo ESC – β se activará cuando el ángulo de deslizamiento supere los 3°, y se desactivará cuando llegue a 1°. Se toman estos valores para desactivarse pues para dejar cada modo tenemos que asegurarnos de que no se va a oscilar en los umbrales, y teniendo en cuenta los valores de estas variables cuando se va en línea recta (donde no debería de estar activo el control pues el vehículo responde como el conductor pide en asfalto seco). Más adelante, cuando se enseñe la implementación del modelo en *Simulink* y cuando se presenten los resultados, se volverá a comentar esto, pues tiene otras implicaciones.

6.3 Sistema de Control Superior

Una vez se ha decidido el modo de control a activar, debemos generar la referencia de yaw rate para mejorar la maniobrabilidad y/o la estabilidad.

6.3.1 Modo de maniobrabilidad ESC – γ

Este modo tenía como objetivo que el vehículo describa una trayectoria con el yaw rate de referencia. Como hemos visto en el capítulo 5, las ecuaciones de la dinámica del vehículo son bastante complejas, lo cual hace que calcular referencias con un modelo teórico relativamente completo del vehículo no sea una solución viable, por lo que se va a usar el modelo simplificado de un vehículo con dos grados de libertad, el llamado modelo de la bicicleta. Este modelo se emplea mucho por su simplicidad y su buena aproximación a la realidad, y sus ecuaciones [13] son:

$$\begin{bmatrix} \dot{\beta} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} -2 \cdot \frac{C_{ad} + C_{at}}{m \cdot V_x} & 2 \cdot \frac{-l_1 \cdot C_{ad} + l_2 \cdot C_{at}}{m \cdot V_x^2} - 1 \\ 2 \cdot \frac{-l_1 \cdot C_{ad} + l_2 \cdot C_{at}}{I_{zz}} & -2 \cdot \frac{l_1^2 \cdot C_{ad} - l_2^2 \cdot C_{at}}{I_{zz} \cdot V_x} \end{bmatrix} \begin{bmatrix} \beta \\ r \end{bmatrix} + \begin{bmatrix} \frac{2 \cdot C_{ad}}{m \cdot V_x} \\ \frac{2 \cdot l_1 \cdot C_{at}}{I_{zz}} \end{bmatrix} \cdot \delta + \begin{bmatrix} 0 \\ \frac{1}{I_{zz}} \end{bmatrix} \cdot M_z \quad (6.1.)$$

Donde los parámetros son:

- C_{ad} y C_{at} : rigidez a la deriva de los neumáticos delanteros y traseros (cornering stiffness)
- m : masa del vehículo
- V_x : velocidad longitudinal del vehículo
- l_1 y l_2 : distancia del eje delantero y trasero al centro de gravedad del vehículo
- I_{zz} : momento de inercia del vehículo respecto al eje z (perpendicular al suelo)
- β : ángulo de deslizamiento del vehículo
- r : yaw rate
- δ : ángulo de giro de la rueda frontal
- M_z : momento en el eje z del vehículo

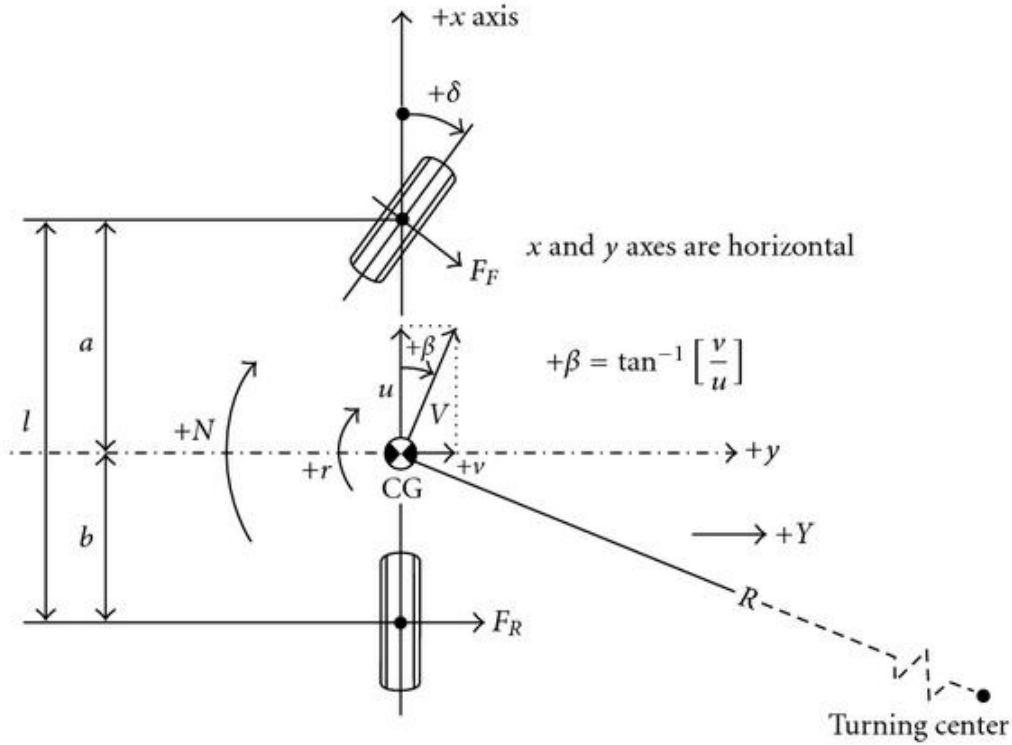


Figura 6-3 Modelo de la bicicleta

A partir del sistema de ecuaciones (6.1), se obtiene el yaw rate de referencia:

$$\gamma_{ref_yaw} = \frac{1}{1 - m \cdot v_x^2 \frac{l_1 \cdot c_{\alpha d} - l_2 \cdot c_{\alpha t}}{2 \cdot c_{\alpha d} \cdot c_{\alpha t} \cdot (l_1 \cdot l_2)^2}} \cdot \frac{v_x}{l_1 + l_2} \cdot \delta \quad (6.2)$$

6.3.2 Modo de estabilidad ESC – β

El objetivo de este modo es, como su propio nombre indica, mejorar la estabilidad del vehículo, para lo cual pretende reducir el ángulo de deslizamiento del vehículo. Para obtener la referencia en este modo de control, se hace un balance de fuerzas en el eje y del vehículo del modelo anteriormente presentado:

$$m \cdot \dot{V}_y = -m \cdot V_x \cdot \gamma + 2 \cdot F_{yd} \cdot \cos \delta + 2 \cdot F_{yt} \quad (6.3)$$

Donde F_{yd} y F_{yt} son las fuerzas en el eje y del vehículo en el eje delantero y el trasero (en la figura 6-3, corresponde a las fuerzas F_F y F_R).

Como estamos suponiendo inestabilidad lateral, tomamos el caso en el que la aceleración longitudinal del vehículo es cercana a cero:

$$\dot{\beta} = -\gamma + \frac{2 \cdot F_{yt} \cdot \cos \delta + 2 \cdot F_{yt}}{m \cdot V_x} \quad (6.4)$$

De esta manera, podemos obtener la referencia de este modo:

$$\gamma_{ref_lateral} = K_{\beta} \cdot \beta + \frac{2 \cdot F_{yd} \cdot \cos \delta + 2 \cdot F_{yt}}{m \cdot V_x} \quad (6.5)$$

El parámetro K_{β} [13] debe tomar un valor mayor que cero, y se toma como 3.

De esta manera, ya tenemos las expresiones que nos indican el yaw rate de referencia del vehículo mediante las ecuaciones (6.2) y (6.5).

6.4 Sistema de Control Inferior

Una vez tenemos la referencia de velocidad angular que debe seguir el vehículo, tenemos que actuar sobre el mismo para conseguir llegar a ella, corrigiendo por tanto su dinámica. Para calcular el momento necesario a aplicar al vehículo, tomamos como elemento un controlador proporcional, propuesto por Hallowell y Ray [14]:

$$M_z = K \cdot I_{zz} \cdot (\dot{\gamma}_{ref} - \dot{\gamma}) \quad (6.6)$$

En esta ecuación, conocemos todos los términos excepto la ganancia del controlador proporcional K, la cual hay que sintonizar. Para ello, se realizan varias pruebas en simulación y se escoge el valor que obtenga unos resultados globales mejores entre todos los ensayados. Para el vehículo en el que se está probando este esquema de control, se ha tomado como ganancia 0.7. Se ha escogido este controlador por su sencillez y rapidez, aunque no se asegura que se alcancen las referencias.

Una vez calculado el momento correctivo que tenemos que generar en el vehículo, volvemos al capítulo quinto, en el cual obtuvimos las ecuaciones para pasar de este momento al par de cada rueda, las ecuaciones (5.14) y (5.15). Para obtener estas ecuaciones, recordamos que supusimos un reparto equitativo entre el eje delantero y el trasero a la hora de generar el momento adicional.

Con estas ecuaciones, tenemos el par que habría que sumar o restar al que pide el conductor para cada rueda. Para que sea más efectivo el par que se entrega, lo aplicamos de manera diferencial, pero siguiendo un esquema diferente al del capítulo anterior. En este caso, debido a que las referencias se calculan en función del estado actual del vehículo, y no con datos experimentales, las referencias se ajustan a la dinámica que debería de tener el vehículo, en vez de la que debemos evitar. Por tanto, el esquema para aplicar este par diferencial sería ahora:

- Si $M_z > 0$ y el control está activado (es necesario girar más hacia la izquierda):
 - o Sumo el par calculado a las ruedas de la derecha, y lo resto a las de la izquierda.
- Si $M_z < 0$ y el control está activado (es necesario girar más hacia la derecha):
 - o Resto el par calculado a las ruedas de la derecha, y lo sumo a las de la izquierda.
- Si el control está desactivado:
 - o Aplicar directamente el par que pide el conductor, haciendo un reparto equitativo entre las cuatro ruedas.

De esta manera, ayudamos al vehículo a girar como la referencia calculada indica, mejorando por tanto su comportamiento dinámico. Siguiendo esta línea, no es necesario diferenciar entre si el vehículo es sobrevirador o subvirador, pues estos datos se introducen en las ecuaciones necesarias para diseñar este controlador (6.2), y se ajusta automáticamente para cada caso. Más concretamente, se introduce si el vehículo tiende o no al sobreviraje al introducir la distancia de los ejes delantero y trasero al centro de gravedad: en nuestro caso, la distancia del eje trasero al centro de gravedad es inferior al delantero, por lo que se comprueba efectivamente esta condición.

6.4.1 Control de tracción

Con todo lo anterior, ahora tenemos el par que se debe aplicar a cada rueda. Sin embargo, aún tenemos una capa más de seguridad en este esquema de control, el control de tracción, basado en el MTTE (Maximum Transmissible Torque Estimation). El desarrollo se enseñó en el capítulo quinto: *Sistema de control de estabilidad basado en lógica difusa*, más concretamente en el subcapítulo cuarto, *Par máximo transmisible*.

6.5 Cálculo de Fuerzas en los Neumáticos

En un principio, ya tenemos todo lo necesario para poder implementar este controlador en el modelo de *Simulink* del FOX o de cualquier otro vehículo similar. Sin embargo, a la hora de implementarlo en vehículo real, debemos

recordar que no tenemos todos los parámetros que necesitamos, pues no hay tantos sensores. Estos parámetros son, principalmente, las fuerzas de cada rueda.

Para poder estimar las fuerzas de las ruedas, necesitamos tomar un modelo. En este caso, se va a tomar el modelo de Dugoff [15], principalmente por su sencillez, pues es necesario un modelo que proponga ecuaciones de rápida resolución dado que el tiempo de cálculo de la ECU es finito y pequeño.

Las fuerzas longitudinal y lateral en cada rueda se calculan como:

$$F_x = C_\lambda \cdot \frac{\lambda}{1+\lambda} \cdot f(\sigma) \quad (6.6)$$

$$F_y = C_\alpha \cdot \frac{\tan \alpha}{1+\lambda} \cdot f(\sigma) \quad (6.7)$$

Donde C_λ y C_α son las rigideces longitudinal y lateral, respectivamente. Para obtener estos valores, se toman los datos directamente del modelo del vehículo.

λ es el denominado slip ratio, y para calcularlo se dispone de la siguiente ecuación (ligeramente modificada respecto a la ecuación 2.3 para aclarar los términos):

$$\lambda = \frac{r \cdot \omega_w - v_x}{\max\{r \cdot \omega_w, v_x\}} \quad (6.8)$$

ω_w es la velocidad angular de la rueda, y r el radio efectivo de la rueda. Este radio se diferencia del radio de la rueda normal porque es el radio que tiene la rueda una vez deformada, es decir, cuando está soportando el peso del vehículo. En principio, puede variar, pero lo aproximamos por el radio de la rueda por simplicidad en los cálculos.

Volviendo a las ecuaciones (6.6) y (6.7), faltan dos términos por explicar:

El primero de ellos es α , que es el ángulo de deriva de cada rueda. Este ángulo se define para las ruedas delanteras (α_d) y traseras (α_t) como:

$$\alpha_d = \delta - \frac{v_y + l_1 \cdot \gamma}{v_x} \quad (6.9)$$

$$\alpha_t = - \frac{-v_y - l_2 \cdot \gamma}{v_x} \quad (6.10)$$

El segundo término es $f(\sigma)$, una función que depende de la variable σ

$$f(\sigma) = \begin{cases} (2-\sigma) \cdot \sigma & \text{si } \sigma < 1 \\ 1 & \text{si } \sigma \geq 1 \end{cases} \quad (6.11)$$

Donde σ se calcula como:

$$\sigma = \frac{\mu \cdot F_z \cdot (1+\lambda)}{2 \cdot [(C_\lambda \cdot \lambda)^2 + (C_\alpha \cdot \tan \alpha)^2]^{0.5}} \quad (6.12)$$

F_z es la carga vertical sobre cada rueda, y μ el coeficiente de rozamiento de la carretera. Para el caso del asfalto seco, toma el valor de 0.8 aproximadamente [15].

7 MONTAJE DE LOS CONTROLADORES EN SIMULINK

Para implementar el sistema de control en Simulink, hicimos una primera implementación con bloques que, aunque hacía que las simulaciones tardaran más, era más visual a la hora de ver el flujo de datos, mientras que en el segundo controlador se optó por una implementación también en bloques, para mantener esa visualización del proceso, pero con trozos de código en vez de bloques grandes, de manera que se gana en velocidad de simulación. En este capítulo se presentará el recorrido desde que se obtuvo el modelo, hasta que se consiguió un modelo en el que se diferenciaban claramente los bloques de control de los del vehículo.

7.1 Diagrama de bloques del vehículo

Lo primero que se hizo fue usar etiquetas “GOTO” y “FROM” de Simulink, para tener las variables accesibles sin tener que usar cables que cruzaran todo el modelo. Además, se emplearon los bloques “TO WORKSPACE” para poder tener los datos importantes en el workspace, y así poder presentar y comparar los datos de una manera más clara.

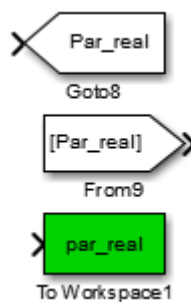


Figura 7-1 Bloques Goto, From y To Workspace

Una vez tenemos todas las variables que queremos con sus etiquetas, lo primero que se hizo fue comprobar que el coche funcionaba sin controlador. Para simular un conductor pisando el acelerador, se usó un controlador del tipo PI [1] para que el modelo calculara automáticamente el par que necesita para intentar mantenerse a una velocidad de referencia v_{ref} . Para calcularlo, necesita la velocidad actual y la velocidad de referencia, como se ve en las siguientes imágenes:

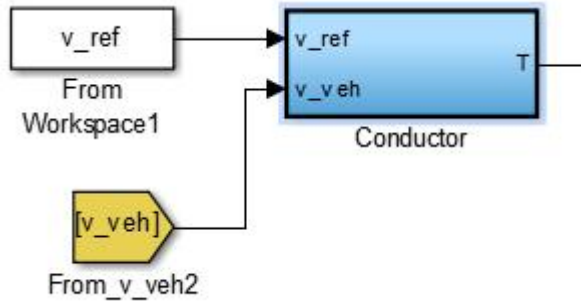


Figura 7-2 Bloque para el cálculo del par que pide el conductor

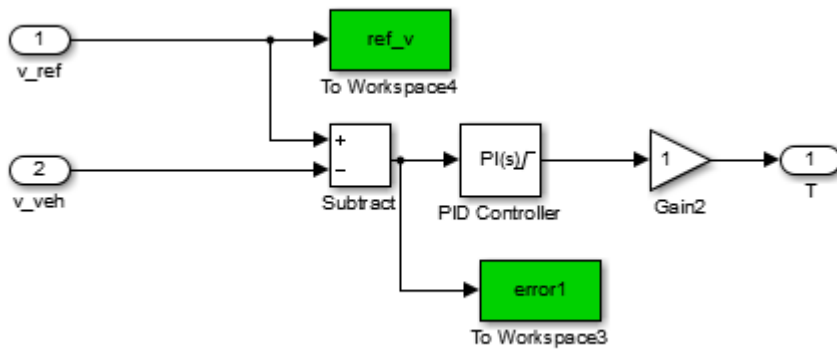


Figura 7-3 Interior del bloque “Conductor”

Una vez teníamos todas las señales con sus respectivas etiquetas, se redujo el modelo completo a un subsistema llamado “Coche”, para tenerlo más recogido, pues las variables internas del vehículo no las necesitamos, y las que sí son importantes las tenemos ya etiquetadas.

Ahora necesitamos meterle a este nuevo subsistema la trayectoria que se quiera hacer. Como se explicó con anterioridad cuando se introdujo el simulador, simplemente se mete el valor instantáneo del ángulo de giro del volante a probar. Uniendo esto a lo anterior, obtenemos un modelo sin controlar que podemos usar para luego comparar los resultados del controlador. A este modelo hay que añadirle una saturación de ± 78 Nm, para que el “conductor” no pida más par del que puedan dar los motores.

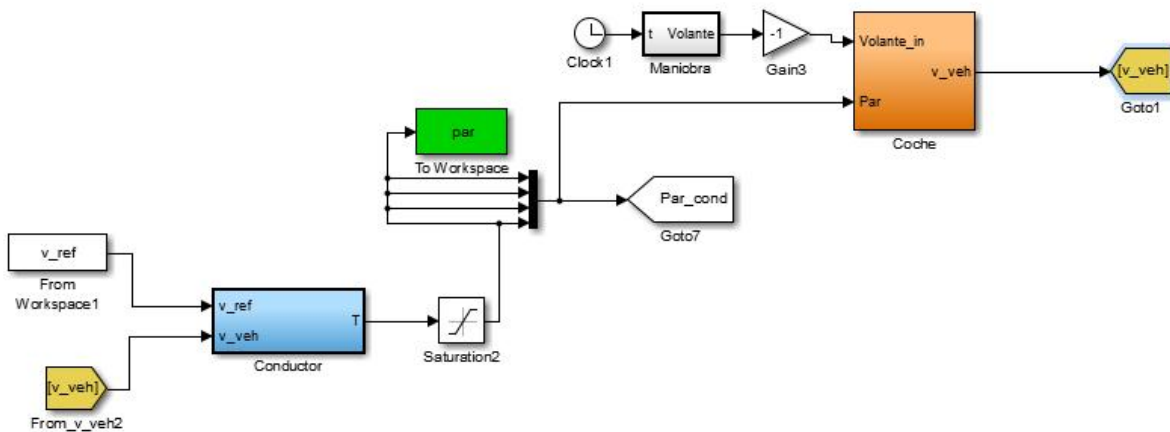


Figura 7-4 Modelo Coche sin control

Una vez tenemos este modelo, empezamos a introducir el controlador en bloques, de manera que se identifique cada parte del controlador. Como lo único que hace el controlador es modificar el par que recibe cada rueda, debemos habilitar que sea una entrada al coche, de manera que se modifica el modelo anterior para ello, añadiendo “Manual Switch” para poder escoger qué par queremos aplicarle al vehículo (conductor o calculado) y qué maniobra queremos realizar.

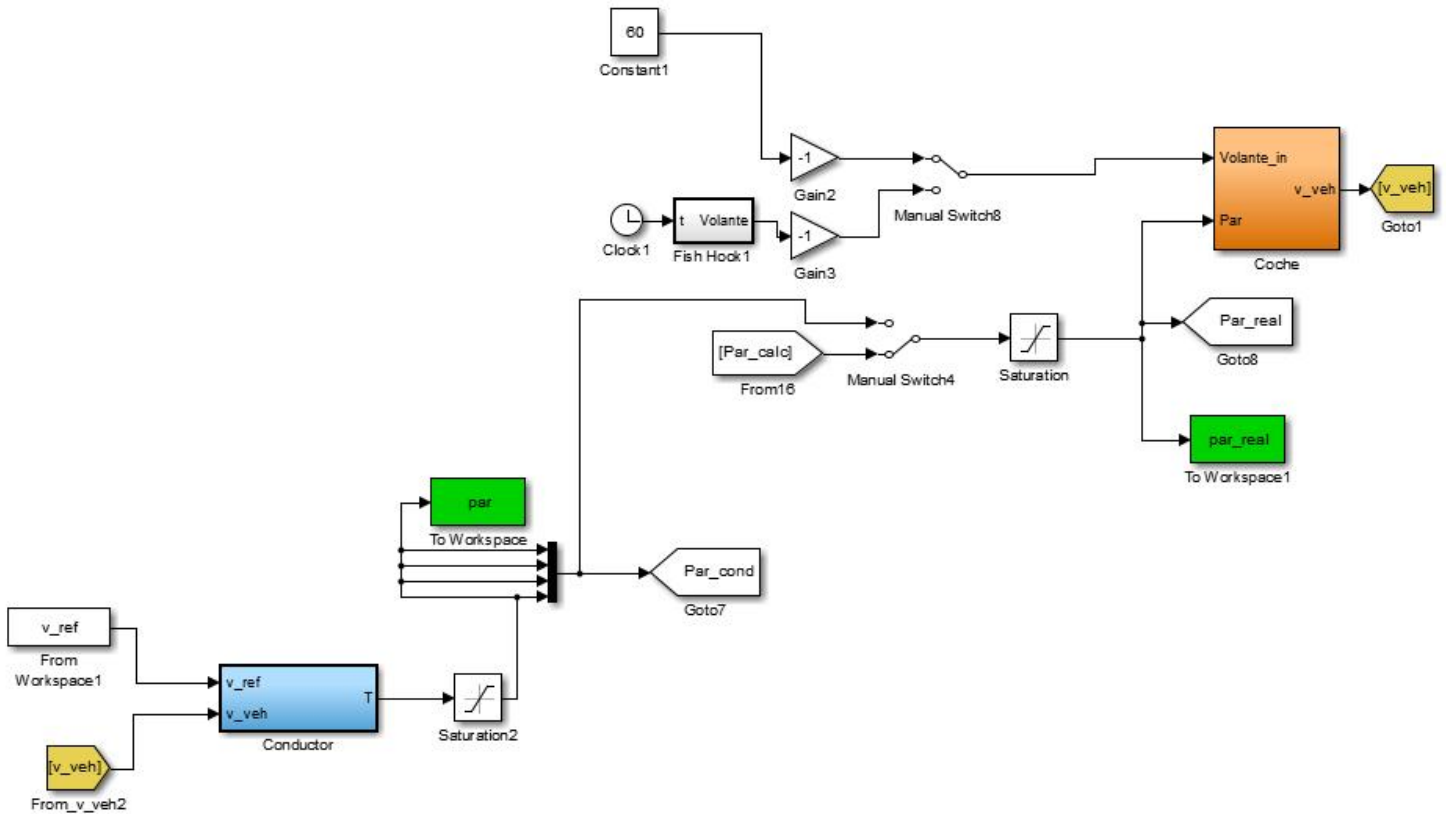


Figura 7-5 Modelo final del simulador con todas las señales

Debido a que en este trabajo se muestran dos controladores diferentes, es importante señalar que sólo uno actúa en cada simulación: o bien el de lógica difusa, o bien el basado en un controlador proporcional. Para cambiar de uno a otro, simplemente se cambia la etiqueta *Goto* que lleva la señal *Par_calc*, seleccionando un control u otro. De todas formas, se han creado dos modelos de *Simulink*, uno para cada controlador, pues no hay interacción alguna entre ellos, y el quitar bloques que no necesitamos hace que los cálculos tarden menos.

7.2 Controlador de Lógica Difusa

7.2.1 Diagrama de Bloques

Una vez conseguido el diagrama de bloques del vehículo, pasamos a la implementación del primer controlador en bloques bien diferenciados para poder hacer un seguimiento de los errores que iban surgiendo. Por tanto, obtenemos el siguiente diagrama de bloques:

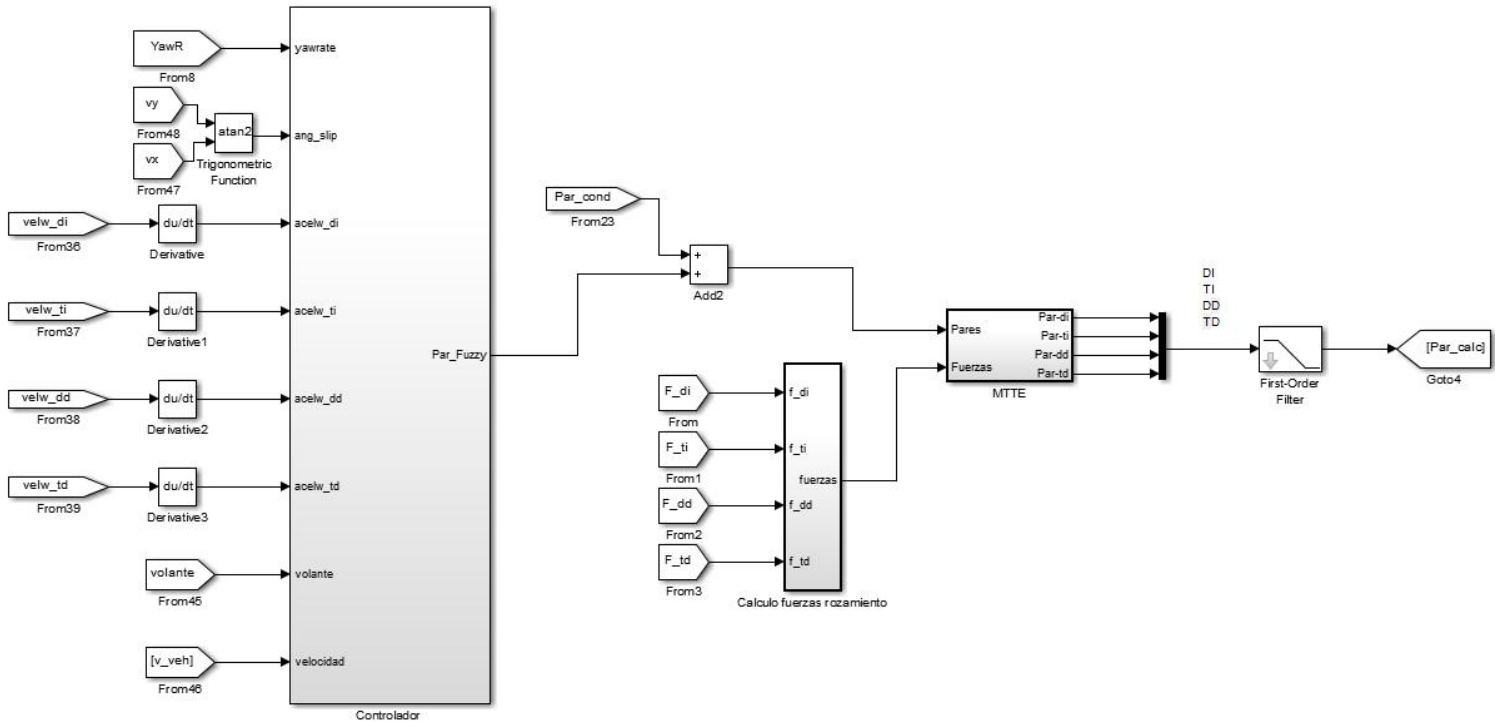


Figura 7-6 Diagrama de bloques del controlador

Explicamos un poco esta figura. Las entradas del sistema de control son, como antes se dijo, Yaw Rate real, ángulo de deslizamiento del vehículo, aceleración angular de cada rueda (calculada como la derivada de cada velocidad angular), el ángulo de giro del volante y la velocidad del vehículo. Con todas estas entradas, se calcula el par que hay que añadir/sustraer a cada rueda respecto al par que pide el conductor. Una vez tenemos el par modificado de cada rueda, se pasa por el subsistema MTTE para calcular el par final que se aplica a cada rueda. Al final se añade un filtro para evitar posibles ruidos en el par calculado.

A continuación, explico el interior del subsistema “Controlador”.

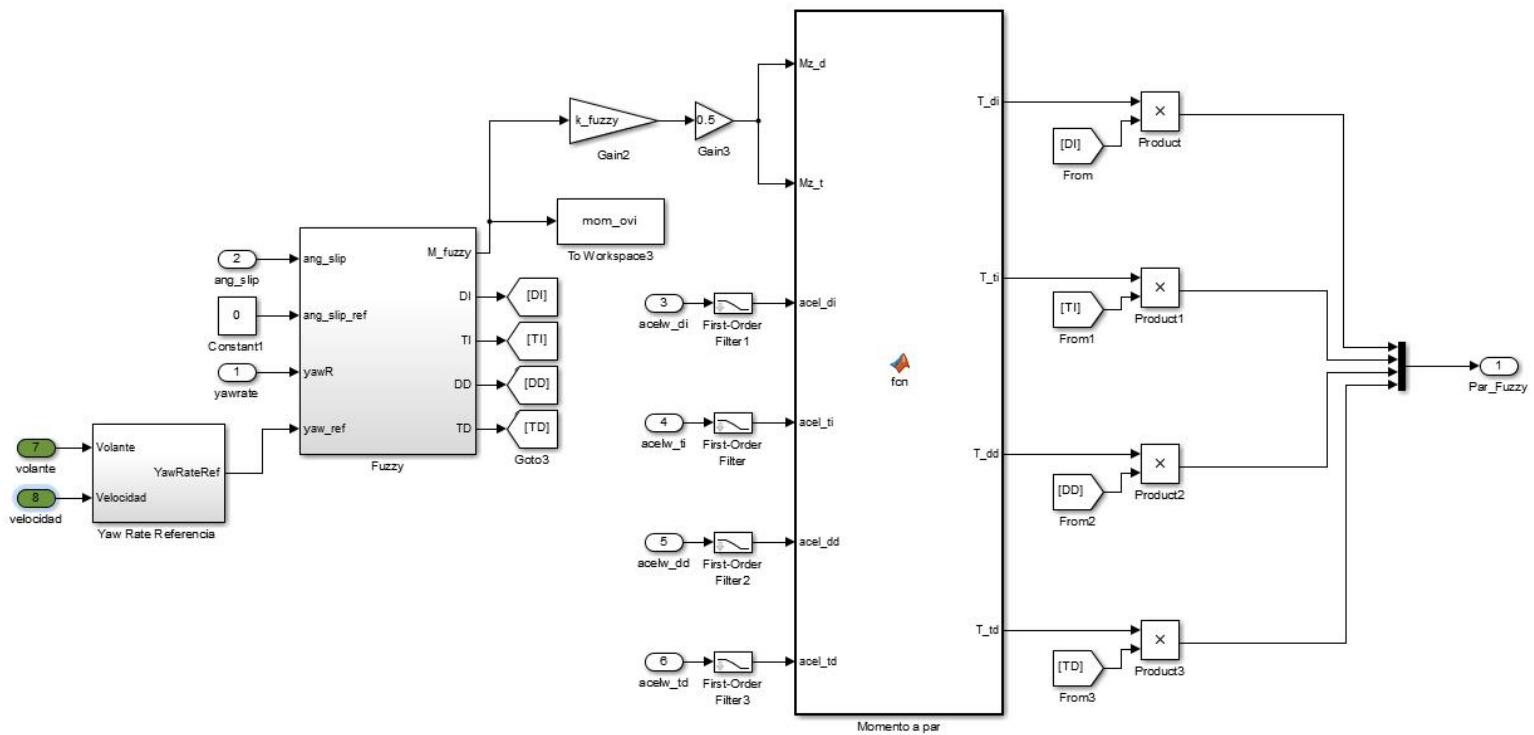


Figura 7-7 Bloque Controlador

En este bloque, como se observa en la figura 7-7, se incluye la generación de la referencia de Yaw Rate, el bloque de Lógica Difusa, el controlador de nivel inferior para pasar de momento a par, y el cálculo del signo del par adicional para cada rueda.

Ya se explicó en el desarrollo teórico del controlador, pero es conveniente recordar que el momento que calcula la lógica difusa pertenece al intervalo $[-1,1]$, de manera que se multiplica por la ganancia k_{fuzzy} , y luego se reparte equitativamente entre cada eje.

De izquierda a derecha, voy a comentar el contenido de los bloques de la figura anterior.

El interior del bloque que genera la referencia de yaw rate consiste en un bloque que mira el fichero donde está tabulado el yaw rate del coche a baja velocidad en función del ángulo de giro del volante, y lo multiplica por la velocidad.

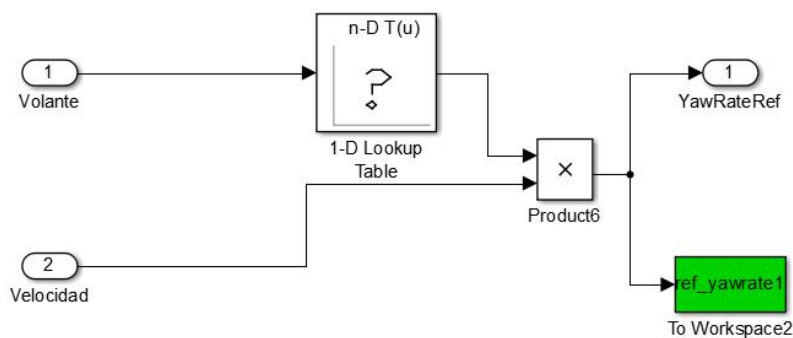


Figura 7-8 Bloque Generador de Referencia de Yaw Rate

El siguiente bloque, “Fuzzy”, consiste en la implementación de la lógica difusa en función de los errores de yaw rate y ángulo de deslizamiento, y como comparte las entradas del cálculo del signo del par que se añade a cada rueda, se hizo todo en el mismo bloque para simplificar.

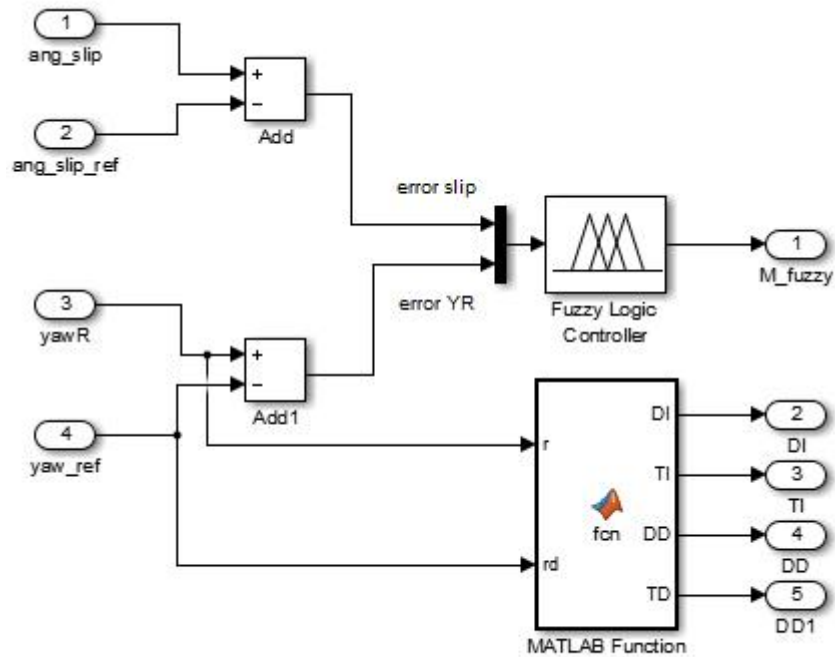


Figura 7-9 Bloque “Fuzzy”

El funcionamiento es sencillo, se da al bloque de lógica difusa los errores que necesita, y da la salida *M_fuzzy*, que es el momento normalizado antes mencionado. Este bloque de lógica difusa tiene como argumento el nombre del fichero *.fis* en el que están puestas las reglas lógicas, que se explicaron en el subcapítulo 5.3.

Para el cálculo del signo del par, almacenado en *DI*, *TI*, *DD* y *TD* (rueda Delantera Izquierda, Trasera Izquierda, Delantera Derecha, Trasera Derecha), se emplea el siguiente código de Matlab, donde *r* es el yaw rate del vehículo, y *rd* el yaw rate de referencia. Como se comentó en el desarrollo teórico, calcula el signo del par que hay que añadirle a cada rueda en función del yaw rate actual y el deseado:

```
function [DI, TI, DD, TD] = fcn(r, rd)
%#codegen
DI=1; TI=1; DD=1; TD=1; %si error=0, saldra esto
if ((r-rd)>0)
    if (r>0 && rd>0)
        DI=-1; TI=-1; DD=1; TD=1;
    end
    if (r<0 && rd<0)
        DI=1; TI=1; DD=-1; TD=-1;
    end
    if (r>0 && rd<0)
        DI=1; TI=1; DD=-1; TD=-1;
    end
    if (r>0 && rd==0)
        DI=-1; TI=-1; DD=1; TD=1;
    end
end

end
if ((r-rd)<0)
    if (r>0 && rd>0)
        DI=1; TI=1; DD=-1; TD=-1;
    end
    if (r<0 && rd<0)
        DI=-1; TI=-1; DD=1; TD=1;
    end
    if (r<0 && rd>0)
```



```

        DI=-1;TI=-1;DD=1;TD=1;
    end
    if (r<0 && rd==0)
        DI=-1;TI=-1;DD=1;TD=1;
    end
end
end

```

Volviendo a la figura 7-7, se explica a continuación el interior del bloque “Momento a par”, que consiste en otra función de Matlab en la que se incluyen las ecuaciones del balance (5.14) y (5.15). Por reducir la nomenclatura, $acel_di$ es la aceleración angular de la rueda delantera izquierda, y así con todas las aceleraciones.

```

function [T_di,T_ti,T_dd,T_td] =
fcn(Mz_d,Mz_t,acel_di,acel_ti,acel_dd,acel_td)
%#codegen
%acel_ang se compone de di ti dd td
r_d=0.25; %radio ruedas delanteras
r_t=0.28; %radio ruedas traseras
Iyyf=0.2334; %Momento de inercia de las ruedas delanteras
Iyyr=0.27; %Momento de inercia de las ruedas traseras
t_d=1.5538; %Anchura del eje delantero
t_t=1.4865; %Anchura eje trasero

T_di=Mz_d*(r_d/t_d)+Iyyf*acel_di; %Motor delantero izquierdo
T_dd=Mz_d*(r_d/t_d)+Iyyf*acel_dd; %Motor delantero derecho
T_ti=Mz_t*(r_t/t_t)+Iyyr*acel_ti;
T_td=Mz_t*(r_t/t_t)+Iyyr*acel_td;

```

Respecto al segundo sumando de las ecuaciones arriba mostradas cabe destacar que, al ser tan pequeños en valor, son casi despreciables, lo cual permite que no haya que filtrar antes de derivar la velocidad angular de las ruedas.

Por último, si se multiplican las variables DI, TI, DD y TD por el par que se acaba de calcular, se obtiene el par que hay que sumar a cada rueda.

Falta por explicar el último bloque, MTTE. Antes de este bloque, hay un subsistema que calcula las fuerzas de rozamiento, dato que necesita el MTTE. Simplemente consiste en coger las componentes x e y de las fuerzas de cada rueda según el eje respecto al que giran, y calcular su módulo. No se incluye la fuerza del eje z pues es la fuerza normal de cada rueda, y no es útil para el cálculo de las fuerzas de rozamiento. Estas fuerzas se obtienen del modelo de Simmechanics, no es necesario calcularlas.

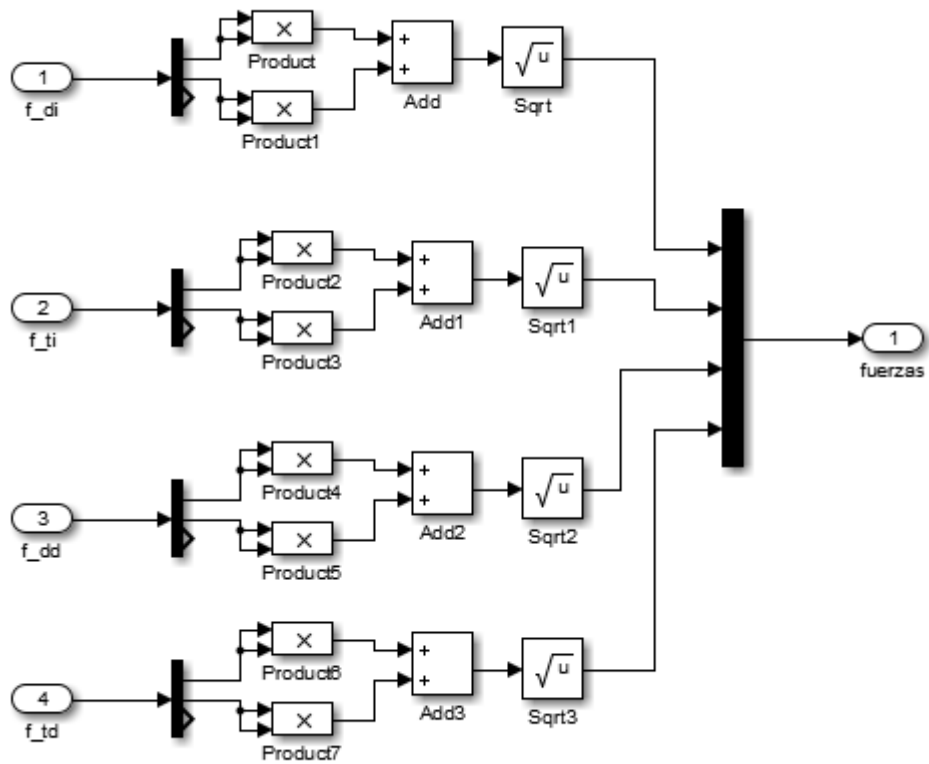


Figura 7-10 Bloque de Cálculo de las Fuerzas de Rozamiento

El interior del bloque MTTE se muestra a continuación:

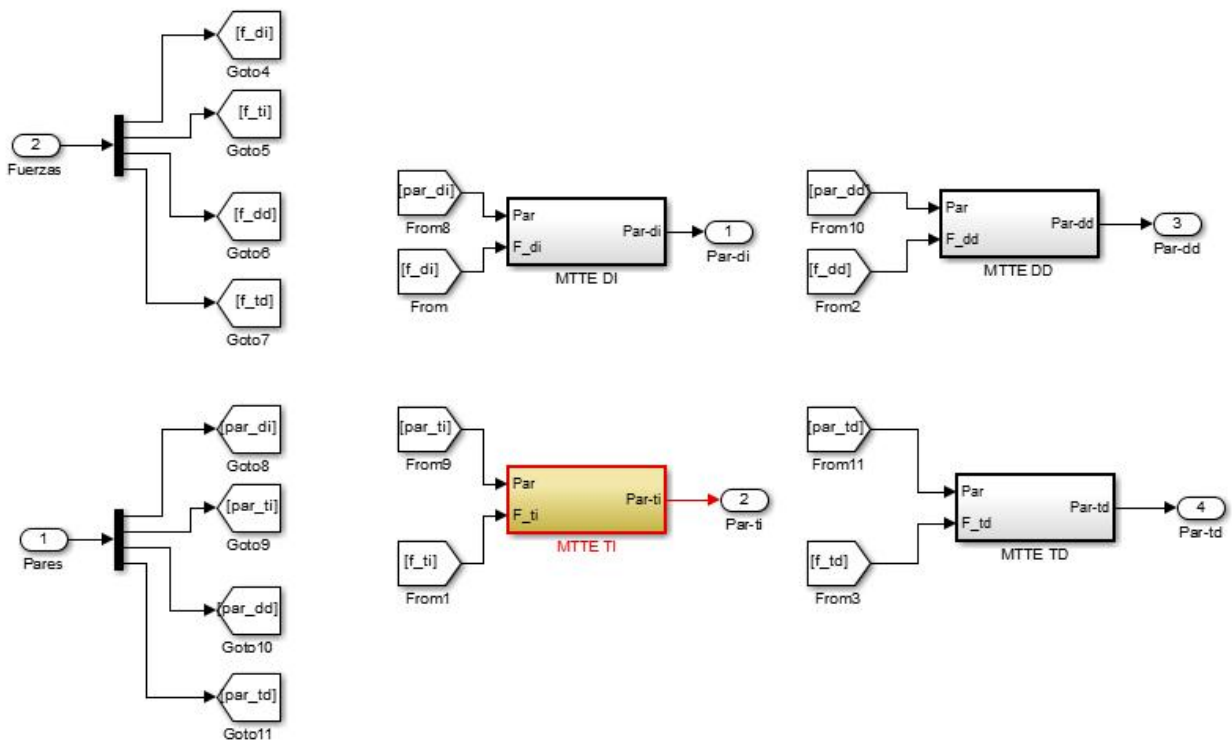


Figura 7-11 Bloque MTTE

Se observa que este bloque está modularizado para cada rueda, de manera que lo primero que se hace es separar las fuerzas y los pares en una señal por rueda. Luego, se tiene un bloque de cálculo del par máximo transmisible de cada rueda. Son todos iguales, implementan la ecuación (5.22), donde alfa se tomó como 0.9.

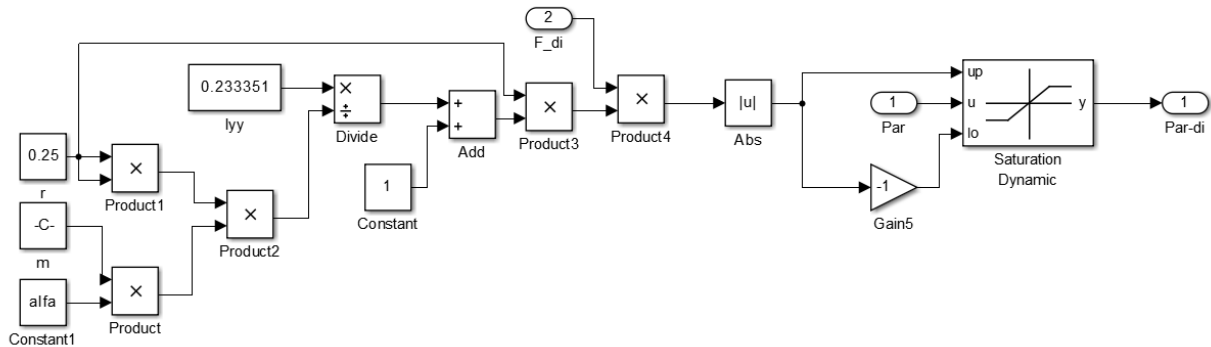


Figura 7-12 Cálculo MTTE

La parte final del bloque consiste en un bloque de saturación dinámica, donde se toma como valor de saturación máxima el par máximo transmisible calculado, y como valor de saturación mínimo el mismo con el signo cambiado. Si el par que se calculó que debería recibir cada rueda es mayor (o inferior) al máximo (o mínimo) calculado aquí, el par de la rueda será el máximo (o mínimo) transmisible aquí calculado.

7.2.2 Implementación Lógica Difusa en Simulink/Matlab

Antes se ha visto (Figura 7-9) que en Simulink existe un bloque para implementar controladores de lógica difusa, al que hay que pasarle como argumento un fichero *.fis* que contenga la lógica difusa. Para crear este fichero, se usa el *toolbox* de Matlab para lógica difusa.

Para abrir este *toolbox*, se debe introducir en Matlab el comando `>> fuzzyLogicDesigner`, lo cual abre una ventana en la que podemos implementar nuestro sistema de lógica difusa.

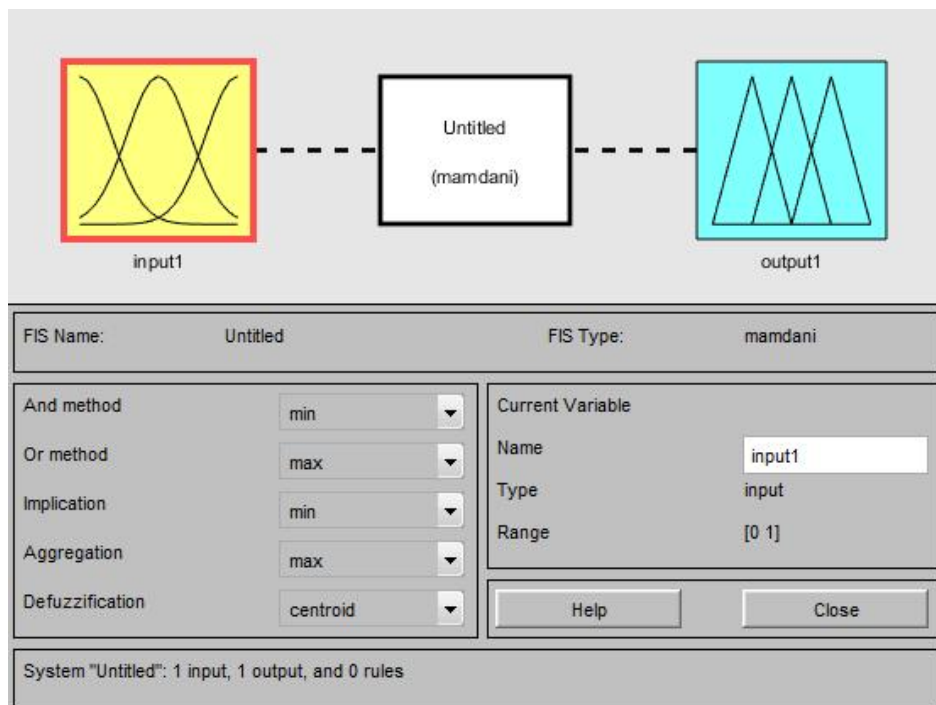


Figura 7-13 Ventana para diseñar sistemas empleando Lógica Difusa

Se ven tres bloques principales, input1, Untitled y output1. El primero, contiene las funciones de pertenencia de la entrada 1, el segundo, contiene las reglas lingüísticas que queremos implementar, y el último contiene las

funciones de pertenencia de la salida. Como nuestro sistema tiene dos entradas y una salida, debemos habilitar otra entrada y renombrarlas, obteniendo la Figura 7-14.

Una vez tenemos las entradas definidas, tenemos que modificarlas para implementar la lógica difusa desarrollada en el capítulo anterior. En las figuras 7-15, 7-16 y 7-17 se muestran las funciones de pertenencia de la variable de entrada slip, yaw rate y las de la variable de salida, respectivamente.

En la figura 7-18, se muestra cómo se implementan las reglas lingüísticas, y en la figura 7-19, se muestra el procedimiento que habría que seguir para calcular la salida, lo cual hace Matlab internamente.

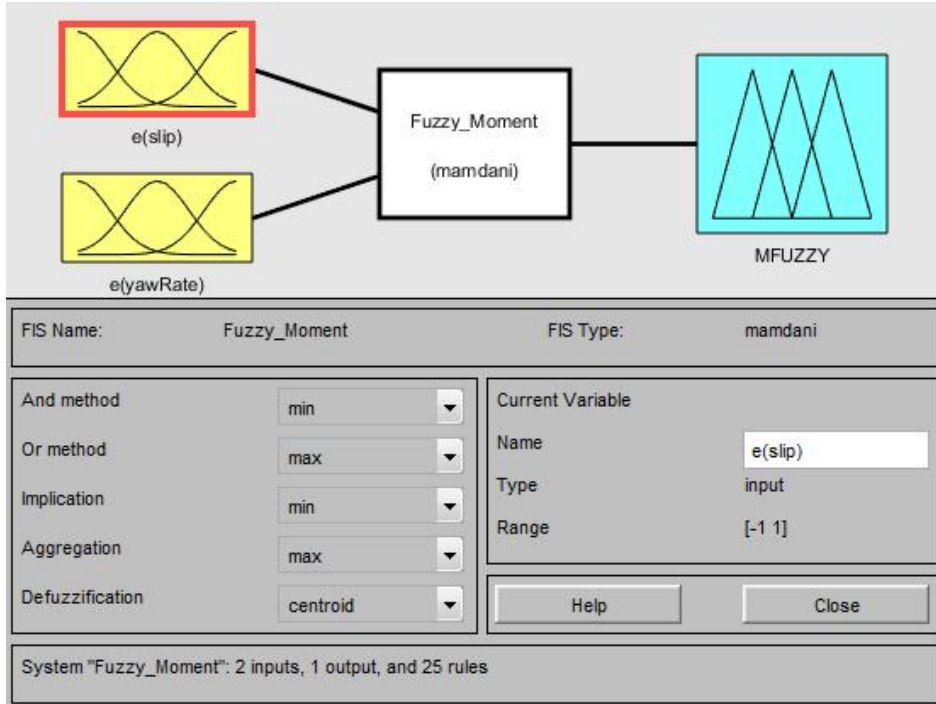


Figura 7-14 Ventana principal controlador Lógica Difusa



Figura 7-15 Funciones de pertenencia del error de deslizamiento



Figura 7-16 Funciones de pertenencia del error de yaw rate

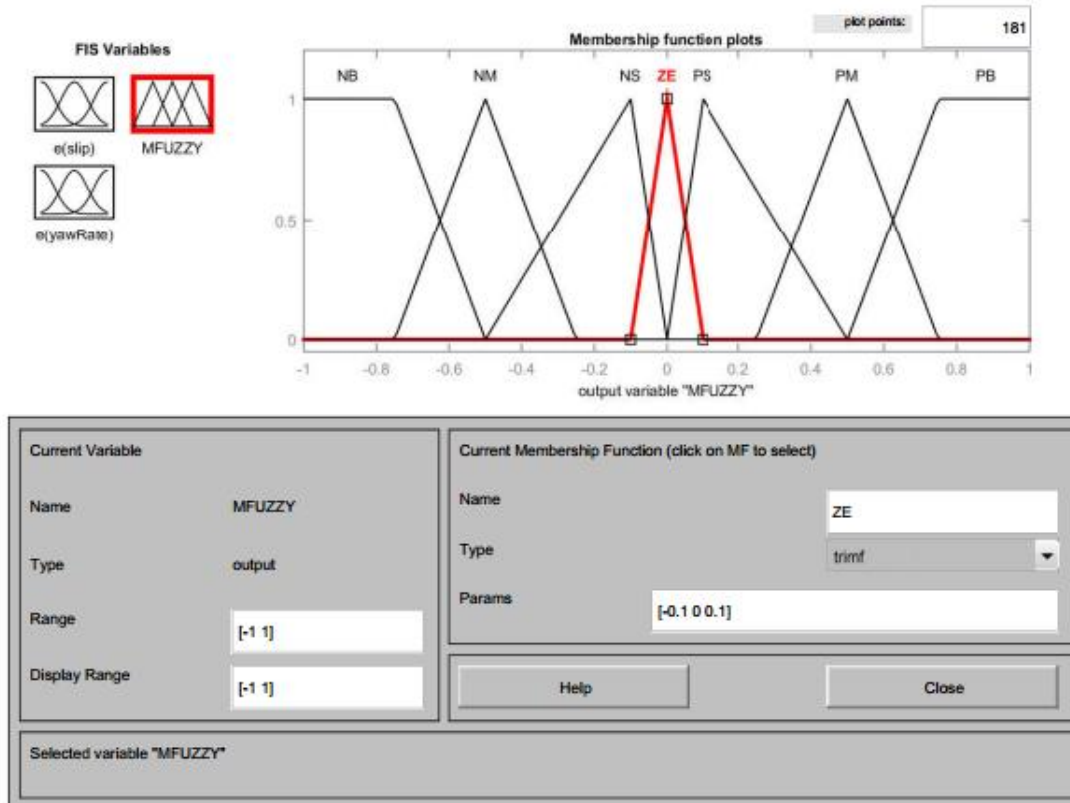


Figura 7-17 Funciones de pertenencia de la salida

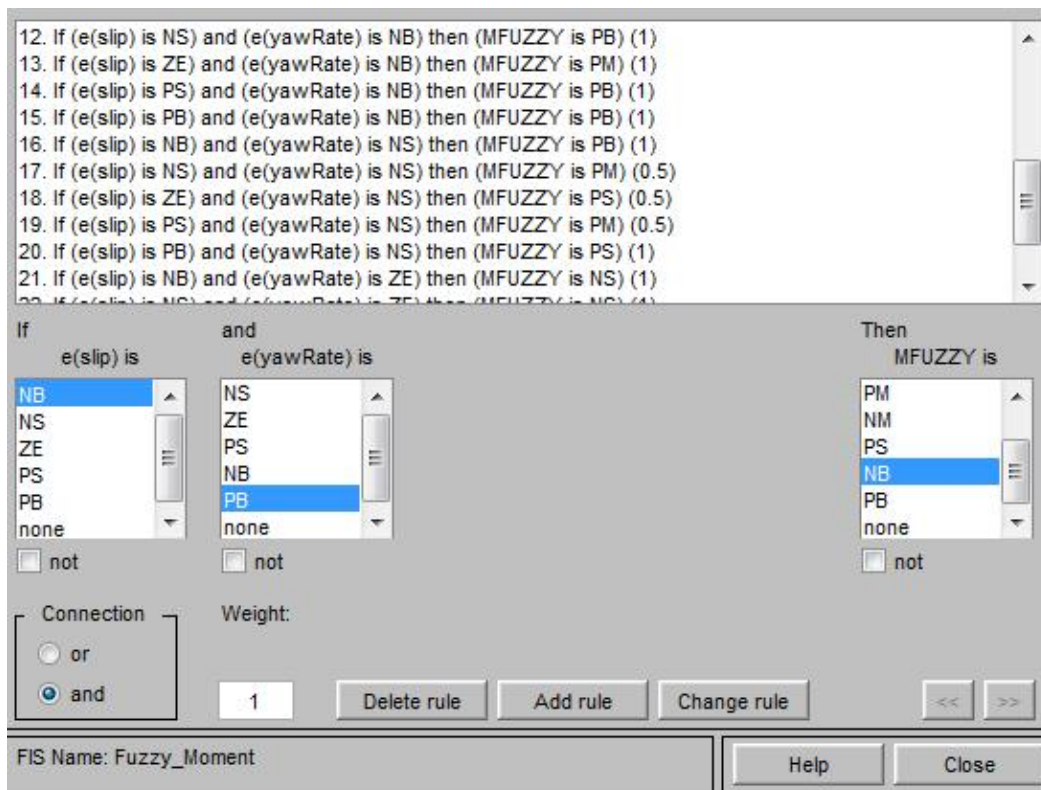


Figura 7-18 Implementación de reglas lingüísticas según el método Mamdani



Figura 7-19 Obtención de la salida de lógica difusa

7.3 Controlador Proporcional

Ahora vamos a implementar en *Simulink* el segundo controlador propuesto. Como ya se comentó con anterioridad, va a haber una diferencia principal en este apartado respecto al montaje del anterior controlador, pues en este caso se ha decidido usar bloques que contienen código de *Matlab* en vez de bloques de álgebra para que las simulaciones tarden menos tiempo, y para que luego sea más inmediato su paso a lenguaje C.

Para ello, recordamos el esquema general mostrado en la figura 6-1, en el que se presentaban tres bloques fundamentales del controlador: Supervisor, Superior e Inferior. En la figura 7-20, puede observarse ese esquema levemente modificado con todas las señales que el sistema de control necesita, así como los respectivos bloques *To Workspace* para luego poder visualizar ordenadamente los resultados.

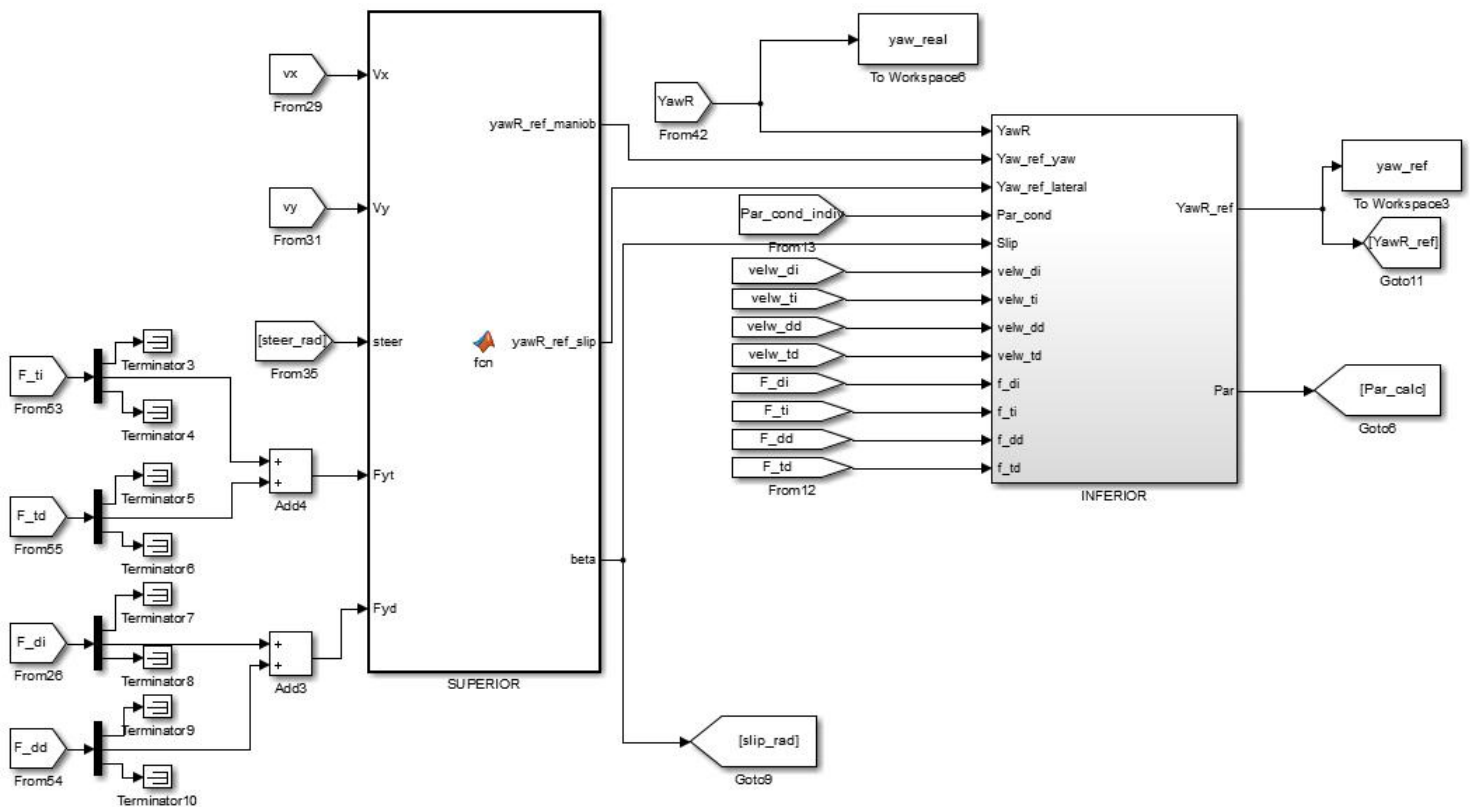


Figura 7-20 Diagrama de bloques del segundo controlador en *Simulink*

La principal diferencia de la figura 7-20 con la 6-1 es que no existe un bloque Supervisor diferenciado del resto. Esto es así pues para comprobar si tenemos que estar en un modo de control o en otro, uno de los parámetros que se evalúan es la diferencia entre el yaw rate actual y el de referencia. Por tanto, necesitamos estar generando estas referencias continuamente, esté activo o no el control, pues si no el sistema funcionará mal. En el subsistema Inferior sí es necesario saber si el control está activo o no, y en qué modo, por lo que las funciones del Supervisor las adopta el bloque Inferior.

A continuación, se van a explicar los dos grandes bloques del esquema, el Superior y el Inferior.

El bloque Superior se ocupaba de calcular las referencias de ambos modos, siguiendo las ecuaciones (6.2) y (6.5). Recordamos que para realizar estos cálculos, se necesita la velocidad longitudinal del vehículo, la velocidad lateral del vehículo, el ángulo de giro del volante, y las fuerzas laterales tanto en el eje trasero como en el delantero. Por tanto, su implementación en código es inmediata:

```
function [yawR_ref_maniob,yawR_ref_slip,beta]= fcn(Vx,Vy,steer,Fyt,Fyd)
```

```
M=0; %Inicializo a cero
m=400.238; %Masa del vehículo con dos ocupantes
Izz=1047.51412; %Momento de inercia del vehículo respecto al eje Z
l1=1.482; %Distancia del eje frontal al centro de gravedad
l2=1.048; %Distancia del eje trasero al centro de gravedad
Cd=70072; %Coeficiente de rigidez a la deriva de los neumáticos delanteros
Ct=91501.36; %Coeficiente de rigidez a la deriva de los neumáticos traseros
Kb=3; %Valor estrictamente positivo
beta=atan2(Vy,Vx); %Ángulo de deslizamiento con signo
```



```

%Cálculo del Yaw Rate de referencia en el modo de maniobrabilidad
yawR_ref_maniob=(1/(1-(m*(l1*Cd-12*Ct)*Vx^2)/(2*Cd*Ct*(l1+l2)^2)))*Vx*(-steer)/(l1+l2);
%Cálculo del Yaw Rate de referencia en el modo de estabilidad lateral
yawR_ref_slip=Kb*beta+(2*Fyd*cos(-steer)+2*Fyt)/(m*Vx);
end

```

Se observa que la distancia del eje trasero al centro de gravedad es un 41% superior al del eje delantero. Como se ha ido comentando a lo largo del trabajo, esto se debe a que el almacenaje de la energía eléctrica se produce en la parte posterior del vehículo, desplazando el centro de gravedad hacia la parte trasera del vehículo, y provocando su tendencia a sobrevirar.

A continuación, procedemos a explicar el bloque Inferior, que como ya se ha comentado, es el que más modificaciones va a tener respecto al análisis teórico. Recordamos que el objeto de este subsistema es, teniendo como entradas la referencia de yaw rate deseada, y el yaw rate del vehículo, calcular el momento correctivo que es necesario aplicar al coche, y luego traducir este momento a par.

Si vemos dentro de este bloque, obtenemos la figura 7-21, en la cual vemos las dos partes antes comentadas: el cálculo del momento correctivo y la traducción del mismo a par.

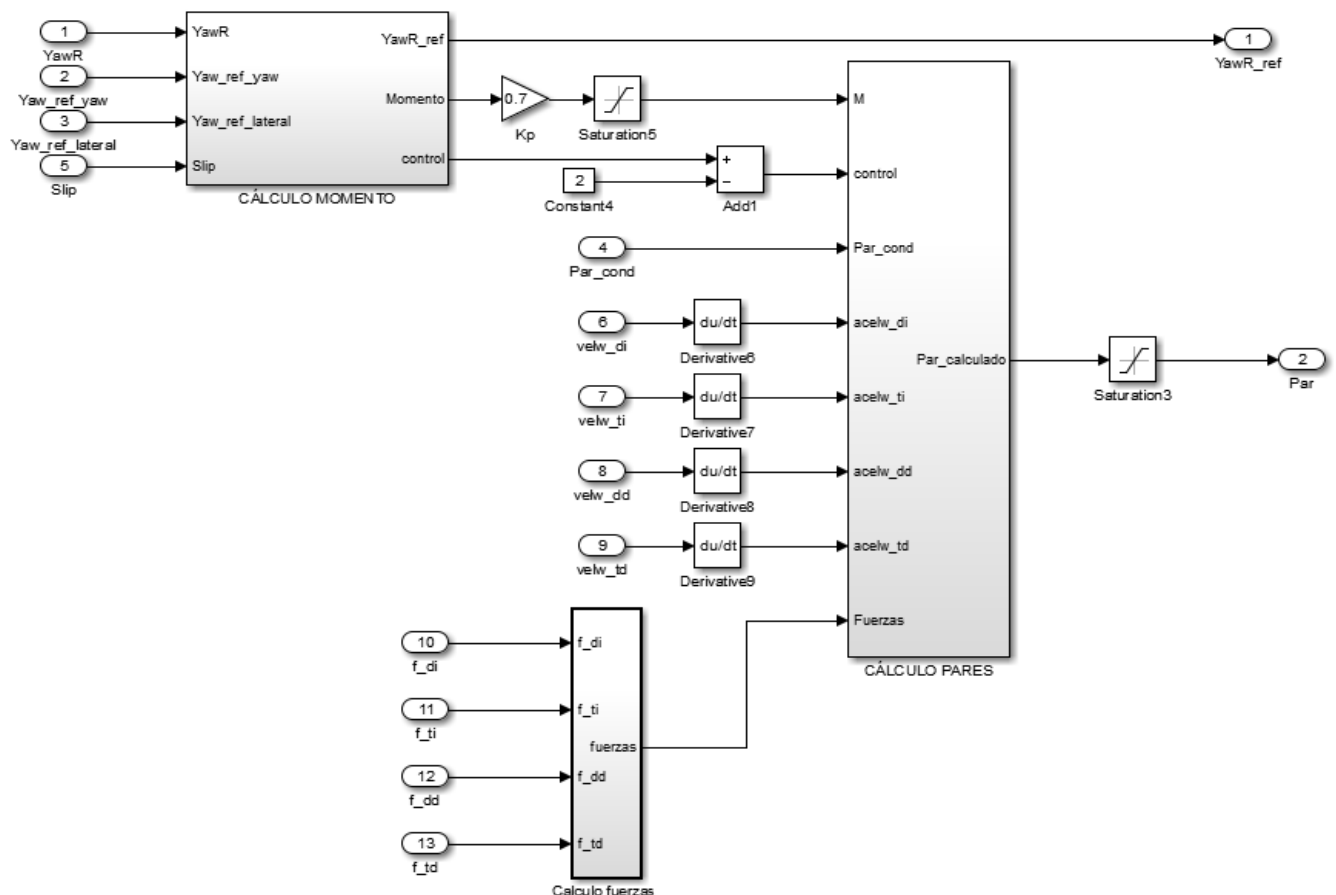


Figura 7-21 Subsistema Inferior

Primero vamos a explicar el bloque de Cálculo de Momento, que se observa en la figura 7-22.

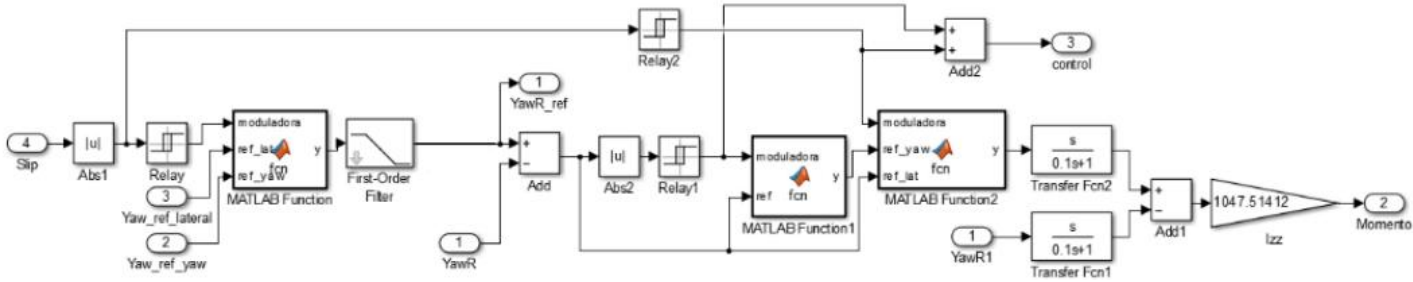


Figura 7-22 Bloque Cálculo Momento, dentro del subsistema Inferior

Su funcionamiento es el siguiente: tomamos el valor absoluto del ángulo de deslizamiento (para que funcione tanto con maniobras hacia la izquierda y hacia la derecha), y lo pasamos por un bloque *Relay*, que simula una histéresis: si no se ha llegado al umbral de activación de 3°, el valor de salida de este bloque es 1. Si se ha llegado, el valor de salida de este bloque es 2. Para evitar que, si el vehículo se queda oscilando en torno a este valor de ángulo de deslizamiento, y se pueda inestabilizar el vehículo ante muchos cambios de referencias casi instantáneos, ponemos que para que la salida cambie de 2 a 1, el ángulo haya descendido a 1°. A continuación, se muestra la figura 7-23, en la cual se puede apreciar la importancia de este bloque de histéresis:

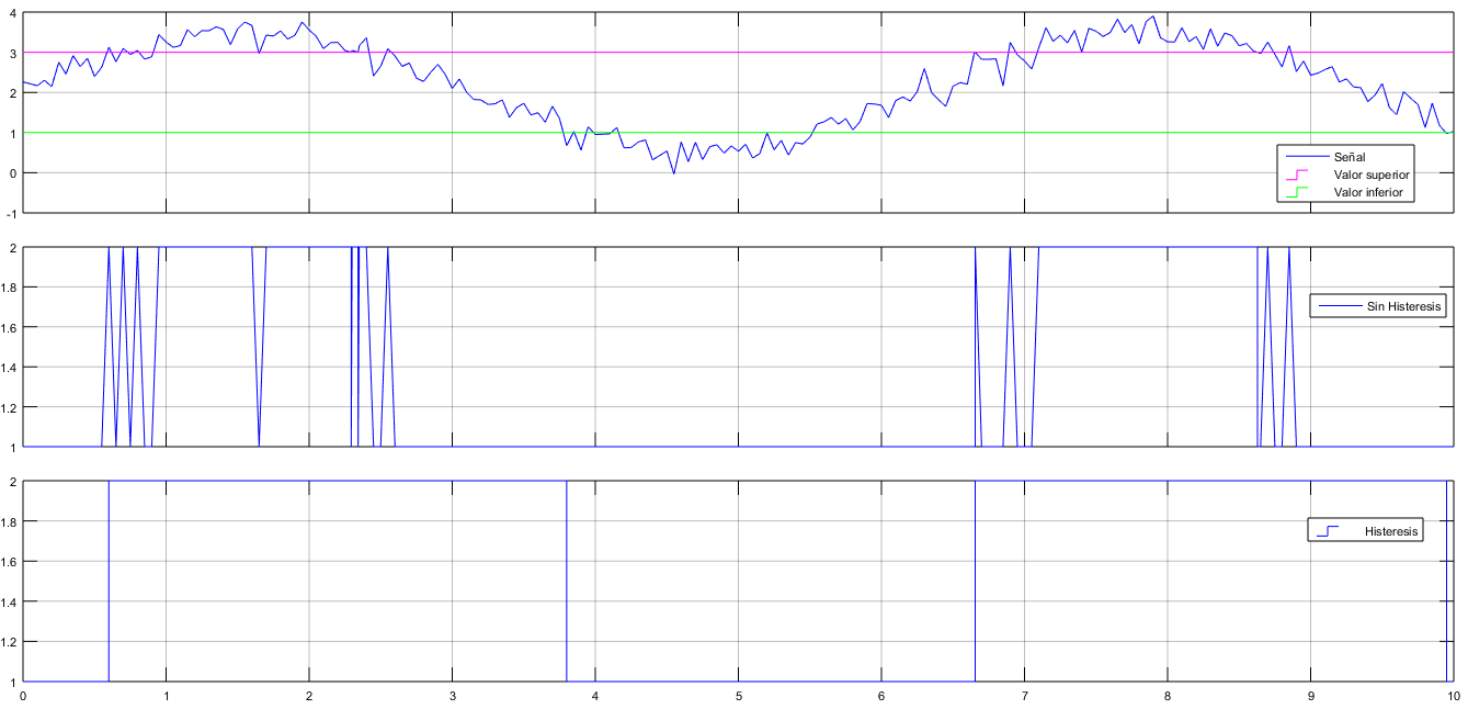


Figura 7-23 Importancia Histéresis

Para obtener la figura 7-23, se ha procedido a hacer la siguiente simulación: una señal con ruido que oscila en torno al valor 2, intentando emular una posible evolución del ángulo de deslizamiento del vehículo. En la primera gráfica se ve la variación de esta señal; en la segunda, se ve la salida que obtendríamos si pusiéramos lo indicado en teoría: cuando se superen los 3°, se activa el segundo modo de control, y cuando sea menor de 3°, que se active el primer modo de control; en la tercera gráfica se ve el resultado de poner la histéresis antes comentada: vale 1 hasta que no llegamos a 3°, cuando se activa vale 2, y para desactivarse tiene que llegar el ángulo a 1°, volviendo a valer 1.

Se observa claramente que cuando usamos el bloque *Relay*, obtenemos una señal que dice qué control está activo mucho más limpia que en el caso que se pone simplemente el bloque *Switch*. Esos cambios tan bruscos de la segunda gráfica pueden hacer que el sistema se haga inestable pues estaríamos cambiando casi instantáneamente de modo de control y ninguno sería capaz de corregir bien la dinámica del vehículo. Por último sobre esta

histéresis, se ha tomado como ángulo de desactivación del segundo modo de control 1° pues si poníamos 2°, se corre el riesgo de que la evolución del ángulo de deslizamiento suba rápidamente a 3°, y se repitan cambios muy seguidos. Como de todas formas es un sistema de seguridad, no se considera un problema que este segundo modo de control esté activo hasta que se asegure que el vehículo ha modificado correctamente su dinámica.

Volviendo a la figura 7-22, ya tenemos la salida del bloque *Relay*, 1 si tenemos activo el primer modo de control (maniobrabilidad) o 2 si tenemos activo el segundo modo (estabilidad lateral). Por tanto, ahora debemos escoger entre ambos modos: para ello se crea la siguiente función, en la que se escoge la referencia que vamos a usar en función de la salida de la histéresis:

```
function y = fcn(moduladora, ref_yaw, ref_lat)
%#codegen
y=0;%Inicialización
if(moduladora==1) %Activado modo maniobrabilidad
    y=ref_yaw;
elseif(moduladora==2) %Activado modo estabilidad lateral
    y=ref_lat;
end
```

A continuación, tenemos un bloque que filtra este yaw rate de referencia: se ha puesto para evitar que las transiciones entre un modo de control y otro sean bruscas, aunque la histéresis corrige esto también.

A continuación, calculamos el error de yaw rate como referencia menos valor real, y le hacemos el valor absoluto para entrar en otro bloque de histéresis. En este caso, se va a evaluar el error, y los límites van a ser 0.08 rad/s y 0. Cuando el error supere los 0.08 rad/s, la señal de salida valdrá 2, y cuando se desactive (error 0), valdrá 1. En este caso, se ha tomado como valor de desactivación 0, el cual podría parecer muy bajo, pero cuando se vean los resultados de las simulaciones se ve que hace que el controlador deje de actuar progresivamente cuando nos acercamos a la referencia, haciendo que el par modificado tienda al que pida el conductor, y evitando así cambios bruscos en los pares.

De la misma manera que antes, con la salida del relé, ahora tenemos que escoger entre la referencia antes calculada (la señal moduladora vale 2) y 0, que sería cuando el controlador no debe actuar (señal moduladora vale 1). Con el siguiente código se consigue este resultado:

```
function y = fcn(moduladora, ref)
y=0;
if(moduladora==1) %no hay error suficiente para activar el control, la
referencia va a ser 0
    y=0;
elseif(moduladora==2) %hay error suficiente, mantengo la señal de referencia
antes calculada
    y=ref;
end
```

Ahora ya queda la última posible transición entre un modo de control. Hasta el momento, primero hemos visto si el modo que debería activarse es el de estabilidad lateral (ESC- β) o el de maniobrabilidad (ESC- γ) analizando el ángulo de deslizamiento, y luego hemos tenido que decidir si se debe activar el control seleccionado o no en función del error de yaw rate. Sin embargo, esto no es suficiente, pues se puede dar el remoto caso en el que el error del ángulo de deslizamiento sea 5°, y el error en yaw rate sea 0, caso en el que debería saltar el modo de estabilidad lateral. Sin embargo, como lo último que se ha evaluado es el error en yaw rate, la referencia será 0. Para evitar esto, se vuelve a poner otra histéresis, en la cual se evalúa si el ángulo de deslizamiento supera los límites: caso en el que se activará el modo de control ESC- β , o si se activará el anteriormente calculado (ESC- γ , ESC- β o 0). Por tanto, siguiendo el mismo método hasta ahora usado, empleo el siguiente código para obtener, al fin, el yaw rate de referencia que se quiere, que vale 0 si no hay control activo:

```
function y = fcn(moduladora, ref_yaw, ref_lat)
y=0;%Inicialización
if(moduladora==1) %No se sabe qué control está activado, depende de los
cálculos anteriores
    y=ref_yaw;
elseif(moduladora==2) %Activado modo estabilidad lateral
    y=ref_lat;
```

end

Con esto, ya obtenemos al fin el yaw rate que va al controlador proporcional propuesto en la ecuación (6.6) para calcular el momento necesario.

A continuación, se nos presenta un nuevo problema: hay que calcular las derivadas del yaw rate real y del yaw rate de referencia para calcular el momento necesario. Sin embargo, hemos visto que las señales llevan ruido, lo cual hace que calcular una derivada sea complicado y ofrezca unos resultados pobres. En vez de usar el bloque de *Simulink* de derivadas, vamos a usar una función de transferencia para ello, como se ve en la figura 7-22. Esta función de transferencia es:

$$G_{derivada}(s) = \frac{s}{\tau \cdot s + 1} \quad (7.1)$$

Como puede verse, si hacemos la constante de tiempo pequeña, es como si aplicáramos el operador s para derivar, el cual no se puede usar directamente pues sería un sistema no causal. Por tanto, tenemos que escoger una τ de manera que podamos derivar la señal, sin perder información por el camino. Se escogerá el valor de $\tau=0.1s$ pues la dinámica del vehículo (más concretamente el yaw rate) tiene un orden de magnitud de segundos, y al escoger una constante de tiempo diez veces o más inferior a la del sistema, evitamos perder información al filtrarlo, pues al fin y al cabo actúa como filtro. En el primer controlador presentado, no teníamos este problema pues con la lógica difusa nos evitábamos el tener que derivar estas señales.

En la figura 7-24 puede apreciarse el efecto de derivar directamente o de aplicar la función de transferencia propuesta a una señal del tipo de las peores de las que podemos obtener en este trabajo.

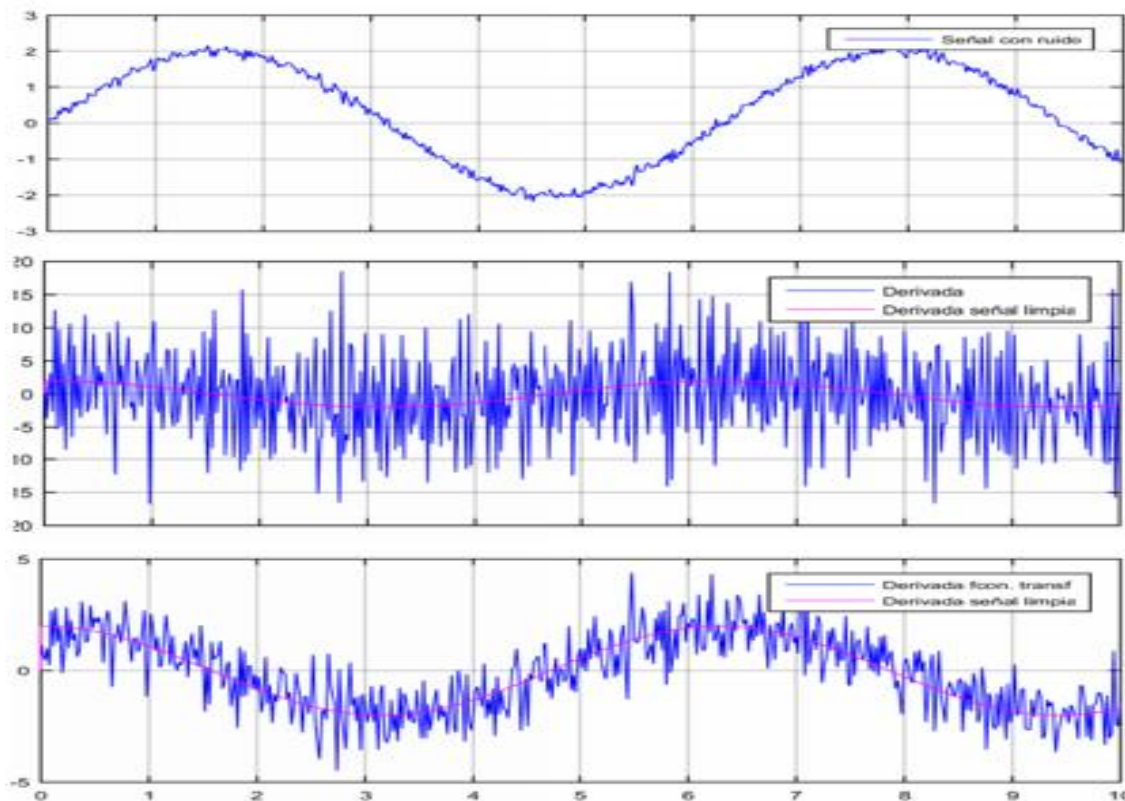


Figura 7-24 Derivadas de una señal con ruido

En la primera gráfica se muestra la señal con ruido que podemos obtener al mover senoidalmente el volante; en la segunda gráfica se muestra en azul la salida de derivar directamente esta señal, y en rosa el valor que debería salir realmente; por último, en la tercera gráfica se muestra el resultado de derivar usando esta función de transferencia. Se observa que cuando se deriva directamente, sale un resultado muy malo, llegando a obtener errores de hasta 18 unidades, pero al hacerlo mediante el segundo método, obtenemos un error inferior, en torno a 2 unidades. Por tanto, aunque no es un método perfecto, es una solución con un error aceptable. Hay que tener en cuenta que luego estas señales que son derivadas se restan, de manera que si las tenemos con poco ruido, gran

parte del mismo se anula, pero si el ruido es muy grande (gráfica del medio), podemos obtener resultados catastróficos, pues luego esta resta se multiplica por el momento de inercia del vehículo, lo cual hará que se amplifique mucho este ruido. Sin embargo, en el modelo se aprecia que se calculan otras derivadas para obtener las aceleraciones angulares de las ruedas, y se hace directamente con los bloques *Derivative* de *Simulink*, pues estas señales no presentan tanto ruido en el simulador.

Cuando ya se tiene el momento necesario calculado (aplicando la ecuación (6.6)), se decide saturarlo, pues a veces se pide un momento muy grande, lo cual resulta en que los motores saturen. Esto se evita con la saturación del momento máximo a 480 Nm, que es un valor elevado para este vehículo, pero que en un momento dado podría necesitarse.

Por último, volvemos a la figura 7-21, en la que se mostraba el contenido del subsistema de control inferior, y abrimos el bloque de Cálculo de Pares (figura 7-25). En él, tenemos dos códigos, el que pasa de momento del vehículo a par de cada rueda, haciendo que cada eje genere el 50% de lo que hay que generar, y luego el MTTE.

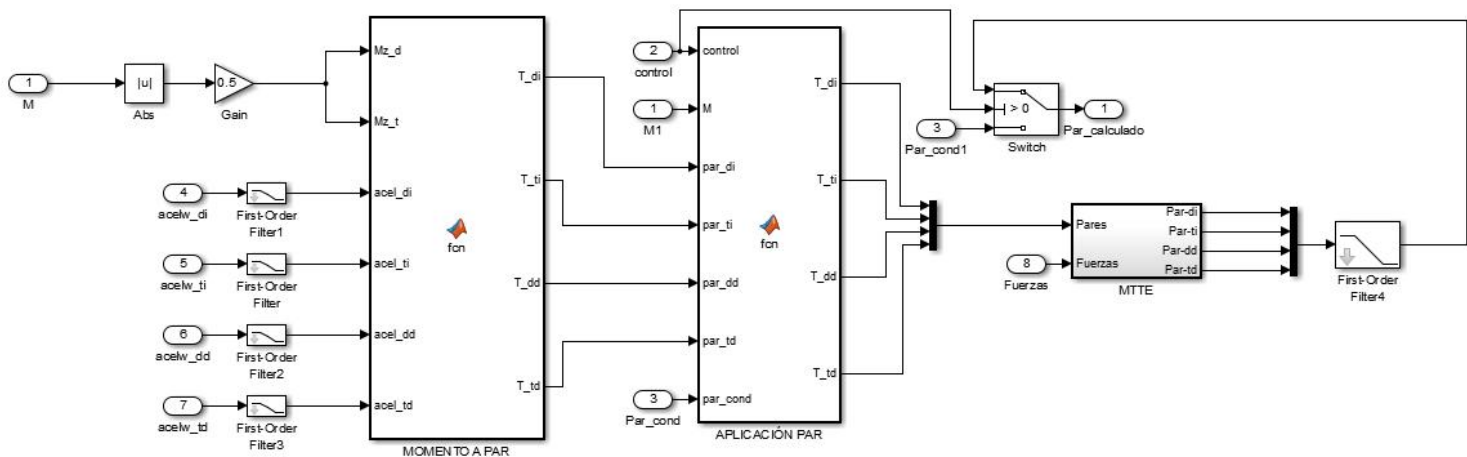


Figura 7-25 Interior bloque Cálculo Pares, del subsistema Inferior

El bloque momento a par es exactamente el mismo que en el controlador de lógica difusa, de manera que no lo explicamos de nuevo, recordamos que lo que se hacía era un balance en los ejes del vehículo para calcular el par que debe darse a cada rueda.

A continuación, vemos el código del bloque Aplicación Par, que consiste en que si el control se ha activado (la variable *control* vale 1), se aplica el par diferencial siguiendo el siguiente código que se basa en la explicación teórica dada anteriormente, y si el control está desactivado (vale cero), no se le suma nada al par del conductor. Para obtener esta variable *control*, lo que se ha hecho es adaptar la suma de dos de las salidas de los relés de la figura 7-22, concretamente la del primer relé (en la que se escoge si se activa el modo de control de maniobrabilidad o el de estabilidad lateral) y la del segundo (en el que se escoge si el control está activo o no). Por tanto, analizamos todos los casos que puede valer dicha suma, y nos damos cuenta de que cuando el control está desactivado, la suma vale 2, y si está activado, vale o bien 3 o bien 4. Por tanto, si restamos 2 a esta suma, obtenemos una variable que vale 0 si no hay control activo, y mayor que cero si está activo. El código que implementa esto es:

```
function [T_di, T_tj, T_dd, T_td] = fnc(control,M,par_di, par_tj, par_dd,
par_td,par_cond)
```

```
T_di=par_cond;T_tj=par_cond; %por defecto, reparto equitativo par
T_dd=par_cond;T_td=par_cond; %por defecto, reparto equitativo par
if(control>0) %si el control esta activado, modifico el par
    if(M>0)
        T_di=par_cond-par_di; T_tj=par_cond-par_tj;
        T_dd=par_cond+par_dd; T_td=par_cond+par_td;
    elseif(M<0)
        T_di=par_cond+par_di; T_tj=par_cond+par_tj;
```

```

        T_dd=par_cond-par_dd; T_td=par_cond-par_td;
    end
end
end

```

A continuación, el bloque MTTE es el mismo que el que se empleó en el primer controlador de lógica difusa, de manera que no volvemos a repetir su implementación, simplemente consiste en introducir las ecuaciones pertinentes. Para que este bloque funcione, antes se vio otro bloque que no estaba explicado, el de Cálculo de fuerzas, que vuelve a ser el mismo que en el anterior controlador (se muestra en la figura 7-21).

Por último, se ha puesto un switch que hace que cambiemos entre el par calculado y el que pide el conductor en función de si el control está activo o no.

Con esto termina la implementación de este segundo controlador.

7.4 Códigos auxiliares

A lo largo de este capítulo se están viendo que en muchos bloques aparecen unos parámetros que son variables del workspace de Matlab. Se ha tomado la decisión de usar estas variables porque así es más fácil modificarlas para sintonizarlas sin tener que buscarlas en el modelo de Simulink, pues están todas en el mismo lugar. Además, hay algunas variables que el modelo del vehículo necesita para poder funcionar, como los coeficientes del modelo de Burckhardt de la rueda. Todos se incluyen en un fichero .m que se tiene que ejecutar antes de cualquier simulación. No se adjunta este código pues son simplemente parámetros que luego se repetirán en el capítulo 8, Implementación del controlador en QNX.

También se ha creado un código que almacena todas las variables de interés en un fichero, de manera que luego sea fácil la comparación entre diferentes simulaciones.

Un ejemplo de este código es:

```

save DataDEFCIRC acely dss pos_x pos_y v_sslip v_yaw volante_vis vvel Pos_IMU
par par_real slip_ang_ruedas slip_ruedas mom_oví yaw_ref

```

En DataDEFCIRC se guardan todos los valores de la simulación, de manera que modificando el nombre, podemos guardar varias simulaciones y luego se pueden comparar. Para compararlas, se ha creado otro código, que se presentará en el siguiente capítulo.

8 RESULTADOS EN SIMULACIÓN

En este capítulo se van a comparar los resultados de diferentes maniobras sin el controlador y con los controladores activos, de manera que podemos ver en qué y cuánto mejora el comportamiento del vehículo con el control diseñado. En el primer controlador se va a permitir que el vehículo tenga par negativo en momentos puntuales pues el par que pide el conductor en las simulaciones a veces lo es, lo que no se permite es que una rueda tenga casi siempre un par negativo. Sin embargo, en el segundo se va a permitir el par negativo en las ruedas dado que es algo que algunos vehículos de la misma tipología están empezando a admitir, pues en esos momentos el par negativo hace que los motores actúen como generadores, con lo cual podría obtenerse energía siempre y cuando las baterías no estén llenas.

8.1 Presentación de resultados

Para ver la mejora de los parámetros del vehículo con el controlador activo, se van a realizar dos simulaciones por cada experimento. La primera, sin el controlador, y la segunda con él activado, de manera que es fácil ver las mejoras. Por tanto, cuando se hable de mejoría, se entiende que es lo que mejora el vehículo con el controlador activo en comparación con el vehículo sin el controlador.

El procedimiento a seguir es el siguiente: primero se presentarán unas gráficas de las variables más importantes en cada simulación y, a continuación, se presentarán unos datos numéricos para saber de qué mejoras porcentuales estamos hablando. Estos datos serán:

- **Par entregado:** Suma de todos los pares de todas las ruedas durante la simulación.
- **Ratio par/distancia:** Cociente entre la distancia recorrida y el par total entregado.
- **Mejora porcentual de la aceleración transversal:** Mejora máxima instantánea de la aceleración transversal al vehículo.
- **Mejora porcentual del ángulo de deslizamiento del coche:** Mejora máxima instantánea del ángulo de deslizamiento del coche.
- **Mejora porcentual del ángulo de deslizamiento de cada rueda:** Mejora máxima instantánea del ángulo de deslizamiento de cada rueda, se toma el mayor de las cuatro ruedas.
- **Mejora rég. perm. porcentual del ángulo de deslizamiento de cada rueda:** Mejora en el régimen permanente del ángulo de deslizamiento de cada rueda, se toma el mayor de las cuatro ruedas.
- **Mejora porcentual de deslizamiento de cada rueda:** Mejora máxima en el deslizamiento porcentual de las ruedas, se toma el mayor de las cuatro ruedas.
- **Mejora porcentual de pérdida de velocidad del coche:** Mejora máxima de la pérdida de velocidad del vehículo, pues no es capaz de mantener la velocidad de referencia.
- **Error régimen permanente de velocidad del coche:** Error en el régimen permanente de la velocidad del vehículo. Es un dato puesto para aclarar la mejora porcentual, pues puede aportar poca información.

8.2 Giro brusco hacia la izquierda

8.2.1 Descripción maniobra

Esta maniobra consiste en describir un escalón de 60° en el sentido antihorario en el volante, de manera que el vehículo, que partía de describir una trayectoria recta, pase a describir una circunferencia hacia la izquierda. Se tomará como velocidad inicial del vehículo 60 km/h (16.67 m/s), pues se entiende que es una situación que podría darse en la realidad, y durante la duración del experimento, que es una vuelta completa, se intentará mantener dicha velocidad.

En la figura 7-1 se muestra la entrada del volante empleada, donde se aplica un filtro al principio para suavizar esta maniobra, pues si se introducen picos tan grandes se generan problemas en el simulador a la hora de calcular la derivada de esta señal.

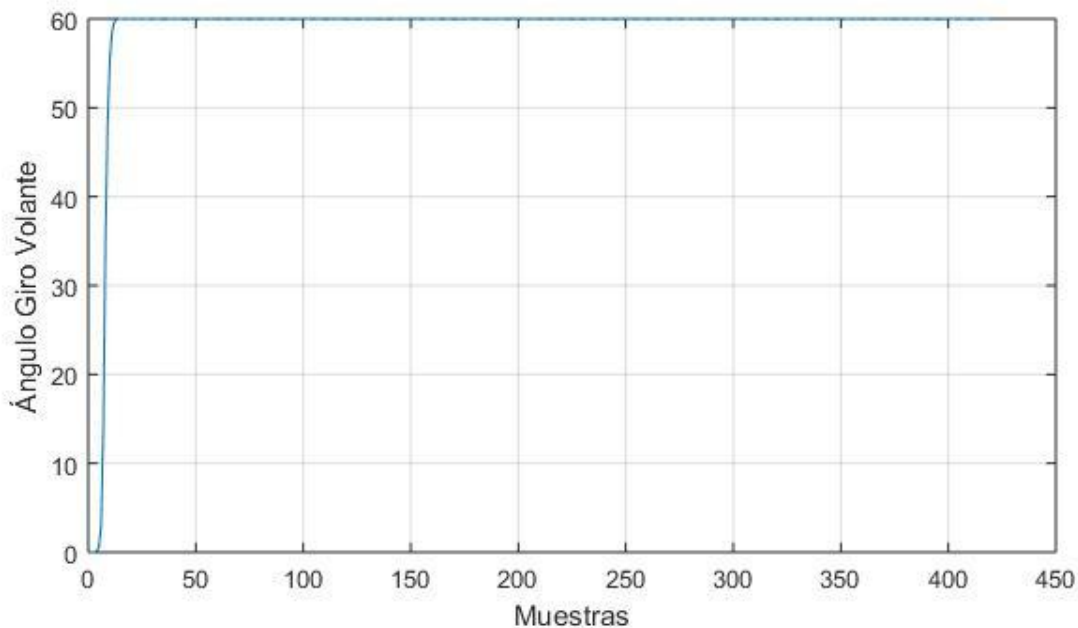


Figura 8-1 Entrada del volante primera simulación

El objeto de esta maniobra es poner al vehículo en una situación relativamente extrema, pues el giro es muy brusco. Para aumentar la brusquedad de la simulación, se podría aumentar la velocidad del vehículo, poniéndolo a 120 km/h, pero a velocidades tan altas los resultados de simulación del FOX difieren respecto a la realidad, por eso se toma esta velocidad, pues está en el rango de velocidades en el que funciona bien.

8.2.2 Resultados Controlador lógica difusa

En el eje x se representa el tiempo, pero no medido en segundos, sino en muestras, de manera que no se le pone tal título al eje para no confundir:

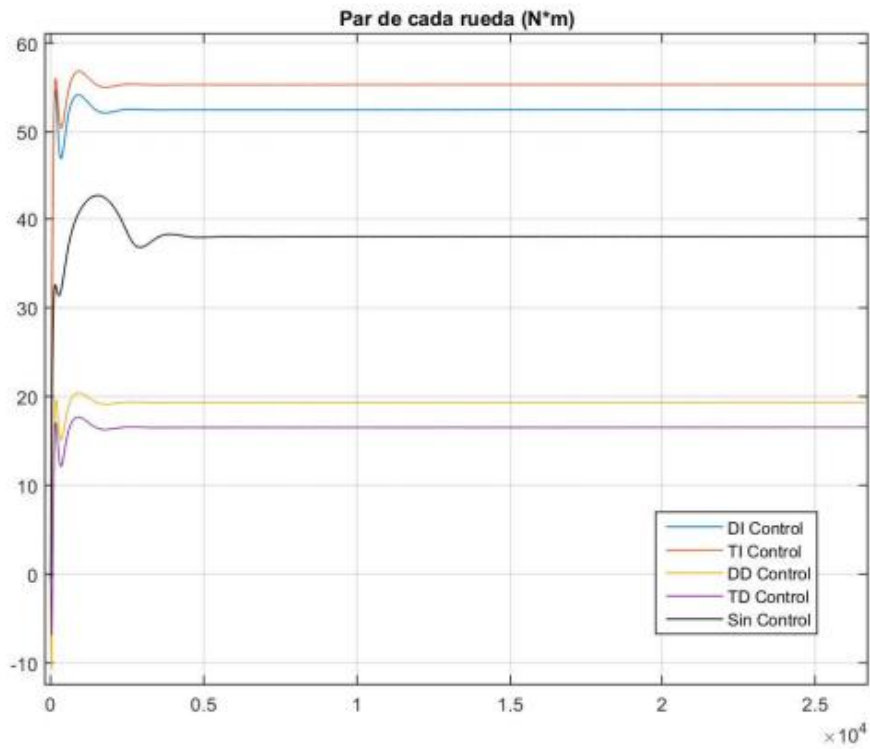


Figura 8-2 Par de cada rueda Exp.1 Control 1

Por último, se observa que la distribución de par varía mucho respecto al caso original, haciendo posible que algunos parámetros varíen, como se mostrará a continuación.

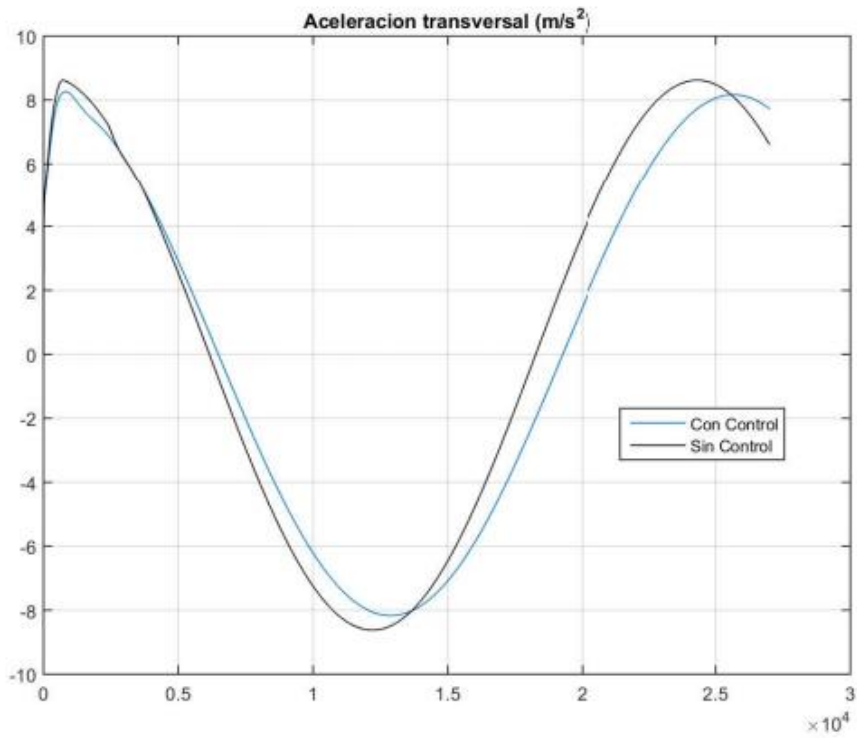


Figura 8-3 Aceleración transversal Exp.1 Control 1

En cuanto a la aceleración transversal, se observa que se reduce, lo cual es bueno pues hace que el vehículo sea más controlable al estar sufriendo menores fuerzas. Hay un tramo en el que la aceleración es mayor, pero en el cómputo global, es menor.

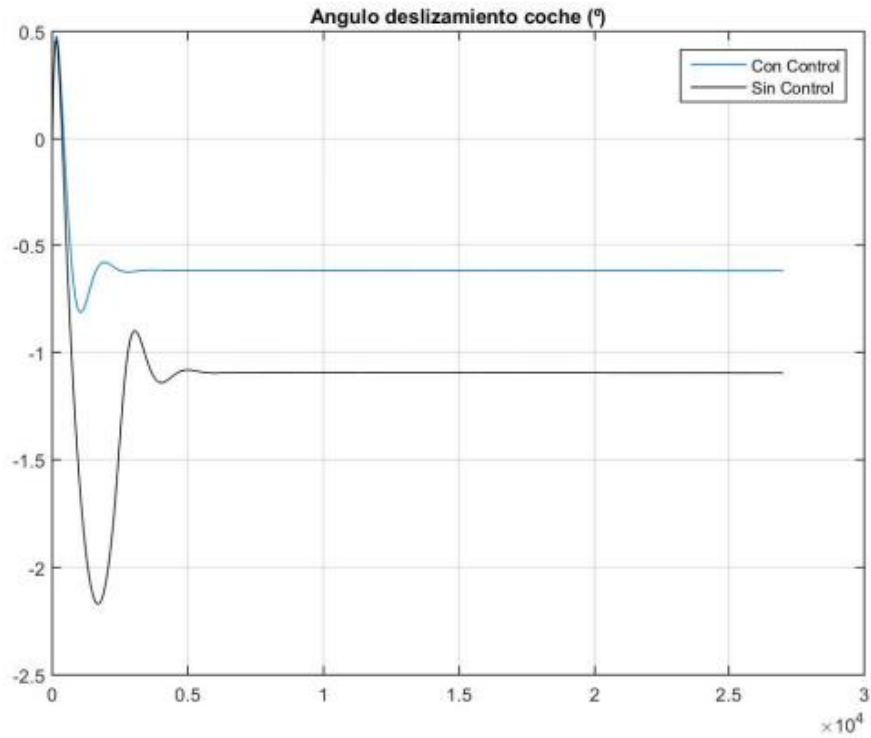


Figura 8-4 Ángulo de deslizamiento del coche Exp.1 Control 1

El ángulo de deslizamiento del vehículo se reduce mucho, lo cual es muy positivo, pues si el coche desliza menos, se puede controlar mejor manualmente. Esta es la principal mejora del controlador diseñado.

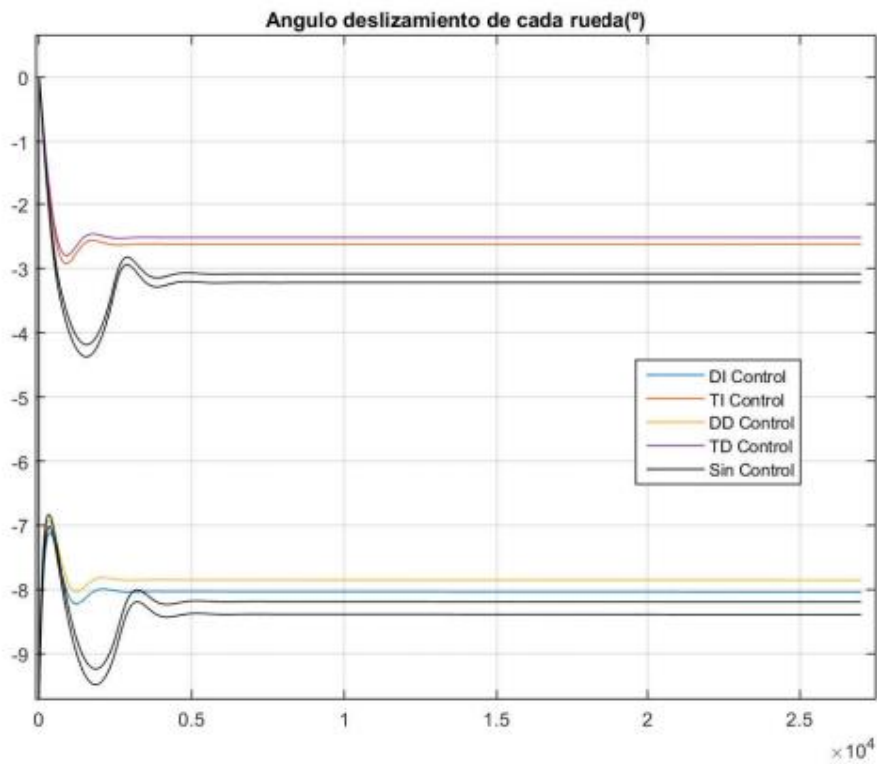


Figura 8-5 Ángulo de deslizamiento de las ruedas Exp.1 Control 1

Se observa que el ángulo de deslizamiento de cada rueda se reduce bastante, lo cual también hace que el vehículo sea más fácil de controlar.

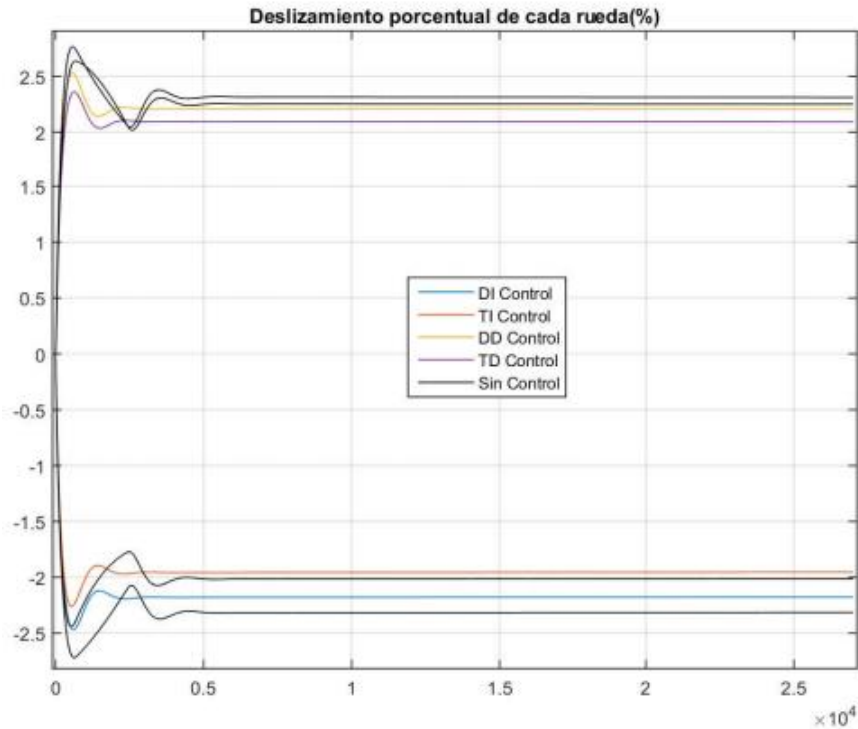


Figura 8-6 Deslizamiento porcentual ruedas Exp.1 Control 1

El deslizamiento porcentual de las ruedas mejora algo, pero tampoco demasiado. Es una mejora secundaria pues el conductor no se dará cuenta de tal efecto.

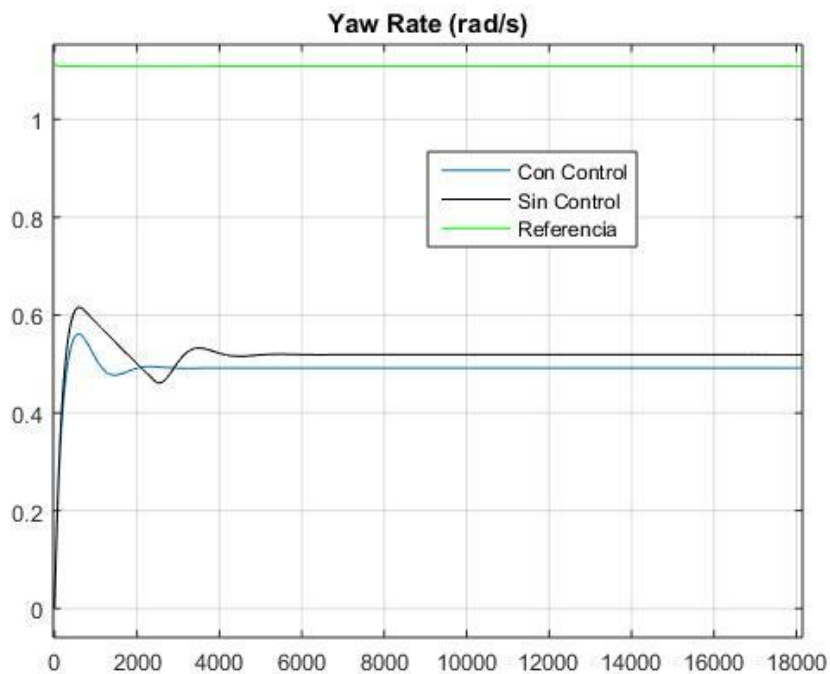


Figura 8-7 Yaw rate Exp.1 Control 1

Se observa que el yaw rate del vehículo se estabiliza antes, lo cual hace que la trayectoria descrita sea más suave. Esto provoca que el vehículo gire menos, pero viendo que estamos hablando de una diferencia muy pequeña, y que el ángulo de deslizamiento mejora bastante, es aceptable. Cuando se presentó el vehículo y el sistema de control, se dijo que el problema principal era que el coche (teóricamente) giraba demasiado por la distribución de peso, y en la figura 8-7 se observa que se reduce la velocidad angular, por lo cual se consigue mejorar el

comportamiento. Sin embargo, no es capaz de llegar a la referencia marcada, pues se puede apreciar que es muy alta. Más adelante, en la siguiente simulación, se verá que esta referencia toma valores más realizables cuando el vehículo describe trayectorias menos extremas (con giros menos bruscos). Por otra parte, se observa que el vehículo tiene un comportamiento más alejado de la referencia que sin el control, esto se debe a que se decidió que en la acción de control tenga más peso la mejora del ángulo de deslizamiento que el yaw rate, pues el primero es difícilmente mejorable por el conductor, mientras que para cambiar el yaw rate simplemente debe girar más o menos el volante. La mejoría del ángulo de deslizamiento del vehículo recordamos que se puede observar claramente en la figura 8-4.

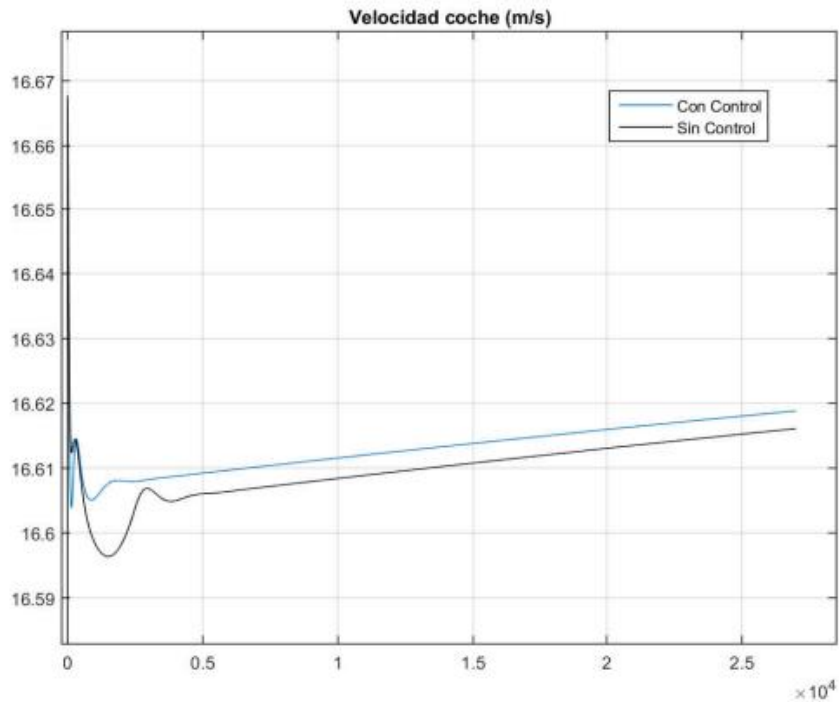


Figura 8-8 Velocidad del vehículo Exp.1 Control 1

La velocidad del vehículo es superior en la mayor parte de la conducción a la que se tiene sin el controlador, de manera que es algo que mejora, aunque no mejore una cantidad muy significativa.

Por último, se muestra en la figura 8-9 la trayectoria que describe el vehículo con y sin el controlador.

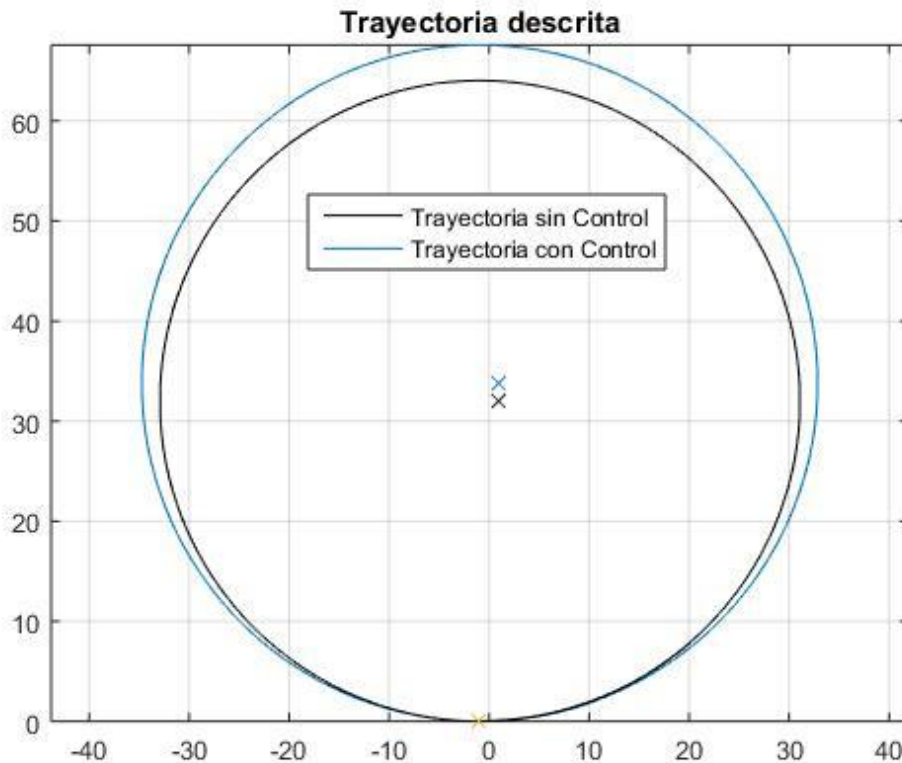


Figura 8-9 Trayectorias descriptas Control 1

Se puede apreciar que el radio de la trayectoria descrita por el vehículo controlado ha aumentado en 2 metros. Teniendo en cuenta que el radio de giro está en torno a los 30 metros, concluimos que este cambio de trayectoria no es un cambio que afecte negativamente al conductor.

A continuación se muestran algunos datos numéricos obtenidos de esta comparación:

Par total entregado sin control:	4108202.46 Nm
Par total entregado con control:	3867701.53Nm
Ratio par/distancia sin control:	19079.94 Nm/m
Ratio par/distancia con control:	18221.73Nm/m
Mejora porcentual de aceleracion transversal:	5.253 %
Mejora porcentual de ángulo de deslizamiento del coche:	62.467 %
Mejora máxima porcentual de ángulo de deslizamiento de cada rueda:	33.132 %
Mejora rég. perm. porcentual de ángulo de deslizamiento de cada rueda:	18.364 %
Mejora porcentual de deslizamiento de cada rueda:	9.219 %
Mejora porcentual de pérdida de velocidad del coche:	0.045 %
Error régimen permanente de velocidad del coche Sin Control:	0.051 m/s
Error régimen permanente de velocidad del coche Con Control:	0.048 m/s

Se observa que se entrega menos par para recorrer una distancia similar, lo cual hace que este controlador reduzca el consumo energético del vehículo, algo que ya se ha comentado en numerosas ocasiones que es muy

positivo. Esto es debido a que se aprovecha mejor la energía entregada, pues al deslizar hasta un 62 % menos, es necesario menor par para poder realizar la trayectoria que queremos.

Por otra parte, se observa que el control del coche mejora pues todos los índices relacionados con el deslizamiento mejoran. Esto es muy importante pues puede hacer que en maniobras más extremas no se pierda tanto el control del vehículo.

8.2.3 Resultados controlador proporcional

En primer lugar, mostramos la comparación de par con este controlador y sin él:

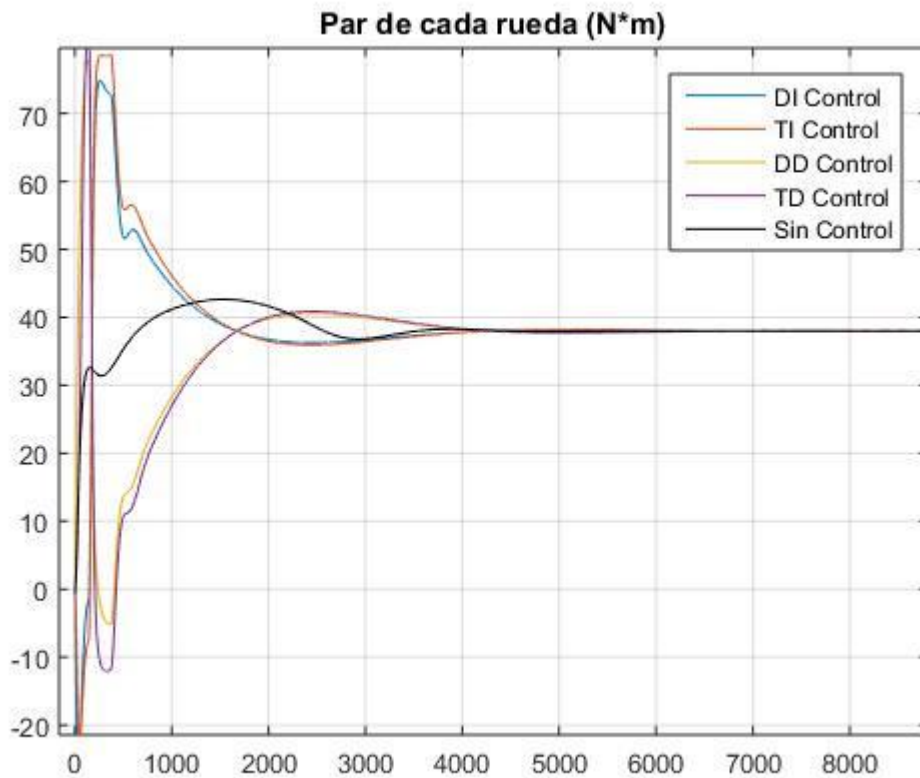


Figura 8-10 Par de cada rueda Exp.1 Control 2

Se observa que este controlador modifica el par del vehículo sobre todo al principio de empezar el giro, cuando se da el volantazo, mientras que en el régimen permanente del giro no tiene efecto. Por tanto, se esperan mejoras en el transitorio, no en el permanente. Si comparamos con el primer controlador (Figura 8-2), se observa claramente que el primero actúa durante toda la trayectoria, mientras que este no lo hace, pues no está siempre activo.

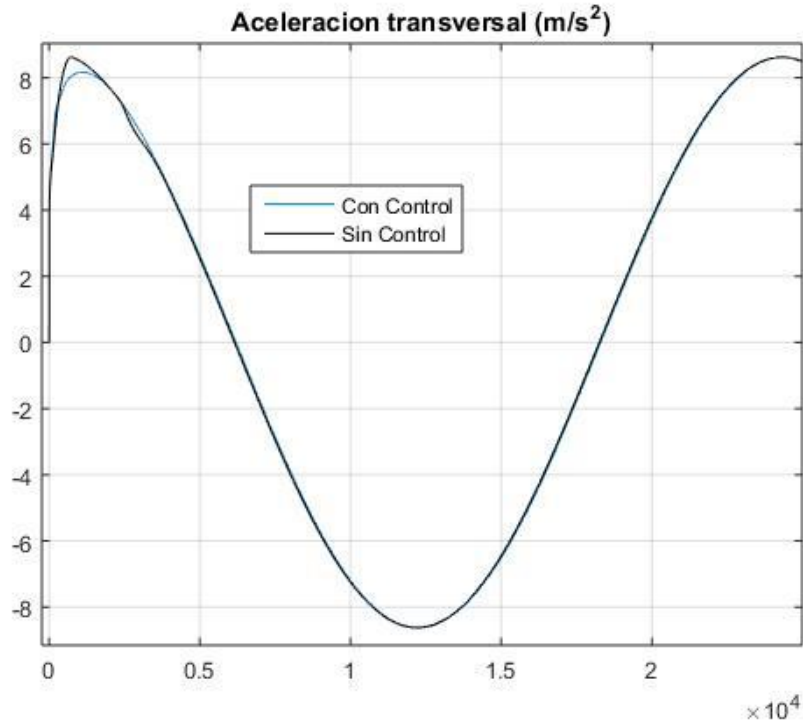


Figura 8-11 Aceleración transversal Exp. 1 Control 2

En la figura 8-11 se ha mostrado la evolución de la aceleración transversal del vehículo. En consonancia con lo explicado en la gráfica que mostraba el par, se observa una mejora de la aceleración transversal al principio de la curva. Esto es muy beneficioso para el conductor pues se traduce directamente en una mejora de la maniobrabilidad, por lo que, si el conductor fuese una persona real, tendría mejor control durante este primer tramo y, por consiguiente, podría modificar la trayectoria para mejorar aún más los resultados.

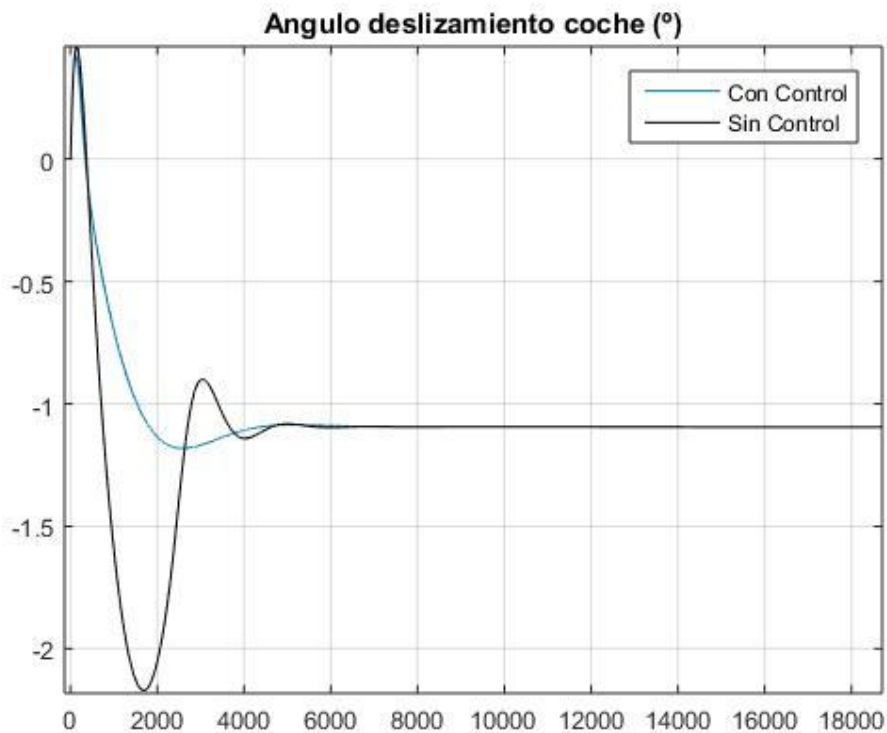


Figura 8-12 Ángulo de deslizamiento del coche Exp. 1 Control 2

Observando la figura 8-12, se ve claramente cómo mejora el ángulo de deslizamiento del vehículo durante el

transitorio, reduciendo las sobreoscilaciones y su valor máximo, lo cual es muy beneficioso. Si se compara con la figura 8-10, se observa claramente que la mejora se produce cuando la distribución de par en las ruedas no es equitativa, sino que el controlador la modifica.

A continuación se muestra el ángulo de deslizamiento de cada rueda, variable relacionada con esta anterior:

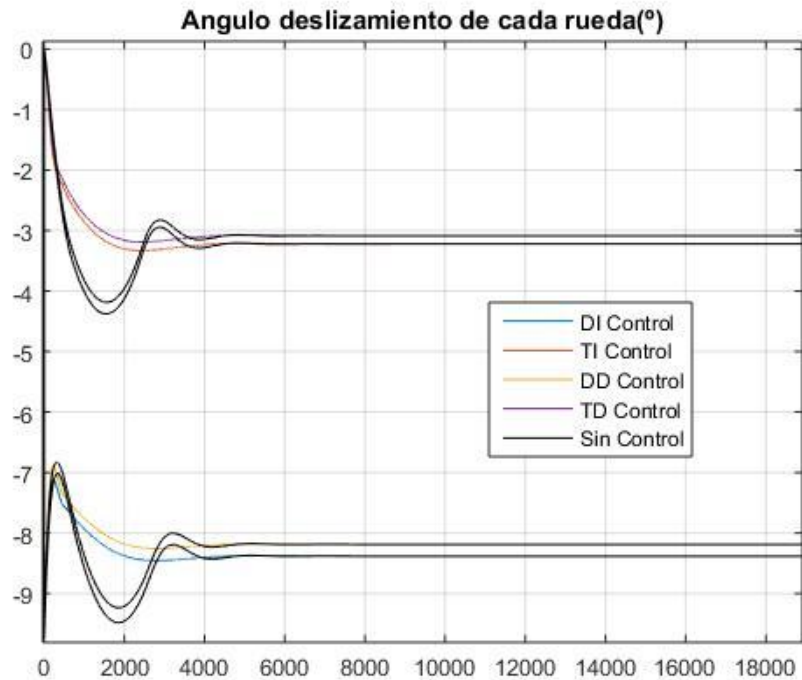


Figura 8-13 Ángulo de deslizamiento de las ruedas Exp. 1 Control 2

Siguiendo la misma tónica de resultados obtenidos con este controlador, volvemos a apreciar mejoras en el transitorio del experimento, reduciendo la sobreoscilación y, por tanto, mejorando la maniobrabilidad del conductor.

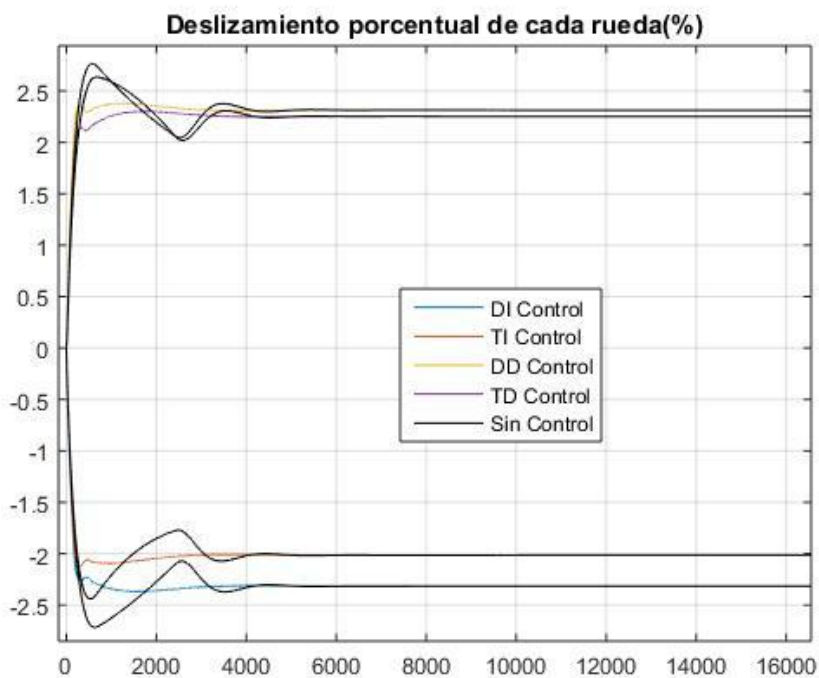


Figura 8-14 Deslizamiento porcentual ruedas Exp.1 Control 2

En la figura 8-14 podemos ver la evolución del deslizamiento porcentual de cada rueda. De un primer vistazo, se observa lo mismo que se ha ido comentando hasta ahora: mejoras en el régimen transitorio, y desactivación del control en el régimen permanente.

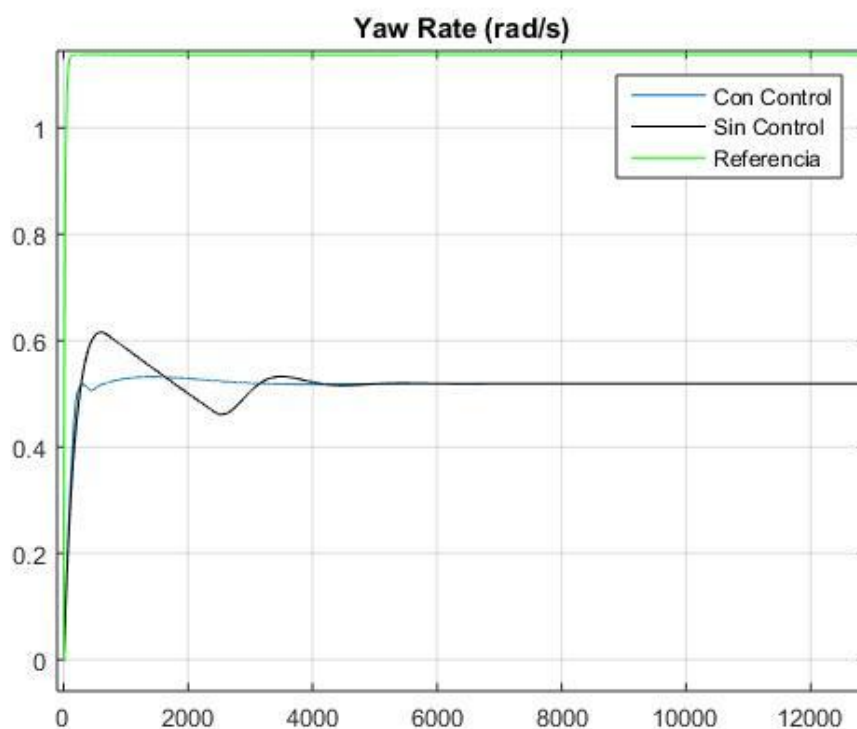


Figura 8-15 Yaw Rate Exp.1 Control 2

Se puede apreciar en la figura 8-15 una clara mejoría del yaw rate en términos de su derivada: la curva del yaw rate es mucho más suave, por lo que se mejora también la maniobrabilidad, pues el vehículo empieza a describir la circunferencia con una velocidad angular más constante que sin el control. Si observamos la referencia, vemos que, a pesar de haber sido calculada de manera diferente, es similar a la empleada con el primer controlador.

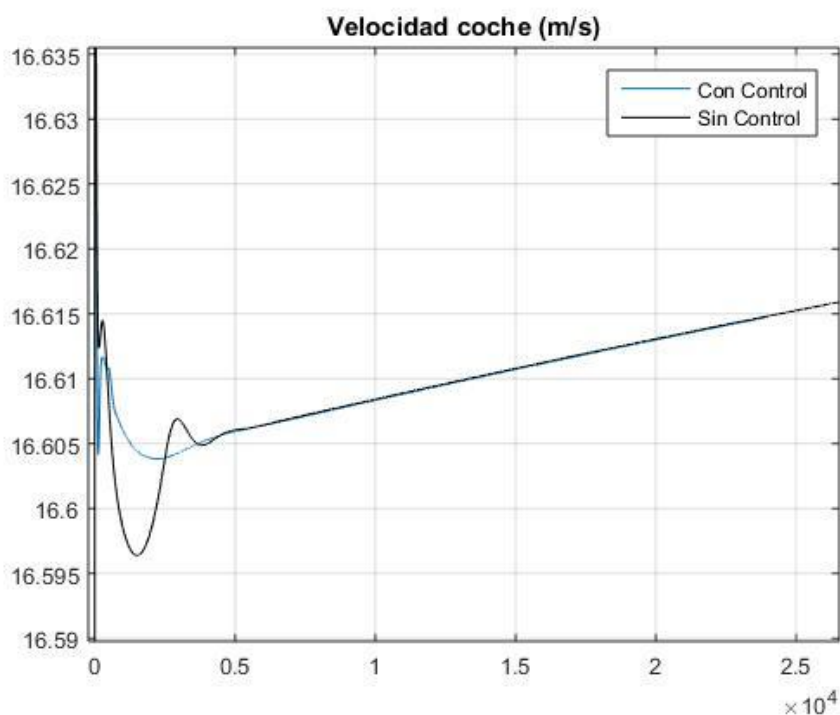


Figura 8-16 Velocidad del vehículo Exp. 1 Control 2

De la misma manera que ocurría en el primer controlador, la velocidad no mejora mucho, por lo que no se considera una gran mejora. Sin embargo, se muestra el resultado para ver que tampoco empeora: se mantiene similar a la obtenida durante la simulación sin el controlador activado.

Por último, mostramos la trayectoria descrita por el vehículo con este controlador:

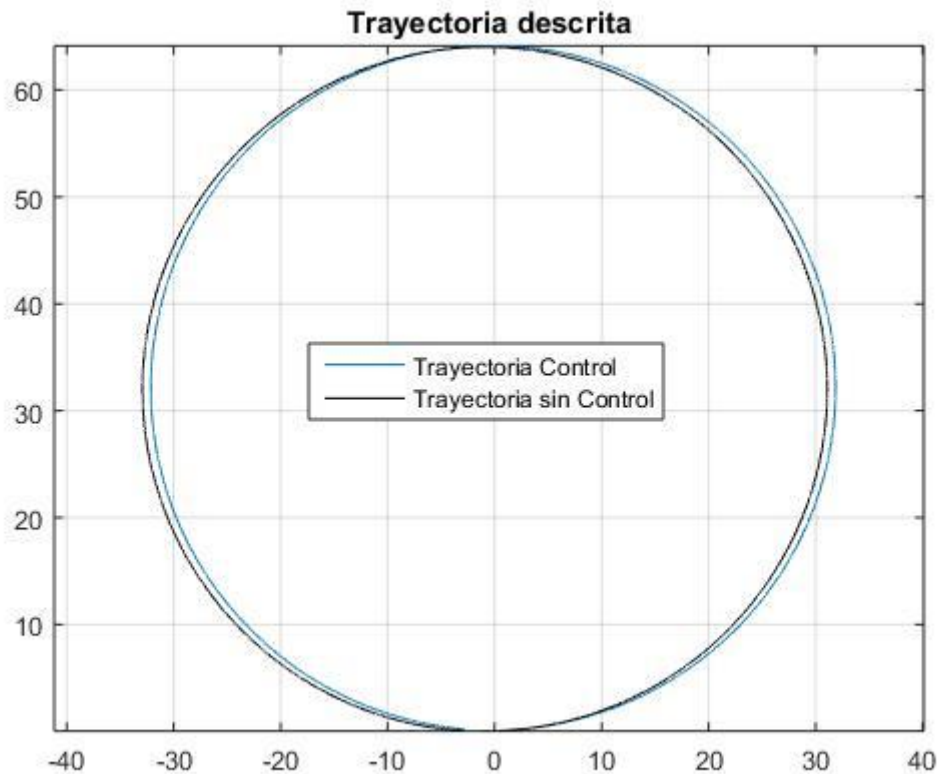


Figura 8-17 Trayectorias descritas control 2

Como se observa, la trayectoria descrita es prácticamente idéntica, existe una variación máxima de 90 centímetros entre ambas, por lo cual se puede concluir que el conductor tiene el control prácticamente total del vehículo mientras el controlador consigue mejorar su dinámica. Sin embargo, aunque el conductor tiene prácticamente el control, se aprecia que las trayectorias descritas son diferentes. El motivo de ello es que al principio (punto [0,0]) es donde el controlador actúa más, haciendo que se empiece a describir una circunferencia de radio mayor (compensando así el subviraje), sin embargo, luego deja de actuar, por lo que se describe una trayectoria con el mismo radio que sin controlador, pero desplazada hacia la derecha por la actuación del control al comienzo.

A continuación se muestran datos concretos sobre estas mejoras obtenidas:

Par total entregado sin control:	3652148.63 Nm
Par total entregado con control:	3627121.50 Nm
Ratio par/distancia sin control:	18165.10 Nm/m
Ratio par/distancia con control:	18032.72Nm/m
Mejora porcentual de aceleracion transversal:	10.589 %
Mejora porcentual de ángulo de deslizamiento del coche:	61.617 %
Mejora máxima porcentual de ángulo de deslizamiento de cada rueda:	30.232 %
Mejora rég. perm. porcentual de ángulo de deslizamiento de cada rueda:	0.000 %

Mejora porcentual de deslizamiento de cada rueda:	23.077 %
Mejora porcentual de pérdida de velocidad del coche:	0.090 %
Error régimen permanente de velocidad del coche Sin Control:	0.051 m/s
Error régimen permanente de velocidad del coche Con Control:	0.051 m/s

Se observa que se reduce el par que se entrega al vehículo para realizar una trayectoria muy similar, lo cual es bueno de cara a una gestión eficiente de la energía del vehículo. Por otro lado, se ve que las mejoras se dan principalmente en el transitorio de esta maniobra, y en el permanente no tiene efecto este controlador, pues así se diseñó, y es un buen resultado que las simulaciones así lo indiquen.

Por último, cabe destacar que el par entregado con este controlador es superior al par entregado por el otro controlador, lo cual es debido a que esta simulación es más larga. De todas formas, se puede ver que el ratio par/distancia con control es aproximadamente similar en ambos.

8.3 Doble cambio de carril a 60 km/h

8.3.1 Descripción maniobra

La maniobra empleada en esta simulación es un doble cambio de carril, en el cual el conductor parte del carril de la izquierda, cambia al carril de la derecha y, una vez allí, vuelve al carril del que partió.

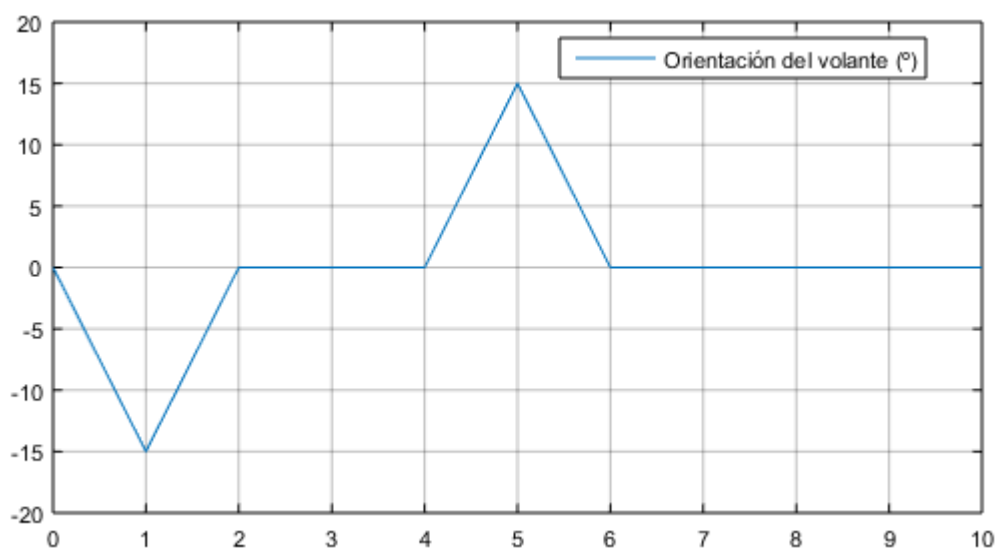


Figura 8-18 Giro del volante en la maniobra de cambio de carril

El objetivo de esta maniobra es ver cómo actúan los controladores en una situación en la cual va a existir poco deslizamiento y el vehículo va a responder bastante bien.

Esta maniobra se va a realizar a 60 km/h para ambos controladores.

8.3.2 Resultados controlador lógica difusa

En primer lugar, se va a mostrar el par de cada rueda, y siguiendo la misma tónica de anteriores gráficas, en negro se mostrará el par que pedía el conductor en la simulación sin el controlador, y en un color diferente, el par que se aplica a cada rueda con el controlador activo.

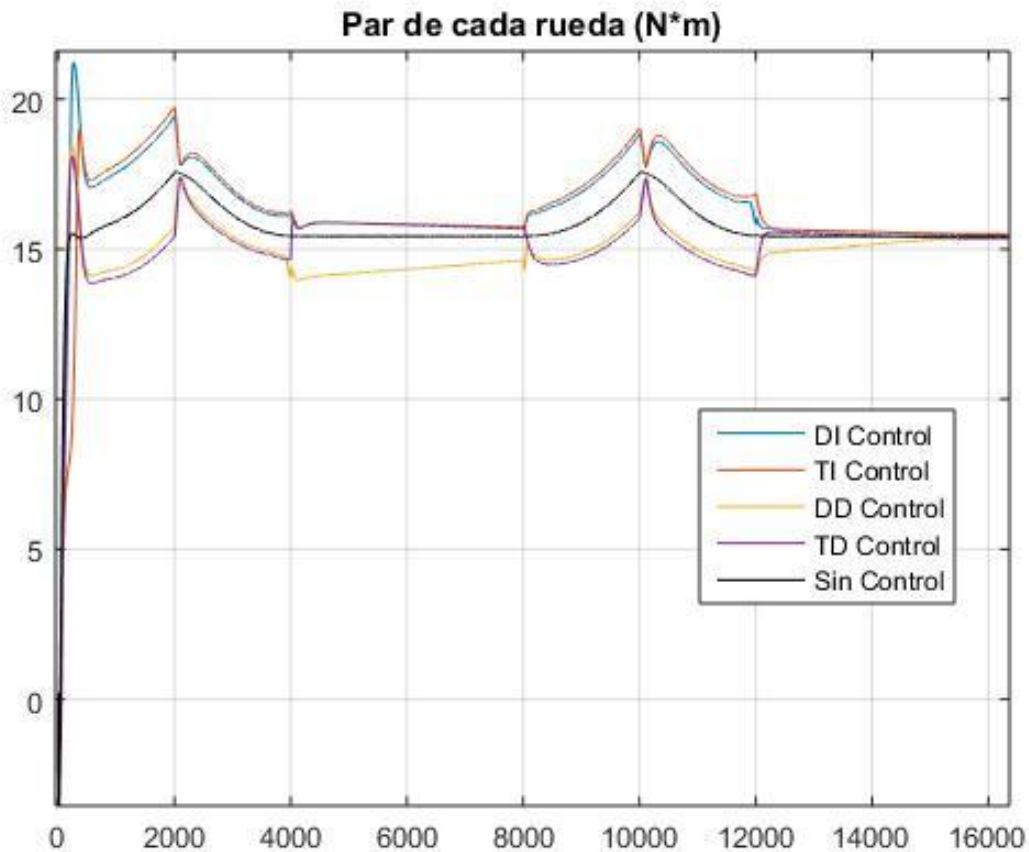


Figura 8-19 Par de cada rueda Exp.2 Control 1

Observando la figura sobre estas líneas, se puede apreciar claramente los cambios en la trayectoria. En la primera parte, se observa que se está realizando un cambio de carril, hasta que la evolución del par de cada rueda se vuelve más horizontal, y luego ocurre algo similar con el siguiente cambio de carril. En el tramo en el que el vehículo va por el carril de la derecha, se observa que el par no es exactamente el que pide el conductor, sino que está levemente modificado en cada rueda, esto se debe a los errores en yaw rate y en ángulo de deslizamiento respecto a las referencias, que luego se mostrarán.

Por otra parte, puede verse que se vaya al carril de la izquierda o al de la derecha, se da siempre más par a las ruedas izquierdas, lo cual también viene influido por los errores antes comentados. Además, el par que se añade y se sustrae a cada rueda es relativamente pequeño, por lo que no vamos a esperar grandes mejorías en los parámetros analizados.

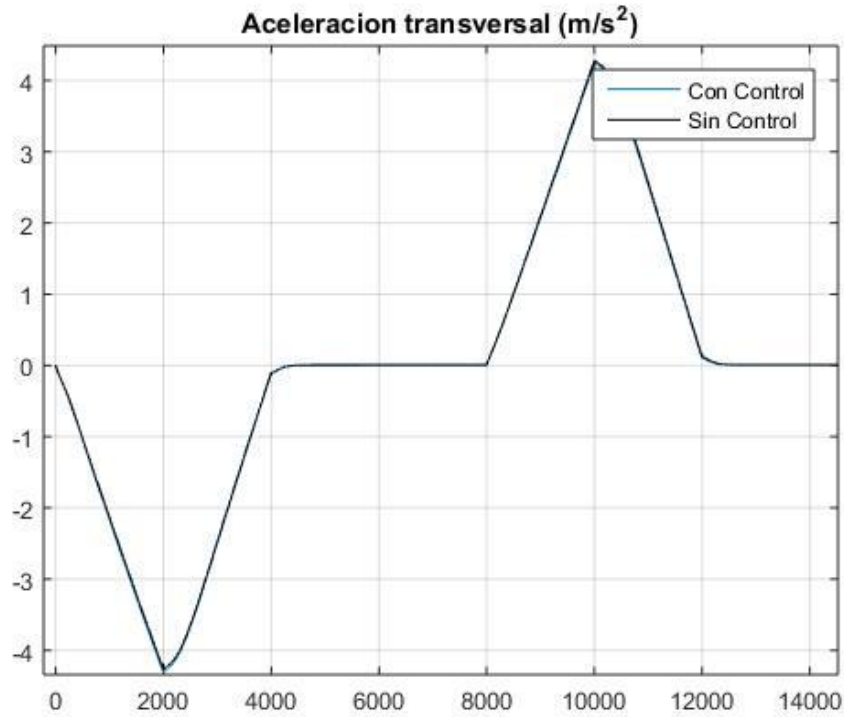


Figura 8-20 Aceleración transversal Exp.2 Control 1

Como se comentó en la figura del par, no se aprecia una gran mejoría en cuanto a la aceleración de cada rueda, pues recordamos que estamos haciendo una maniobra suave, por lo que el controlador actúa acorde a la maniobra.

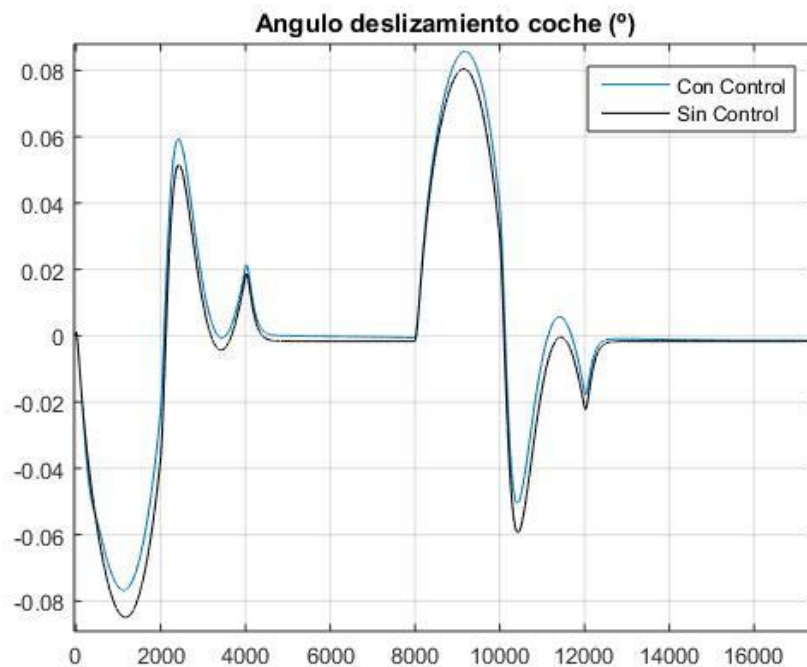


Figura 8-21 Ángulo deslizamiento coche Exp.2 Control 1

En cuanto al ángulo de deslizamiento, se observa que hay zonas en la que mejora y otras en el cual empeora un poco, aunque por lo general mejora.

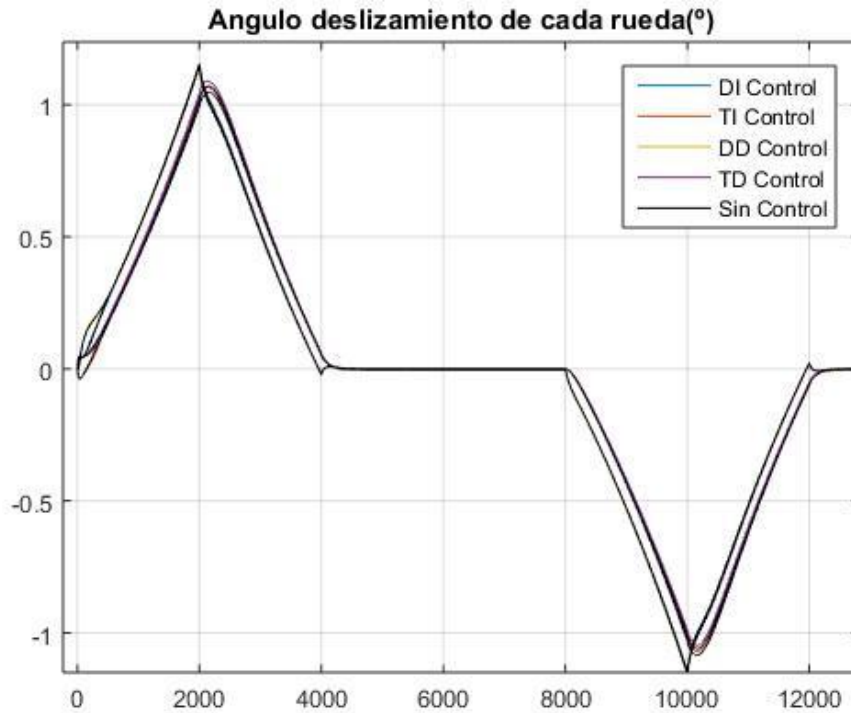


Figura 8-22 Ángulo de deslizamiento de las ruedas Exp.2 Control 1

En la figura 8-22 se muestran los ángulos de deslizamiento de cada rueda. No obtenemos mejoras muy significativas en este experimento, pues hay poco margen de mejora y el controlador actúa poco, lo cual es bueno pues permite que el usuario tenga el control total prácticamente del vehículo en trayectorias suaves.

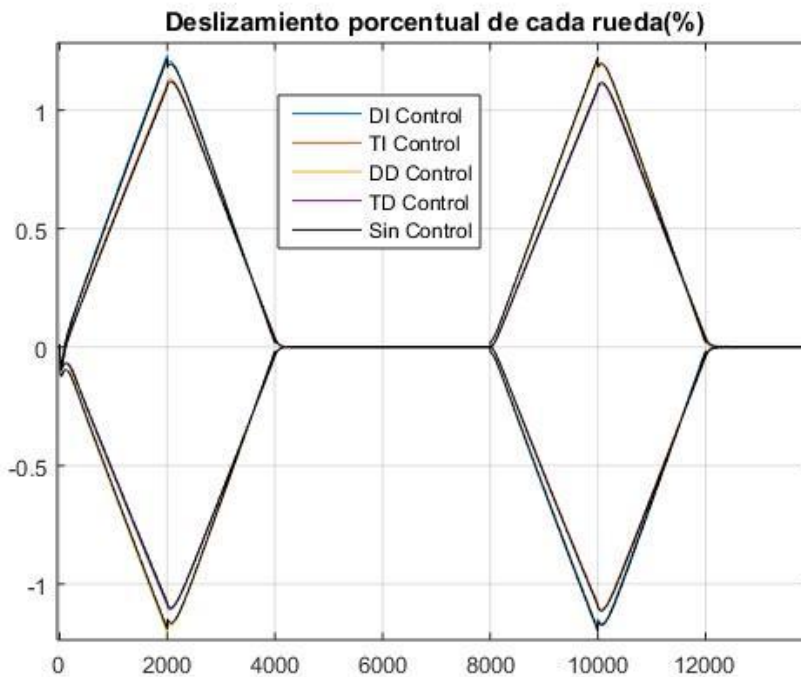


Figura 8-23 Deslizamiento porcentual ruedas Exp.2 Control 1

Al igual que en el experimento anterior, el ángulo de deslizamiento de cada rueda disminuye poco, aunque probablemente no mejore lo suficiente como para que el conductor se percate, pues estamos hablando de magnitudes muy pequeñas.

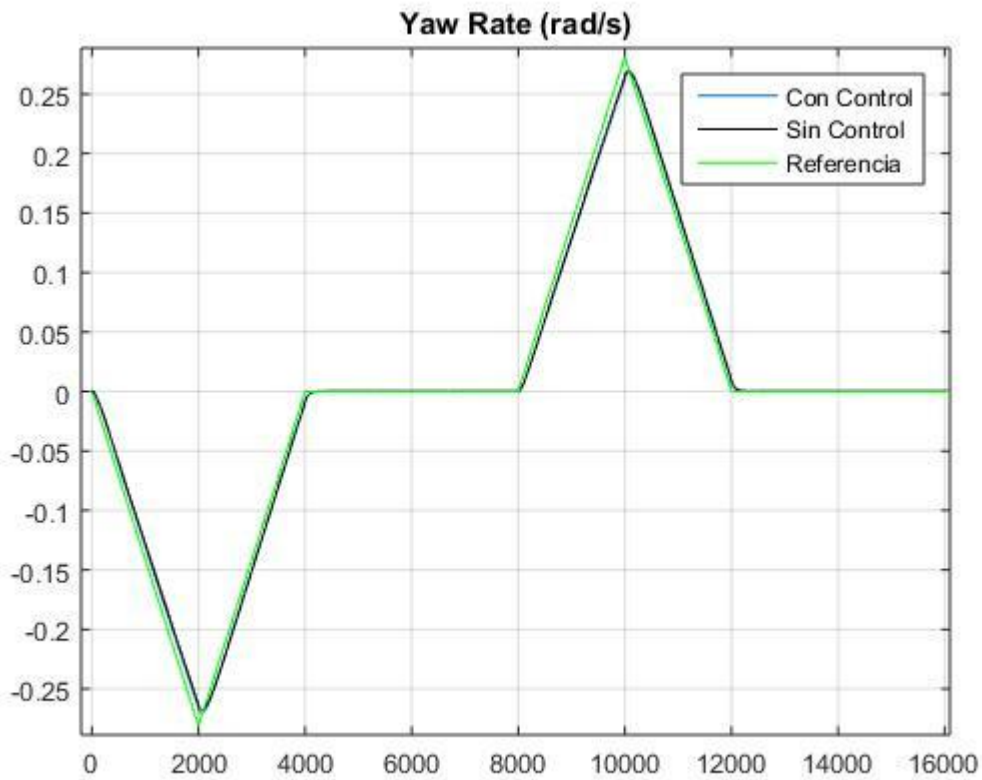


Figura 8-24 Yaw Rate Exp.2 Control 1

En la figura 8-24 se muestra que el vehículo, como se ha podido ir viendo, tiene un comportamiento bastante bueno sin control, pues la referencia se acerca mucho a la curva del yaw rate. Si hacemos zoom, obtenemos la figura 8-25:

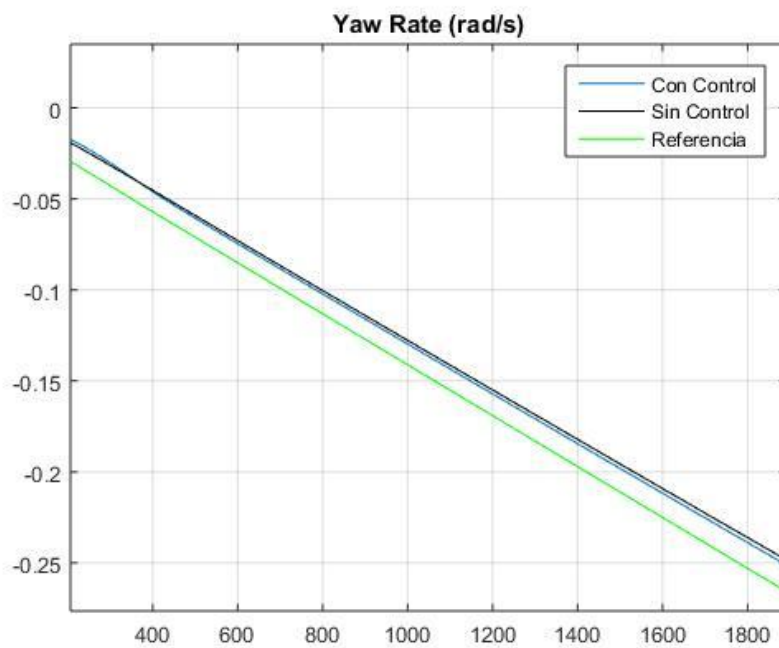


Figura 8-25 Yaw Rate aumentado Exp.2 Control 1

En esta figura se aprecia que el controlador hace que el vehículo se acerque más a la referencia que el caso sin control, lo cual es bueno, pero al ser un margen tan pequeño, la mejora es también pequeña.

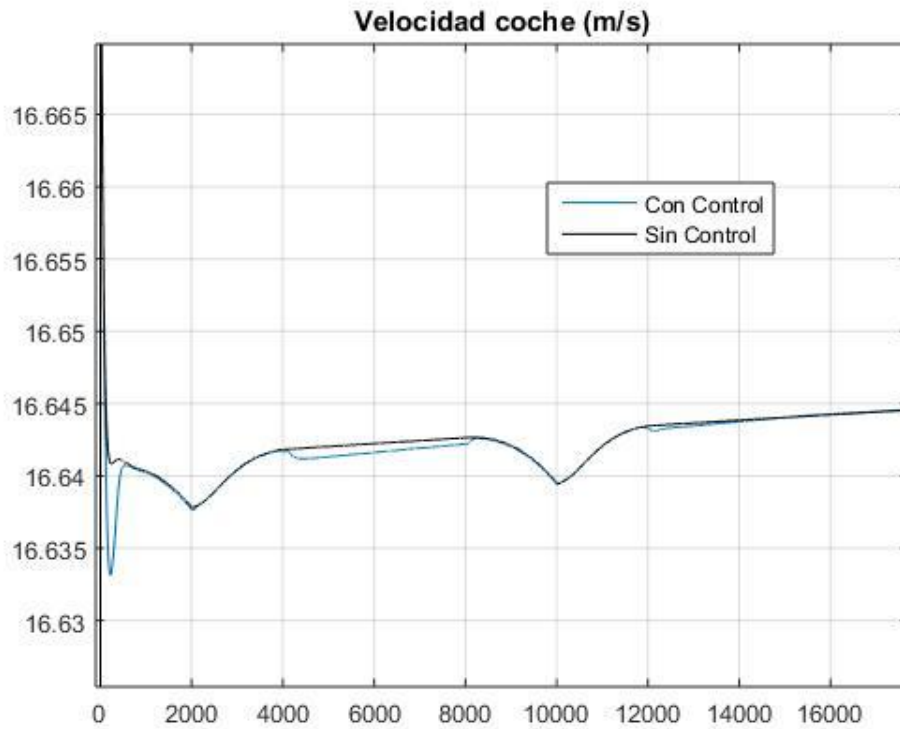


Figura 8-26 Velocidad del vehículo Exp.2 Control 1

Por último, en la figura 8-26, se observa la velocidad del vehículo, la cual es inferior en un 0.03% como máximo en el recorrido, por lo cual no es considerado como mal funcionamiento.

A continuación se presentan algunos resultados numéricos:

Par total entregado sin control:	1253655.34 Nm
Par total entregado con control:	1253846.46 Nm
Mejora porcentual de aceleracion transversal:	1.206 %
Mejora porcentual de ángulo de deslizamiento del coche:	9.704 %
Mejora máxima porcentual de ángulo de deslizamiento de cada rueda:	1.824 %
Mejora rég. perm. porcentual de ángulo de deslizamiento de cada rueda:	6.001 %
Mejora porcentual de deslizamiento de cada rueda:	0.937 %
Mejora porcentual de pérdida de velocidad del coche:	-0.03 %
Error régimen permanente de velocidad del coche Sin Control:	0.022 m/s
Error régimen permanente de velocidad del coche Con Control:	0.022 m/s

Estos datos informan de que el efecto del controlador es relativamente bajo en este experimento, el cual tenía como objeto ver qué ocurría si se efectuaba una maniobra suave a una velocidad no muy alta. Por tanto, podemos concluir que este controlador cumple con los objetivos: mejora el comportamiento del vehículo en casos más extremos, y casi no actúa en casos de conducción normal.

8.3.3 Resultados controlador proporcional

A continuación, se van a mostrar los resultados de esta misma maniobra, pero con el segundo controlador propuesto. Se espera que ocurra algo similar a lo que ocurrió con el anterior controlador, que actúe muy poco.

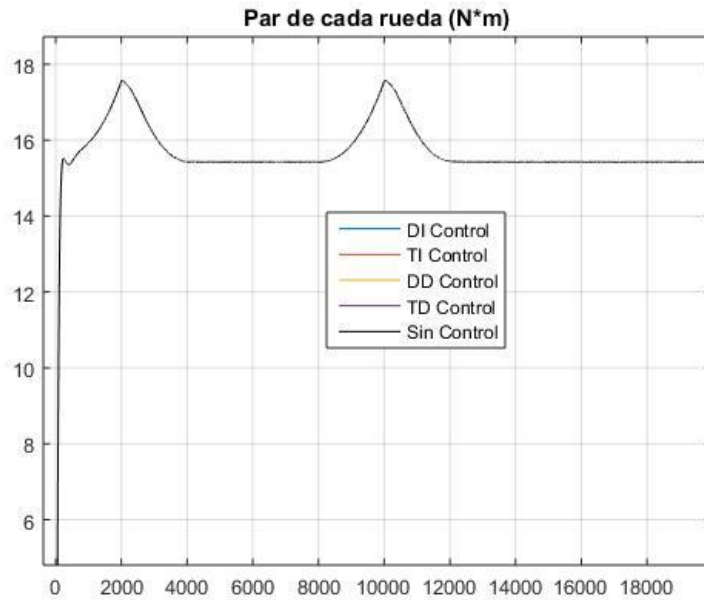


Figura 8-27 Par de cada rueda Exp.2 Control 2

En la figura 8-27 observamos que el reparto del par a cada rueda es exactamente igual al caso en el que no hay control. Esto es debido a que los errores de yaw rate y ángulo de deslizamiento son inferiores a los umbrales para los que salta el control. Por tanto, se considera un éxito esta situación, pues el controlador no actúa al no percibir que haya nada que arreglar. En la siguiente figura se mostrará estas dos variables para demostrar que, efectivamente, el error es inferior al umbral de actuación (cabe recordar que la referencia del ángulo de deslizamiento era 0°):

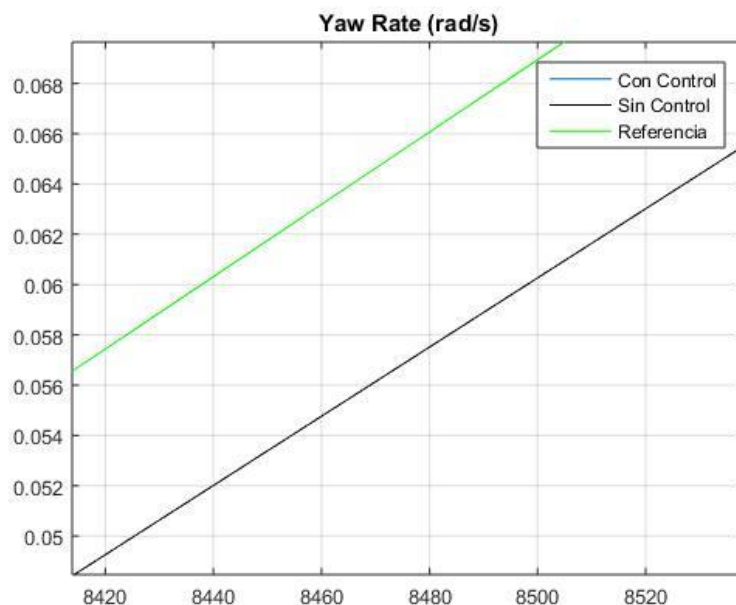


Figura 8-28 Yaw Rate Exp.2 Control 2

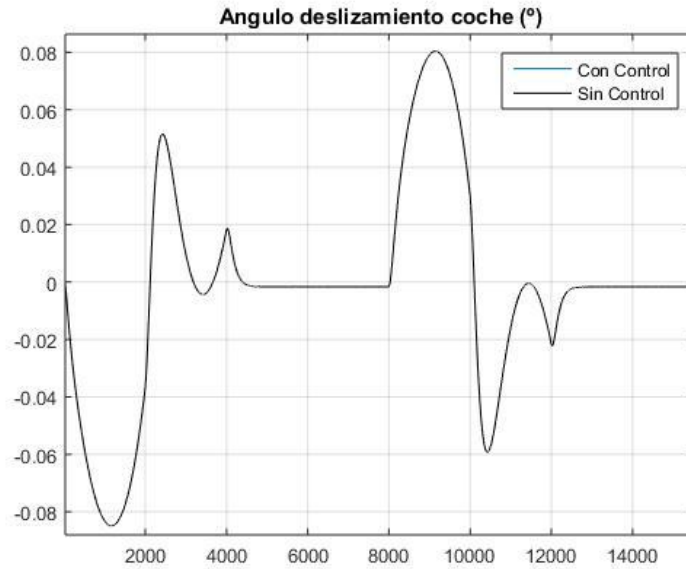


Figura 8-29 Ángulo de deslizamiento Exp.2 Control 2

Aunque se ha llegado al resultado que se quería, vamos a hacer una leve modificación en el experimento para no tener únicamente un resultado en el que el controlador actúe: se repetirá esta misma maniobra a 100 km/h. Se realizará esta maniobra para los dos controladores para así poder compararlos.

8.4 Doble cambio de carril a 100 km/h

8.4.1 Descripción maniobra

Esta maniobra, como se ha comentado, va a ser la misma que en la figura 8-18, pero aumentando la velocidad a 100 km/h.

8.4.2 Resultados controlador lógica difusa

Empezamos visualizando el par de cada rueda en la figura 8-30:

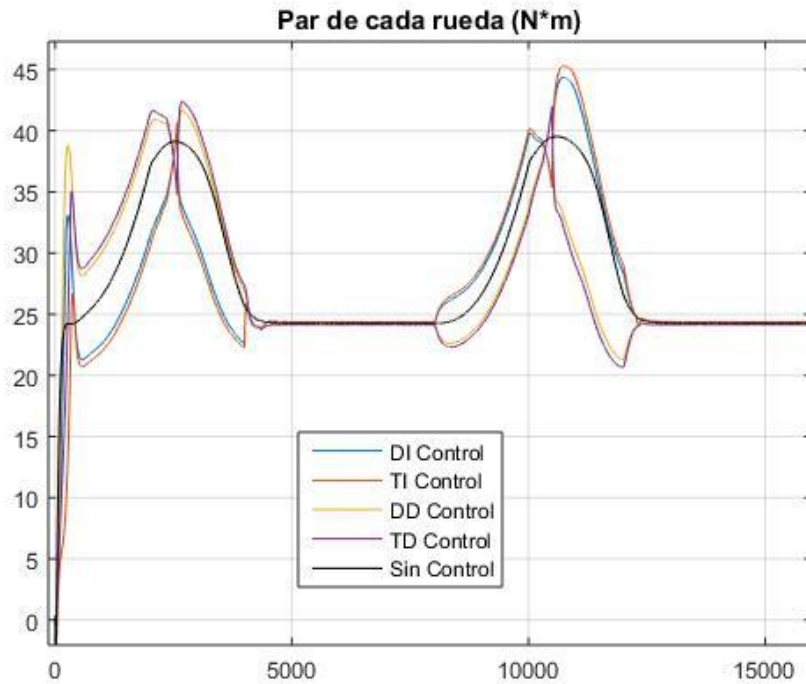


Figura 8-30 Par de cada rueda Exp. 3 Control 1

Respecto a la distribución de par poco podemos ver que van a existir cuatro zonas bien diferenciadas en los resultados: una primera en la que se realiza un primer giro, donde el par empieza a variar en cada rueda en función de lo que el controlador calcula; una segunda zona en la que el par de las ruedas coincide prácticamente con el par pedido por el conductor en el caso en el cual no hay control, pues se está realizando un tramo recto; y los tramos tercero y cuarto, que son los simétricos en cuanto a trayectoria de los dos primeros.

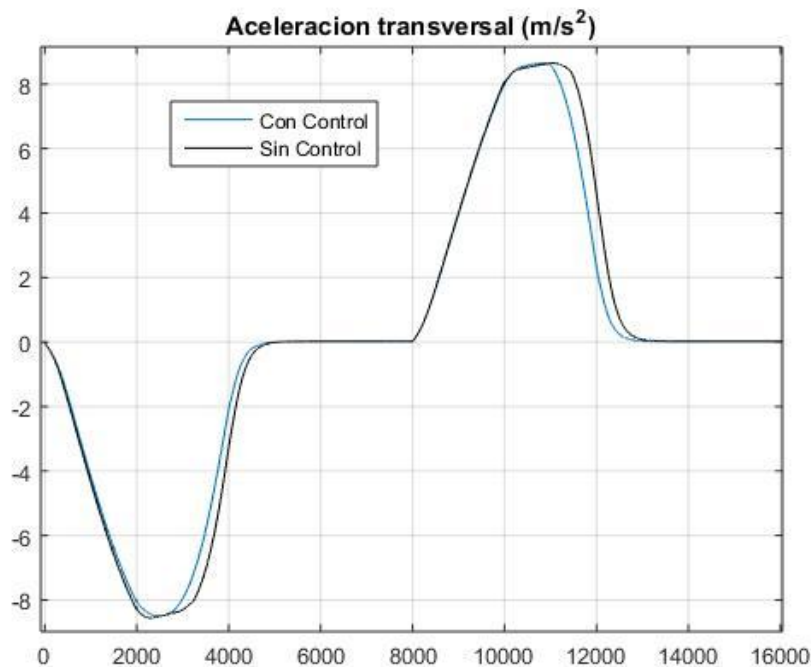


Figura 8-31 Aceleración transversal Exp. 3 Control 1

Analizando la aceleración transversal del vehículo durante la maniobra, podemos apreciar que la mayor reducción se consigue durante los tramos en los que se está terminando cada cambio de carril. Además, se aprecia que se llega al cero de aceleración antes del caso sin controlar, lo cual es positivo.

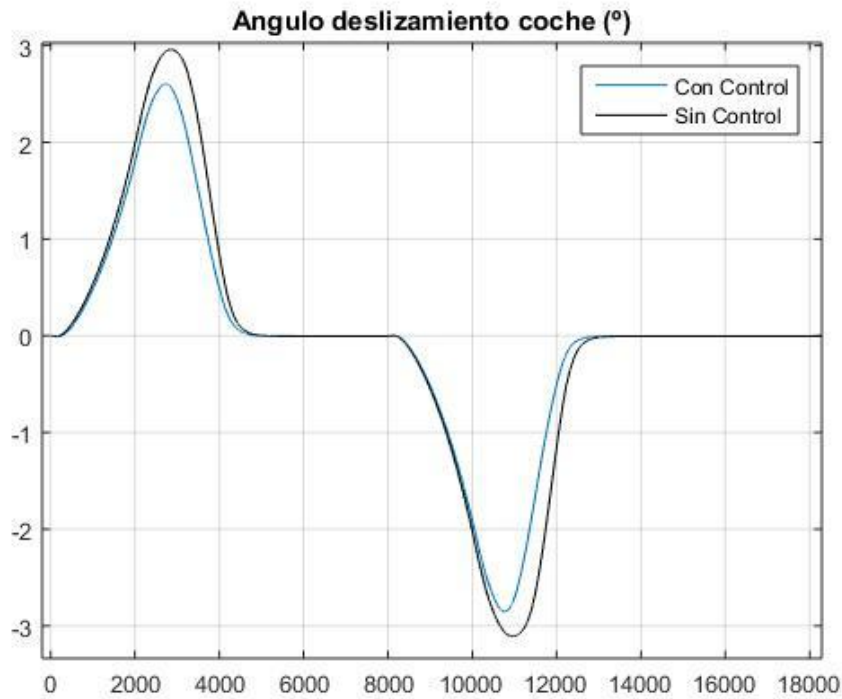


Figura 8-32 Ángulo de deslizamiento del vehículo Exp. 3 Control 1

Basándonos en la figura 8-32, podemos afirmar que se consigue reducir el ángulo de deslizamiento del vehículo, lo cual era una de las metas del controlador. Si hacemos un análisis más fino, vemos algo que ya se percibió al estudiar la aceleración transversal, y es que las mayores mejoras se producen a partir de la mitad de cada cambio de carril, lo cual es coherente.

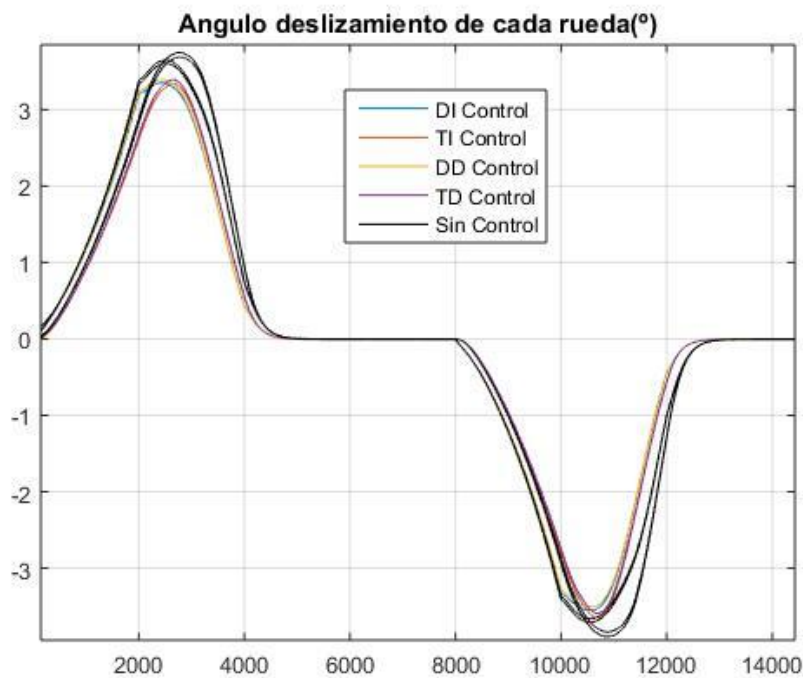


Figura 8-33 Ángulo de deslizamiento de cada rueda Exp. 3 Control 1

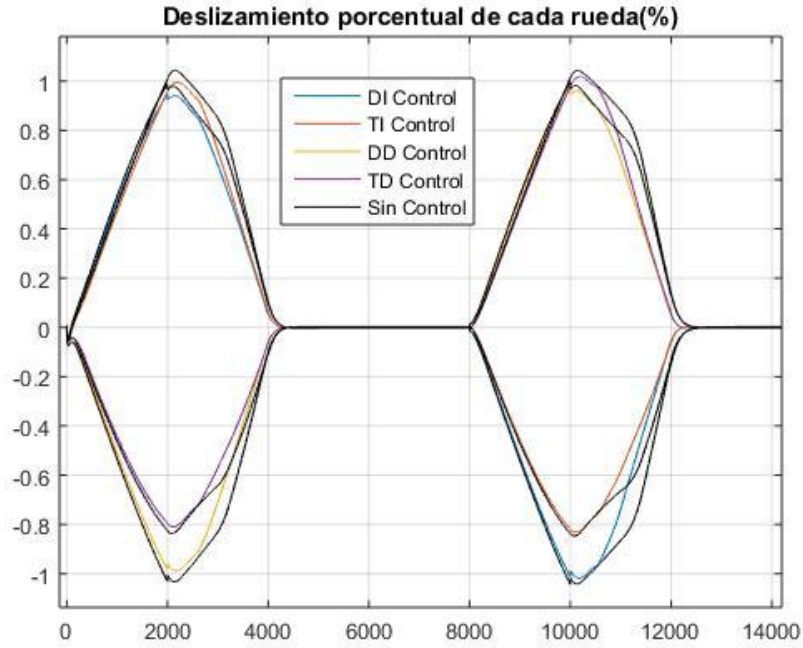


Figura 8-34 Deslizamiento porcentual de cada rueda Exp. 3 Control 1

De las figuras 8-33 y 8-34 poco se puede añadir, pues se muestra que efectivamente mejora el deslizamiento tanto en ángulo como porcentual de cada rueda, algo esperado viendo los resultados anteriores.

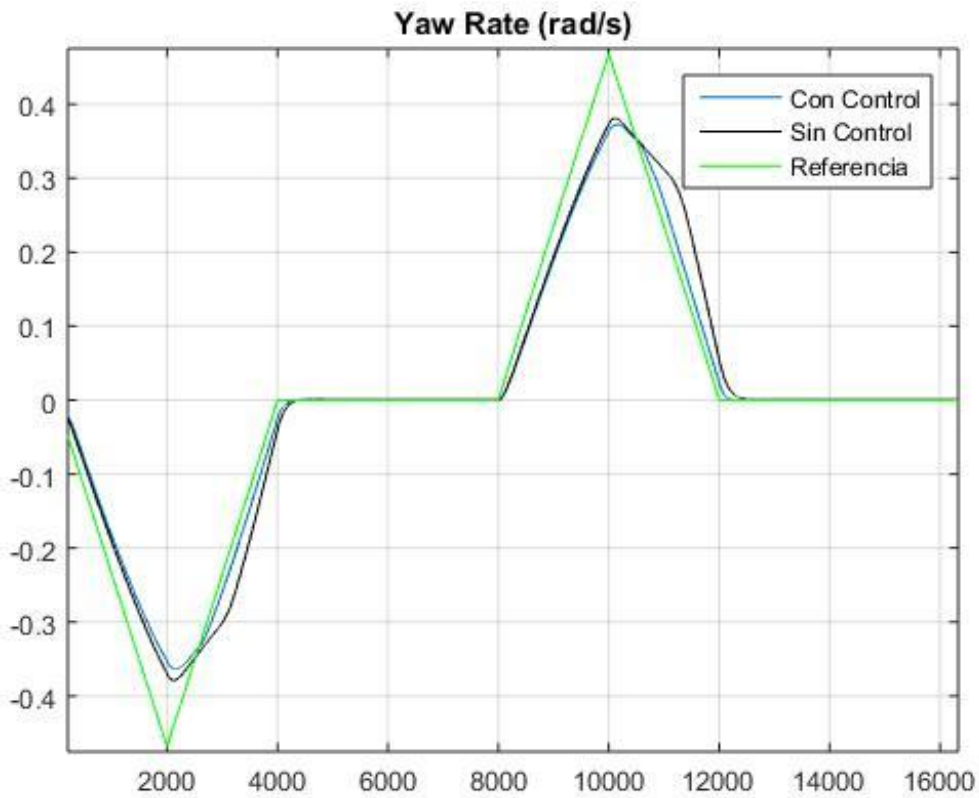


Figura 8-35 Yaw Rate Exp. 3 Control 1

Respecto al Yaw Rate del vehículo vemos que al empezar cada cambio de carril el controlador no es capaz de hacer que se llegue a la referencia, pero no es un gran problema pues la diferencia respecto al caso sin control es muy pequeña. Sin embargo, si nos fijamos a partir de la mitad de cada cambio de carril (muestras 2000-4000 y

10000-12000) podemos apreciar que casi se alcanza la referencia, y se ve claramente que mejora, reduciendo el sobreviraje pues hay menos diferencia entre la referencia y el valor real.

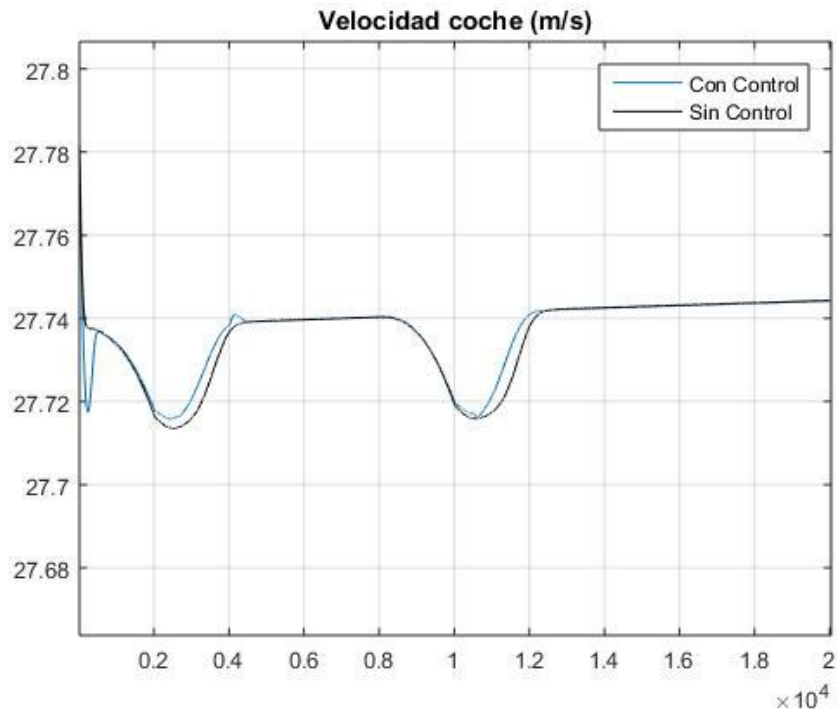


Figura 8-36 Velocidad del vehículo Exp. 3 Control 1

Como última gráfica se muestra la variación de la velocidad del vehículo, la cual vemos que no ofrece grandes cambios pues la diferencia máxima es de 0.02m/s o 0.072 km/h.

Para concluir con esta simulación, mostramos a continuación algunos datos numéricos para ofrecer una mejor comparación:

Par total entregado sin control:	2140052.28 Nm
Par total entregado con control:	2140052.28 Nm
Mejora porcentual de aceleración transversal:	64.286 %
Mejora porcentual de ángulo de deslizamiento del coche:	32.000 %
Mejora máxima porcentual de ángulo de deslizamiento de cada rueda:	28.013 %
Mejora rég. perm. porcentual de ángulo de deslizamiento de cada rueda:	0.000 %
Mejora porcentual de deslizamiento de cada rueda:	22.222 %
Mejora porcentual de pérdida de velocidad del coche:	0.049 %
Error régimen permanente de velocidad del coche Sin Control:	0.061 m/s
Error régimen permanente de velocidad del coche Con Control:	0.061 m/s

8.4.3 Resultados controlador proporcional

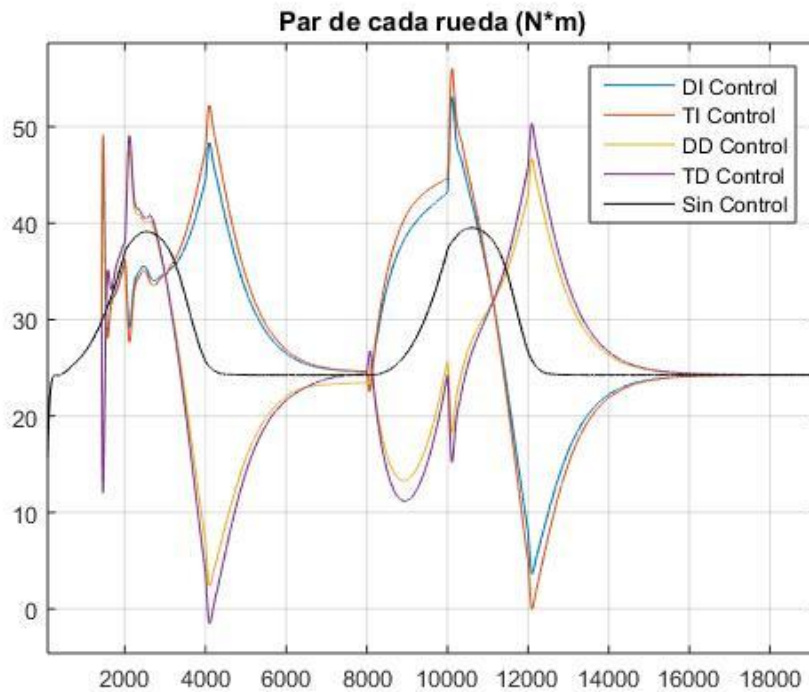


Figura 8-37 Par de cada rueda Exp. 3 Control 2

En cuanto al par que se entrega a cada rueda, podemos observar que al empezar la maniobra de cambiar al carril derecho se empieza dando más par a las ruedas izquierdas, de manera que se puede girar más, y una vez ya ha pasado esta primera parte, se reduce el par de estas ruedas para intentar estabilizar más el vehículo. Por otra parte, el par de estas ruedas tiende al par que pide el conductor una vez ha transcurrido un cierto tiempo desde que se hizo la maniobra: a velocidades menores, este tiempo se reduce.

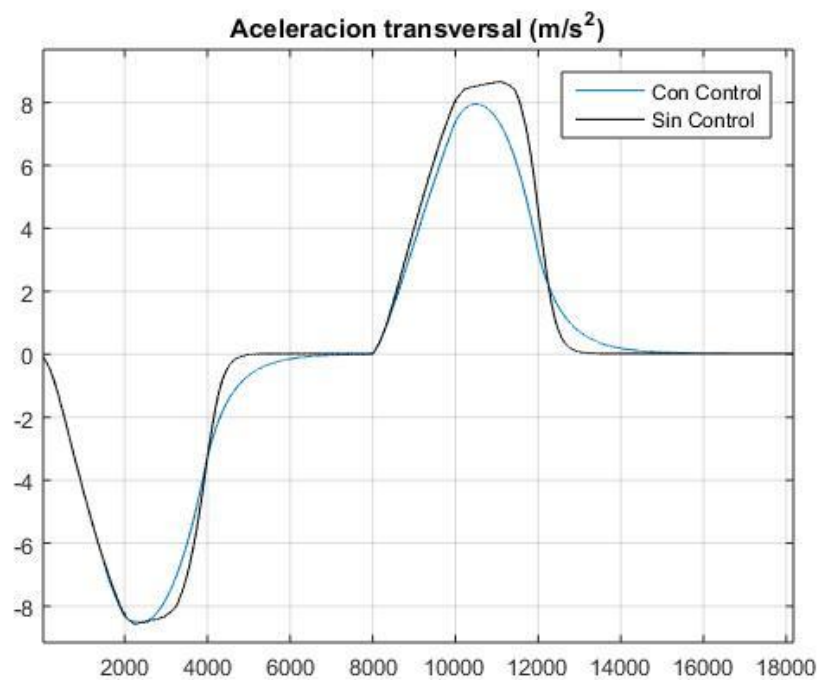


Figura 8-38 Aceleración transversal Exp. 3 Control 2

En la figura 8-38 observamos que se obtienen resultados importantes en cuanto a aceleración lateral, pues no sólo conseguimos reducirla, sino que su evolución es más suave, lo cual es bueno para el conductor pues sentirá menos los efectos de la aceleración.

Si se analiza con más profundidad este resultado, observamos que cuando la aceleración lateral del vehículo se reduce hasta llegar a cero, en el caso controlado se tarda más en llegar a este valor, lo cual tiene sentido, pues al ir a una velocidad tan alta, el vehículo tiene mucha inercia, y si se está dando más par a unas ruedas que a otras, el cambio a una situación en la que las cuatro ruedas tienen el mismo par es más lento. Esto se puede comprobar también en la figura del par, pues se observa que el vehículo tarda en recibir una distribución de par uniforme en las cuatro ruedas. Volviendo al capítulo en el que se explicó la implementación de este controlador, recordamos que se habló de una histéresis para cambiar de control activado a desactivado, en la cual para desactivarse el control se exige un error más pequeño en yaw rate y en ángulo de deslizamiento que para activarse. Si no existiese esto, el vehículo tardaría menos en llegar a cero en algunas variables, pero sería mucho más brusco, y si el objetivo del controlador era mejorar la estabilidad, mientras menos movimientos bruscos haya, mejor.

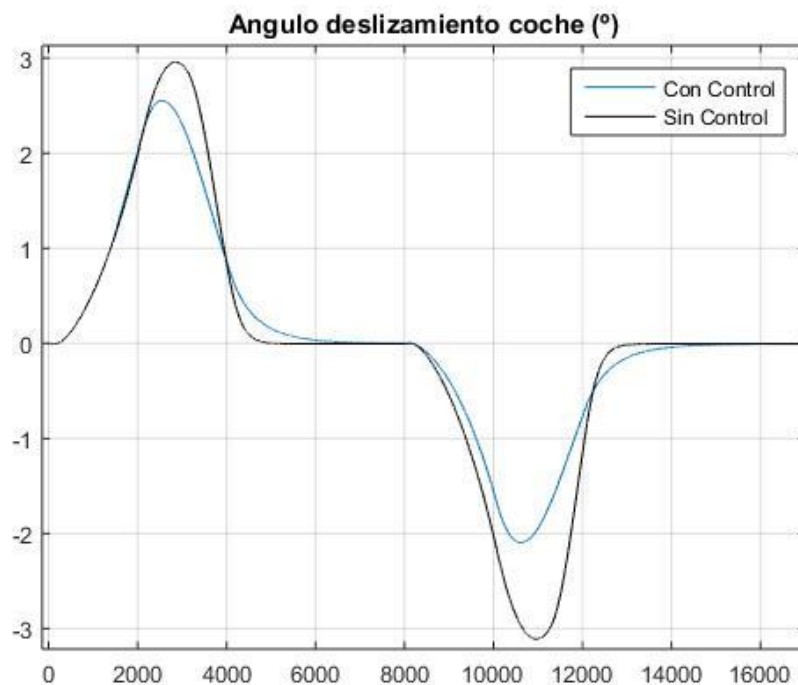


Figura 8-39 Ángulo de deslizamiento del vehículo Exp. 3 Control 2

En la figura 8-39 se observa el ángulo de deslizamiento del vehículo, el cual consigue reducirse considerablemente durante la trayectoria, y mantiene la tendencia antes comentada en los puntos cercanos a cero. Esta mejoría es bastante importante, pues como se dijo antes, el conductor por sí sólo no siempre es capaz de reducir el deslizamiento, y el control lo reduce bastante.

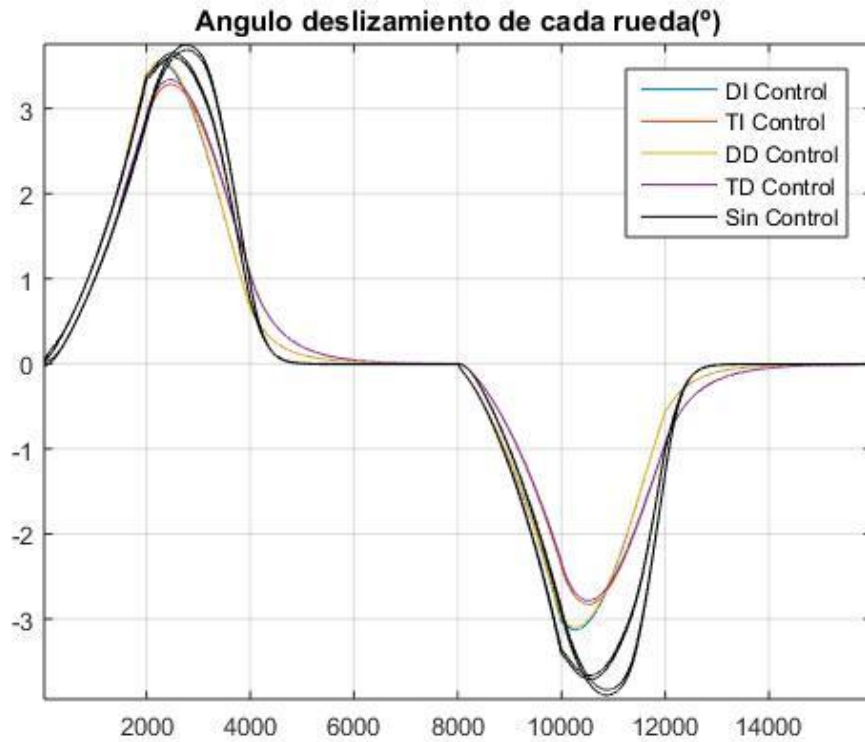


Figura 8-40 Ángulo de deslizamiento de cada rueda Exp. 3 Control 2

Poco cabe destacar sobre este resultado que no se haya comentado antes: reducción de valores máximos y transiciones suaves al cero.

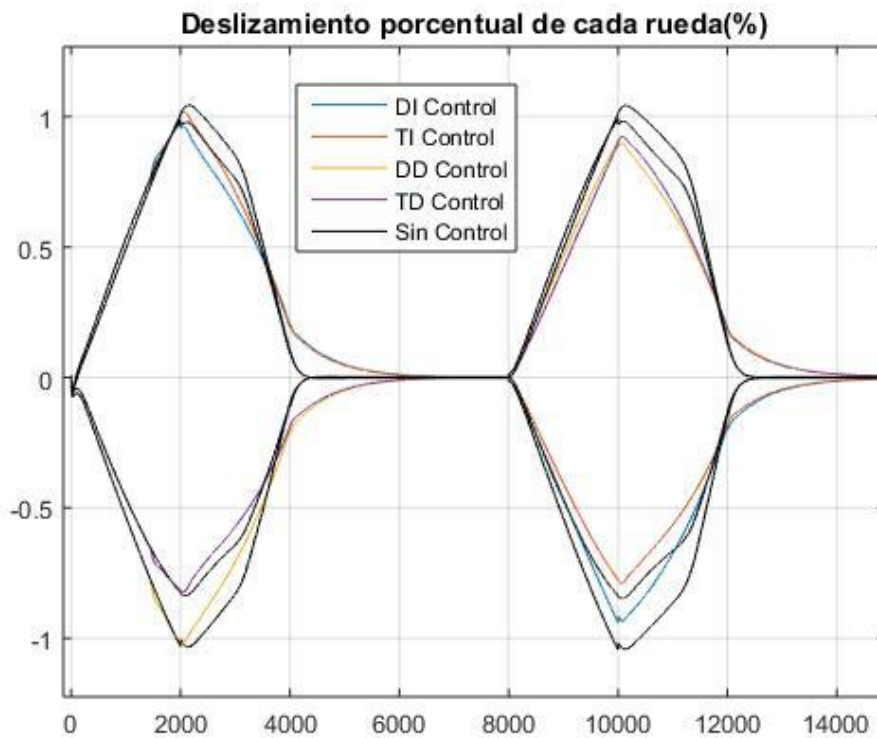


Figura 8-41 Deslizamiento porcentual de cada rueda Exp. 3 Control 2

En cuanto al deslizamiento porcentual de cada rueda, no se observa nada que no se pudiera esperar, pues se reduce esta variable en todo el trayecto salvo en las transiciones.

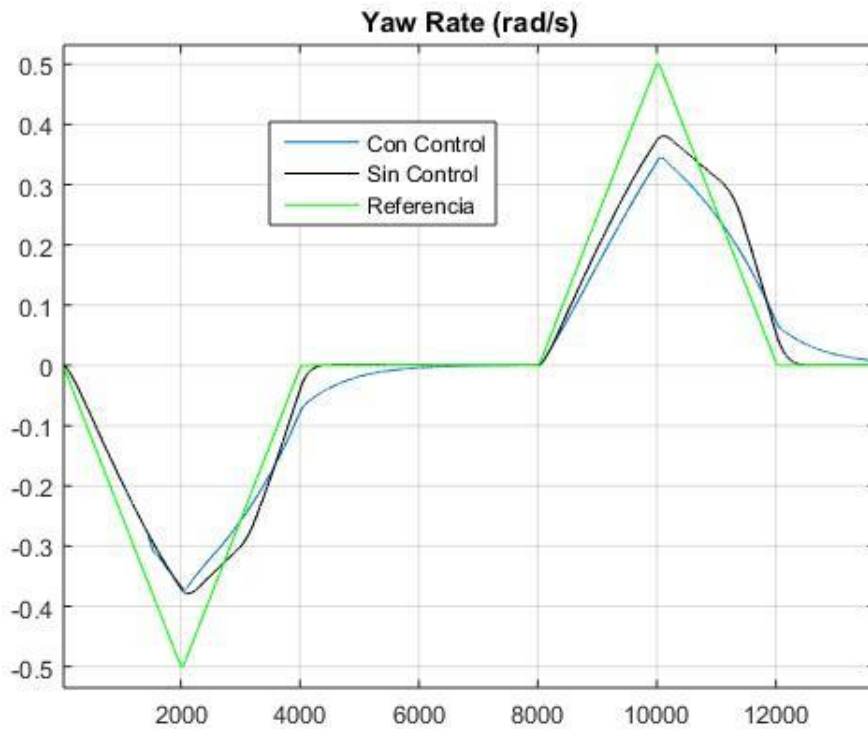


Figura 8-42 Yaw Rate Exp. 3 Control 2

En la figura 8-42 volvemos a ver algo que ocurrió con el primer controlador: se ha conseguido mejorar bastante el ángulo de deslizamiento del vehículo, sin embargo, no se ha sido capaz de llevar al vehículo a la referencia de Yaw Rate. No tiene por qué suponer un gran problema dado que, si se compara con el caso sin control, se observa que a tramos se reduce el sobreviraje (en valor absoluto, la referencia es menor que el valor actual) y en otros, el subviraje (en valor absoluto, la referencia es mayor que el valor actual). Sin embargo, al no ser una mejora presente en toda la simulación, sino en algunas partes, no se considera tan bueno como la mejora del ángulo de deslizamiento.

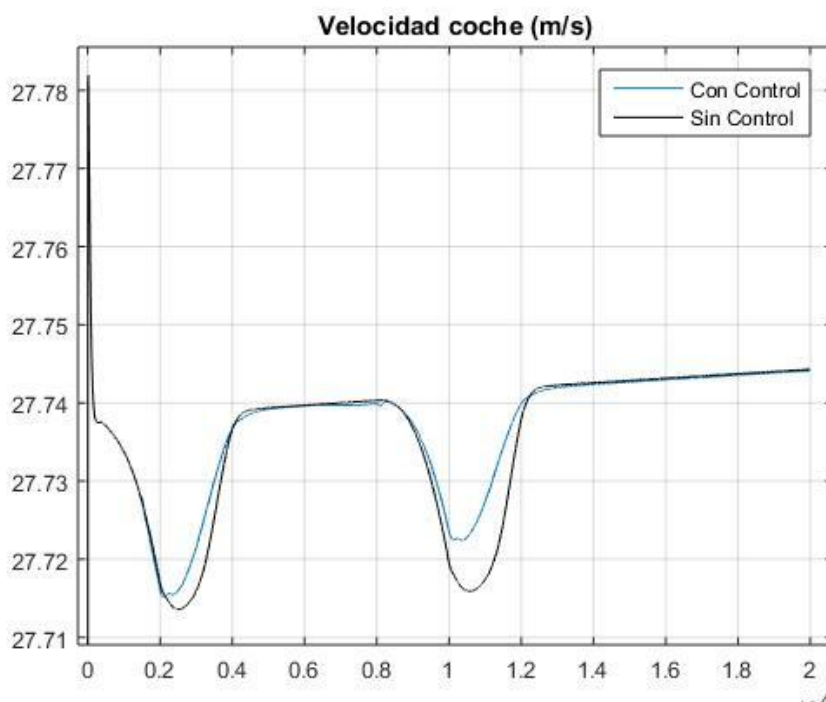


Figura 8-43 Velocidad del vehículo Exp. 3 Control 2

De la misma manera que en el resto de simulaciones, los resultados de la velocidad del vehículo no presentan gran mejoría pues no hay mucho margen de mejora.

Por último, presento a continuación resultados numéricos de las anteriores gráficas:

Par total entregado sin control:	2181084.38 Nm
Par total entregado con control:	2124807.06 Nm
Mejora porcentual de aceleración transversal:	23.401 %
Mejora porcentual de ángulo de deslizamiento del coche:	32.258 %
Mejora máxima porcentual de ángulo de deslizamiento de cada rueda:	31.579 %
Mejora rég. perm. porcentual de ángulo de deslizamiento de cada rueda:	0.000 %
Mejora porcentual de deslizamiento de cada rueda:	18.182 %
Mejora porcentual de pérdida de velocidad del coche:	0.259 %
Error régimen permanente de velocidad del coche Sin Control:	0.061 m/s
Error régimen permanente de velocidad del coche Con Control:	0.061 m/s

Con estos resultados concluimos los resultados obtenidos de este controlador: conseguimos mejorar la mayoría de variables de importancia del vehículo, y las transiciones entre control activo y control total del conductor son suaves, por lo que se da por cumplido el objetivo de este controlador.

8.5 Comparativa entre controladores

A continuación se mostrarán unas gráficas comparando los resultados de los dos controladores y el caso sin control para los experimentos de giro brusco a la izquierda y para los de cambio de carril a 100 km/h, pues son los que aportan más información acerca de la actuación de los diferentes controles. La metodología a seguir será la siguiente: se mostrará una gráfica por variable analizada, y en ella se verán los datos obtenidos en cada simulación con diferentes colores, azul para el primer controlador (el de lógica difusa), rojo para el segundo controlador (controlador proporcional) y negro para el caso sin control activo; tras lo cual se resaltarán las principales diferencias.

8.5.1 Giro brusco hacia la izquierda

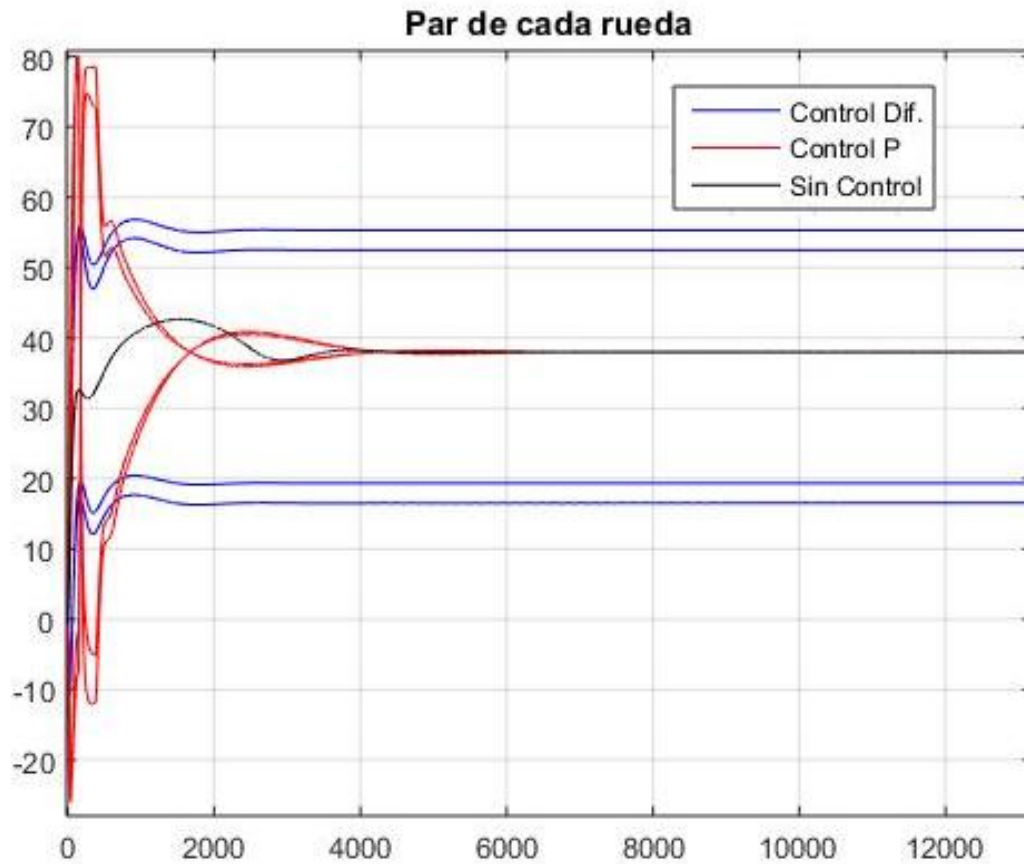


Figura 8-44 Comparativa de par 1

En la figura 8-44 se muestran los diferentes pares otorgados durante el experimento de giro hacia la izquierda. Se observa una diferencia principal: el controlador proporcional actúa mucho durante el transitorio, mientras que en el permanente el conductor es el que tiene el control total del vehículo; sin embargo, el controlador de lógica difusa actúa algo menos en el transitorio, pero a cambio mantiene su actuación durante toda la simulación.

Aunque no se vea debido a los colores, si vemos las figuras 8-2 y 8-10, apreciamos que al principio de las simulaciones se aumenta el par de las ruedas izquierdas, y se reduce el de las situadas a la derecha, por lo cual ambos controladores suman y restan a las mismas ruedas, pero cantidades diferentes. Esto es bueno pues ambos buscan una solución similar, aunque con métodos diferentes.

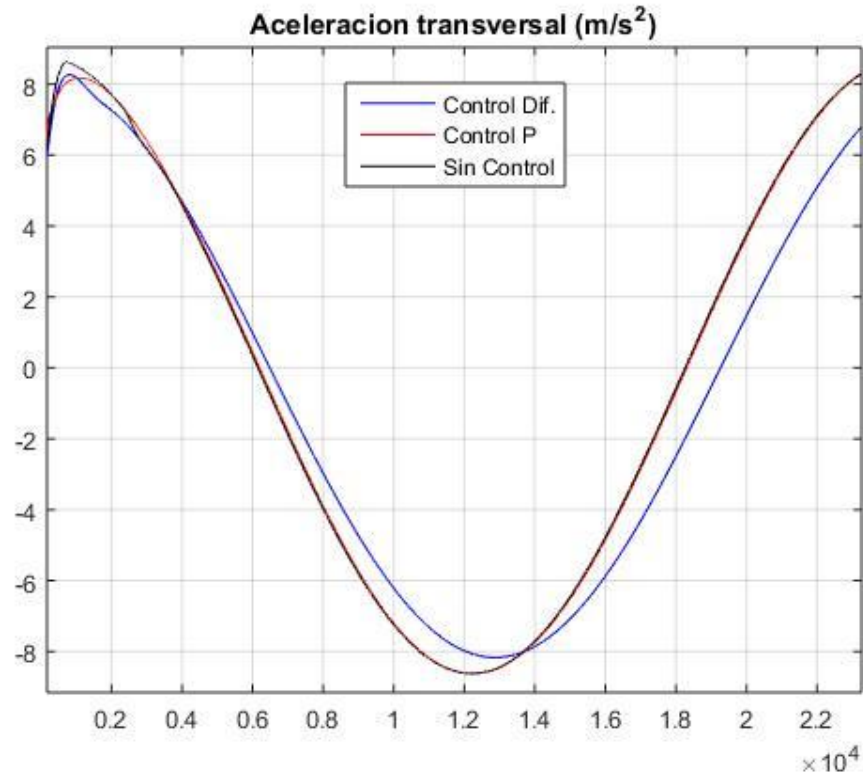


Figura 8-45 Comparativa de aceleración transversal 1

Siguiendo la tónica de lo observado comparando el par, en la figura 8-36 se aprecia claramente que el controlador proporcional tiende a actuar rápidamente al principio de la maniobra, mientras que luego deja de actuar, y el basado en lógica difusa presenta su mayor mejora una vez transcurrido el transitorio inicial.

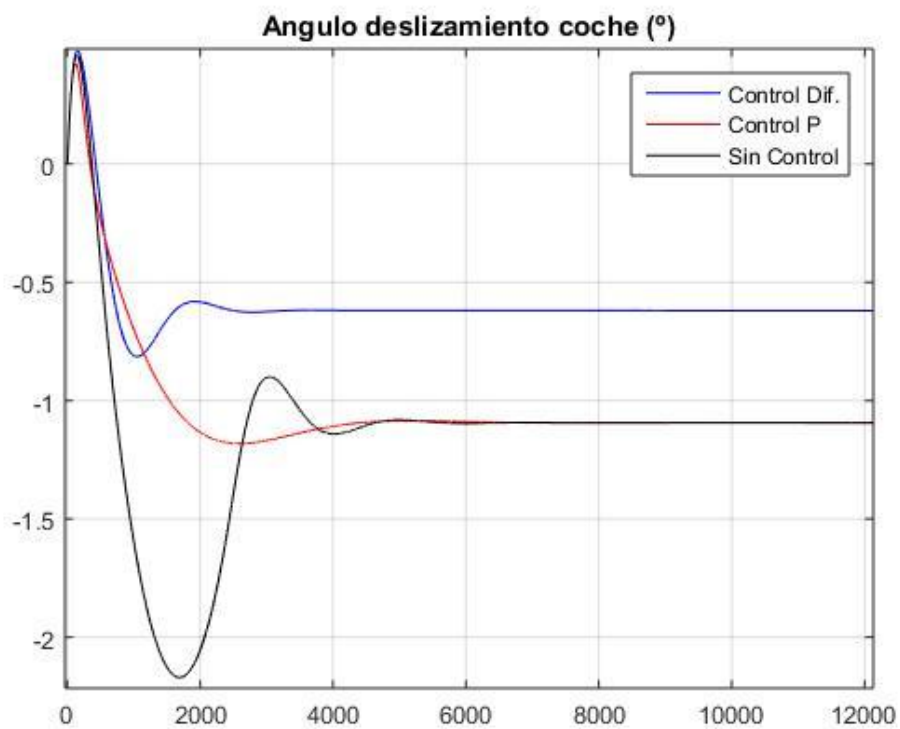


Figura 8-46 Comparativa de ángulo de deslizamiento del vehículo 1

En la figura anterior es en la que mejor se aprecia cómo actúa un controlador en comparación con el otro: el control de lógica difusa mejora en todo momento el ángulo de deslizamiento del vehículo, y lo reduce casi en

todo momento más que el controlador proporcional, el cual parece suavizar el ángulo que tendría el vehículo sin control. Por tanto, se ve claramente que el primer controlador tiende a modificar la dinámica del vehículo completa, mientras que el segundo la suaviza (presenta la menor sobreoscilación).

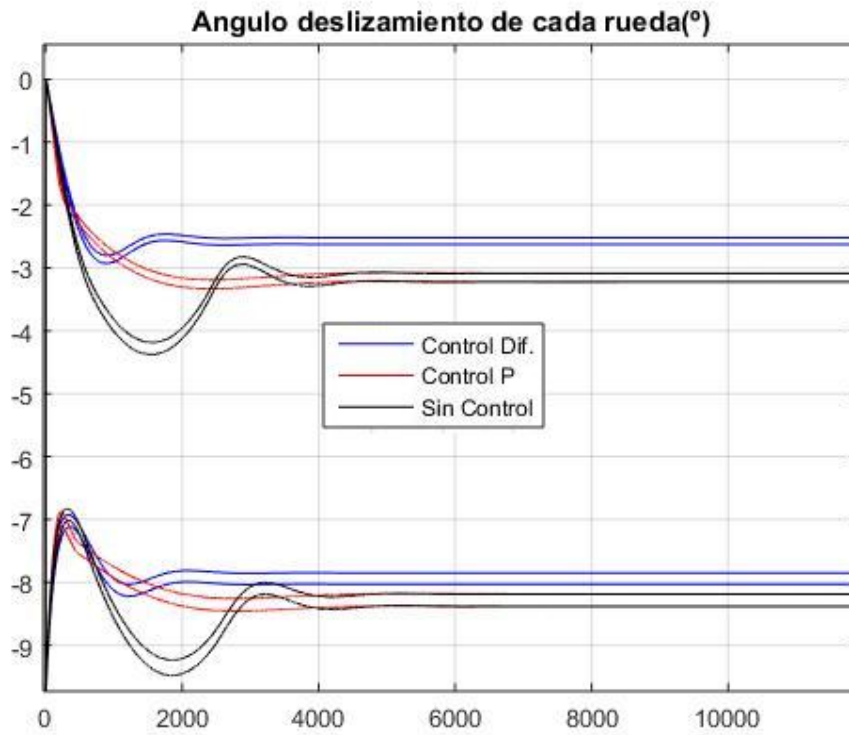


Figura 8-47 Comparativa de ángulo de deslizamiento de las ruedas 1

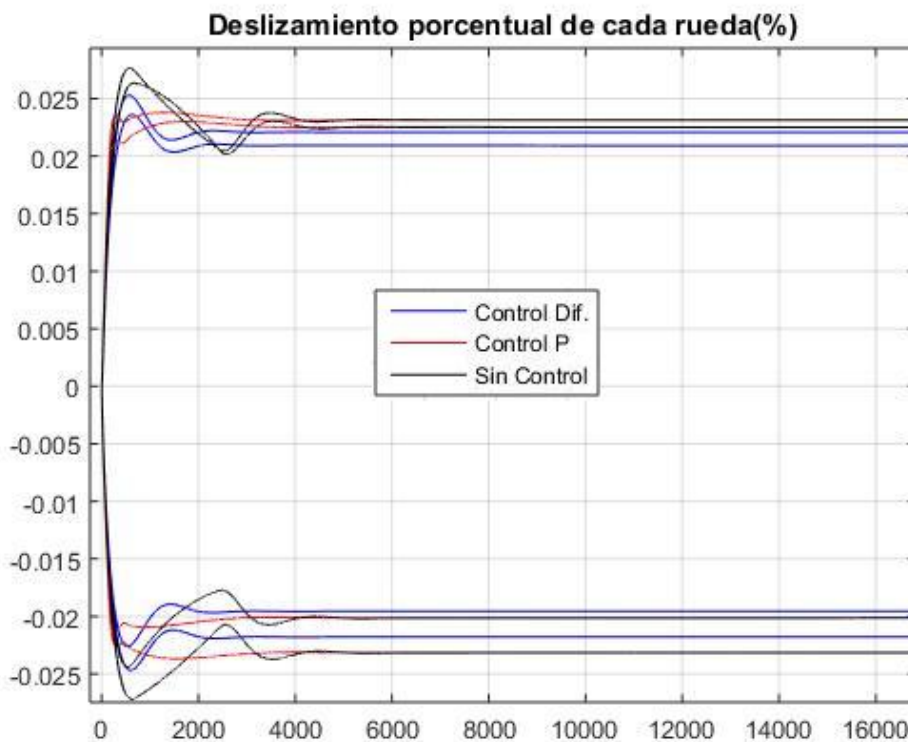


Figura 8-48 Comparativa de deslizamiento porcentual de las ruedas 1

En las figuras 8-47 y 8-48 se observa lo mismo que se comentó respecto al ángulo de deslizamiento del vehículo, un controlador suaviza los resultados, mientras que el otro obtiene mejoras en todo momento.

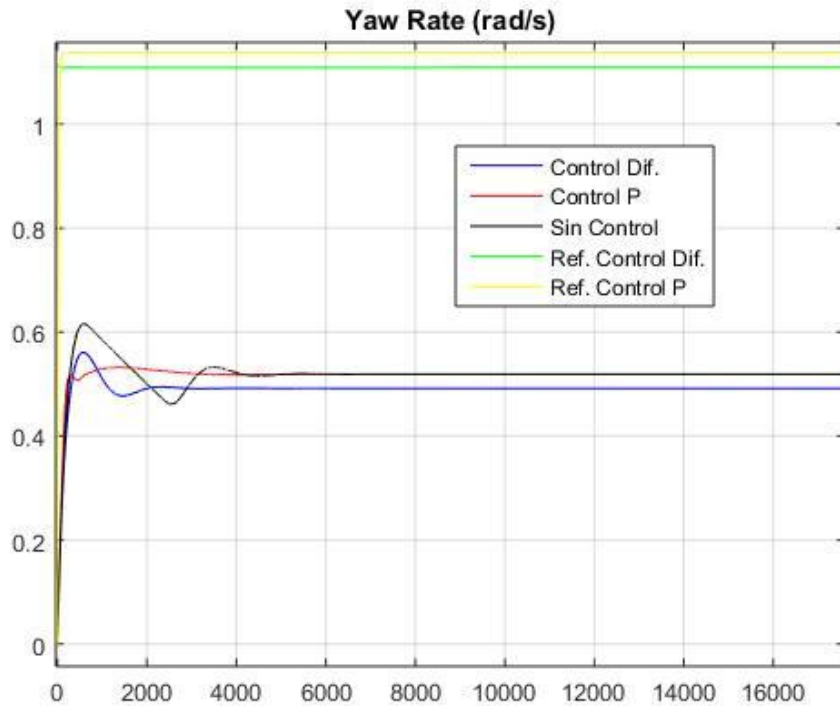


Figura 8-49 Comparativa de Yaw Rate 1

Respecto al yaw rate, podemos observar que ambos controladores tienden a reducirlo, cada uno según lo que se ha estado comentando durante este subcapítulo. En esta imagen podemos observar que las referencias de cada controlador, aún habiéndose obtenido de maneras diferentes, lo cual es positivo. Si analizamos un poco más el por qué son tan altas, podemos concluir que los generadores de referencias consideran que se podría mejorar mucho más el comportamiento del vehículo referido al yaw rate, sin embargo, debido a la limitación de los motores y de las ganancias, este cambio no se puede hacer, pero sí conseguimos que la actuación sea elevada, motivo por el cual mejoran tanto la mayoría de variables analizadas.

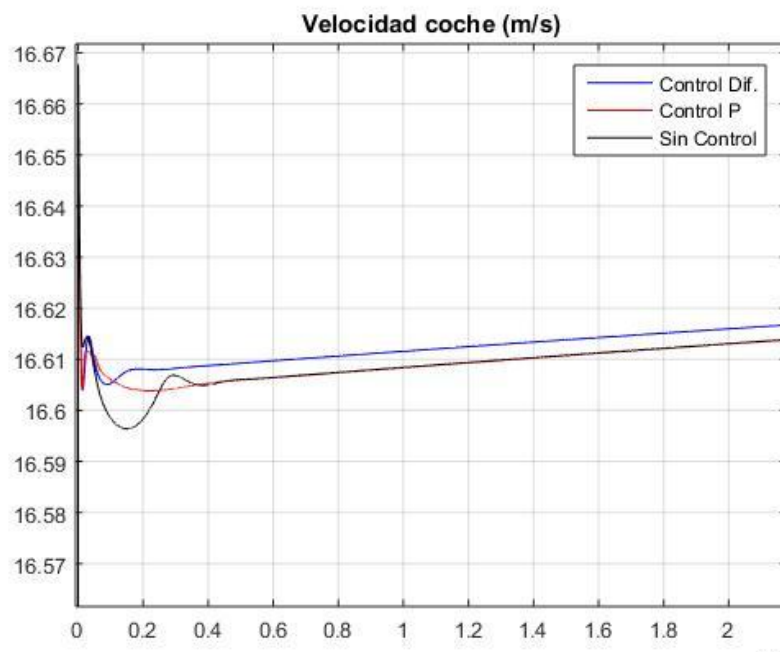


Figura 8-50 Comparativa de velocidad del vehículo 1

Observando la velocidad del vehículo, observamos que no hay gran mejora, pues no hay apenas margen de mejora.

Para concluir, se muestra la figura 8-51, en la que se comparan las trayectorias descritas por el vehículo:

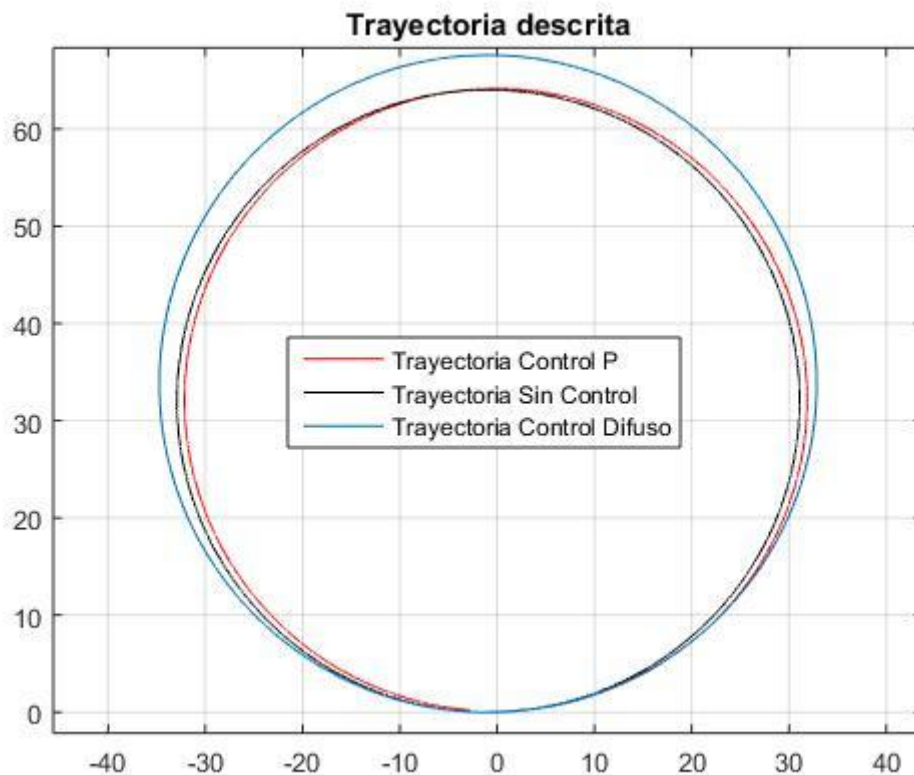


Figura 8-51 Comparativa de trayectorias 1

Se observa claramente en la trayectoria que describe el vehículo (las trayectorias empiezan en el punto $[0,0]$) que el controlador de lógica difusa, al actuar durante toda la simulación, hace que el vehículo tenga un radio de giro mayor, compensando así que el vehículo originalmente viraba demasiado. Esta diferencia es de unos 3 m en el diámetro de su trayectoria, la cual no es muy importante si tenemos en cuenta que el diámetro total está en torno a los 62 m.

Por otra parte, el controlador proporcional no tiene una variación del diámetro superior a los 90 cm. Esto es así pues hacía que cambiara el par fundamentalmente durante el principio de la maniobra, dejando durante el resto de la misma la distribución de par que pedía el conductor, por ello se observa que al principio se describe con el controlador una circunferencia de radio mayor, pero luego el radio es igual al caso sin control, por lo que la circunferencia se ve desplazada hacia la derecha.

8.5.2 Doble cambio de carril a 100 km/h

Vamos a analizar primero una de las gráficas en las que se van a observar los mayores cambios, la de distribución del par:

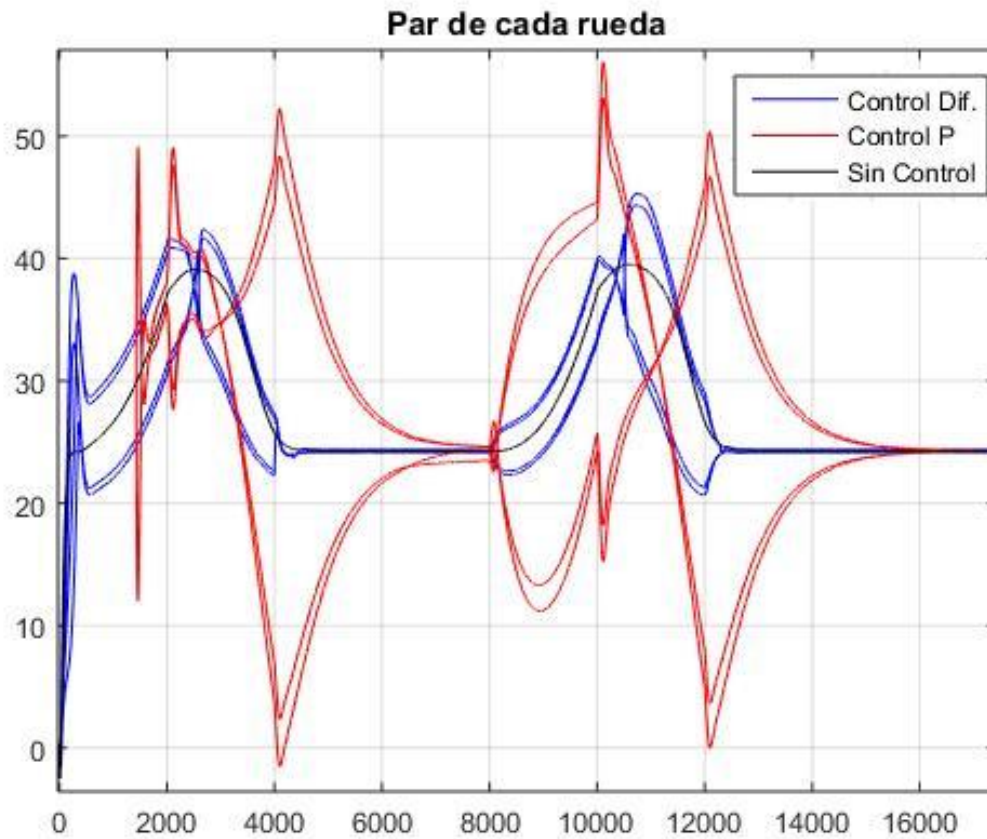


Figura 8-52 Comparativa de par 2

Si observamos detenidamente la figura 8-52, podemos apreciar que los pares que da cada controlador son muy diferentes, lo cual tiene una explicación lógica: el controlador de lógica difusa se diseñó con una restricción que era que no se podía dar par negativo salvo en momentos puntuales, lo cual resultó en que la separación de los pares de las ruedas se limitase, como se aprecia. Sin embargo, en el segundo controlador no existía esta limitación, por lo que es aceptable una mayor diferencia de par en las ruedas. Cuando la velocidad era baja, como en la simulación del giro brusco, esta diferencia no era tan notable, pero ahora al ser casi el doble, se observa mucho mejor.

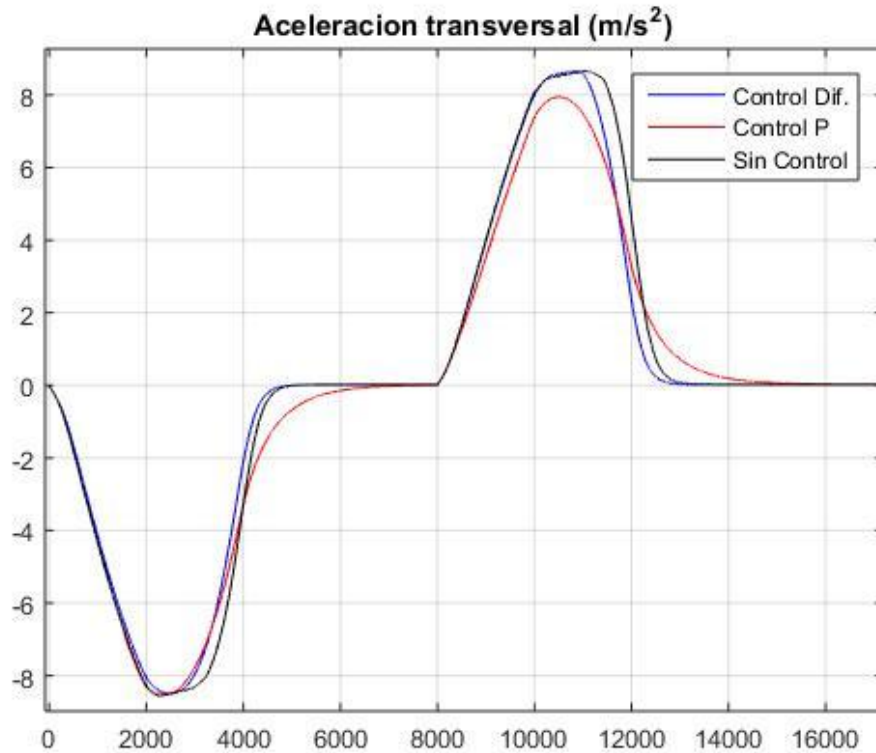


Figura 8-53 Comparativa de aceleración transversal 2

Respecto a la aceleración transversal, vamos a ver que las mejoras del controlador proporcional son las mayores, pues al permitir una diferencia de par mayor, podemos mejorar más los parámetros de la conducción. Una diferencia muy notable entre ambos controladores es el comportamiento al acercarse al cero: el controlador proporcional tarda más en llegar a él por las histéresis anteriormente comentadas, lo cual no ocurre con el otro controlador. De todas maneras, ambas curvas son suaves, de manera que no debe suponer gran diferencia.

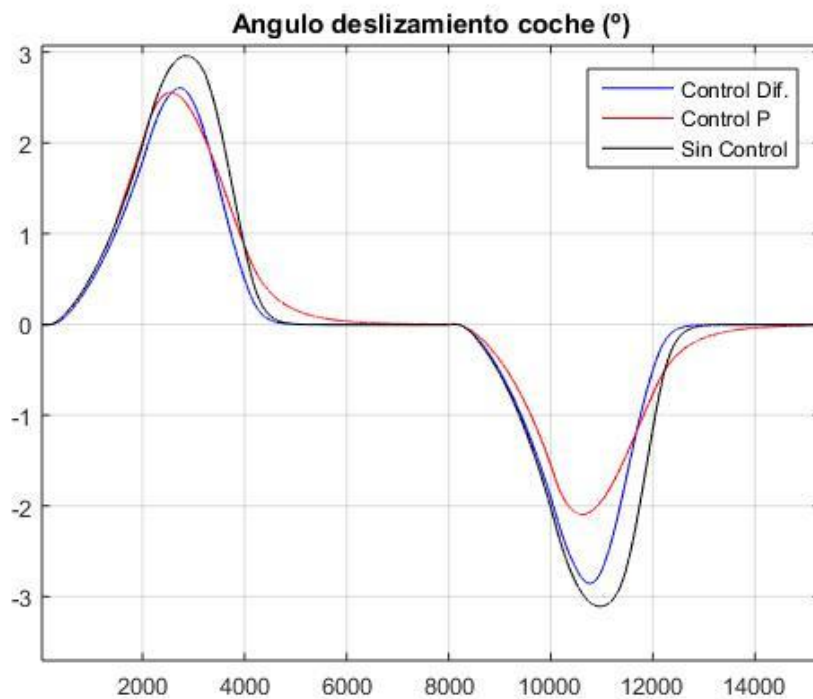


Figura 8-54 Comparativa de ángulo de deslizamiento del vehículo 2

Analizando la figura 8-54, podemos apreciar las diferencias antes comentadas respecto a la aceleración transversal: el controlador proporcional obtiene mejores resultados, pero tarda más en dejar esta variable a cero.

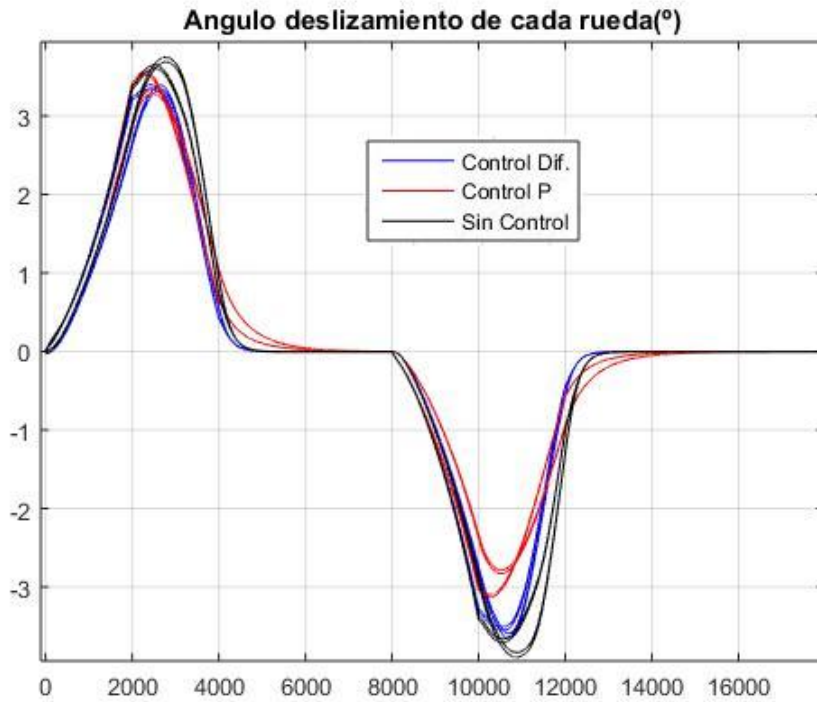


Figura 8-55 Comparativa de ángulo de deslizamiento de las ruedas 2

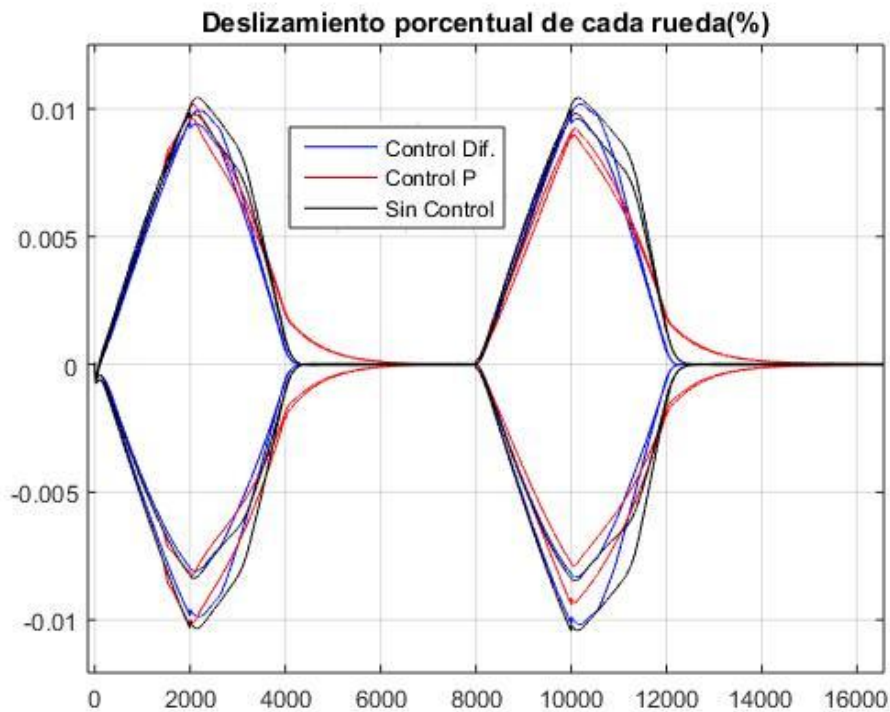


Figura 8-56 Comparativa de deslizamiento porcentual de las ruedas 2

Respecto a las figuras 8-55 y 8-56 observamos, como cabía esperar, lo mismo que se viene comentando en este subapartado.

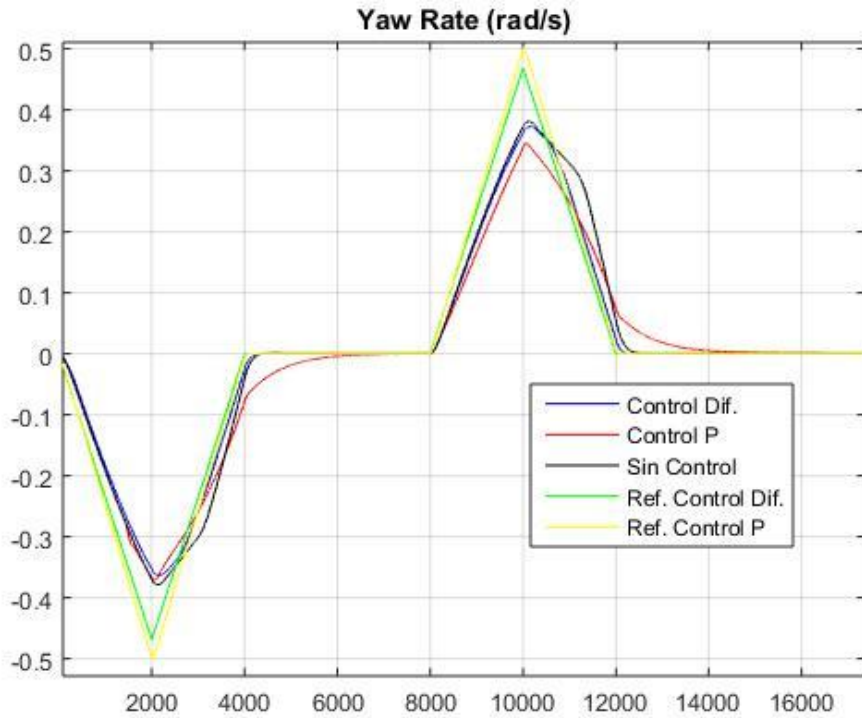


Figura 8-57 Comparativa de Yaw Rate 2

El Yaw Rate del vehículo permite un análisis mayor al resto de variables. Por una parte, vemos que el control difuso es el que consigue acercarse más a su referencia, lo cual tiene sentido pues los controladores se diseñaron con el objetivo de mejorar tanto el yaw rate como el ángulo de deslizamiento, pero a veces es complicado que al mejorar una variable mucho mejore la otra. Esto es lo que ocurre en este caso: al mejorar el deslizamiento del vehículo más en el controlador proporcional, el yaw rate no es capaz de mejorar tanto. Por otra parte, se observa que las referencias de cada controlador, a pesar de ser calculadas de manera diferente, son muy parecidas, como ya se vió anteriormente. Si ampliamos:

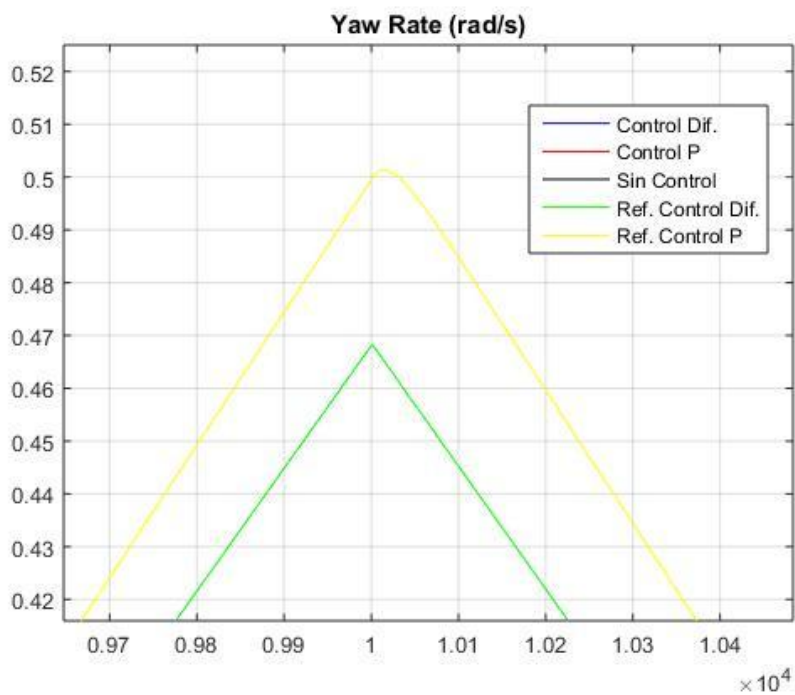


Figura 8-58 Comparativa de Yaw Rate aumentada 2

Observamos que la diferencia de ambas referencias está en torno a un 6%, por lo que se puede decir que ambas son aceptables.

Por último, se muestra la comparación de la velocidad del vehículo, la cual como se viene viendo durante todo este trabajo no presenta grandes diferencias:

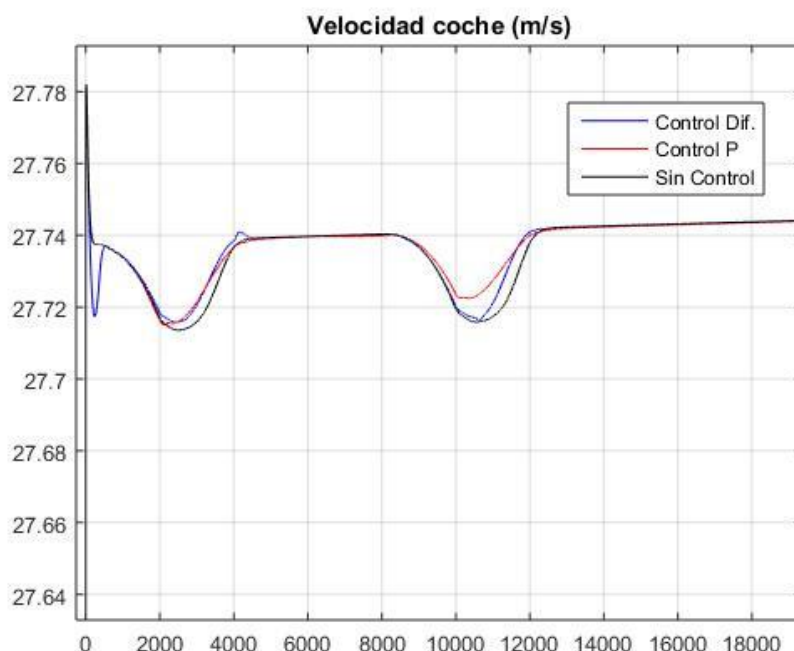


Figura 8-59 Comparativa de velocidad del vehículo 2

A la vista de estos resultados, podemos concluir lo que se está observando desde las primeras figuras de este subcapítulo: el primer controlador, el de lógica difusa, modifica y mejora la dinámica del vehículo durante toda la conducción, mientras que el segundo controlador, el proporcional, la suaviza en los transitorios de las maniobras. Además, a mayores velocidades el controlador proporcional reduce más el ángulo de deslizamiento que el de lógica difusa. Por tanto, según el ámbito en el que se vaya a emplear el vehículo y el tipo de conductor que vaya a usarlo, será más interesante un controlador u otro.

En un principio, se propone el controlador de lógica difusa a una conducción más relajada y orientada principalmente a conductores noveles, pues el sistema mejora la dinámica del vehículo, dándole menos peso a las acciones de la persona al volante; por el contrario, se propone el controlador proporcional a los conductores más agresivos y con más experiencia, pues tienen prácticamente el control del vehículo y lo que hace es suavizar su comportamiento. No obstante, serían necesarias pruebas reales para determinar exactamente en qué condiciones interesa implementar el primero o el segundo.

9 MONTAJE DEL CONTROLADOR DE LÓGICA DIFUSA EN C (QNX)

El vehículo FOX tiene una ECU en la que está instalado QNX, un sistema operativo en tiempo real. Por tanto, necesitamos pasar del modelo de bloques de Simulink a un código en lenguaje C que pueda entender la ECU. Para ello, se dan varios pasos. El primero es sustituir todo el diagrama de bloques del controlador por una función de Matlab en la que se le pase los datos del vehículo, y calcule el par de cada rueda. Una vez esto funcione, se procede a codificar lo mismo en C, de manera que metiéndole manualmente entradas se obtenga la salida, y luego se modifica este código de manera que sea insertable en el código que existe para el cálculo del par de cada rueda del FOX.

En este capítulo se va a codificar el algoritmo del controlador de lógica difusa únicamente, y cuando se haya probado en el vehículo real, se procederá a implementar el segundo controlador, para así aplicar todo lo aprendido en el primer controlador en el segundo.

9.1 Código Matlab

Queremos conseguir un modelo en el que todo el sistema de control esté implementado en una única función, de manera que luego podamos transcribir el código a lenguaje C. Por tanto, se busca lo mostrado en la siguiente figura.

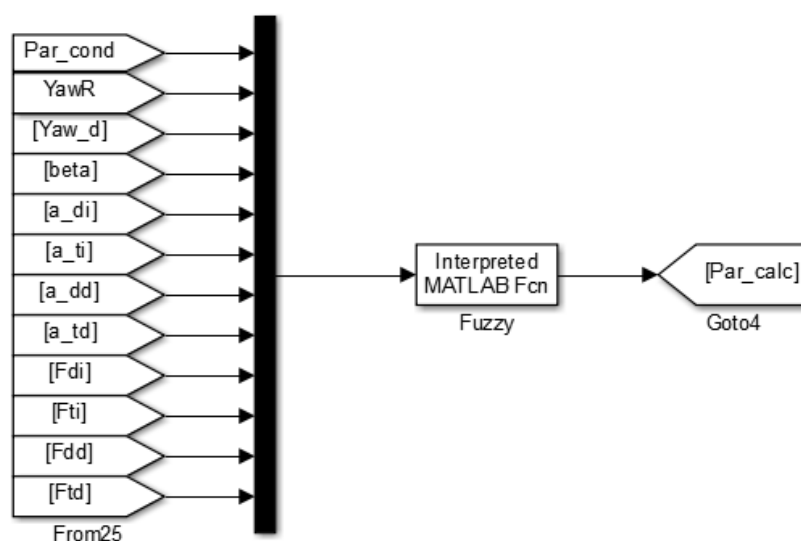


Figura 9-1 Implementación buscada

Esto es sencillo para todas las partes que se basan en una ecuación, de manera que lo único que hay que hacer es implementarlas en código. El problema viene al codificar la parte de lógica difusa, pues implementarla en código es muy larga, aunque no sea muy complicado. Como hay 25 reglas, hay que repetir muchas veces lo mismo, de manera que se propone otra solución, que consiste en tabular la nube de puntos de la salida de lógica difusa en función de las entradas, mostrada en la figura 5.8.

Para ello, se guardan los datos de cada par de entrada (las entradas van de -1 a 1) y la salida obtenida, de manera que obtenemos una tabla con tres columnas, entrada 1, entrada 2 y salida. Para tomar estos datos, se muestrean las entradas con paso 0.01, de manera que se obtienen 40000 combinaciones. De esta manera, obtenemos las columnas a1, a2 y a3, y si tenemos una dupla de errores, vemos en qué altura de la matriz se encuentra dicha dupla, y obtenemos la salida. Como son muchos datos, para que se ocupe menos memoria en la ECU, se ha redondeado la salida del controlador a 4 decimales, pues son más que suficientes para la precisión que necesitamos. A continuación muestro en una tabla un esquema sobre cómo están tabulados los datos para mayor claridad:

Tabla 9-1 Tabulación de Lógica Difusa

Fila	Error Yaw Rate	Error Slip	Salida
1	a	a	S1
2	a	b	S2
3	a	c	S3
4	b	a	S4
5	b	b	S5
6	b	c	S6
7	c	a	S7
8	c	b	S8
9	c	c	S9

Lo primero que hay que ver es que tanto el error de Yaw Rate como el error de Slip están de -1 a 1 con paso 0.01, por lo que se entiende que en la tabla existan los valores a, b y c repetidos en las columnas de los errores, aunque para simplificar, se ha supuesto que en este caso solo hay tres valores para cada error.

De esta manera, si tenemos un error en Yaw Rate c, y un error en Slip b, se debe buscar en la primera fila hasta dar con el error b, que es la 7 y luego buscar en la segunda columna si a la misma altura de ese error c (fila 7) tenemos el error de Slip b. Como en esa fila tenemos error en slip a, debemos buscar el siguiente error c en la primera columna, pasando a la fila 8, momento en el que encontramos que el error de Slip de la fila 8 es b, el buscado, por tanto, la salida es S8. Como se aprecia, es sencillo y rápido.

A continuación, hay que comparar si los resultados obtenidos aproximando la lógica difusa por una nube de puntos son buenos. Para ello, se crea la siguiente función, que compara los resultados que daría el bloque de lógica difusa y los resultados obtenidos mirando en la tabla la salida en función del error de deslizamiento y de yaw rate. a1, a2 y a3 no se muestran en el código pues son muchos números y no aportan nada.

```
function [tabla,fuzzy,t_tabla,t_fuzzy] = TablaFuzzy(e_yaw,e_beta)

a1=[]; %datos de error de yaw rate
a2=[]; %datos de error de slip
```

```

a3=[]; %salida de lógica difusa
%Inicializo por seguridad
fuzzy=0;
tabla=0;

%primero veo los resultados analiticos
tic;
fismat=readfis('Fuzzy_Moment');
fuzzy=evalfis([e_beta,e_yaw],fismat);
t_fuzzy=toc;

%pongo mismo numero de decimales para evitar errores
tic; %para medir el tiempo que tarda cada metodo
e_yaw=round(e_yaw,2);
e_beta=round(e_beta,2);

%ahora tenemos que ver si nos salimos del rango de la tabla (0,1) para los
%errores
if abs(e_yaw)>1
    e_yaw=1;
end
if abs(e_beta)>1
    e_beta=1;
end
for i=1:length(a1) %busco primera coincidencia en
primera columna
    if (a1(i)== e_yaw)
        %fprintf('found\n')
        if((a2(i) == e_beta)) %buscando en la misma altura de
la primera columna, veo si encuentro el segundo error
            tabla=a3(i); %lo encuentro, lo guardo en tabla
            % fprintf('FIN\n');
            break;
        end
    end
end
end
t_tabla=toc;
%saco tambien el resultado real para comparar resultados
fprintf('\nLa tabla es %f veces mas rapida\n',t_fuzzy/t_tabla)
fprintf('Resultado tabla - Resultado analitico:\t%f\n',fuzzy-tabla);

```

De esta manera, al ejecutarlo y pasarle una dupla de valores, nos informa sobre el tiempo que tarda cada método y el resultado. Por ejemplo, ejecutando la siguiente línea obtenemos:

```
>> [tabla,fuzzy,t_tabla,t_fuzzy] = TablaFuzzy(-1,0.2)
```

```

La tabla es 67.034767 veces mas rapida
Resultado tabla - Resultado analitico: -0.000029
tabla = 0.7941
fuzzy = 0.7941
t_tabla = 6.2323e-04
t_fuzzy = 0.041

```

De manera que podemos observar que el error es del orden de 0.1 milésima, lo cual, al multiplicarlo por 400 (la ganancia del controlador para pasar de esta salida a momento), el error es del orden de una centésima, lo cual es

despreciable.

Es muy importante apreciar que empleando el método de la tabla, se tarda hasta 67 veces menos (variará según el caso) en calcular la salida que haciéndolo analíticamente, y dado que el error es tan pequeño, se acepta este método pues consigue mejorar el tiempo de cálculo, algo que es fundamental en un sistema de control de par. Si nos vamos al final de la tabla, se tarda el doble en calcular el resultado mediante la tabla que analíticamente, pero viendo que podemos conseguir una mejora de hasta 67 veces el tiempo de cálculo, es una pérdida asumible.

Una vez nos hemos asegurado de las mejoras de tabular el controlador de lógica difusa, procedemos a crear el código que contiene el sistema completo de control.

Hay que matizar dos cosas sobre la búsqueda del valor en la tabla. La primera de ellas es que en el ejemplo anterior, le pasamos dos errores que eran exactamente igual a los que estaban en la tabla, de -1 a 1 con dos decimales. El simulador nos va a dar números que no tienen dos decimales únicamente, tienen más, de manera que hay que redondearlo. Para ello, lo que hacemos es ver si el número de la tabla menos el error es inferior o igual a 0.005, la mitad del paso con el que se tomaron los datos.

El segundo matiz es que podemos tener errores mayores de 1 en valor absoluto, aunque no sería lo deseable. Cuando al bloque de lógica difusa que hemos usado se le introduce un error superior a 1, calcula la salida como si el error fuese 1, de manera que tenemos que hacer lo mismo en la tabla, comprobar si el número se sale del intervalo [-1,1], y si es así, sustituirlo por 1 o -1. No debería llegarse a este punto, pues con el controlador evitamos que estos errores sean tan grandes, pero a veces en momentos puntuales pueden ocurrir si las maniobras son muy extremas.

Al final del código, se implementa una saturación del par que se va a aplicar a cada rueda, escogiendo el mínimo en valor absoluto del par calculado, el par máximo transmisible y el par máximo del motor, y luego se multiplica por el signo que tenía el par. Con esto, evitamos que se den pares que puedan dañar los motores.

Con todo esto, el código queda como sigue:

```
function [ out ] = CodigoQNX( in )
%Esta es la versión de Matlab del código, para comprobar si obtenemos
%los mismos resultados, luego se codificará en C.
%estos son los datos experimentales redondeados, no se muestran porque
son números simplemente y ocupan mucho espacio:
a1=[];
a2=[];
a3=[];
%asigno las entradas
Par_cond=in(1);
yaw_r=in(2);
yaw_d=in(3);
beta=in(4);
acel_di=in(5);
acel_ti=in(6);
acel_dd=in(7);
acel_td=in(8);
F_di=in(9);
F_ti=in(10);
F_dd=in(11);
F_td=in(12);

%Datos
r_d=0.25; %Radio de las ruedas delanteras
Jw_d=0.2334; %Momento de inercia de las ruedas delanteras
r_t=0.28; %Radio de las ruedas traseras
Jw_t=0.27; %Momento de inercia de las ruedas traseras
Iyyf=0.2334; %Momento de inercia de las ruedas delanteras
Iyyr=0.27; %Momento de inercia de las ruedas traseras
t_d=1.5538; %Anchura del eje delantero
t_t=1.4865; %Anchura eje trasero
m=400.238; %Masa del vehlo con dos ocupantes
```

```
Par_max=78; %Limitacion real del coche
```

```
e_yaw=yaw_r-yaw_d;
e_beta=beta-0; %Referencia es 0
```

%En esta parte, tenemos que hacer una saturacion de las entradas, de manera que si obtenemos un error de yaw rate o slip mayor de un radian, debemos hacer que el error valga 1 (o -1, segun el caso). La logica difusa toma errores mayores de uno como si fueran uno, asi que es lo que hacemos para la tabla

```
if abs(e_yaw)>1
    e_yaw=sign(e_yaw)*1;
end
if abs(e_beta)>1
    e_beta=sign(e_beta)*1;
end
```

```
%Ahora buscamos en la tabla los errores y la salida que corresponde
for i=1:length(a1) %busco primera coincidencia en primera columna
    if (abs(e_yaw-a1(i))<=0.005) %veo si el dato esta a menos de 0.005
        del dato de la tabla, entonces lo tomo por bueno
        if(abs(e_beta-a2(i))<=0.005) %buscando en la misma altura de la
            primera columna, veo si encuentro el segundo error
                mom_norm=a3(i); %lo encuentro, lo guardo
                break;
            end
        end
    end
end
```

```
%Paso a Momento
Momento=407*mom_norm;
%Reparto equitativo entre ejes
Mz_d=0.5*Momento;
Mz_t=0.5*Momento;
```

```
%Paso de Momento a Par
Par_di=Mz_d*(r_d/t_d)+Jw_d*acel_di; %Motor delantero izquierdo
Par_ti=Mz_t*(r_t/t_t)+Jw_t*acel_ti;
Par_dd=Mz_d*(r_d/t_d)+Jw_d*acel_dd; %Motor delantero derecho
Par_td=Mz_t*(r_t/t_t)+Jw_t*acel_td;
```

```
%Aplico par diferencial
DI=1;TI=1;DD=1;TD=1; %si error=0, saldra esto
```

```
if ((yaw_r-yaw_d)>0)
    if((yaw_r>0) && (yaw_d>0))
        DI=-1;TI=-1;DD=1;TD=1;
    end
    if ((yaw_r<0) && (yaw_d<0))
        DI=1;TI=1;DD=-1;TD=-1;
    end
    if ((yaw_r>0) && (yaw_d<0))
        DI=1;TI=1;DD=-1;TD=-1;
    end
    if ((yaw_r>0) && (yaw_d==0))
        DI=-1;TI=-1;DD=1;TD=1;
    end
end
```

```

end
end
if ((yaw_r-yaw_d)<0)
    if((yaw_r>0) && (yaw_d>0))
        DI=1;TI=1;DD=-1;TD=-1;
    end
    if ((yaw_r<0) && (yaw_d<0))
        DI=-1;TI=-1;DD=1;TD=1;
    end
    if ((yaw_r<0) && (yaw_d>0))
        DI=-1;TI=-1;DD=1;TD=1;
    end
    if ((yaw_r<0) && (yaw_d==0))
        DI=-1;TI=-1;DD=1;TD=1;
    end
end

%Sumo los pares adicionales
Par_di=Par_cond+DI*Par_di;
Par_ti=Par_cond+TI*Par_ti;
Par_dd=Par_cond+DD*Par_dd;
Par_td=Par_cond+TD*Par_td;

%Aplico MTTE
Par_diMAX=(Jw_d/(0.9*m*r_d^2)+1)*r_d*F_di;
Par_tiMAX=(Jw_t/(0.9*m*r_t^2)+1)*r_t*F_ti;
Par_ddMAX=(Jw_d/(0.9*m*r_d^2)+1)*r_d*F_dd;
Par_tdMAX=(Jw_t/(0.9*m*r_t^2)+1)*r_t*F_td;

%Escojo el par que aplico
Par_DI=sign(Par_di)*min(abs([Par_di,Par_diMAX,Par_max]));
Par_TI=sign(Par_ti)*min(abs([Par_ti,Par_tiMAX,Par_max]));
Par_DD=sign(Par_dd)*min(abs([Par_dd,Par_ddMAX,Par_max]));
Par_TD=sign(Par_td)*min(abs([Par_td,Par_tdMAX,Par_max]));

%Asigno variables de salida
out=[Par_DI, Par_TI, Par_DD, Par_TD];
end

```

A continuación tenemos que ver si la simulación del modelo con bloques y este modelo en el que el controlador es un código es la misma o muy similar, en cuyo caso se validará y se procederá a codificar el controlador en lenguaje C.

9.2 Comparaciones de la simulación del modelo de bloques y el modelo en código

Se ha comprobado anteriormente que en la aproximación de la salida del controlador como una nube de puntos existe un error tan pequeño que no debería afectar al funcionamiento del vehículo. Ahora vamos a comprobar si existen diferencias entre el controlador como modelo de bloques, y el controlador como se va a implementar en QNX.

Para ello, vamos a realizar la simulación del giro a la izquierda pues no presenta saturación como en el caso de la maniobra Fish Hook. Tras varias pruebas, se decidió cambiar la ganancia del controlador de 400 a 407, pues es con esa ganancia con la que se obtienen menores diferencias.

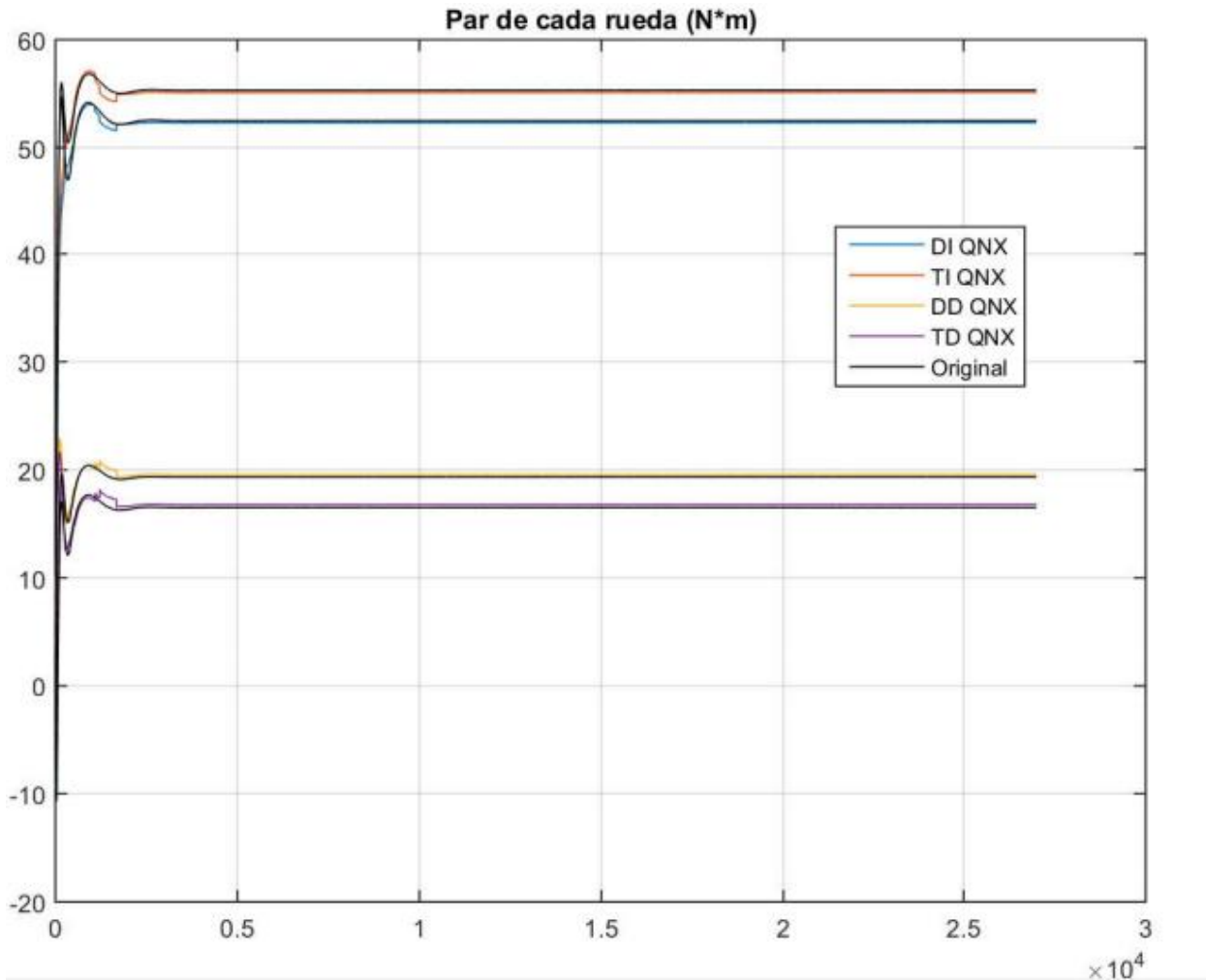


Figura 9-2 Comparación entre implementaciones del par

Como se observa en la figura 9-2, el par es muy parecido, el error en el régimen permanente es muy pequeño. Al principio, se observan algunos cambios muy pequeños, pero son prácticamente despreciables. Esto se debe a los filtros empleados en el modelo de bloques, que no se implementan en el código. Además, hay una pequeña diferencia respecto al tiempo de muestreo del bloque de lógica difusa y el tiempo de muestreo del controlador en código, por ello se aprecia alguna diferencia.

A continuación se mostrarán algunas gráficas de variables del controlador, van a ser menos que las mostradas en el capítulo de simulaciones pues queremos ver sólo las más importantes.

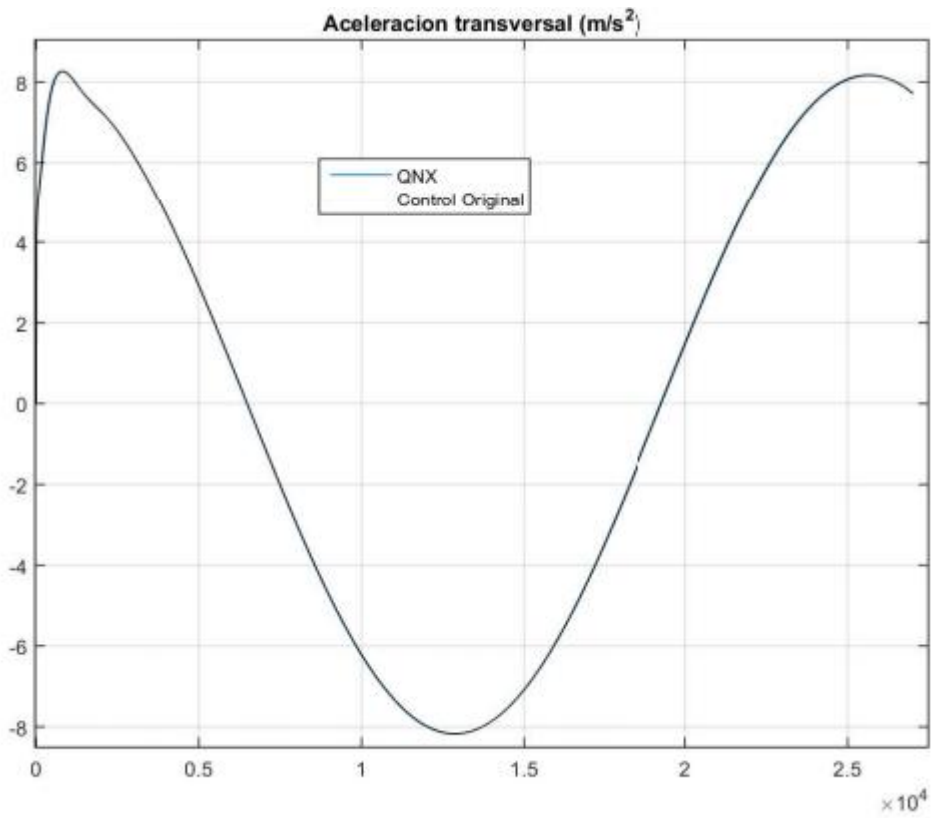


Figura 9-3 Comparación entre implementaciones de la aceleración transversal

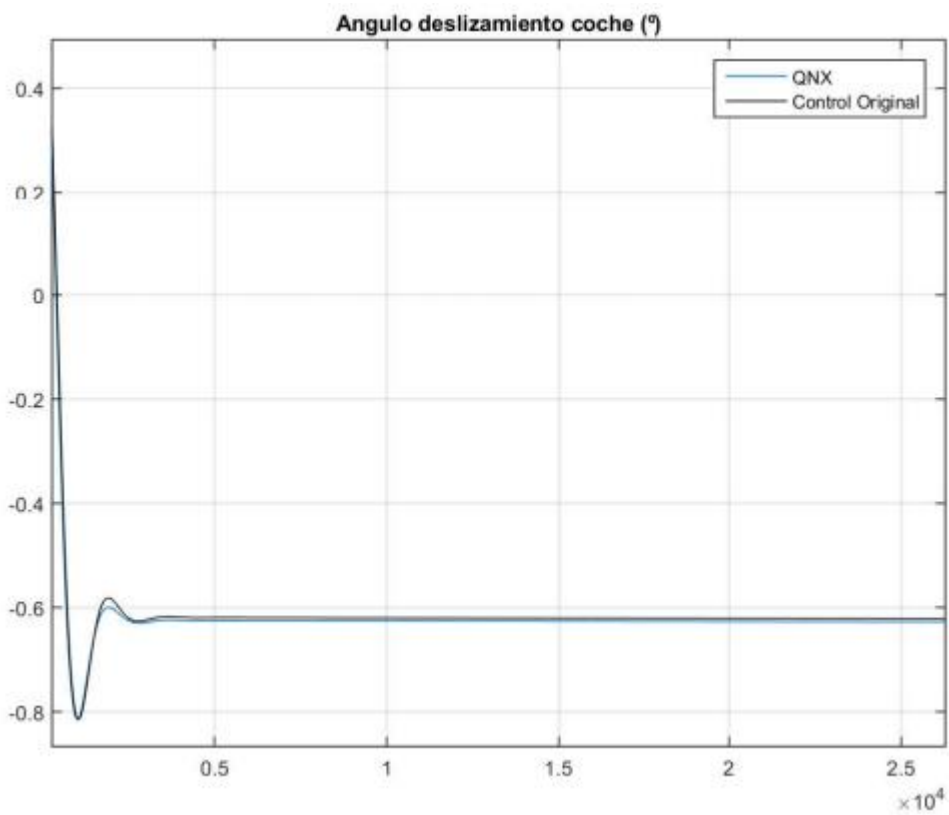


Figura 9-4 Comparación entre implementaciones del ángulo de deslizamiento del coche

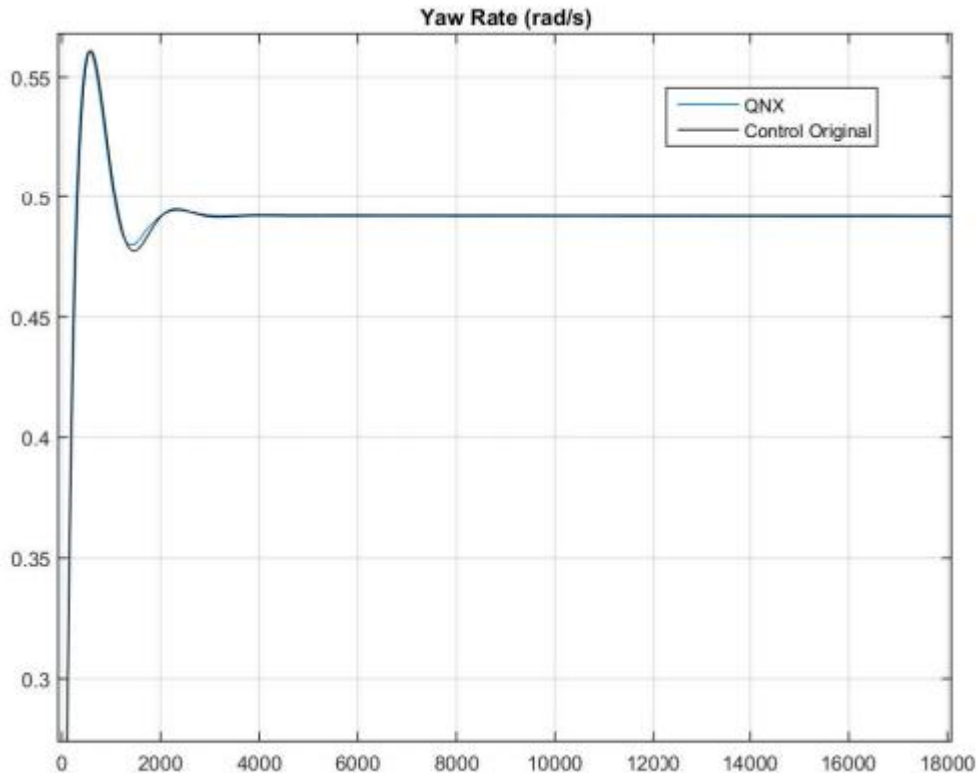


Figura 9-5 Comparación entre implementaciones del Yaw Rate

Como se aprecia con las figuras 9-1 a 9-5, los cambios son muy pequeños, en muchos casos inapreciables.

A continuación se presentan algunos datos numéricos para ver que, como se observa en las gráficas, las diferencias son despreciables.

Par total entregado con control para QNX:	3870495.38 Nm
Par total entregado con control original:	3867701.53 Nm
Ratio par/distancia con control para QNX:	18249.20 Nm/m
Ratio par/distancia con control original:	18221.73 Nm/m
Diferencia porcentual de aceleracion transversal:	0.074 %
Diferencia porcentual de ángulo de deslizamiento del coche:	-0.000 %
Diferencia máxima porcentual de ángulo de deslizamiento de cada rueda:	-0.074 %
Diferencia rég. perm. porcentual de ángulo de deslizamiento de cada rueda:	0.251 %
Diferencia porcentual de deslizamiento de cada rueda:	-0.000 %
Diferencia porcentual de perdida de velocidad del coche:	-0.000 %
Error régimen permanente de velocidad del coche Con Control para QNX:	0.048 %
Error régimen permanente de velocidad del coche Con Control original:	0.048 %

Como estos resultados demuestran, los errores son muy pequeños, la mayoría son del 0%, siendo el mayor de un 0.25%. Por tanto, concluimos que es una buena implementación.

9.3 Código lenguaje C

Una vez se ha tomado como bueno el código de Matlab, se procede a codificarlo en lenguaje C. Para ello, hay que hacer dos cosas principalmente:

- Pasar de las funciones de Matlab a las de C, como el valor absoluto (ahora se llama fabsf), y crear algunas nuevas, como calcular el mínimo de tres números y ver el signo de un número.
- Crear un archivo en el que esté guardada la tabla con los datos del controlador, de manera que al principio del programa se cargue el archivo, se tengan los datos en memoria, y cuando se acabe el programa se borre la tabla de la memoria. El archivo se cerrará en cuanto se carguen los datos, para tener ocupada la menor cantidad de memoria posible.

Las funciones creadas son muy sencillas, de manera que no merece la pena explicarlas, pues calcular el mínimo de tres números y ver el signo de un número del tipo float es muy sencillo.

Respecto a la tabla, hay que tener en cuenta que QNX no permite crear tablas de 40401 componentes de longitud, pues es demasiado grande, así que se tiene que hacer reserva dinámica de memoria.

Como es de vital importancia el tiempo que se tarda en ejecutar el código, se ha creado un código que permite calcular el tiempo que se tarda en cargar el archivo (se hace al principio del programa únicamente) y el tiempo que se tarda en calcular los cuatro pares cada vez que se ejecuta. Para ello, se usa la función `clock()`, la cual guarda en una variable el tiempo actual. Llamando varias veces a esta función, podemos ver que se tardan 6 centésimas de segundo en abrir y cargar la tabla en memoria, y los cálculos tardan menos de 1 centésima, por lo que se ha conseguido un buen tiempo de ejecución.

En el código mostrado a continuación no se muestra el procedimiento de calcular el tiempo, pues lo que se quiere conseguir en este apartado es un código en lenguaje C que responda igual que el de Matlab.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <unistd.h>
#include <math.h>
#include <fcntl.h>

/*****
ESTE ARCHIVO CALCULA EL PAR A APLICAR EN CADA RUEDA
*****/
float min(float x, float y, float z); //Calcula el minimo de tres numeros
int sign(float v); //Devuelve el signo del numero v

/*MAIN*/
void main(void)
{
    /*Reserva dinamica de memoria, en QNX no puedo crear una matriz tan
grande
del tipo float matriz[3][40401]; hay que hacerlo asi*/
    float *vec_yaw; //Vector para guardar
entrada e_yaw de fuzzy
    float *vec_slip; //Vector para guardar
entrada e_slip de fuzzy
    float *vec_salida; //Vector para guardar
salida de fuzzy
    vec_yaw=malloc(40402*sizeof(float)); //Reserva dinamica de
memoria para entrada primera de fuzzy
    vec_slip=malloc(40402*sizeof(float)); //Reserva dinamica de
memoria para entrada segunda de fuzzy
    vec_salida=malloc(40402*sizeof(float)); //Reserva dinamica de
memoria para salida de fuzzy
    /*Variables para el archivo*/
```

```

int i=0;
int j=0;
FILE *fp_fuzzy;
/*Datos del coche*/
float r_d=0.25; //Radio de las ruedas delanteras
float Jw_d=0.2334; //Momento de inercia de las ruedas
delanteras
float r_t=0.28; //Radio de las ruedas traseras
float Jw_t=0.27; //Momento de inercia de las ruedas traseras
float t_d=1.5538; //Anchura del eje delantero
float t_t=1.4865; //Anchura eje trasero
float m=400.238; //Masa del vehlo con dos ocupantes
float Par_max=78; //Limitacion real del coche
/*Variables para el controlador*/
float e_yaw; //error del yaw rate
float e_beta; //error del angulo de deslizamiento
float beta,yaw_r,yaw_d; //angulo deslizamiento, yaw real, yaw
deseado
float mom_norm; //Salida del controlador de -1 a 1
float Momento; //Momento extra que es necesario para
corregir error
float Mz_d,Mz_t; //Momento extra necesario en cada eje
float Par_di, Par_ti; //Par calculado para las ruedas de la
izquierda
float Par_dd, Par_td; //Par calculado para las ruedas de la
derecha
float DI, TI, DD, TD; //Coeficientes para aplicar par diferencial
float Par_diMAX,Par_tiMAX; //Resultado MTTE en las ruedas de la
izquierda
float Par_ddMAX,Par_tdMAX; //Resultado MTTE en las ruedas de la
derecha
float Par_DI,Par_TI; //Par que se aplica a las ruedas de la
izquierda
float Par_DD,Par_TD; //Par que se aplica a las ruedas de la
derecha
//Para asignar el par: T_voltios=T_calc/Relacion_ParPedal;
/*Datos del coche, en el vehículo se pasan con el nombre que esta
comentado, pero aquí se dan valores para poder ver si el código compila*/
float acel_di=10; //acel_ang_est_FL-Aceleracion angular rueda
float acel_ti=10; //acel_ang_est_RL
float acel_dd=0; //acel_ang_est_FR
float acel_td=10; //acel_ang_est_RR
float Par_cond=40; //Par_inicial_met1[i]; i=0,1,2,3
float F_di=10; //Ffricc_FL
float F_ti=10; //Ffricc_RL
float F_dd=10; //Ffricc_FR
float F_td=110; //Ffricc_RR

/*Cargo tabla*/
fp_fuzzy = fopen ( "Datos_Fuzzy.txt", "r" );
if (fp_fuzzy==NULL) {printf ("File error"); exit (1);}

for (i=0;i<40401;i++)
{
fscanf(fp_fuzzy, "%f", &vec_yaw[i]); /*Cargo el dato de entrada
error_yaw*/
}
for (i=0;i<40401;i++)
{
fscanf(fp_fuzzy, "%f", &vec_slip[i]); /*Cargo el dato de entrada
slip*/
}

```



```

    }
    for (i=0;i<40401;i++)
    {
        fscanf(fp_fuzzy, "%f", &vec_salida[i]); /*Cargo el dato de salida
momento*/
    }
    printf("\n");
    fclose (fp_fuzzy);

    /*LOGICA DIFUSA*/
    e_yaw=yaw_r-yaw_d; /*Error es real-deseado*/
    e_beta=beta-0; /*Referencia es 0*/
    //e_yaw=3;e_beta=-0.123456; //Pruebas para asegurar funcionamiento
    /*float fabsf(float i); devuelve el valor absoluto en decimal del numero
decimal i*/
    if (fabsf(e_yaw)>1) { /*por si me salgo de los limites*/
        e_yaw=sign(e_yaw)*1;}
    if (fabsf(e_beta)>1){ /*por si me salgo de los limites*/
        e_beta=sign(e_beta)*1;}
    for (i=0;i<40401;i++) /*busco primera coincidencia en primera
columna*/
    {
        if (fabsf(e_yaw-vec_yaw[i])<=0.005) /*veo si el dato esta a
menos de 0.005 del dato de la tabla, entonces lo tomo por bueno */
        {
            if(fabsf(e_beta-vec_slip[i])<=0.005) /*buscando en la misma
altura de la primera columna, veo si encuentro el segundo error*/
            {
                mom_norm=vec_salida[i]; /*lo encuentro, lo guardo*/
                break; /*dejo de buscar, ya lo encontré*/
            }
        }
    }

    /*CALCULO MOMENTO EXTRA QUE ES NECESARIO*/
    Momento=407*mom_norm; /*la ganancia es de 407, en
simulador era 400*/
    /*reparto el momento necesario equitativamente entre ejes*/

    Mz_d=0.5*Momento;
    Mz_t=0.5*Momento;

    /*CALCULO PAR EXTRA NECESARIO CADA RUEDA*/
    Par_di=Mz_d*(r_d/t_d)+Jw_d*acel_di; /*Motor delantero izquierdo*/
    Par_ti=Mz_t*(r_t/t_t)+Jw_t*acel_ti; /*Motor trasero izquierdo*/
    Par_dd=Mz_d*(r_d/t_d)+Jw_d*acel_dd; /*Motor delantero derecho*/
    Par_td=Mz_t*(r_t/t_t)+Jw_t*acel_td; /*Motor trasero derecho*/

    /*APLICACION PAR DIFERENCIAL*/
    DI=1;TI=1;DD=1;TD=1; /*Si no hay error, o es
despreciable, no habra par diferencial*/
    if ((yaw_r-yaw_d)>0)
    {
        if((yaw_r>0) && (yaw_d>0)){
            DI=-1;TI=-1;DD=1;TD=1;}
        if ((yaw_r<0) && (yaw_d<0)){
            DI=1;TI=1;DD=-1;TD=-1;}
        if ((yaw_r>0) && (yaw_d<0)){
            DI=1;TI=1;DD=-1;TD=-1;}
        if ((yaw_r>0) && (yaw_d==0)){
            DI=-1;TI=-1;DD=1;TD=1;}
    }

```

```

}
if ((yaw_r-yaw_d)<0)
{
    if((yaw_r>0) && (yaw_d>0)){
        DI=1;TI=1;DD=-1;TD=-1;}
    if ((yaw_r<0) && (yaw_d<0)){
        DI=-1;TI=-1;DD=1;TD=1;}
    if ((yaw_r<0) && (yaw_d>0)){
        DI=-1;TI=-1;DD=1;TD=1;}
    if ((yaw_r<0) && (yaw_d==0)){
        DI=-1;TI=-1;DD=1;TD=1;}
}

/*ADICION PAR ADICIONALES CON SU SIGNO*/
Par_di=Par_cond+DI*Par_di;
Par_ti=Par_cond+TI*Par_ti;
Par_dd=Par_cond+DD*Par_dd;
Par_td=Par_cond+TD*Par_td;

/*CALCULO PAR MAXIMO TRANSMISIBLE SEGUN MTTE*/
Par_diMAX=(Jw_d/(0.9*m*pow(r_d,2))+1)*r_d*F_di;
Par_tiMAX=(Jw_t/(0.9*m*pow(r_t,2))+1)*r_t*F_ti;
Par_ddMAX=(Jw_d/(0.9*m*pow(r_d,2))+1)*r_d*F_dd;
Par_tdMAX=(Jw_t/(0.9*m*pow(r_t,2))+1)*r_t*F_td;

/*SELECCION PAR A APLICAR, SERA EL MINIMO DE: CALCULADO, RESULTADO MTTE,
MAXIMO PAR DEL MOTOR*/
Par_DI=sign(Par_di)*min(fabsf(Par_di),fabsf(Par_diMAX),fabsf(Par_max));
Par_TI=sign(Par_ti)*min(fabsf(Par_ti),fabsf(Par_tiMAX),fabsf(Par_max));
Par_DD=sign(Par_dd)*min(fabsf(Par_dd),fabsf(Par_ddMAX),fabsf(Par_max));
Par_TD=sign(Par_td)*min(fabsf(Par_td),fabsf(Par_tdMAX),fabsf(Par_max));

/*para evitar pares negativos*/
if (Par_DI<=0) Par_DI=0;
if (Par_TI<=0) Par_TI=0;
if (Par_DD<=0) Par_DD=0;
if (Par_TD<=0) Par_TD=0;
printf("%f %f %f %f\n",Par_DI,Par_TI,Par_DD,Par_TD);

/*LIBERO MEMORIA DE LOS DATOS, NO PUEDE ESTAR EN EL BUCLE*/
free(vec_yaw);free(vec_slip);free(vec_salida);
}

/*FUNCION PARA CALCULAR MINIMO DE TRES NUMEROS*/
float min(float x, float y, float z)
{
if (x < y)
if (x < z) return x;
if (y < z) return y;
return z;
}

/*FUNCION PARA VER EL SIGNO DE UN NUMERO*/
int sign(float v)
{
return v > 0 ? 1 : (v < 0 ? -1 : 0);
}

```

9.4 Código C para QNX

Esta parte consiste únicamente en cambiar el nombre de las variables de mi código para adaptarlo al código que se ocupa del reparto de par del FOX, de manera que no aporta nada nuevo de cara al desarrollo de este trabajo, pues es repetir código pero con diferente nombre, así que no se mostrará el código.

El código que hay que tocar del FOX tiene dos partes bien diferenciadas, una primera en la que se inicializan las variables, y que es donde vamos a hacer que se cargue la tabla de datos, y luego una segunda parte que es un bucle que se repite hasta que se llegue al final del experimento. En este bucle se va a incluir todo el cálculo del par. Una vez se ha acabado el experimento, se libera la memoria de la tabla.

Por tanto, el pseudocódigo es el siguiente:

```
Inicializo variables
Abro archivo con la tabla del controlador de lógica difusa
Hago reserva dinámica de memoria para la tabla
Cargo la tabla
Cierro el archivo que contenía la tabla
Mientras la variable fin (global) valga cero
    Recibo datos del coche
    Cálculos
    Pido acceso a memoria compartida
    Asigno par calculado a cada rueda
    Dejo libre el acceso a memoria compartida
Vuelco las variables medidas en un archivo para su posterior análisis
Borro memoria reservada dinámicamente para la tabla
```

10 CONCLUSIONES Y TRABAJO FUTURO

10.1 Conclusiones

Para concluir este trabajo, se resumirá todo lo visto durante el mismo:

- El primer controlador, el cual estaba basado en lógica difusa y usaba referencias obtenidas de datos del modelo del vehículo a baja velocidad, actúa tanto en el régimen permanente como en el transitorio de las diferentes maniobras realizadas, salvo en las trayectorias rectas (obviando las transiciones de curvas a rectas), en las cuales el conductor tiene el control total sobre el mismo.

Además, al diseñar este controlador se han tenido en cuenta las limitaciones reales del vehículo FOX, para intentar obtener unos resultados en simulación lo más parecidos posibles a los datos que se obtendrán al probarlo en el vehículo en la realidad.

Por último, tras haber visto todos los resultados de las simulaciones, se concluye que este controlador mejora principalmente el ángulo de deslizamiento tanto en valor como en suavidad, y no es capaz de obtener resultados tan buenos con el yaw rate.

- El segundo controlador mostrado se basaba en un control proporcional, y calculaba las referencias a seguir en función de las ecuaciones de la dinámica del vehículo.

En cuanto a su actuación, cabe destacar que el controlador entra en juego principalmente en los transitorios de cada maniobra, lo cual es bueno pues permite al conductor tener el control absoluto del vehículo durante el resto de la maniobra, pero modifica la dinámica del vehículo lo suficiente para que sea más fácil de conducir.

Para terminar con este controlador, se ve que mejoraba principalmente el ángulo de deslizamiento del vehículo, aunque también era capaz de mejorar el yaw rate, en menor medida que esta otra variable pero más que el anterior controlador.

- A pesar de que cada modo ha calculado las referencias de las variables de control, ambas han resultado ser muy similares en los experimentos llevados a cabo. Además, en las simulaciones más bruscas, estas referencias se distanciaban mucho del comportamiento del vehículo, haciendo que la actuación fuese más agresiva y así se mejorase bastante. Sin embargo, en las simulaciones con trayectorias más suaves, las referencias prácticamente coincidían con los valores obtenidos al quitar los controladores. Con todo

esto, podemos decir que ambas referencias son buenas y son compatibles con diferentes tipos de conducción.

- Respecto al consumo energético, se ha observado que ambos controladores son capaces de reducir el par otorgado a las ruedas realizando la misma maniobra, lo cual se traduce en un ahorro de energía. Esta mejora de la eficiencia energética es muy importante, pues uno de los principales problemas actuales de los vehículos eléctricos es la autonomía de los mismos.

Por tanto, ambos controladores son buenos en simulación, pues no mejoran únicamente las variables controladas, sino todas las relacionadas con ellas, disfrutando de una mayor mejoría la aceleración lateral y un menor consumo energético.

10.2 Trabajo futuro

De cara a seguir trabajando en esta temática, se presentan los siguientes objetivos para seguir ahondando en esta temática:

- Depuración de errores del código de la ECU del FOX para poder ver resultados reales del primer controlador (actualmente se está probando este código y hay errores que deben arreglarse, como el cálculo de la velocidad a partir de la posición que da el GPS, la cual tiene un error máximo de un par de metros, suficiente para que el ángulo de deslizamiento no se calcule correctamente y, por tanto, la acción de control aplicada no sea la correcta).
- Codificación en lenguaje C del segundo controlador una vez funcione como se suponía el primer controlador, evitando así caer en los mismos fallos.
- Desarrollo de un controlador que minimice el consumo de energía del vehículo, usando para ello una caracterización energética de los motores en cuanto a pérdidas y hallando así el punto de funcionamiento óptimo de los mismo en función del par demandado.
- Comprobación de los efectos de variar el porcentaje del momento que se debe generar en cada eje una vez se tiene el momento global que se debe aplicar. Actualmente se está generando la mitad en cada eje, pero esta proporción podría variar dinámicamente según si se detecta sobreviraje o subviraje.
- Estimación del coeficiente de deslizamiento de las diferentes superficies por las que puede circular el vehículo.

REFERENCIAS

- [1] D. Marcos Rodríguez, «Contributions to power management and dynamics control in hybrid vehicles.», Sevilla, 2014.
- [2] D. Limón Marruedo, «Control predictivo de sistemas no lineales con restricciones: estabilidad y robustez», Sevilla, 2001.
- [3] F. Rodríguez Rubio y M. J. López Sánchez, «Control Adaptativo y Robusto», Sevilla, 1996.
- [4] G. J. Klir y T. A. Folger, *Fuzzy Sets, Uncertainty and Information*, Prentice Hall, 1988.
- [5] V. Zanten, «Bosch ESP systems: 5 years of experience,» *SAE Technical Paper*, nº 2000-01-1633, 2000, DOI:10.4271/2000-01-1633, 2000.
- [6] J. Park, H. Jeong, I. G. Jang y S.-H. Hwang, «Torque Distribution Algorithm for an Independently Driven Electric Vehicle Using a Fuzzy Control Method,» *Energies*, nº 8, pp. 8537-8561, DOI:10.3390/en8088537, 2015.
- [7] B. L. Boada, M. J. L. Boada y V. Díaz, «Fuzzy-logic applied to yaw moment control for vehicle stability,» *Vehicle System Dynamics*, pp. 753-770 43:10, DOI: 10.1080/004231105001289984, 2005.
- [8] J. Kiumars, «Stability Control of Electric Vehicles with In-wheel Motors,» Universidad de Ontario, 2010.
- [9] J. Espinosa y J. Vandewalle, «Fuzzy modelling and identification, a guide for the user,» Katholieke Universiteit Leuven, 1997.
- [10] G. Hernández Rodríguez y C. Bordons Alba, «Control de Estabilidad basado en MPC para un Vehículo Eléctrico con Motores en Rueda,» Sevilla, 2015.
- [11] D. Yin, S. Oh y Y. Hori, «A Novel Traction Control for EV Based on Maximum Transmissible Torque Stimation,» *IEEE Trans. Ind. Electron.*, vol. 56, nº 6, pp. 2086-2094, 2009.
- [12] J. Bravo Riera y P. Luque Rodríguez, «Análisis del comportamiento dinámico en vehículos con tracción independiente a las cuatro ruedas,» Oviedo, 2014.
- [13] J. Yoon, S. Yim, W. Cho, B. Koo y K. Yi, «Design of an unified chassis controller for rollover prevention, manoeuvrability and lateral stability,» *Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility*, nº 48:11, pp. 1247-1268, 2010.
- [14] S. Hallowell y L. Ray, «All-wheel driving using independent torque control of each wheel,» *American Control Conference, 2003. Proceedings of the 2003*, vol. 3, pp. 2590-2595, 2003.

- [15] M. M. EmranHasan, M. M. Ektesabi y A. Kapoor, «An Investigation into Differential Torque Based Strategies for Electronic Stability Control in a In-Wheel Electric Vehicle,» *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, n° 7, pp. 327-336, 2013.