



UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

DEPARTAMENTO DE INGENIERÍA MECÁNICA

PROYECTO FINAL DE GRADO

GRADO EN TECNOLOGÍAS INDUSTRIALES

**“Diseño de un estimador de transferencia de carga
de una furgoneta 4T/8,5T utilizando redes
neuronales”**

AUTOR: DON SERGIO IBISATE HERRAIZ

TUTORAS: DOÑA DRA. BEATRIZ LÓPEZ BOADA

DOÑA DRA. M^a JESÚS LÓPEZ BOADA

JUNIO 2015



ÍNDICE:

Contenido:

CAPÍTULO 1: INTRODUCCIÓN.....	10
1.1 OBJETIVOS DEL PROYECTO.....	10
1.2 ESTRUCTURA DEL PROYECTO	11
CAPÍTULO 2: VEHÍCULOS PESADOS	14
2.1 PROBLEMÁTICA DE VUELCO.....	14
2.2 SISTEMA DE SUSPENSION	16
2.2.1 Funciones del sistema de suspensión	16
2.2.2 Elementos que forman el sistema de suspensión.....	17
2.2.3 Tipos de sistemas de suspensión	21
2.3 TRANSFERENCIA DE CARGA	22
2.4 CALCULO DE LA TRANSFERENCIA DE CARGA NORMALIZADA.....	24
2.4.1 Alternativas para el cálculo de la transferencia de carga	25
CAPÍTULO 3: REDES NEURONALES	30
3.1 REDES NEURONALES BIOLÓGICAS	30
3.2 REDES NEURONALES ARTIFICIALES	32
3.2.1 Redes preceptrón multicapa	34
CAPÍTULO 4: TRUCKSIM	42
4.1 SIMULACIONES EN TRUCKSIM	43
4.2 MANIOBRAS SIMULADAS EN TRUCKSIM.....	43
4.2.1 Proceso de diseño de las maniobras	43
4.2.2 Double Lane Change.....	53
4.2.3 J-Turn.....	53
4.3 ANÁLISIS DE LOS RESULTADOS OBTENIDOS EN LAS SIMULACIONES.....	54
4.3.1 Maniobras de J-Turn	55
4.3.2 Maniobras de Double Lane Change (DLC).....	57
CAPÍTULO 5: APRENDIZAJE DE LA TRANSFERENCIA DE CARGA UTILIZANDO REDES NEURONALES EN MATLAB	60
5.1 REDES PERCEPTRON MULTICAPA.....	62
5.1.1 Creación de la red	62
5.1.2 Simulación de la red neuronal.....	62
5.1.3 Entrenamiento de la red neuronal.....	63



5.2 ANÁLISIS DE LOS RESULTADOS OBTENIDOS TRAS EL ENTRENAMIENTO DE LA RED NEURONAL	65
5.2.1 Comparativa de resultados en función del algoritmo de entrenamiento de la red neuronal	65
5.2.2 Comparativa entre los errores cometidos en el entrenamiento de la red neuronal.	69
5.2.3 Graficas asociadas a las simulaciones en Matlab.....	72
CAPÍTULO 6: CONCLUSIONES Y FUTUROS TRABAJOS	76
6.1 Conclusiones.....	76
6.2 Futuros trabajos	77
CAPÍTULO 7: BIBLIOGRAFÍA.....	80



Índice de figuras:

Figura 2.1: Amortiguador de fricción.....	18
Figura 2.2: Elementos que componen un amortiguador hidráulico monotubo.....	19
Figura 2.3: Comparativa amortiguador Monotubo VS Bitubo.....	20
Figura 2.4: Ballesta.....	21
Figura 2.5: Muelle helicoidal.....	21
Figura 2.6: Cojinetes neumáticos.....	21
Figura 2.7: Suspensión Rígida	21
Figura 2.8: Suspensión Semi-Independiente	21
Figura 2.9: Suspensión Independiente	22
Figura 2.10: Modelo de transferencia lateral de carga.....	23
Figura 2.11: Comportamiento del vehículo para valores de transferencia de carga a partir de 1	26
Figura 2.12: Comportamiento del vehículo para valores de transferencia de carga comprendidos entre 0 y +1	26
Figura 2.13: Comportamiento del vehículo para valores de transferencia de carga de 0.....	27
Figura 2.14: Comportamiento del vehículo para valores de transferencia de carga comprendidos entre 0 y -1	28
Figura 2.15: Comportamiento del vehículo para valores de transferencia de carga a partir del valor de -1	28
Figura3.1: Red neuronal biológica	30
Figura 3.2: Red neuronal biológica y sinapsis	31
Figura 3.3: Esquema de funcionamiento de una neurona.....	33
Figura 3.4: Esquema de la arquitectura de una red neuronal	33
Figura 3.5: Función de activación lineal.....	36
Figura 3.6: Función de activación sigmoideal	36
Figura 3.7: Función de activación tangente hiperbólica	36
Figura 4.1: Vehículo de trabajo con Trucksim.....	42
Figura 4.2: Vista general del menú principal de Trucksim	44
Figura 4.3: Selección del número de ejes de la unidad principal del vehículo	44
Figura 4.4: Elección del tipo de vehículo	45
Figura 4.5: Elección del procedimiento de maniobra	47



Figura 4.6: Elección de categoría y título de la maniobra.	47
Figura 4.7: Menú de diseño de los parámetros específicos de la maniobra	48
Figura 4.8: Definición del tipo y título del giro	48
Figura 4.9: Definición del tiempo y valor de ángulo de giro	49
Figura 4.10: Elección tipo y características de la carretera	49
Figura 4.11: Elección plots “Vehicle Motion	50
Figura 4.12: Elección plots “Tire Forces”.	50
Figura 4.13: Elección plots “Steering”.	51
Figura 4.14: Menú principal de las simulaciones.....	52
Figura 4.15: Menú resultados variables de estudio de la maniobra simulada	52
Figura 4.16: Maniobra J-Turn.....	55
Figura 4.17: Comienzo de vuelco asociado a la pérdida de adherencia rueda calzada en una de las maniobras desechadas	56
Figura 4.18: Vuelco total asociado a la pérdida de adherencia rueda calzada en una de las maniobras desechadas.....	56
Figura 4.19: Maniobra Double Lane Change.....	57
Figura 4.20: Situación de inestabilidad y pérdida de adherencia rueda calzada en DLC a 140 km/h, invasión de línea divisora de carril	58
Figura 5.1: Red neuronal con una única capa, R elementos de entrada y S neuronas.....	60
Figura 5.2: Proceso de ajuste de pesos en el aprendizaje supervisado.	61
Figura 5.3: Arquitectura de la red neuronal implementada	64
Figura 5.4: Comparativa salidas objetivo vs salidas obtenidas por algoritmo Levenber-Marquardt	65
Figura 5.5: Comparativa salidas objetivo vs salidas obtenidas por algoritmo de Regularización Bayesiana o automatizada	66
Figura 5.6: Comparativa salidas objetivo vs salidas obtenidas por algoritmo del Gradiente Conjugado Escalado	67
Figura 5.7: Histograma de errores durante la simulación.....	72
Figura 5.8: Evolución del error cuadrático medio durante la simulación.....	72
Figura 5.9: Evolución del entrenamiento durante la simulación.....	73
Figura 5.10: Regresión asociadas a las salidas en función de los distintos tipos de datos empleados	74



Índice de tablas:

Tabla 2.1: vehículos según tipo, que han intervenido en los accidentes acaecidos en el año 2013	15
Tabla 3.1: Comparativa Cerebro / Computador	32
Tabla 4.1: Rango de velocidades simuladas para la maniobra de Double Lane Change	53
Tabla 4.2: Rango de velocidades y ángulos simulados para la maniobra de J-Turn.....	54
Tabla 5.1: Comparativa de errores cometidos en el entrenamiento de la red neuronal en función de el algoritmo empleado y la maniobra simulada	69
Tabla 5.2: Comparativa errores en función algoritmo de entrenamiento y paquete de datos .	71



Glosario:

V_x	<i>Velocidad longitudinal [m/s].</i>
A_x	<i>Aceleración longitudinal [m/s²].</i>
A_y	<i>Aceleración lateral [m/s²].</i>
F_x	<i>Carga en los neumáticos en dirección longitudinal [N].</i>
$F_{z,l}$	<i>Carga vertical sobre los neumáticos izquierdos [N].</i>
$F_{z,r}$	<i>Carga vertical sobre los neumáticos derechos [N].</i>
$k_{u,i}$	<i>Resistencia a la rodadura.</i>
$\Phi_{u,i}$	<i>Ángulo de balanceo de la masa no suspendida [rad].</i>
$\Phi_{s,i}$	<i>Ángulo de balanceo de la masa suspendida.</i>
l_w	<i>Distancia del eje x a cada una de las ruedas del vehículo [m].</i>
m_u	<i>Peso de la masa no suspendida [kg].</i>
m_s	<i>Peso de la masa suspendida [kg].</i>
z_r	<i>Distancia desde el centro de la masa no suspendida al eje de rodadura.</i>
$h_{c,g}$	<i>Altura del centro de gravedad de la masa suspendida sobre el suelo [m].</i>
h_r	<i>Distancia del centro de la masa suspendida al eje de rodadura [m].</i>
h_u	<i>Altura del centro de gravedad de la masa no suspendida sobre el suelo [m].</i>
h	<i>Altura del engranaje.</i>
μ_x	<i>Coefficiente de fricción de la rueda.</i>
ψ	<i>Ángulo de guiñada [rad].</i>
$\dot{\psi}$	<i>Velocidad de guiñada [rad/s].</i>
$\dot{\Phi}$	<i>Velocidad de balanceo [rad/s].</i>
T	<i>Anchura efectiva del vehículo [m].</i>
g	<i>Aceleración de la gravedad [m/s²].</i>
H	<i>Matriz Hessiana.</i>
J	<i>Matriz Jacobiana.</i>
e	<i>Vector de errores obtenidos en el entrenamiento de la red neuronal.</i>
α	<i>Tasa de aprendizaje de la red neuronal.</i>
N	<i>Número de patrones.</i>



- $x(n)$ Vector de entradas a la red.
 $s(n)$ Vector salidas objetivo red neuronal.
 $y(n)$ Vector salida obtenida de la red neuronal.
 $w(n)$ Vector de pesos asociados.
 $F(n)$ Función de activación.
 g Gradiente.

Subíndices:

- l Lado izquierdo del vehículo
 r Lado derecho del vehículo
 x Componente longitudinal.
 y Componente lateral.
 z Componente vertical.
 r -rear Eje trasero del vehículo.
 f -front Eje delantero del vehículo.



CAPÍTULO 1: INTRODUCCIÓN

CAPÍTULO 1: INTRODUCCIÓN

La transferencia de carga es un elemento clave para el estudio de la seguridad de los vehículos automóviles, y especialmente en los vehículos pesados, ya que estos al tener el centro de gravedad más alto que los turismos convencionales son más susceptibles al vuelco. Por esta razón la investigación de nuevos sistemas para paliar esta circunstancia en la industria es constante. Se está trabajando en la implementación de nuevos y sofisticados sistemas para la mejora de la seguridad de los vehículos y en especial de los pesados.

Una de estas ramas de investigación es la de la mejora de la estabilidad de los vehículos mediante la limitación de la transferencia lateral de carga, que es la que origina la falta de estabilidad frente al balanceo. Se pretende limitar la transferencia lateral de carga mediante la investigación de nuevos y mejorados tipos de suspensiones principalmente y de otros elementos que participan activamente en el control de la estabilidad del vehículo.

Para el correcto diseño e implementación de estos nuevos sistemas es preciso conocer al detalle las características del vehículo en que van a ser implantados, además del comportamiento de este en situaciones adversas como giros pronunciados a altas velocidades o calzadas con poca adherencia. Con estos datos y también los obtenidos en situaciones ordinarias de la conducción del vehículo dispondremos de la información del comportamiento del vehículo, que es la clave a la hora de diseñar e innovar en nuevos elementos de seguridad.

Las redes neuronales son técnicas no paramétricas altamente implantadas en los diversos campos de la ciencia e ingeniería, por su alta capacidad para la resolución de problemas complejos, que las técnicas tradicionales son incapaces de resolver o es muy complicado.

Este tipo de redes son una forma de inteligencia artificial, que permiten emulando el comportamiento del cerebro humano obtener un modelo complejo que relaciona un conjunto de variables de salida con un conjunto de variables de entrada, es decir dados unos valores de entrada la red es capaz de predecir cuál será el valor de la salida, disponiendo de una serie de observaciones de las variables. Estas observaciones serán empleadas como patrón de entrenamiento de la red para que esta sea capaz de aprender y después predecir las salidas del modelo deseado. Las capacidades de la red dependerán enormemente de esta fase de entrenamiento.

1.1 OBJETIVOS DEL PROYECTO

El objetivo principal perseguido en este proyecto es el del diseño de un estimador de la transferencia de carga para una furgoneta 4T / 8,5T mediante el uso de redes neuronales con la herramienta Matlab.

Para la realización de esta tarea primero se calcularán una serie de parámetros necesarios para la creación y entrenamiento de la red, estos parámetros serán resultado de la simulación de una serie de maniobras realizadas por la furgoneta en la herramienta Trucksim, esta herramienta nos arrojará una serie de valores deseados de una serie de variables necesarias para el cálculo

de la transferencia de carga a partir de la simulación mencionada anteriormente, con estos valores calcularemos la transferencia de carga para cada instante de tiempo y cada maniobra.

Tras la obtención de los parámetros deseados entrenaremos la red neuronal en Matlab mediante las entradas obtenidas en Trucksim y con los valores de salida calculados en el cálculo de la transferencia de carga, y esta a partir de este entrenamiento será capaz de aprender y después de predecir o estimar el valor de la transferencia de carga para cualquier maniobra que pertenezca al rango de maniobras estudiado.

1.2 ESTRUCTURA DEL PROYECTO

Este proyecto consta de siete capítulos, el primero de ellos es en el que nos encontramos ahora, dispone de una introducción al mismo y tras esto se divide en dos partes, los objetivos del proyecto anteriormente mentados y la estructura del proyecto en el que describimos brevemente el contenido de cada capítulo.

El segundo capítulo versa sobre los vehículos pesados, concretamente la furgoneta que será estudiada y se definen las características principales de los vehículos pesados, este consta de cuatro apartados, el 2.1 en el cual hablaremos de la problemática de vuelco asociada a este tipo de vehículos. En el apartado 2.2 explica y define los sistemas de suspensión haciendo especial hincapié en la importancia de estos y sus funciones principales y elementos. El apartado 2.3 introduce y explica el concepto de transferencia de carga, y finalmente en el 2.4 se ven reflejadas las ecuaciones empleadas para el cálculo de la transferencia de carga normalizada.

El capítulo tercero de este proyecto versa sobre las redes neuronales, consta de un primer apartado el 3.1 en el cual se expone el funcionamiento de las redes neuronales biológicas y el 3.2 explica el diseño, funcionamiento de las redes neuronales artificiales, arquitectura, así como sus usos y características principales, además se habla del tipo de red que usaremos durante el transcurso de este proyecto, las redes perceptrón multicapa, las cuales serán definidas y explicado su funcionamiento y utilidades.

El cuarto capítulo trata las simulaciones con el programa ingenieril Trucksim, consta de tres apartados, en el primero 4.1 se explican las características principales de la herramienta, el apartado 4.2 expone las maniobras realizadas dentro de la herramienta, como se crearon y simularon y sus características principales y las variables de estudio. En el apartado 4.3 se explicarán los resultados obtenidos en función de la maniobra realizada y después se evaluarán estos resultados.

El capítulo quinto versa sobre las redes neuronales artificiales en la herramienta Matlab, está dividido en 3 apartados, en el primero 5.1 se explica cómo se procede para la creación, entrenamiento y simulación de la red neuronal. El capítulo 5.2 se habla sobre la implementación de la red neuronal, haciendo especial hincapié en el código empleado para la creación de la red neuronal. En el 5.3 analizaremos los resultados obtenidos tras la creación, simulación y entrenamiento de la red, y los evaluaremos comparándolos con los resultados objetivo en función del algoritmo de entrenamiento empleado, prestaremos especial atención al valor del error medio cuadrático obtenido por cada método de cálculo.



El penúltimo capítulo, el sexto, expone las conclusiones obtenidas tras la realización del proyecto y enumera una serie de posibles futuros trabajos a partir de este.

Finalmente el último capítulo, el séptimo enumera la bibliografía consultada para la realización de este proyecto.



CAPÍTULO 2: VEHÍCULOS PESADOS

CAPÍTULO 2: VEHÍCULOS PESADOS

Se define vehículo pesado como aquel tipo de vehículo que permite el transporte de personas o mercancías además de poseer y cumplir una de las siguientes características definidas por el Real Decreto RD 818/2009 [9481] en el que se aprueba el Reglamento General de Conductores, existiendo distintas limitaciones en los vehículos autorizados a conducir en función del tipo de permiso que posea el conductor:

1. El vehículo sea de más de 3.500 Kg de M.M.A. (Masa máxima autorizada)
2. Disponer de más de 9 plazas incluyendo la del conductor.

En este proyecto, dentro de la gran variedad de vehículos pesados que existen nos centraremos en las furgonetas.

En la edición de 2013 de la Reglamentación Sobre Vehículos Pesados, Prioritarios, Especiales, de Transporte de Personas y Mercancías y Tramitación Administrativa de la DGT (Dirección General de Tráfico) se define furgoneta como aquel vehículo de cuatro ruedas o más, concebido y construido para el transporte de mercancías, cuya cabina está integrada en el resto de la carrocería, y con un máximo de 9 plazas incluyendo la del conductor. Será considerado un vehículo pesado cuando su M.M.A. sea superior a 3.500 kilogramos.

2.1 PROBLEMÁTICA DE VUELCO

La gran envergadura de los vehículos pesados los hace muy susceptibles al vuelco, dado que su centro de gravedad está a una altura elevada. Esta problemática sumando al enorme uso de este tipo de medios de transporte tanto para el transporte de personas como para el de mercancías han provocado un trabajo constante en medidas y elementos activos para prevenir el vuelco, mejorar la seguridad y reducir el número de accidentes en este tipo de vehículos.

Pese a ser uno de los medios de transporte con menor siniestralidad, el vuelco de un vehículo pesado al ser los que mayor número de personas transporta suele implicar un mayor número de víctimas y un mayor impacto social y mediático, además tras el vuelco debido a su gran masa y envergadura es un peligro mayor para el resto de usuarios de la vía.

Tabla 2.1: vehículos según tipo, que han intervenido en los accidentes acaecidos en el año 2013. **Fuente:** Anuario estadístico de accidentes de 2013, DGT.

PEATÓN Y TIPO DE VEHÍCULO	NUMERO DE VEHICULOS TOTAL GENERAL		
	Total	Con un solo vehículo	Con dos o más vehículos
Peatones	12.665	12.068	596
Vehículos de dos o tres ruedas:			
Bicicleta o triciclo sin motor	6.208	1.588	4.620
Ciclomotor	7.574	1.899	5.675
Coche de minusválido	12	5	7
Moto	20.701	5.741	14.960
Vehículos ligeros:			
Turismos S.P. hasta 9 plazas	1.368	237	1.131
Turismos otros	101.957	21.333	80.624
Ambulancia	97	17	80
Máquina obras y agrícola	105	31	74
Tractor agrícola	283	74	209
Furgoneta	8.093	1.585	6.508
Camiones de -3.500 Kg.	1.964	297	1.667
Vehículos pesados:			
Camiones de + 3.500 Kg.	1.874	314	1.560
Vehículos articulados	1.078	256	822
Bus escolar	45	6	39
Otros bus	1.795	704	1.091
Otros vehículos:			
Tren	46	14	32
Carro	11	2	9
Otros	1.428	390	1.038
No especificados	365	58	307
TOTAL VEHÍCULOS	155.004	34.551	120.453

Como se puede observar en la tabla 2.1 los vehículos pesados pese a ser los más sensibles a vuelco y otras circunstancias que los hacen más peligrosos a priori que los turismos convencionales, apenas producen o se vieron involucrados en un 3% de los accidentes que sucedieron durante el año 2013, debido a la especialización de los conductores de este tipo de vehículos y a las medidas técnicas incorporadas a estos.

A pesar de esta circunstancia, se sigue trabajando e investigando para mejorar la seguridad de este tipo de servicios. Uno de los elementos claves para la prevención de vuelcos es el sistema de suspensión.

2.2 SISTEMA DE SUSPENSION

Para poder prevenir los vuelcos, el vehículo en cuestión ha de ser capaz de resistir los momentos de vuelco generados por las cargas y la falta de adherencia en la calzada o firme. El principal y más importante elemento antivuelco del vehículo es el sistema de suspensión, dado que el funcionamiento de este sistema con los muelles y amortiguadores que lo componen generan momentos entre las masas suspendidas (elementos soportados por el chasis del vehículo) y las no suspendidas (resto de elementos del vehículo) como respuesta a los movimientos de balanceo de las masas, lo cual hace del sistema de suspensión un elemento crítico en el comportamiento frente a vuelco de cualquier vehículo, y en los pesados aún más ya que se encarga de enlazar ambas masas.

2.2.1 Funciones del sistema de suspensión

El sistema de suspensión de cualquier vehículo tiene dos funciones principales. La primera es la de absorber las reacciones producidas en las ruedas al pasar sobre algún terreno con irregularidades, evitando que se transmitan a la carrocería y así asegurando el confort de los pasajeros y conductor del vehículo. La segunda función es la de asegurar el contacto de las ruedas con el suelo para evitar vuelcos, y obtener una estabilidad y direccionalidad óptima del vehículo que permita tener un alto grado de seguridad, tanto para el conductor como para el resto de ocupantes del vehículo en cualquier tipo de desplazamiento.

La absorción de las reacciones debidas a las irregularidades del terreno se realizará de la siguiente manera: las reacciones producidas por pequeñas irregularidades del terreno son absorbidas por los neumáticos, debido a las características elásticas de estos. Por otro lado las reacciones producidas por irregularidades mayores serán absorbidas por los elementos elásticos del sistema de suspensión. A la hora del diseño de una suspensión se ha de prestar especial atención a que no se produzcan balanceos excesivos en la suspensión para lo cual se incorporan los amortiguadores que se encargan de limitar y eliminar las oscilaciones producidas en los elementos elásticos de la suspensión.

El sistema de suspensión del vehículo también debe cumplir las siguientes funciones:

1. Soportar el peso y carga del vehículo
2. Control de la dirección del vehículo en todo momento.
3. Conservación de la adherencia rueda-calzada.
4. Resistir el par motor y de frenada.
5. Resistir las aceleraciones producidas en las curvas.
6. Asegurar la estabilidad de la frenada.
7. Asegurar la correcta alineación de las ruedas así como conservar el paralelismo entre ejes y la perpendicularidad del bastidor.
8. Absorber las oscilaciones producidas por la conducción del vehículo.
9. Mantenimiento de la altura óptima del vehículo.

Como se ha comentado en el apartado anterior el principal objetivo de las suspensiones es la de proporcionar la seguridad necesaria al vehículo para su correcto y óptimo funcionamiento. La mayor preocupación en los vehículos pesados es la prevención del vuelco, lo cual implica la necesidad de que los neumáticos estén en constante contacto con el suelo, se mide este contacto calculando la diferencia de desplazamiento entre ambos.

A continuación se describen los elementos que forman el sistema de suspensión.

2.2.2 Elementos que forman el sistema de suspensión

La suspensión de cualquier vehículo está compuesta fundamentalmente por dos tipos elementos:

- Elementos amortiguadores.
- Elementos elásticos.

2.2.2.1 Elementos amortiguadores

La función de los elementos amortiguadores como hemos citado anteriormente es la de limitar el movimiento oscilatorio de los elementos elásticos de la suspensión disipando las oscilaciones producidas en estos y mantener los neumáticos en contacto con el terreno. Los amortiguadores ejercen una fuerza de amortiguamiento directamente proporcional a la velocidad de compresión o extensión. A la relación existente entre la fuerza que ejercen y la velocidad de desplazamiento se le llama constante de amortiguamiento.

Básicamente son bombas de aceite, son dispositivos contruidos con un eje cromado y dos tubos de acero uno dentro del otro. El exterior denominado tubo de reserva está lleno de aceite y el interior se denomina tubo de compresión. En uno de los extremos del amortiguador se ancla el vehículo y en el otro se monta un pistón, que se desplazará a lo largo del tubo de compresión, produciendo en función del tipo de desplazamiento del pistón que presione o succione el aceite

a través de las válvulas instaladas en el tubo de compresión, generando dos fuerzas distintas, extensión o compresión, con el fin de:

- Conseguir que el la adherencia entre el vehículo y la calzada nunca se vea reducido a cero (Nunca se despegue la rueda).
- Aportación de seguridad en las curvas.
- Realización de desplazamientos confortables.

Existen distintos tipos de amortiguadores:

- Amortiguadores de fricción, actualmente en desuso. Consisten en dos brazos unidos mediante un tornillo y un disco de fricción. El principal inconveniente de este tipo de amortiguadores es que mientras no se supere la fuerza de fricción el vehículo carece de suspensión. Cuando se ha superado dicha fuerza, la fuerza de amortiguación disminuye con la velocidad, comportamiento contrario al deseado. Además requieren el cambio de piezas con bastante frecuencia por el elevado desgaste que sufren. (Figura 2.1)



Figura 2. 1: Amortiguador de gas

- Amortiguadores hidráulicos, son amortiguadores cilíndricos telescópicos. Constan de un pistón que trabaja dentro de un cilindro en el que hay aceite. En este pistón se disponen una serie de orificios y válvulas precomprimidas que permiten el paso del aceite de una parte a otra del pistón al superar un valor umbral. Por los orificios existe un paso permanente de aceite, sin embargo las válvulas regulan este paso abriéndose o cerrándose en función de la presión existente. (Pueden observarse sus componentes en la figura 2.2) En uno de los extremos del amortiguador se ancla el vehículo y en otro el pistón. Son los amortiguadores más extendidos. Sus funciones principales son:
 - Conseguir que el la adherencia entre el vehículo y la calzada nunca se vea reducido a cero (Nunca se despegue la rueda).
 - Aportación de seguridad en las curvas.
 - Realización de desplazamientos confortables.

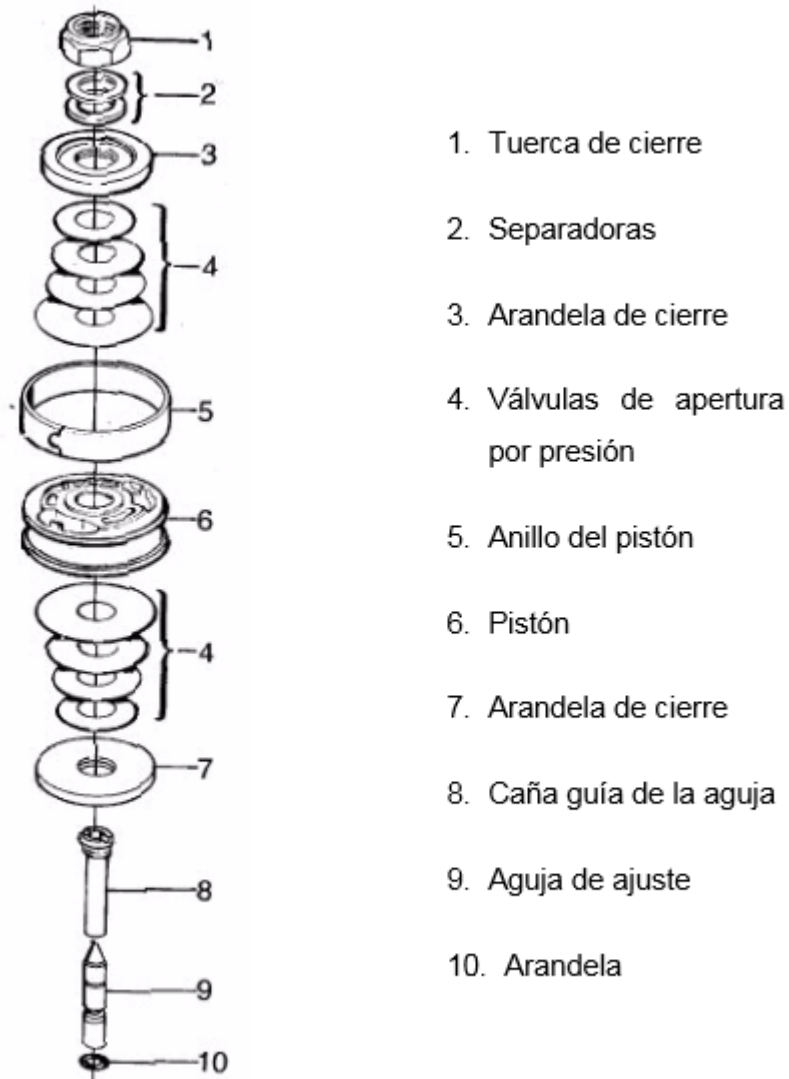


Figura 2.2: Elementos que componen un amortiguador hidráulico monotubo

Existen dos tipos de amortiguadores hidráulicos:

- Amortiguadores monotubo. (Figura 2.3).
- Amortiguadores bitubo. (Figura 2.3).

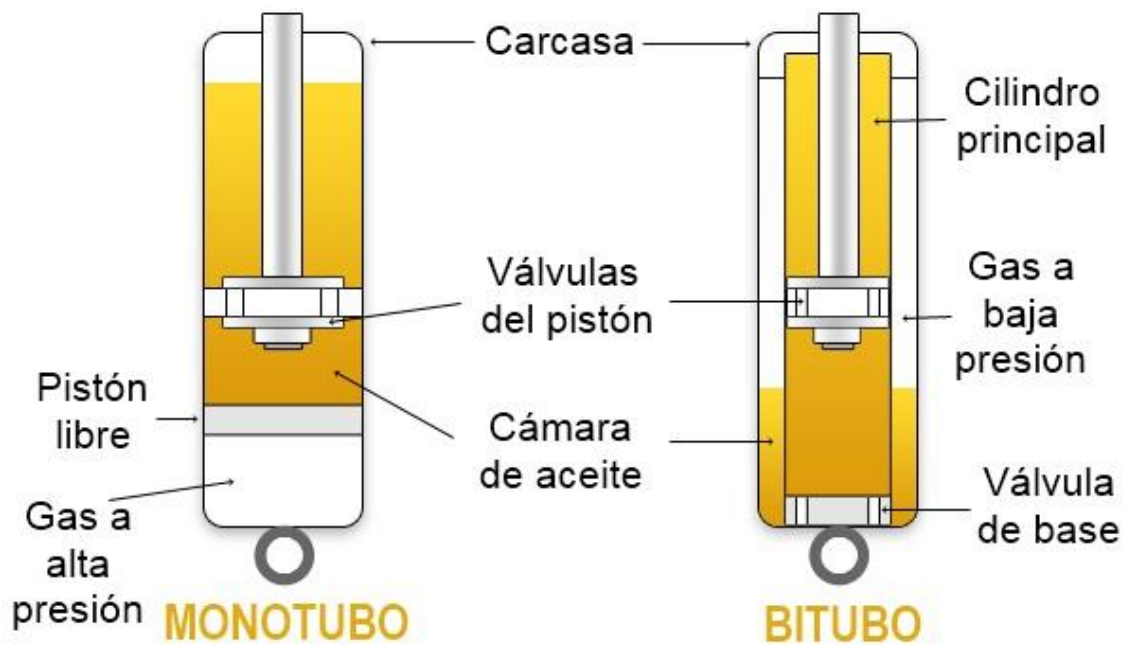


Figura 2.3: Comparativa amortiguador Monotubo VS Bitubo

2.2.2.2 Elementos elásticos

La función principal de los elementos elásticos de la suspensión es la de aislar a la masa suspendida de las irregularidades del terreno transmitidas por las ruedas. Su rigidez es un factor determinante, ya que ante elementos elásticos muy rígidos se obtiene una mayor capacidad direccional del vehículo, pero las irregularidades se verán amortiguadas en menor medida repercutiendo así en el confort de los ocupantes del vehículo en cuestión.

Se distinguen los siguientes elementos elásticos:

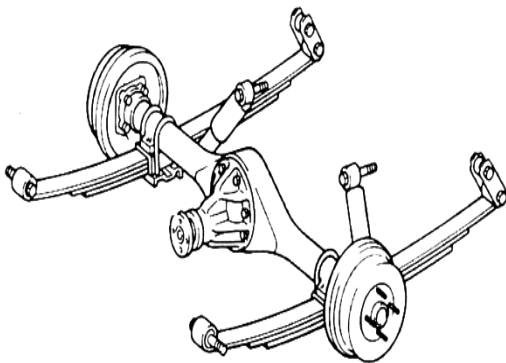
- Neumáticos.
- Ballestas, las cuales consisten en una serie de placas curvas unidas entre sí. (Figura 2.4).
- Muelles helicoidales. (Figura 2.5).
- Brazos torsionales.
- Barras estabilizadoras.
- Cojines neumáticos. (Figura 2.6).

**Figura 2.4:** Ballesta**Figura 2.5:** Muelle helicoidal**Figura 2.6:** Cojinetes neumáticos

2.2.3 Tipos de sistemas de suspensión

En función de la relación entre las ruedas del vehículo distinguiremos los siguientes tipos de sistemas de suspensión:

- Dependiente o rígida, caracterizado por que las ruedas derechas e izquierdas del vehículo están conectadas a un mismo eje sólido, por consiguiente al moverse una de las ruedas la opuesta del eje se moverá en sentido contrario. (Figura 2.7).
- Semi-Independiente, caracterizado por tener en cada rueda un movimiento independiente limitado, se transmite la acción de torsión al eje sólido de conexión. (Figura 2.8).
- Independiente, caracterizado por que cada rueda está conectada al vehículo de forma independiente, permitiéndose así que cada una tenga un movimiento diferente al de las otras. (Figura 2.9).

**Figura 2.7:** Suspensión Rígida**Figura 2.8:** Suspensión Semi-Independiente

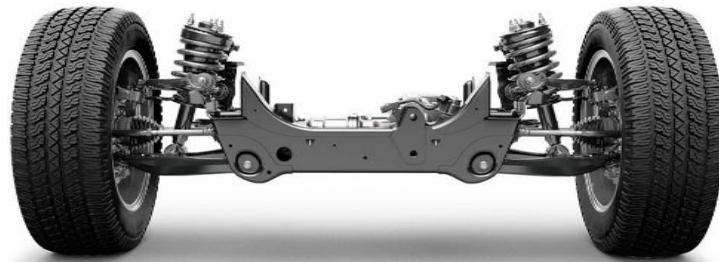


Figura2.9: Suspensión Independiente

También pueden dividirse en:

- Suspensiones pasivas, basadas fundamentalmente en el uso de un sistema muelle-amortiguador que trata de absorber las irregularidades del terreno. Ahora bien, este tipo de suspensiones no alcanzan resultados demasiado satisfactorios. Fundamentalmente porque para alcanzar un nivel óptimo de confort para el pasajero lo idóneo es el uso de una suspensión blanda que absorba prácticamente todas las irregularidades del terreno. Sin embargo a nivel de seguridad no es lo idóneo, ya que con suspensiones blandas se producirá un excesivo balanceo en curvas y además se favorece el cabeceo en las frenadas. Lo que puede resultar peligroso para la seguridad de los pasajeros. Es el sistema de suspensión más implantado debido a su menor costo.
- Suspensiones semi-activas, empleando sistemas regulados que permiten la variación de los mecanismos de suspensión y amortiguación a las necesidades del vehículo.
- Suspensiones activas, cuando se quiere exigir más al sistema de suspensión se recurre a este tipo de suspensiones, necesita de un sistema de suspensión regulable que actúe constantemente sobre cada rueda. Se compone de una serie de sensores y actuadores que hacen de este sistema un sistema complejo que ha de ser controlado por un sistema de control más complejo que para las pasivas.

2.3 TRANSFERENCIA DE CARGA

Se denomina transferencia de carga a la variación de las cargas en las ruedas de un vehículo determinado.

Cuando un vehículo realiza un giro en una curva se produce en él un movimiento de balanceo con respecto al eje longitudinal del mismo (eje x), este movimiento origina una transferencia de carga la cual provoca que las ruedas exteriores del vehículo sufran mayores cargas que las interiores.

Para conseguir estabilidad al balanceo se procede a limitar la transferencia lateral de carga por debajo de los límites que provoquen que las ruedas se despeguen del asfalto, ya que cuanto mayor sea esta mayor es la suma de las de las fuerzas laterales de las ruedas exteriores e interiores.

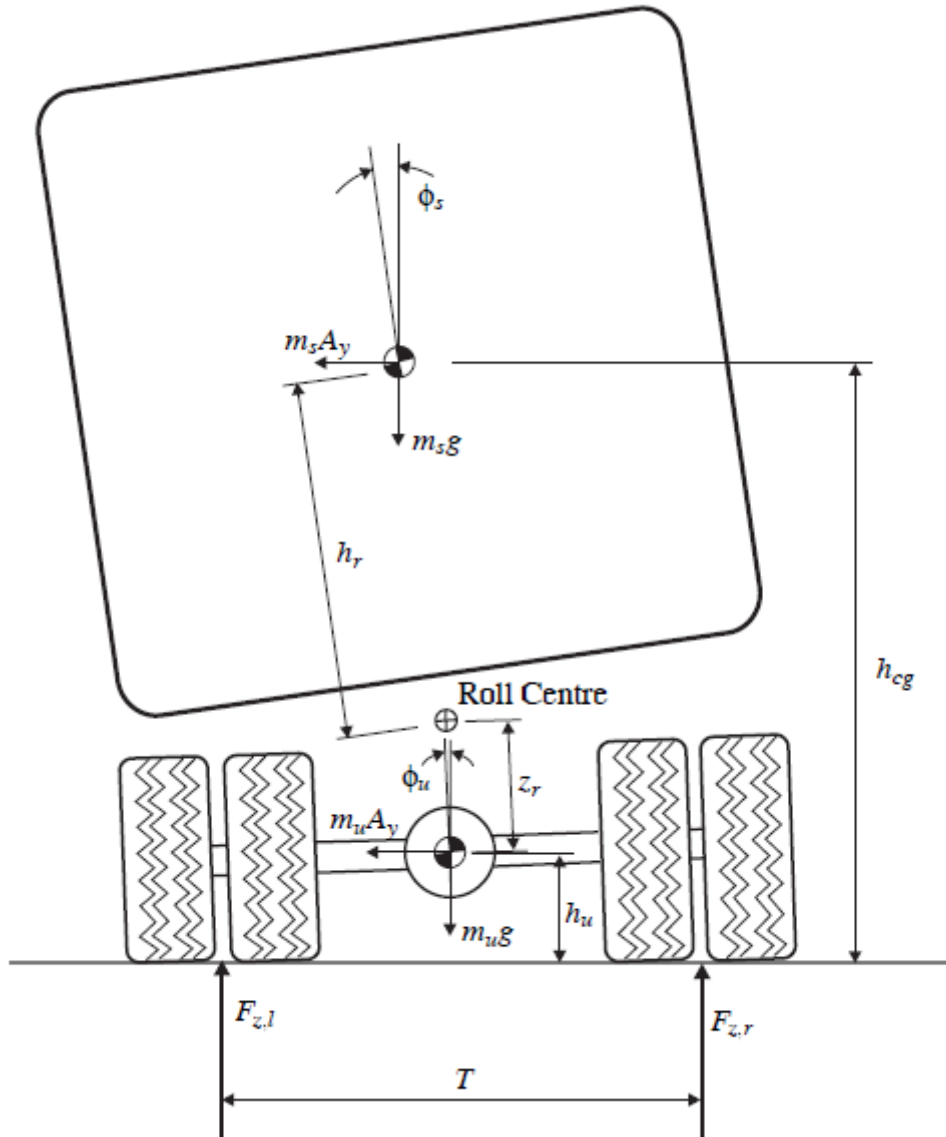


Figura 2.10: Modelo de transferencia lateral de carga. **Fuente:** Roll Dynamics and Lateral Load Transfer Estimation in Articulated Heavy Freight Vehicles: A Simulation Study.

2.4 CALCULO DE LA TRANSFERENCIA DE CARGA NORMALIZADA

Para determinar la transferencia de carga lateral en un vehículo se debe recurrir a la siguiente expresión, la cual representaremos de la siguiente manera:

$$\Delta F_{z,i} = \frac{k_{u,i} \Phi_{u,i}}{l_w} \text{ (ec 2.1)}$$

donde $k_{u,i}$ serán las constantes de rigidez de los amortiguadores del vehículo, $\Phi_{u,i}$ es el ángulo de balanceo de la masa no suspendida y l_w la distancia del eje x del vehículo a cada una de las ruedas. El subíndice i hace referencia a cada eje del vehículo, siendo $i = f$ para referirse al eje delantero del vehículo e $i = r$ para referirse al trasero del mismo.

En condiciones de vuelco del vehículo, la carga en las ruedas interiores se ve reducida a 0, transfiriéndose toda la carga a las ruedas exteriores. De acuerdo con esto la premisa fundamental para evitar vuelcos y conseguir estabilidad de balanceo es que las ruedas estén siempre en contacto con el suelo. Para poder conseguir que las ruedas no se despeguen del firme o calzada se deben obtener valores para la expresión de la transferencia de carga normalizada entre -1 y +1, el cálculo de la transferencia de carga normalizada se realizara mediante la siguiente expresión, la expresión general para el cálculo de la transferencia de carga normalizada:

$$R_i = \frac{\Delta F_{z,i}}{F_{z,i}} \text{ (ec 2.2)}$$

donde $\Delta F_{z,i}$ es el valor de la transferencia de carga lateral entre los ejes y $F_{z,i}$ es la carga total sobre el eje, l se refiere al lado izquierdo del vehículo y r al lado derecho del mismo. En este proyecto se calculará el valor de la transferencia de carga a partir de las fuerzas verticales en los neumáticos del vehículo elegido vehículo calculadas a partir de las simulaciones de Trucksim.

Aplicándose a este proyecto modelo calcularemos la transferencia de carga por eje mediante las siguientes expresiones, (según Roman Kamnik de la Universidad de Glasgow):

Para el eje delantero empleamos la siguiente ecuación:

$$R_f = \frac{F_{z,f,l} - F_{z,f,r}}{F_{z,f,l} + F_{z,f,r}} \text{ (ec 2.3)}$$

Para el eje trasero empleamos la siguiente ecuación:

$$R_r = \frac{F_{z,r,l} - F_{z,r,r}}{F_{z,r,l} + F_{z,r,r}} \text{ (ec 2.4)}$$

Finalmente se obtiene el valor final de la transferencia total de carga en el vehículo de estudio mediante la siguiente expresión:

$$R = \frac{R_r + R_f}{2} \text{ (ec 2.5)}$$

donde R_r es el valor de la transferencia de carga en el eje trasero del vehículo y R_f es el valor de la transferencia de carga en el eje delantero del vehículo.

Se calculará la transferencia de carga normalizada a partir de la fuerza vertical calculada con Trucksim en las ruedas, en dirección del eje z.

Si en algún momento alguno de los valores de la transferencia de carga normalizada llegase a +1 o -1 significaría que el vehículo ha perdido contacto con la calzada en uno de los neumáticos lo cual no significa que el vehículo vaya a volcar pero supone un importante riesgo de que se dé esta circunstancia. El valor de R variará de -1 en el despegue de la rueda izquierda, pasando por 0 cuando no haya transferencia de carga y tenderá a +1 en el despegue de la rueda derecha.

2.4.1 Alternativas para el cálculo de la transferencia de carga

Otra manera más simple de calcular la transferencia de carga en un vehículo es una aproximación del estado estacionario simplificado en términos de la aceleración lateral a_{ym} y la altura del engranaje h se realiza mediante la siguiente expresión:

$$R = 2 \frac{a_{ym} h}{g e_m} \text{ (ec 2.6)}$$

donde a_{ym} es la aceleración lateral y e_m es la media de anchura que viene dada por la siguiente expresión (ef será la anchura del eje delantero y er la del eje trasero del vehículo):

$$e_m = \frac{(e_f + e_r)}{2} \quad (ec\ 2.7)$$

Además de la manera descrita con anterioridad en las ecuaciones 4 y 5 también tenemos la posibilidad de calcular la transferencia de carga, de la siguiente manera:

La carga longitudinal en las ruedas se relaciona con la carga vertical a través de la fricción, el coeficiente μ_x , ya que $F_x = \mu_x F_z$ por tanto la ecuación 2 puede ser calculada alternativamente con la siguiente expresión:

$$R = \frac{\mu_{x,r} - \mu_{x,l}}{\mu_{x,r} + \mu_{x,l}} \quad (ec\ 2.8)$$

Las figuras 2.11, 2.12, 2.13, 2.14 y 2.15 muestran ejemplos del comportamiento del vehículo para los valores de transferencia de carga entre -1 y +1 respectivamente.



Figura 2.11: Comportamiento del vehículo para valores de transferencia de carga a partir de 1



Figura 2.12: Comportamiento del vehículo para valores de transferencia de carga comprendidos entre 0 y +1

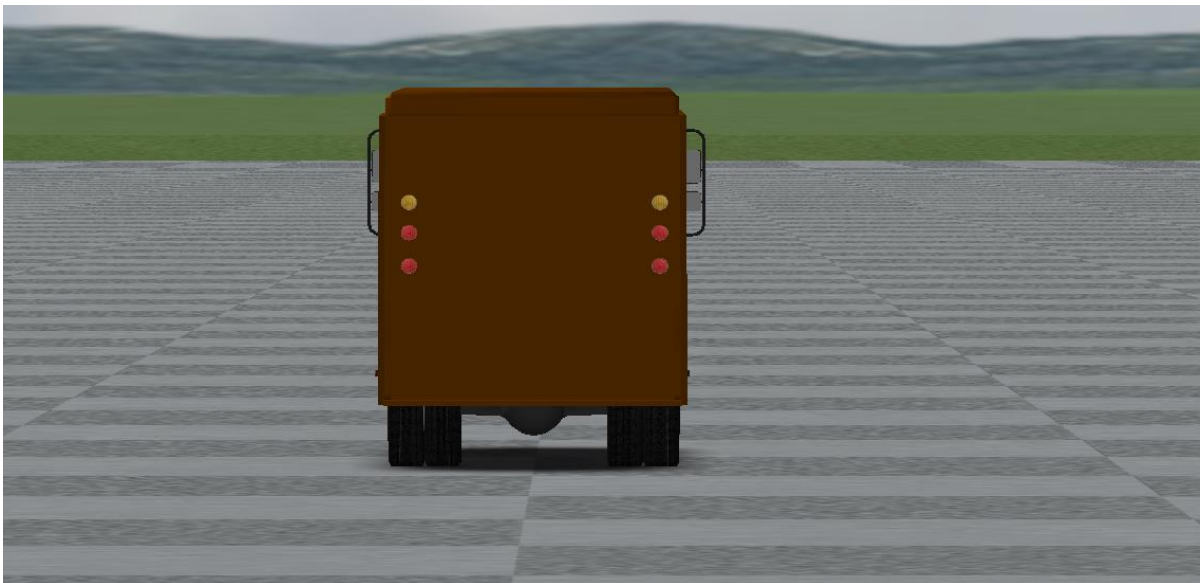


Figura 2.13: Comportamiento del vehículo para valores de transferencia de carga de 0



Figura 2.14: Comportamiento del vehículo para valores de transferencia de carga comprendidos entre 0 y -1



Figura 2.15: Comportamiento del vehículo para valores de transferencia de carga a partir del valor de -1



CAPÍTULO 3: REDES NEURONALES

CAPÍTULO 3: REDES NEURONALES

3.1 REDES NEURONALES BIOLÓGICAS

La red neuronal biológica o circuito neuronal es un conjunto de conexiones sinápticas ordenadas generada como resultado de la unión de las neuronas a otras en sus regiones correspondientes tras la migración neuronal.

El cuerpo humano posee órganos receptores de la información, reflejados en los 5 sentidos, mediante los cuales recibe la información del exterior. El sistema nervioso central recibe esta información, la analiza y procesa y se encarga de enviar impulsos nerviosos a glándulas o fibras musculares para la realización de determinada acción.

La unidad fundamental del sistema nervioso es la neurona. Las neuronas en nuestro organismo se unen unas a otras formando redes. Están compuestas por un núcleo central situado en el soma, un pericarion que alberga los orgánulos y neuritas (generalmente compuesto por un axón que es una ramificación de salida de la neurona y una gran cantidad de dendritas o ramificaciones de entrada).



Figura3.1: Red neuronal biológica

El funcionamiento fundamental de las neuronas es el siguiente: Un estímulo es captado por alguna región sensorial, este es conducido a través de las neuronas que se van conectando unas con otras mediante la sinapsis (zona de contacto e intercambio de información entre una neurona y otra o una neurona y una célula efectora), la información electro-química recogida en la sinapsis llega al núcleo de la neurona a través las dendritas, donde es procesada y se genera

una respuesta que será propagada por el axón, la respuesta a esta señal es conducida por las neuronas y ejecutada mediante una acción motora, que puede ser una contracción muscular o una secreción glandular.

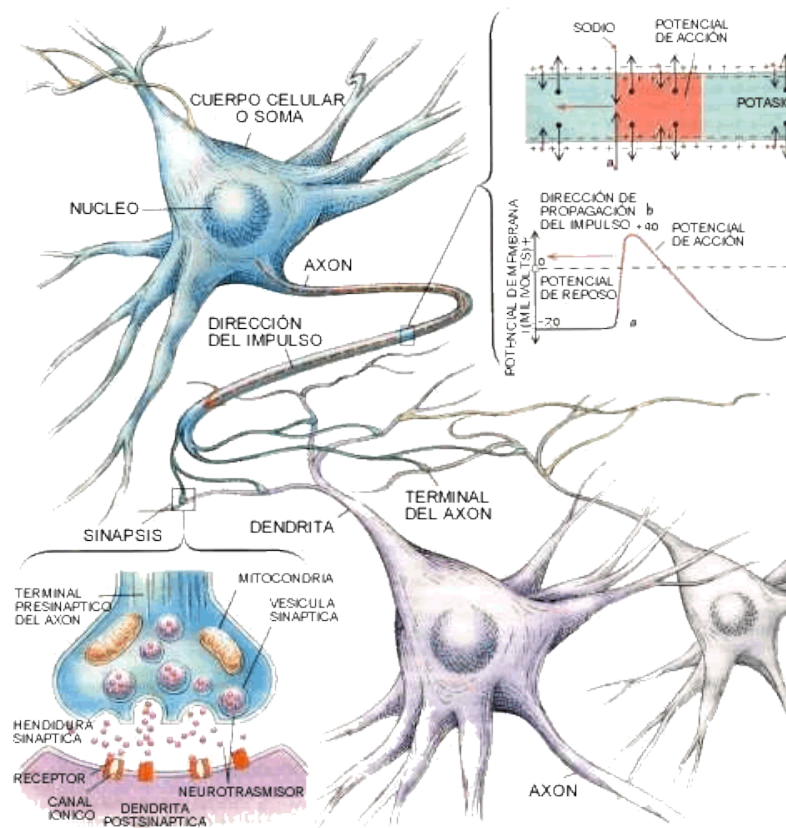


Figura 3.2: Red neuronal biológica y sinapsis.

El funcionamiento de las neuronas arroja una característica fundamental, la plasticidad neuronal por la cual en el espacio sináptico que es un espacio líquido en el cual existe una cierta concentración de iones, y posee unas características eléctricas que le permiten potenciar o inhibir a conveniencia la señal eléctrica que en él se transmite.

Se observa que el sistema neuronal es un conjunto de neuronas conectadas entre sí formando redes, que reciben elaboran y transmiten la información entre ellas o a glándulas o fibras musculares y el sistema es capaz de potenciar o inhibir la información que es transferida en función de la necesidad de realizar una determinada acción gracias a las propiedades del espacio intersináptico. Esta función llamada plasticidad neuronal es de vital importancia ya que poder alterar el peso de cada información en la red neuronal nos permitirá tener la capacidad de aprender y de desarrollar la memoria.

3.2 REDES NEURONALES ARTIFICIALES

Las redes neuronales artificiales están basadas en el funcionamiento del sistema neuronal humano. La función de las redes neuronales artificiales es la de imitar las características propiedades y funcionamiento en la medida de lo posible de las redes neuronales biológicas. Como en estas se componen de una serie de elementos, llamados neuronas también conectados entre sí.

Tabla 3.1: Comparativa Cerebro / Computador. **Fuente:** Martín del Brio & Sanz Molina, 2006.

	CEREBRO	COMPUTADOR
VELOCIDAD DE PROCESO	$\approx 10^{-2}$ seg. (100HZ)	$\approx 10^{-9}$ seg. (GHZ)
ESTILO DE PROCESAMIENTO	Paralelo	Secuencial
NÚMERO DE PROCESADORES	10^{11} - 10^{14}	Pocos
CONEXIONES	10000 por procesadores	Pocas
ALMACENAMIENTO DEL CONOCIMIENTO	distribuido	Direcciones fijas
TOLERANCIA A FALLOS	Amplia	Nula
TIPO DE CONTROL DEL PROCESO	Auto-organizado	Centralizado

Cada neurona recibe un valor de entrada, que transforma en función de una determinada función de activación. La señal transformada es la salida de la neurona.

Las neuronas se conectan entre si formando una arquitectura determinada, normalmente formando capas. Cada conexión tiene un peso específico que pondera cada entrada a la neurona. Así la entrada a una determinada neurona viene determinada por la suma de las salidas de todas las neuronas que estén conectadas a esta, multiplicadas por el peso de la respectiva conexión.

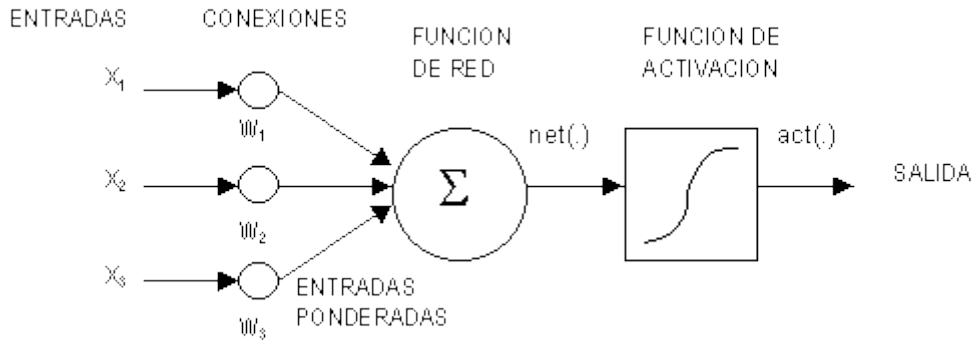


Figura 3.3: Esquema de funcionamiento de una neurona

Como se puede observar en la figura adjunta (Figura 3.3) la neurona recibe una serie de entradas x_1, x_2, \dots, x_n . Estas señales se multiplican por el peso asociado a su conexión con la neurona w_1, w_2, \dots, w_n . Tras este paso se suman estas entradas y se les aplica la función de activación que tras aplicarse generara la señal de salida asociada a esta neurona. Los valores de los pesos son ajustados durante la fase de aprendizaje de la red neuronal.

Las neuronas están interconectadas de acuerdo a una arquitectura determinada, se agrupan en capas, existen dos capas principales, la capa de entrada y la capa de existir, pero también pueden existir una serie de capas intermedias denominadas capas ocultas.

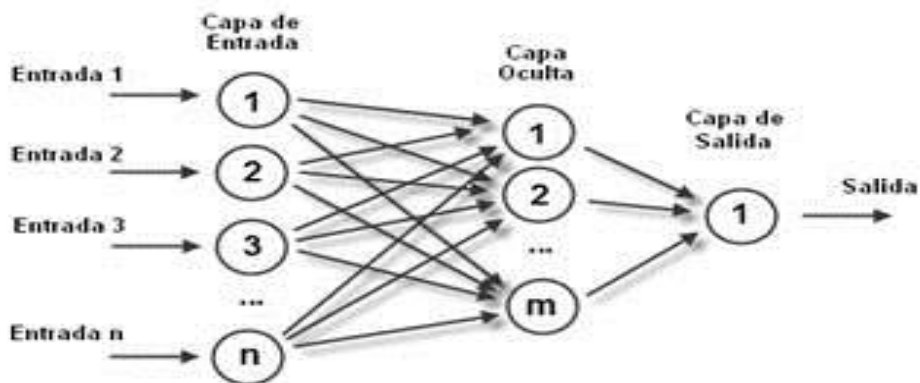


Figura 3.4: Esquema de la arquitectura de una red neuronal

Para ver cómo funciona una red neuronal ha de observarse la Figura adjunta 3.4, así pues las entradas a la red serán introducidas en las neuronas que forman la capa de entrada, que generan una salida tal cual se ha descrito anteriormente, esta señal de salida es propagada a las neuronas de la siguiente capa, de esta manera cada neurona de la segunda capa debe generar una salida de valor:

$$S_{2j} = F_{2j}(X_1 W_{1j}) \quad (ec. 3.1)$$

donde X_1 será el vector de las entradas de la capa 1 y W_{1j} es el vector de los pesos correspondientes a las conexiones que conectan todas las neuronas pertenecientes a la primera capa con la neurona j de la segunda capa. La función F_{2j} es la función de activación de la neurona j de la segunda capa. Este proceso se repetirá para todas las neuronas de la segunda capa. Las salidas generadas son propagadas hasta las neuronas que forman la capa de salida de la red neuronal. Estas generarán las salidas de la red, cada neurona generará una salida de valor:

$$S_{si} = F_{si}(W_{2i} S_2) \quad (ec. 3.2)$$

donde W_{2i} será el vector de pesos asociados a las conexiones de las neuronas de la segunda capa con la neurona i de la capa de salida y S_2 es el vector de salidas de las neuronas pertenecientes a la capa 2, que al mismo tiempo serán las entradas de las neuronas de la capa de salida de la red.

El aprendizaje de una red neuronal artificial consiste en hallar los valores precisos de los pesos de sus conexiones interneuronales para que esta sea capaz de resolver un determinado problema. En general el proceso consistirá en la introducción de una serie de datos patrón e ir ajustando los pesos de las conexiones en función de un determinado criterio. En este proyecto el ajuste del valor de dichos pesos se fundamentará en el error cometido por la red, lo que exige conocer a priori los resultados de salida de la red primaria. Se trata de un entrenamiento supervisado. Así pues el primer paso es la introducción de los patrones, con el correspondiente reajuste del valor de los pesos de las conexiones, y posteriormente se comprueba si la nueva salida se ajusta al criterio impuesto y a la salida buscada, de no ser así repetiremos todo el proceso.

3.2.1 Redes perceptrón multicapa

El perceptrón multicapa es una red neuronal artificial compuesta por múltiples capas, lo cual le permite resolver problemas linealmente dependientes, lo cual es su principal limitación. El perceptrón multicapa puede ser local o totalmente conectado. En el primer caso cada salida de una neurona de la capa " i " es entrada de todas las neuronas de la capa " $i+1$ ", mientras que en el segundo cada neurona de la capa " i " es entrada de una serie de neuronas de la capa " $i+1$ ".

3.2.1.1 Introducción

El perceptrón multicapa con conexiones hacia adelante es una generalización del perceptron simple. Surge como respuesta a los problemas que tenía la red perceptron simple, sobre todo el no poder resolver problemas no linealmente separables.

3.2.1.2 Arquitectura

La arquitectura de la red neuronal perceptron multicapa es característica por agrupar a todas las neuronas que la componen en distintos niveles denominados capas. Se distinguen los siguientes niveles:

- Primer nivel, correspondiente a la capa de entrada, su función es la de propagar las entradas recibidas por el resto de la red neuronal.
- Último nivel, correspondiente a la capa de salida, encargada de proporcionar los valores de salida de la red neuronal.
- Capas intermedias, denominadas capas ocultas, reciben las entradas de la capa de entrada y se realiza un procesamiento no lineal de los patrones recibidos y se entregarán los resultados a la capa de salida de la red neuronal.

Las conexiones de las redes perceptron multicapa son siempre hacia adelante, es decir normalmente todas las neuronas de un determinado nivel o capa se conectan a las neuronas de la capa o nivel inmediatamente posterior. También pueden darse casos de conexiones interneuronales de capas o niveles no consecutivos o inexistencia de conexión entre neuronas de niveles consecutivos, por lo cual el peso asociado a dicha conexión será cero.

Las funciones de activación usualmente empleadas por las redes neuronales de perceptron multicapa suelen ser las siguientes (**Fuente:** Capítulo3, Matlab Toolbox, Fundación Universitaria Konrad Lorenz):

- Función lineal (Figura 3.5):

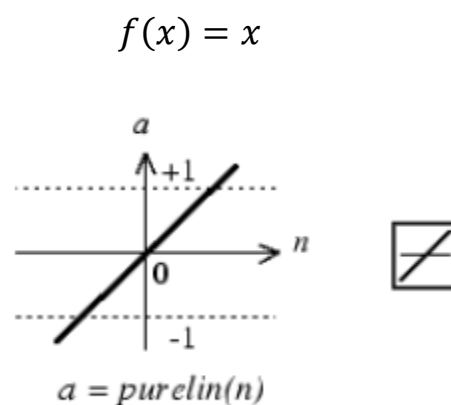


Figura 3.5: Función de activación lineal

- Función sigmoïdal (Figura 3.6):

$$f(x) = \frac{1}{1 + e^{-x}}$$

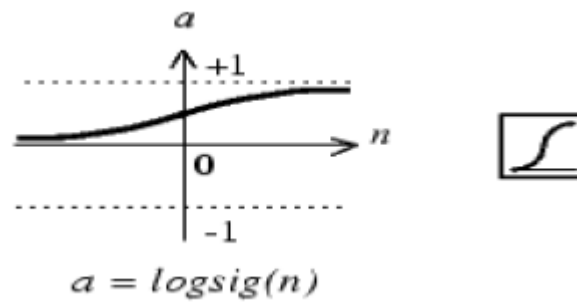


Figura 3.6: Función de activación sigmoïdal

- Función tangente hiperbólica (Figura 3.7):

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

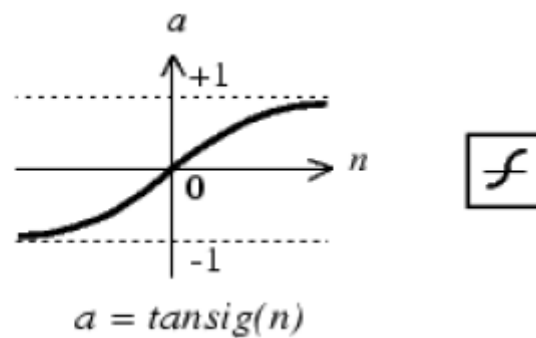


Figura 3.7: Función de activación tangente hiperbólica

3.2.1.3 Algoritmo de retropropagación

El algoritmo de retropropagación es el algoritmo básico de aprendizaje del perceptron multicapa. Mediante él se van adaptando los parámetros de la red, es un tipo de entrenamiento supervisado.

Se plantea el aprendizaje de la red neuronal como un problema de minimización de una determinada función de error, normalmente se emplea como función de error del aprendizaje el error medio cuadrático, expresado en la siguiente expresión (**Fuente:** Matlab's User Guide,):

$$E = \frac{1}{N} \sum_{n=1}^N e(n) \quad (ec\ 3.3)$$

donde $e(n)$ es:

$$e(n) = \frac{1}{2} \sum_{n=1}^t (s(n) - y(n))^2 \quad (ec\ 3.4)$$

donde; $s(n)$ es la salida objetivo e $y(n)$ es la salida obtenida de la red, t es el número de neuronas de salida y N el número de patrones.

Se trata de un problema no lineal y como tal el problema asociado a la minimización de la función de error será resuelta por técnicas de optimización no lineales, basándose en el ajuste paramétrico en función de una determinada dirección. En este caso la dirección elegida será la negativa del gradiente de la función de error. Aunque existen otras formas de resolución no basadas en este método.

Así pues existen dos opciones a partir de este paso, o bien cambiar los parámetros cada vez que introduzcamos un patrón, o cambiarlos cuando se hayan introducido todos los parámetros de entrenamiento por cada ciclo. Siendo así en el primer caso deberemos minimizar $e(n)$ y en el segundo la función E .

La variación asociada a la dirección del gradiente del error se expresará de la siguiente manera:

$$w(n) = w(n - 1) - \alpha \frac{\partial e(n)}{\partial w} \quad (ec\ 3.5)$$

El problema consiste en evaluar la derivada, el parámetro α es la tasa de aprendizaje de la red que influye directamente en la magnitud del desplazamiento en la superficie de la función de error.

Se evalúa el valor del gradiente para los pesos de las conexiones de la última capa oculta hasta la capa de salida de la red neuronal.

Sea $w_{ji}^{C-1}(n)$ el peso de la conexión j de la capa $C-1$ a la neurona i de la capa de salida de la red. El valor del error $e(n)$ viene dado por la ecuación 3.4, y se observa que solo se ve afectado el valor de este por w_{ji}^{C-1} en el error de salida de la neurona i . Por lo que:

$$\frac{\partial e(n)}{\partial w_{ji}^{C-1}} = -(s_i(n) - y_i(n)) \frac{\partial y_i(n)}{\partial w_{ji}^{C-1}} \quad (ec\ 3.6)$$

Por otro lado, la salida de la neurona i es igual a la suma de las entradas transformadas según su función de activación. Aplicando la regla de la cadena y teniendo en cuenta que w_{ji}^{C-1} solo afecta a la entrada de la neurona i porque va multiplicando a_j se tiene que:

$$\frac{\partial y_i(n)}{\partial w_{ji}^{C-1}} = f' \left(\sum_{j=1}^{n_{C-1}} w_{ji}^{C-1} + a_j^{C-1} + u_i^{C-1} \right) a_j^{C-1}(n) \quad (ec\ 3.7)$$

Se define δ asociado a la neurona i de la capa C del patrón n , $\delta_i^C(n)$ como:

$$\delta_i^C(n) = -(s_i(n) - y_i(n)) f' \left(\sum_{j=1}^{n_{C-1}} w_{ji}^{C-1} + a_j^{C-1} + u_i^{C-1} \right) \quad (ec\ 3.8)$$

De este modo el gradiente se expresara mediante la siguiente expresión:

$$\frac{\partial e(n)}{\partial w_{ji}^{C-1}} = \delta_i^C(n) a_j^{C-1}(n) \quad (ec\ 3.9)$$

Este cálculo es extensible para todos los pesos de las conexiones de las neuronas pertenecientes a la capa C-1 con las neuronas de la capa de salida de la red. Se obtienen de manera análoga los valores umbrales.

A continuación se calcula la magnitud de la variación del resto de conexiones existentes entre las capas ocultas de la red neuronal. Se procede de manera análoga a la realizada hasta ahora. Se obtendrá una ley de concurrencia para la modificación de dichos pesos.

Considérese el peso w_{kj}^{C-2} de la conexión de la neurona k de la capa C-2 con la neurona j de la capa C-1. El nuevo valor del peso vendrá expresado por una ecuación semejante a la ecuación 3.5. Solo se desconoce el valor del gradiente, y en este caso concreto w_{kj}^{C-2} influye en la totalidad de las salidas de la red, mediante la siguiente expresión:

$$\frac{\partial y_i(n)}{\partial w_{kj}^{C-2}} = f' \left(\sum_{j=1}^{n_{c-1}} w_{ji}^{C-1} + a_j^{C-1} + u_i^{C-1} \right) \frac{\partial a_j^{C-1}}{\partial w_{kj}^{C-2}} \quad (ec \ 3.10)$$

La salida de la neurona j, únicamente se ve afectada por w_{kj}^{C-2} que multiplica a a_k^{C-2} . Tras aplicar nuevamente la regla de la cadena se obtiene la siguiente expresión:

$$\frac{\partial a_j^{C-1}}{\partial w_{kj}^{C-2}} = f' \left(\sum_{i=1}^{n_{c-2}} w_{kj}^{C-2} + a_k^{C-2} + u_k^{C-2} \right) a_k^{C-2}(n) \quad (ec \ 3.11)$$

Se define el valor δ para las neuronas de la capa C-1 de la red de la siguiente manera:

$$\delta_j^{C-1}(n) = f' \left(\sum_k w_{kj}^{C-2} + a_k^{C-2} + u_j^{C-2} \right) \sum_{i=1}^{n_c} \delta_j^{C-1}(n) w_{ji}^{C-1} \quad (ec \ 3.12)$$

donde el valor de la derivada con respecto al peso w_{kj}^{C-2} será definido como:

$$w_{kj}^{C-2}(n) = w_{kj}^{C-2}(n-1) - \alpha \delta_j^{C-1}(n) a_k^{C-2}(n) \quad (ec \ 3.13)$$

Se observa que la modificación de los pesos de las conexiones entre las neuronas de las capas C-2 y las de la capa C-1 se ven afectadas por la salida de la neurona k de la capa C-2 y el término δ asociado a la neurona a la que llega la conexión. Solo varía la expresión de δ . A partir de este punto es posible la generalización de la actualización de cualquier peso perteneciente a cualquier capa de la red neuronal de acuerdo con la ecuación 3.13. Por esta razón la actualización de uno de los pesos de la conexión de la neurona k de la capa h con la neurona j de la capa $h+1$, estará definida por la siguiente expresión:

$$w_{kj}^h(n) = w_{kj}^h(n-1) - \alpha \delta_j^{h+1}(n) a_k^h(n) \quad (\text{ec 3.14})$$

donde el término δ se define por la siguiente ley de recurrencia:

$$\delta_j^{h+1}(n) = f' \left(\sum_k w_{kj}^h + a_k^h + u_j^{h+1} \right) \sum_{i=1}^{n_{h+2}} \delta_j^{h+2}(n) w_{ji}^{k+1} \quad (\text{ec 3.14})$$

El término δ propaga los errores obtenidos a la salida hacia atrás. Así cada neurona oculta recibe un cierto error o valor δ de todas las neuronas a las que se conecta, y la suma de todos los errores es el error total que comete esta neurona. Para actualizar los valores umbrales se procede de la misma manera obteniéndose la siguiente expresión:

$$u_j^{h+1}(n) = u_j^{h+1}(n-1) - \alpha \delta_j^{h+1}(n) \quad (\text{ec 3.15})$$

En este tipo de entrenamiento se define el número de épocas como el número de veces que se han comparado los ejemplos con las salidas de la red para el ajuste del valor de los pesos de las conexiones.



CAPÍTULO 4: TRUCKSIM

CAPÍTULO 4: TRUCKSIM

Trucksim es una herramienta empleada para la simulación mecánica de un determinado tipo de vehículos, en general se emplea para simulaciones mecánicas en camiones, furgonetas (Figura 4.1), pickup, autobuses, tractores, etc.

TruckSim predice el rendimiento de los vehículos en respuesta a las entradas de control del conductor (dirección, acelerador, frenos, embrague y cambio) en un entorno determinado (geometría de la carretera, los coeficientes de fricción, el viento). Por funcionamiento, nos referimos a movimientos de vehículos, fuerzas y momentos que participan en aceleración, manejo y frenado. Casi cualquier prueba de un vehículo que se lleve a cabo en una pista de pruebas o por carretera puede ser simulada.

Con esta herramienta es posible estudiar las variaciones en el comportamiento del vehículo resultantes de la modificación de cualquiera de los cientos de parámetros del vehículo, entradas de control, entorno de conducción. Pudiendo añadir elementos del vehículo tales como los controles ABS, control de tracción o control de estabilidad en el vehículo y que estos sean empleados para desarrollar los algoritmos de control del mismo.

Trucksim no incluye flexibilidad estructural, acústica, o vibraciones de alta frecuencia. Por lo cual no podrá ser utilizado para el estudio del ruido o deformaciones en la estructura de los vehículos.



Figura 4.1: Vehículo de trabajo con Trucksim

Tampoco se utiliza para simular cargas de componentes individuales para análisis de durabilidad, ya que en la mayoría de los casos los modelos TruckSim tienen ecuaciones internas para el comportamiento del sistema (por ejemplo, el movimiento total de la suspensión), en lugar del comportamiento de los componentes (por ejemplo, la barra de acoplamiento o del brazo de control de movimientos).

4.1 SIMULACIONES EN TRUCKSIM

En este proyecto se realizan dos tipos de maniobras con esta herramienta, se simula con el objetivo de observar y recoger los datos pertinentes del comportamiento de la furgoneta en la maniobra J-Turn y Double Lane Change, que se explican a continuación. Previo al estudio de las maniobras se procede a la selección del vehículo de estudio, en este caso una furgoneta de reparto 4T/8.5T como la mostrada en la Figura 4.1.

4.2 MANIOBRAS SIMULADAS EN TRUCKSIM

Como se ha mencionado anteriormente se realizan dos maniobras para la obtención de los parámetros vehiculares necesarios para la realización de este proyecto, en ambas maniobras se presta especial atención a los resultados de las siguientes variables, ya que son los que posteriormente serán necesarios para la implementación de la red neuronal y así obtener el estimador de la transferencia de carga deseado, estas variables son:

- Velocidad de balanceo (Roll Rate)
- Velocidad de guiñada (Yaw Rate)
- Velocidad en el eje X del vehículo V_x
- Aceleración en el eje X del vehículo A_x
- Aceleración en el eje Y del vehículo A_y
- Fuerzas verticales en los neumáticos en ambos lados del vehículo F_z

4.2.1 Proceso de diseño de las maniobras

Tal cual se mencionó con anterioridad, el primer paso a la hora del diseño de cualquier maniobra en Truksim seguiremos los siguientes pasos:

1. Elección del tipo de vehículo; para lo cual dentro del menú principal (Figura 4.2) del programa elegiremos tal cual se especifica en la figura adjunta 4.3 la opción de tipo de vehículo: “Unidad principal con 2 ejes (lead unit with two axles). A continuación elegiremos el tipo de vehículo entre todos los disponibles con 2 ejes en la herramienta, de la lista elegimos el Delivery: 4T/8.5T tal cual muestra la figura 4.4.

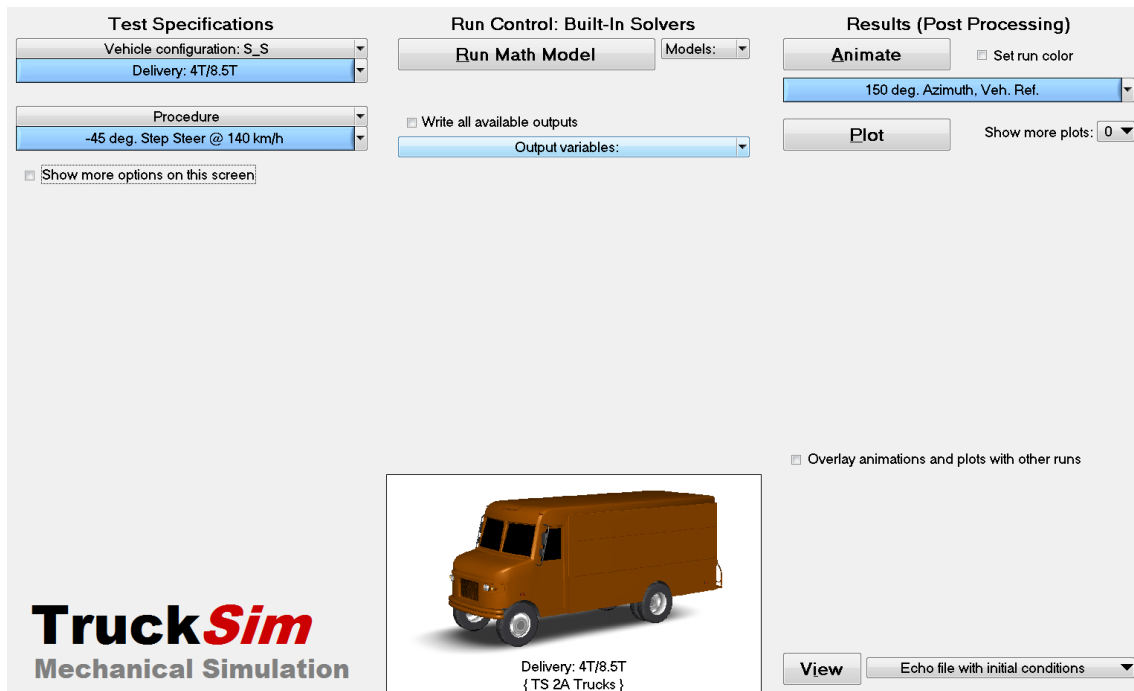


Figura 4.2: Vista general del menú principal de Trucksim

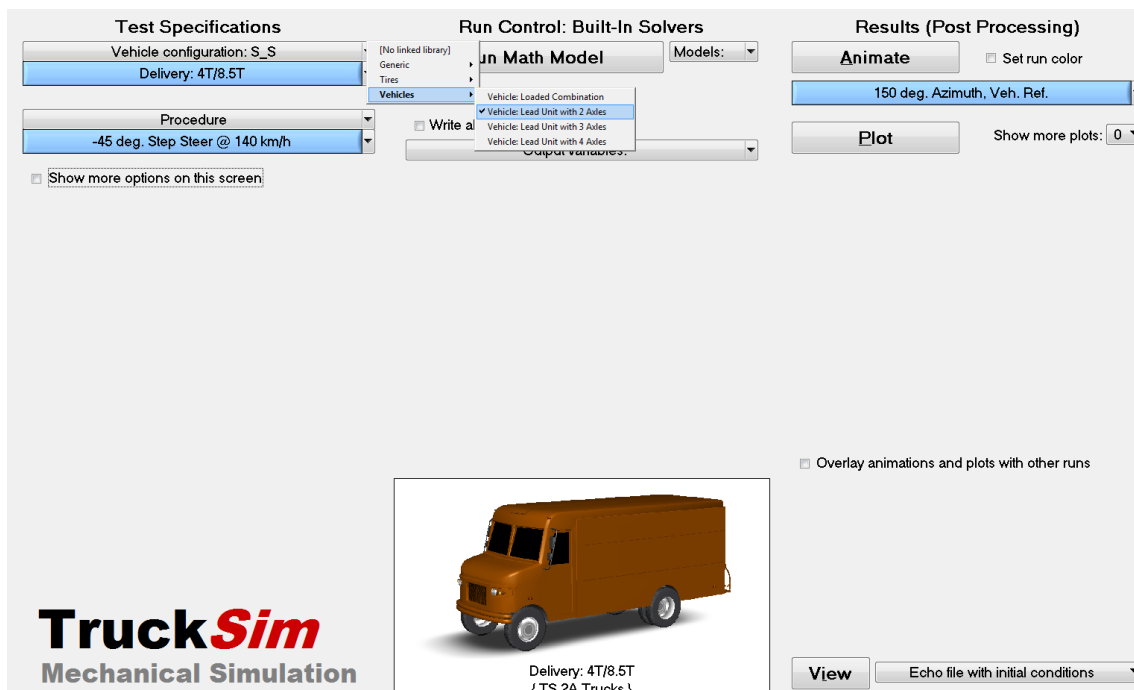


Figura 4.3: Selección del número de ejes de la unidad principal del vehículo

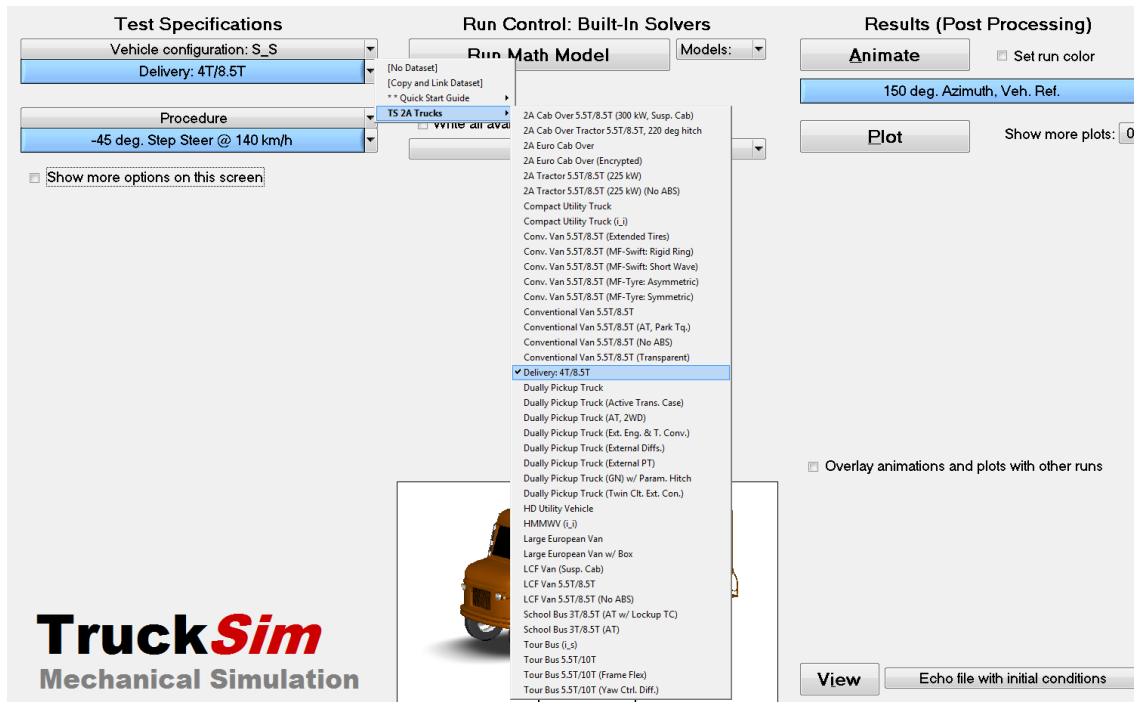


Figura 4.4: Elección del tipo de vehículo

2. Creación de la maniobra deseada; tras los pasos descritos en el paso 1, se procede a la crear la maniobra deseada, para lo cual se realiza lo siguiente, primero se procede a crear la maniobra pulsando en “[Copy and Link Dataset]” tal cual se muestra en la figura 4.5, una vez allí se elige la categoría y se le da el título deseado a la maniobra en cuestión como se muestra en la figura 4.6 (En este ejemplo se crea una maniobra de giro de -45 grados a 40 km/h), una vez creada se pulsa encima de ella y se accede al diseño de los parámetros específicos de la maniobra, tales como la carretera en la que se simulará la maniobra, el ángulo de giro, la velocidad, el tiempo de realización de la maniobra y las variables de estudio que son elegidas para esta maniobra, puede observarse este menú en la figura 4.7.

El primer paso en este menú es la elección de la velocidad a la que se desarrolla la maniobra, se modifica como se observa en la figura 4.7 eligiendo “Constant target speed” e introduciendo el valor deseado, a continuación se procede al diseño del ángulo de giro, para lo cual en la pestaña “Steering” se elige “Control” y dentro de este “Control: Steering (Open Loop)”, después se pulsa en la pestaña “No dataset selected” y allí en [Link to new dataset] y tal cual se muestra en la figura 4.8 se selecciona la categoría del nuevo dataset (en este ejemplo “Step Steer”) y se le otorga un nombre (en este caso “-45 deg. Step Steer”). Tras este paso se define el giro de -45 grados pulsando la pestaña con el nombre “-45 deg. Step Steer” y se accede al menú mostrado en la figura 4.9 en el que se

especificará el rango de tiempos en el que se desea que se produzca el giro modificando el valor de X Axis y el valor de giro deseado modificando el valor de Y Axis como se muestra en la figura mencionada.

Después de definir el giro y la velocidad se retorna al menú de la figura 4.7 para la elección de la carretera donde se realizará la simulación y las variables de salida que deseadas. Para la elección de la carretera se va a la pestaña “Misc” ahí se selecciona “Roads” y dentro de este “Road: 3D Surface”, una vez realizados estos pasos será elegido el tipo de carretera como se muestra en la figura 4.10 pinchando en “1 km Square” / “3D Grid” / “1 km Square”.

Tras elegir la carretera se procede a la selección de las variables de estudio que son necesarios para la creación y entrenamiento de la red neuronal, para ello se selecciona en las pestañas debajo de “Plot Definitions” mostrada en el menú de la figura 4.2.1-6, y tal cual se muestra en las figuras 4.11, 4.12 y 4.13 se eligen los siguientes plots, en “Vehicle Motion”:

- Ax – Long. Accel. Of CG’s.
- Ay – Lat. Accel. Of CG’s.
- Vx – Longitudinal Speed.
- Yaw Rate – Sprung Masses.
- Roll Rate (Body-fixed), Vehicle.

En “Tire Forces” se eligen:

- Fz – Vertical Tire Forces (Left Side).
- Fz – Vertical Tire Forces (Right Side).

Y finalmente en “Steering”:

- Steer_SW – Steering Wheel Angle.

Tras la elección de estos parámetros concluirá la creación de la maniobra deseada.

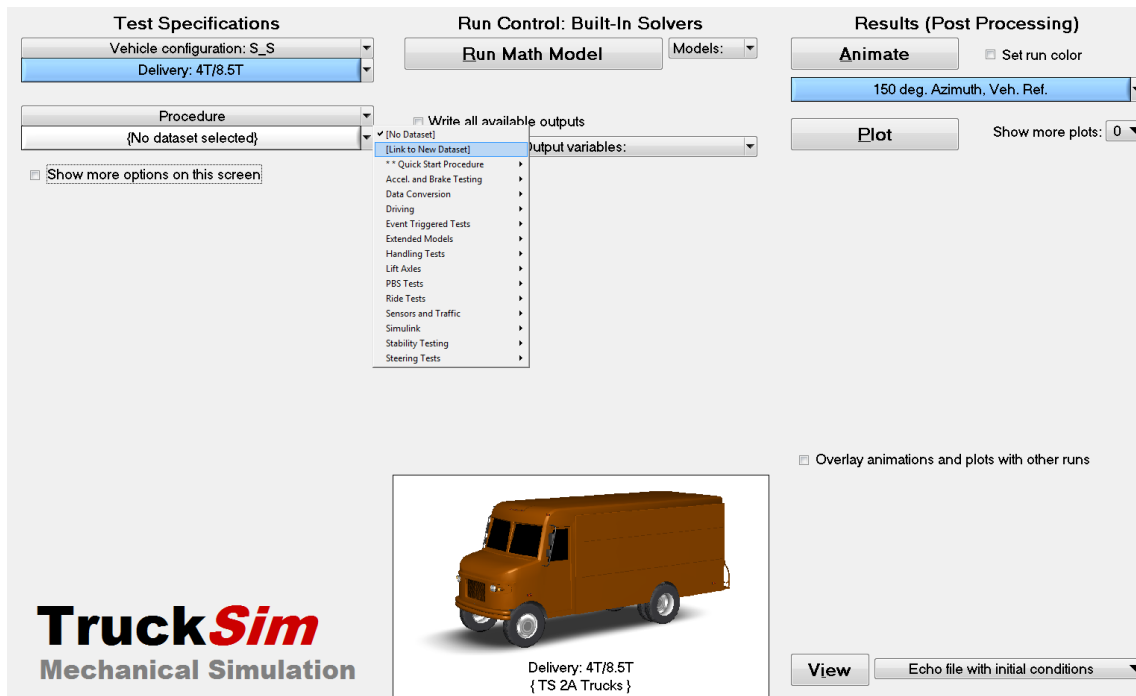


Figura 4.5: Elección del procedimiento de maniobra

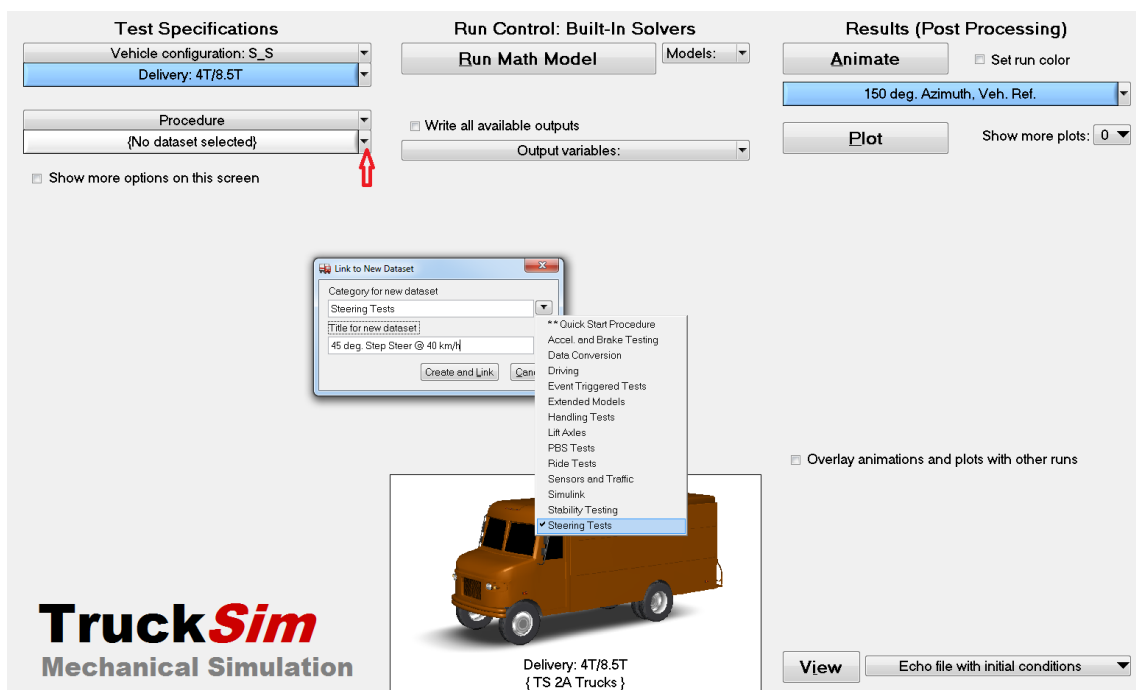


Figura 4.6: Elección de categoría y título de la maniobra

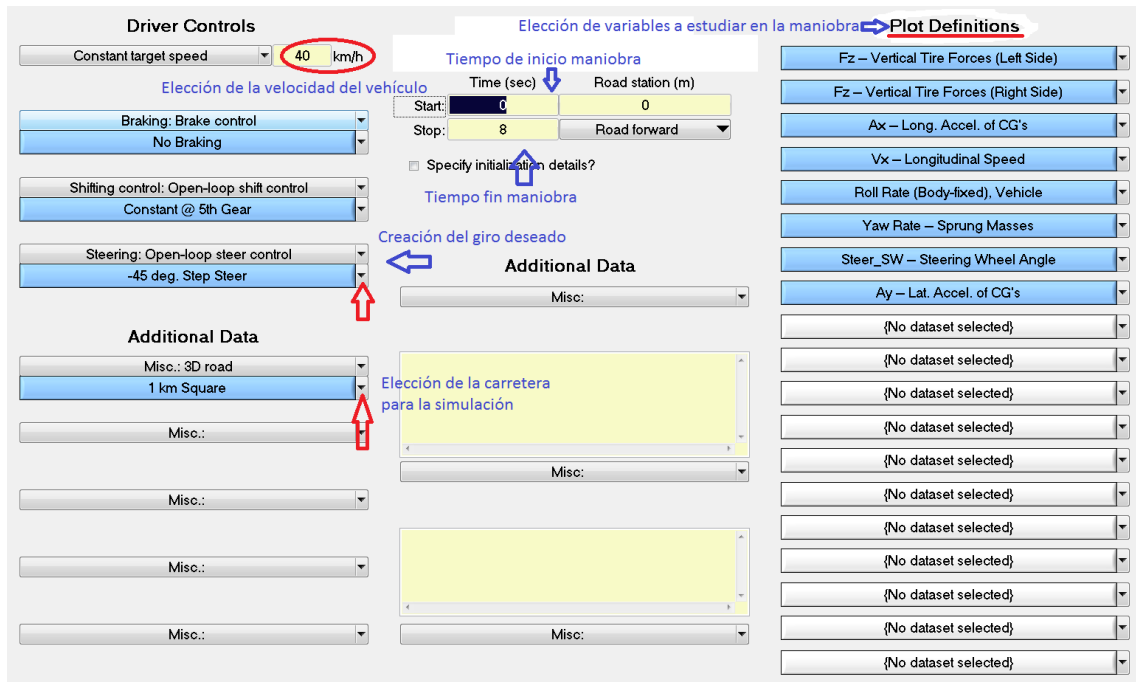


Figura 4.7: Menú de diseño de los parámetros específicos de la maniobra

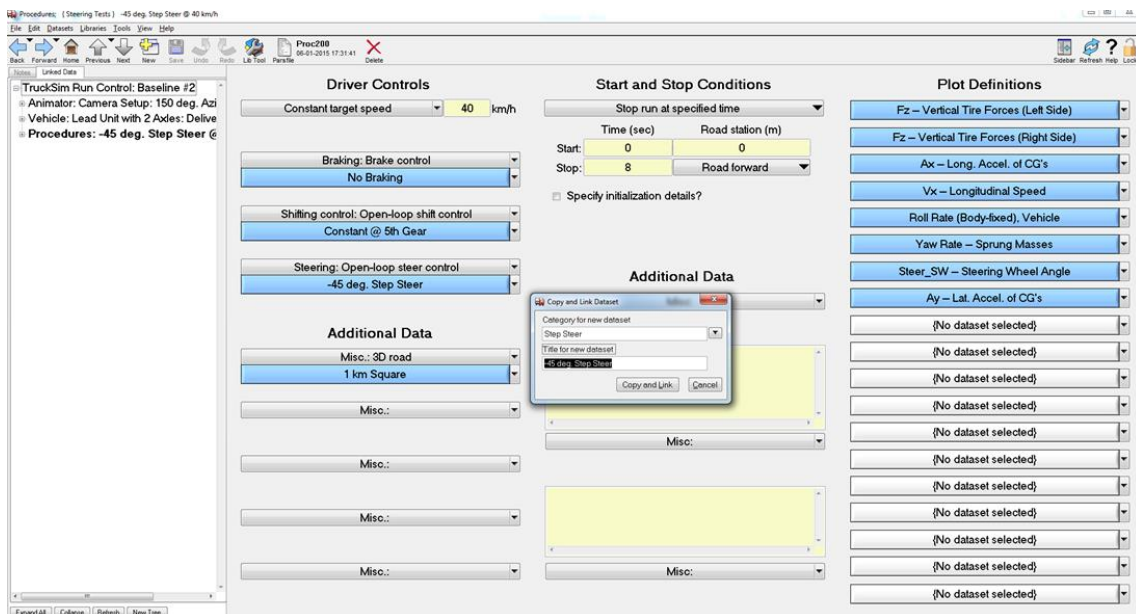


Figura 4.8: Definición del tipo y título del giro

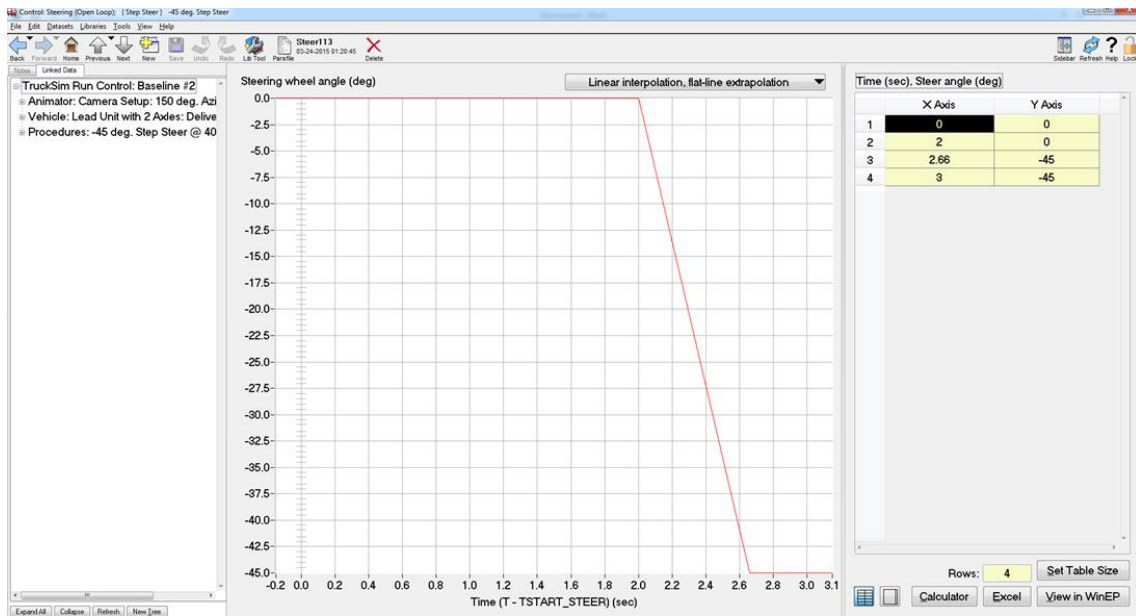


Figura 4.9: Definición del tiempo y valor de ángulo de giro

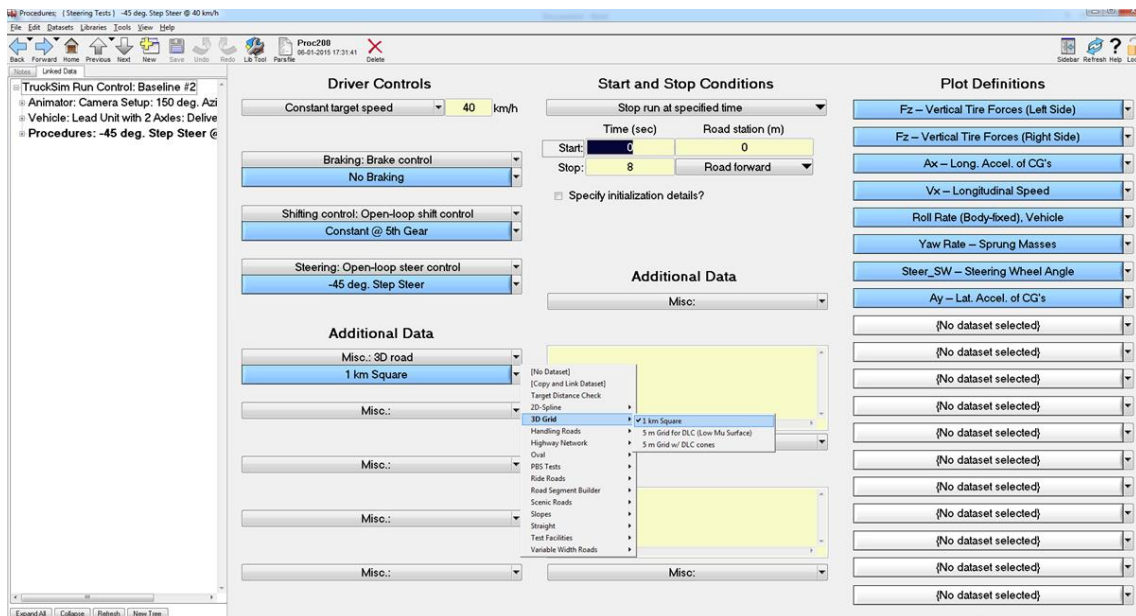


Figura 4.10: Elección tipo y características de la carretera

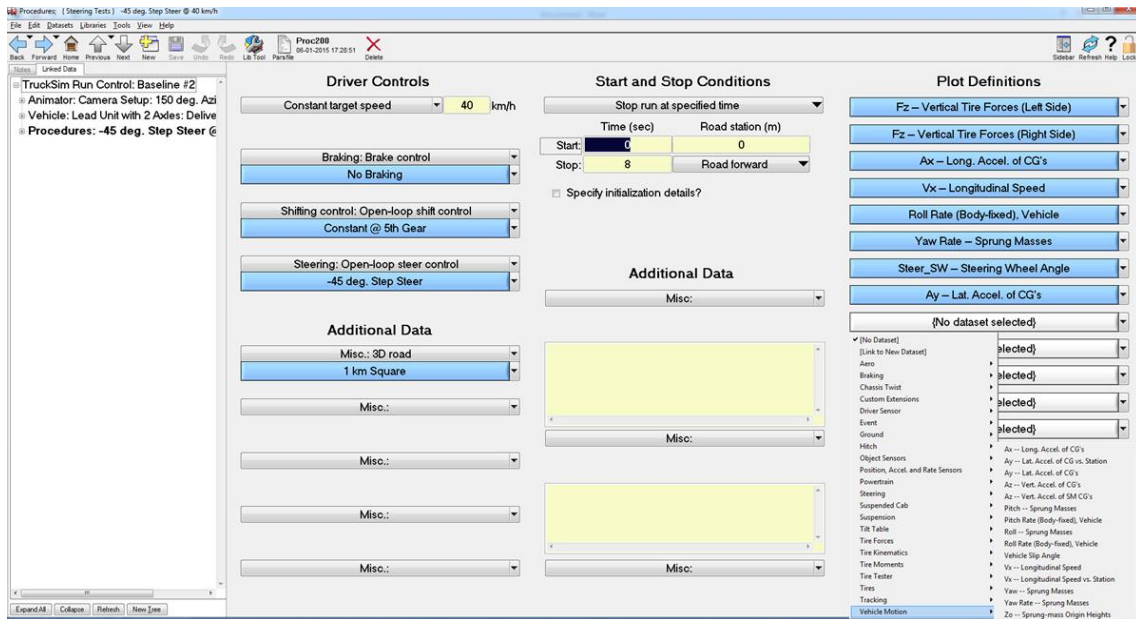


Figura 4.11: Elección plots “Vehicle Motion”

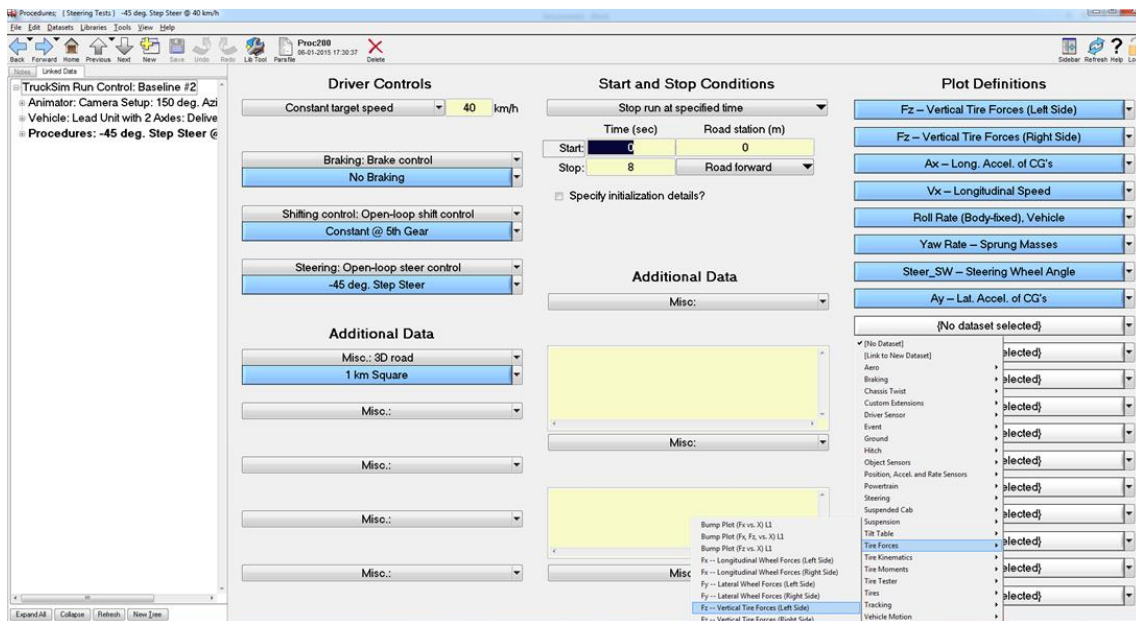


Figura 4.12: Elección plots “Tire Forces”

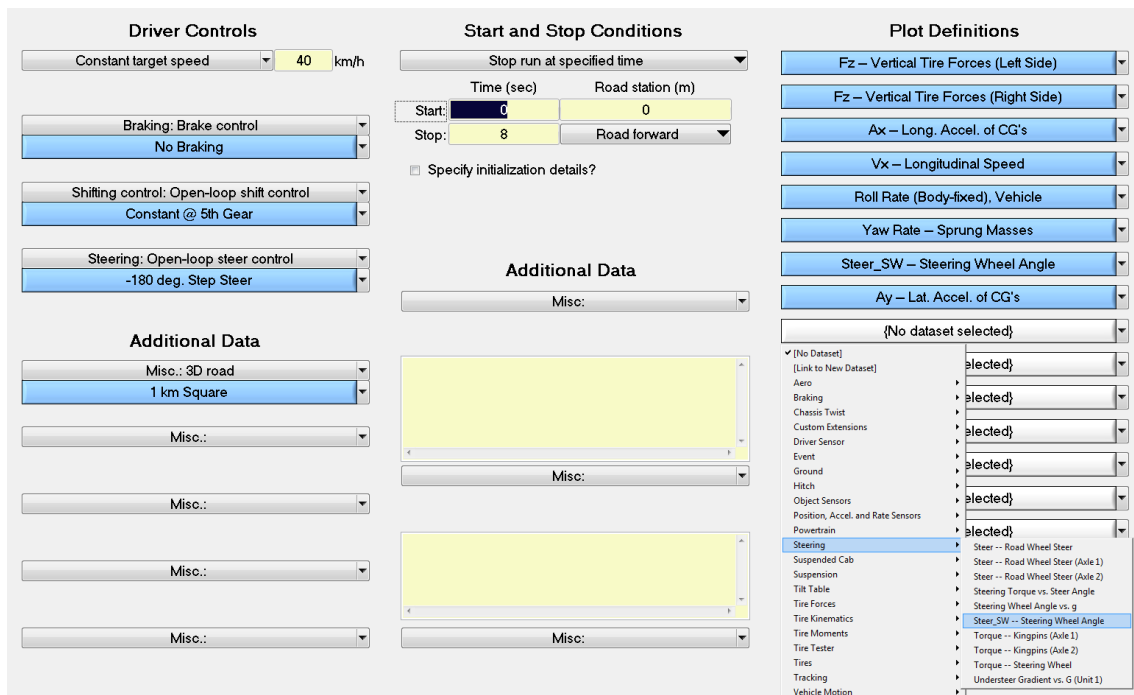


Figura 4.13: Elección plots “Steering”

3. Simulación de las maniobras creadas; tras la creación de todas las maniobras a simular mediante el procedimiento descrito en el anterior apartado, se procede a la simulación de las mismas, para lo cual se selecciona la maniobra a simular y se pulsara la pestaña “Run Math Model” muestra en la figura 4.14, tras pulsarla se simularan las condiciones de nuestra maniobra, se puede observar la maniobra en su totalidad pulsando el botón “Animate” como se muestra en la figura adjunta 4.2.1-13, y los resultados de los parámetros de estudio los obtendremos pulsando el botón “Plot” también mostrado en la figura 4.14. Una vez dentro del menú generado al pulsar “Plot” se guardarán los resultados de las variables de estudio para su posterior utilización en la red neuronal, tal cual se muestra en la figura 4.15.

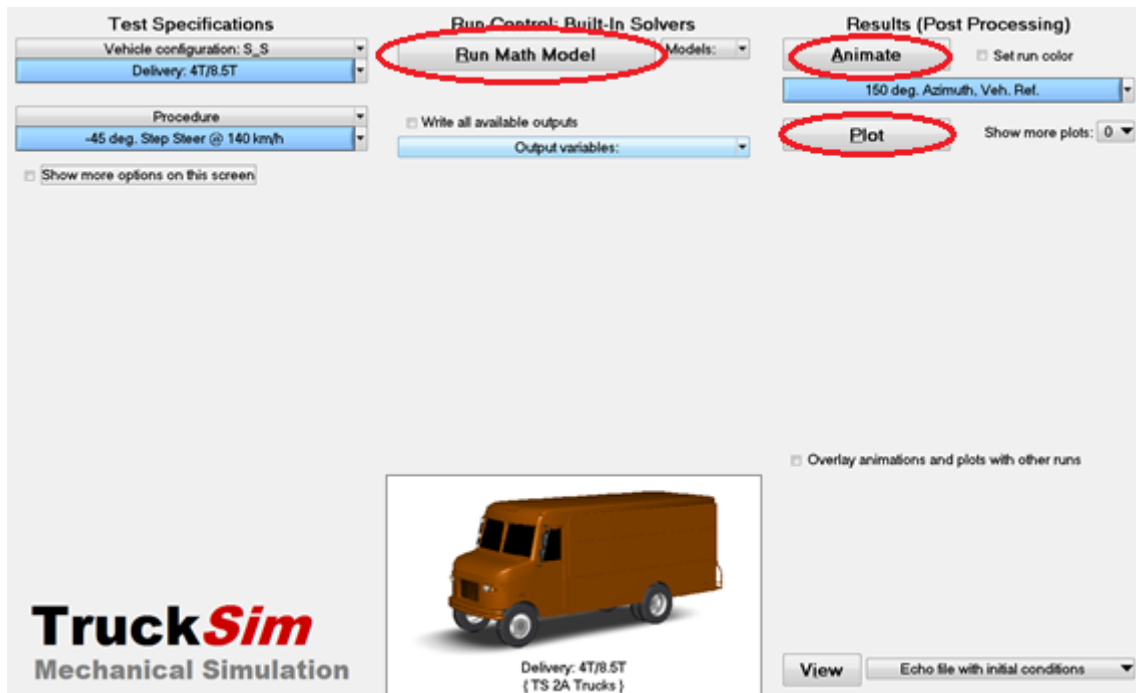


Figura 4.14: Menú principal de las simulaciones

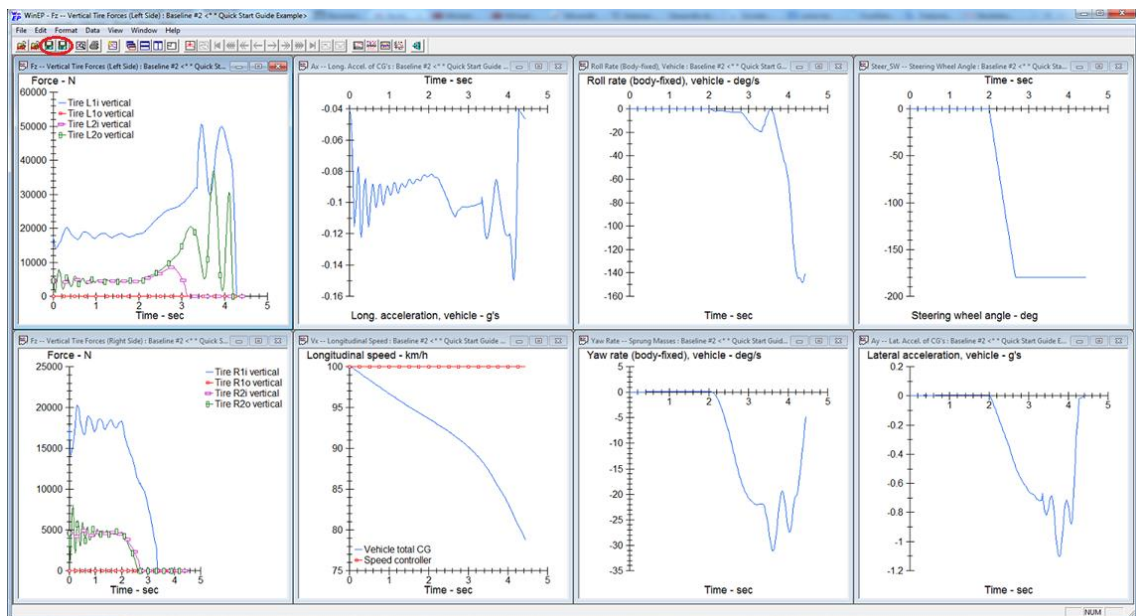


Figura 4.15: Menú resultados variables de estudio de la maniobra simulada

4.2.2 Double Lane Change

Esta maniobra básicamente lo que estudia es el comportamiento del vehículo en un doble cambio de carril, será simulada por la exposición del vehículo a movimientos de las masas no suspendidas y su influencia en la transferencia de carga en el vehículo. Será simulado con velocidades entre 40 y 140 km/h tal cual se muestra en la tabla adjunta (Tabla 4.1) para el estudio de las variables anteriormente mencionadas, y así poder realizar el cálculo de la transferencia de carga en el vehículo de estudio.

Tabla 4.1: Rango de velocidades simuladas para la maniobra de Double Lane Change

Double Lane Change (km/h)
40
60
80
100
120
140

4.2.3 J-Turn

Esta maniobra básicamente lo que busca es el estudio del comportamiento del vehículo en giros, se simulará el siguiente rango de velocidades y ángulos tanto en giros hacia la derecha como hacia la izquierda, tal como se observa en la Tabla 4.2:

Tabla 4.2: Rango de velocidades y ángulos simulados para la maniobra de J-Turn

Velocidad (km/h)	Ángulo de giro izquierda	Ángulo de giro derecha
40	45	-45
40	90	-90
40	135	-135
40	180	-180
40	225	-225
40	270	-270

60	45	-45
60	90	-90
60	135	-135
60	180	-180
60	225	-225
60	270	-270
80	45	-45
80	90	-90
80	135	-135
80	180	-180
80	225	-225
80	270	-270
100	45	-45
100	90	-90
100	135	-135
100	180	-180
100	225	-225
100	270	-270
120	45	-45
120	90	-90
120	135	-135
120	180	-180
120	225	-225
120	270	-270
140	45	-45
140	90	-90
140	135	-135
140	180	-180
140	225	-225
140	270	-270

4.3 ANÁLISIS DE LOS RESULTADOS OBTENIDOS EN LAS SIMULACIONES

Tras el proceso de simulación de las maniobras pertinentes para el diseño entrenamiento y test de la red neuronal se procede al análisis de los resultados de dichas maniobras.

Se destaca que no todas las maniobras que inicialmente se decidieron simular tuvieron resultados satisfactorios, por ello deben ser desechados los resultados de cara a la posterior implementación de la red neuronal. Estos resultados a desechar serán distinguidos en función de la maniobra desempeñada en Truksim:

4.3.1 Maniobras de J-Turn



Figura 4.16: Maniobra J-Turn

En el análisis de las maniobras de giro estándar (Figura 4.16) se observan los siguientes fallos, y las maniobras pertinentes fueron descartadas para la posterior implementación:

- La furgoneta no puede realizar giros mayores a 180 grados (hacia la izquierda) para velocidades a partir de 60 km/h (incluida esta velocidad) ya que pierde contacto con el asfalto de manera significativa provocándose el vuelco del vehículo.
- De manera homónima a la descrita en el anterior punto tampoco podrá para giros mayores a -180 grados (giros hacia la derecha) repitiéndose el mismo comportamiento que el descrito anteriormente.
- Para ángulos de giro de ± 180 grados y a partir de 80 km/h los resultados también serán desechados, dado que nuevamente se incurre en una situación de vuelco del vehículo de estudio.
- Para giros con un ángulo de -135 grados a la velocidad de 100 km/h la furgoneta nuevamente vuelca por lo que los resultados serán desechados, sin embargo para valores de +135 no serán desechados ya que no se produce la situación de vuelco del vehículo.
- Para ángulos de giro de ± 135 grados y velocidades mayores o iguales a 120 km/h los resultados serán desechados por vuelco de la furgoneta.



Figura 4.17. Comienzo de vuelco asociado a la pérdida de adherencia rueda calzada en una de las maniobras desechadas

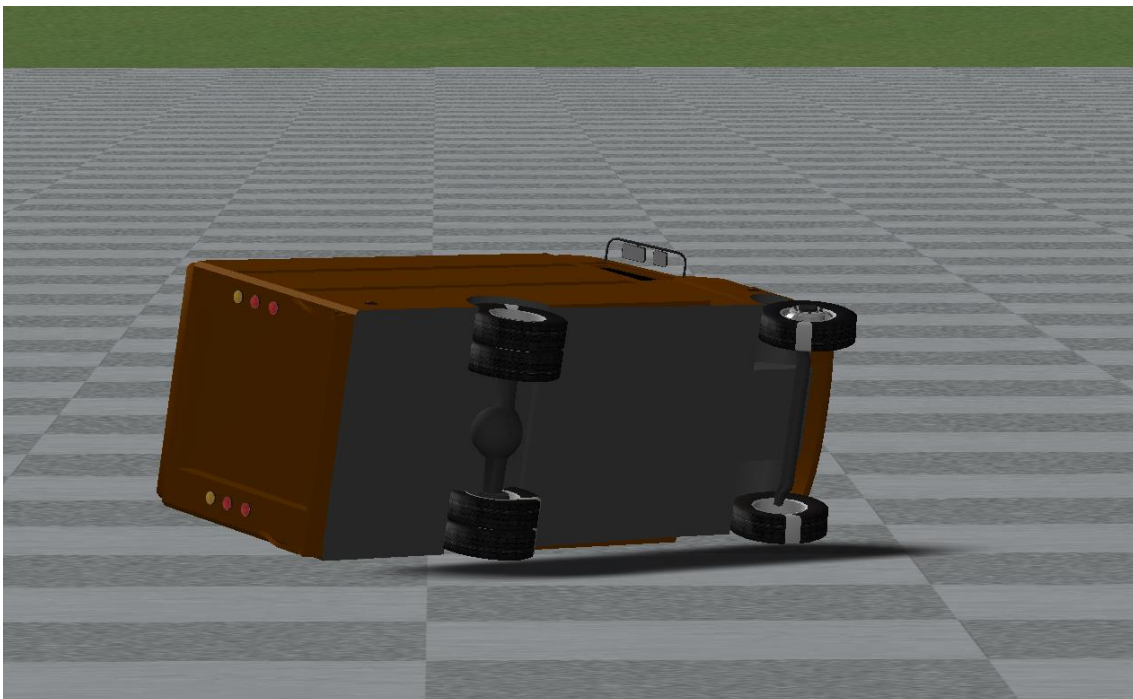


Figura 4.18. Vuelco total asociado a la pérdida de adherencia rueda calzada en una de las maniobras desechadas

El resto de maniobras ensayadas son válidas y sus resultados son correctos y serán utilizados para la implementación de la red neuronal.

4.3.2 Maniobras de Double Lane Change (DLC)

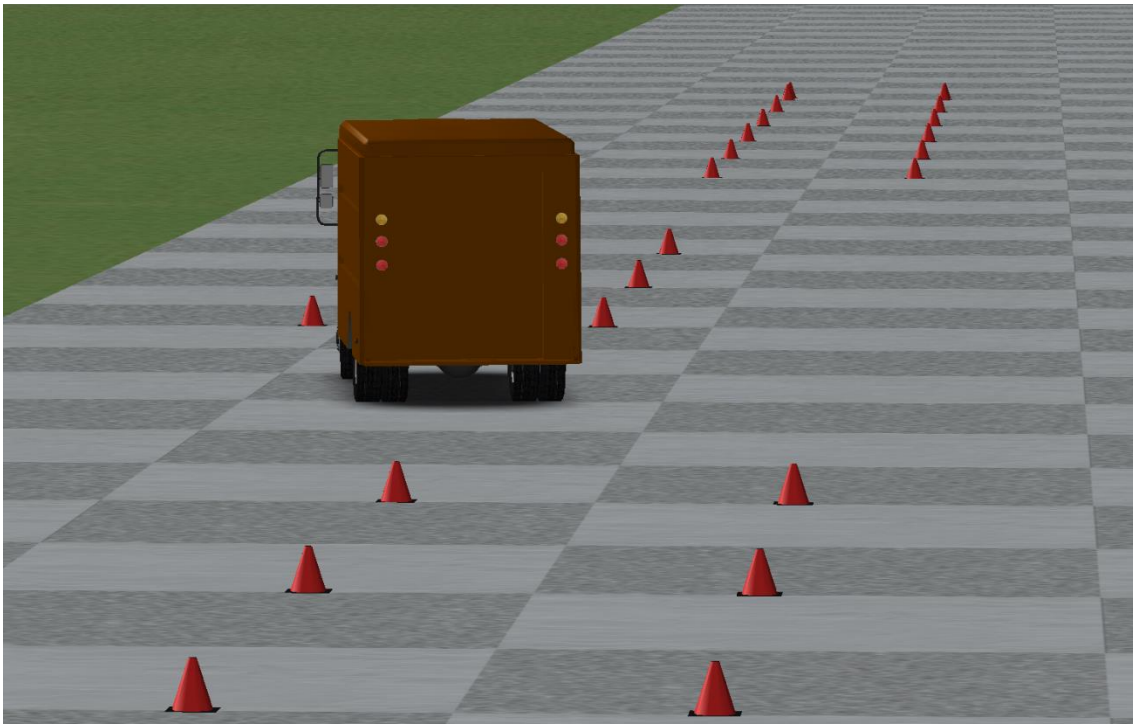


Figura 4.19: Maniobra Double Lane Change

En el análisis de las maniobras de doble cambio de carril (Figura 4.19) observamos solo una maniobra fallida, se trata de la maniobra DLC a la velocidad de 140 km/h, en la que la furgoneta es incapaz de realizar correctamente la maniobra por no tener la estabilidad necesaria a dicha velocidad para realizar el cambio sin invadir las líneas delimitadoras de los carriles.

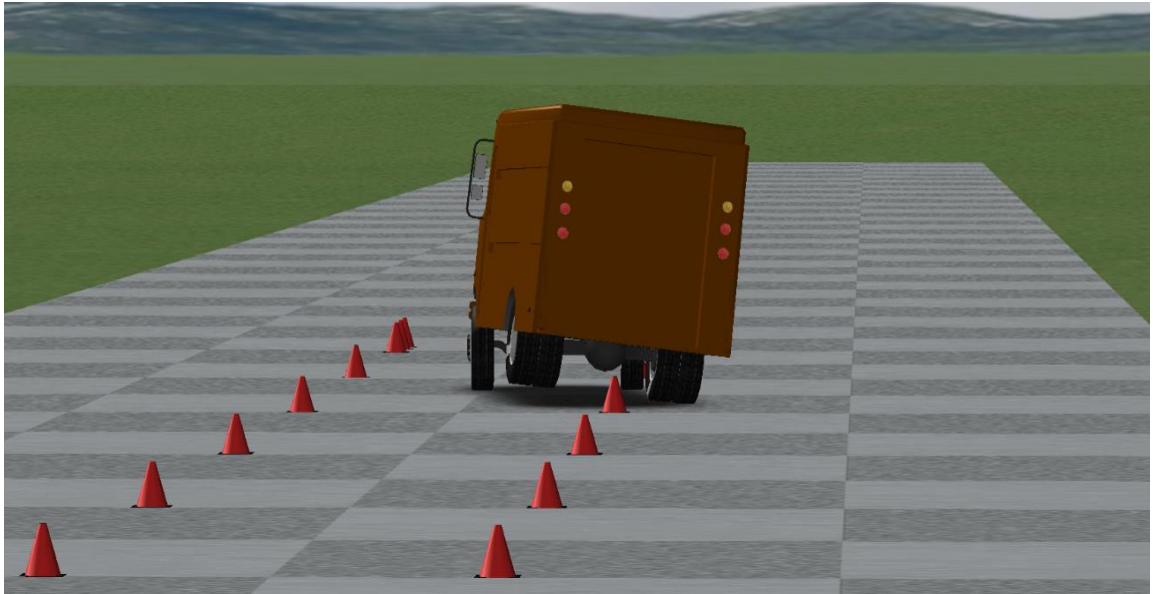


Figura 4.20. Situación de inestabilidad y pérdida de adherencia rueda calzada en DLC a 140 km/h, invasión de línea divisora de carril



CAPÍTULO 5: APRENDIZAJE DE LA TRANSFERENCIA DE CARGA UTILIZANDO REDES NEURONALES EN MATLAB

CAPÍTULO 5: APRENDIZAJE DE LA TRANSFERENCIA DE CARGA UTILIZANDO REDES NEURONALES EN MATLAB

Las redes neuronales artificiales son una técnica de aprendizaje y procesamiento automático inspirada en el funcionamiento del cerebro humano. Se pueden definir las redes neuronales artificiales como una estructura de procesamiento paralelo masivo constituida por unas unidades muy sencillas (denominadas neuronas), que tienen la capacidad de almacenar conocimiento experimental y ponerla a disposición para su uso.

Las redes neuronales artificiales se asemejan a las redes neuronales biológicas en varios aspectos:

1. Las neuronas son elementos simples y altamente interconectados (aunque las neuronas artificiales son mucho más simples).
2. Las conexiones entre las neuronas determinan la función de la red. Estas conexiones, conocidas como pesos (pesos sinápticos), se utilizan para almacenar el conocimiento adquirido.
3. El conocimiento es adquirido a partir de su entorno mediante un proceso de aprendizaje.

ARQUITECTURA

La arquitectura (o topología) de la red hace referencia a la disposición de las neuronas en la red. Las neuronas se organizan formando capas, de modo que la red neuronal puede consistir en una o más capas de neuronas (Figura 5.1).

Cada neurona recibe un conjunto de entradas multiplicadas por su interconexión (peso), que son sumado y operado por una función de transferencia (o función de activación) antes de transmitirse a la siguiente capa o como salida de la red.

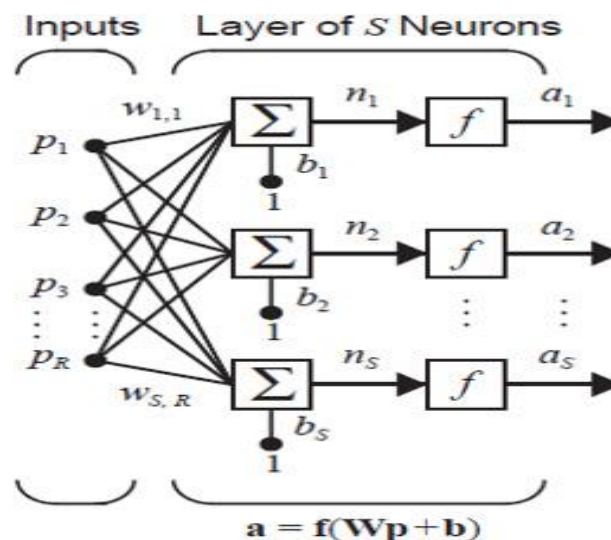


Figura 5.1: Red neuronal con una única capa, R elementos de entrada y S neuronas

APRENDIZAJE

En numerosas ocasiones, se suelen también clasificar en función del tipo de aprendizaje que realizan (Figura 5.2). Así, se distingue entre:

- Aprendizaje supervisado, donde un conjunto conocido de datos de entrada-salida se utiliza para iterativamente ajustar los pesos de la red.
- Aprendizaje no supervisado, en donde únicamente se disponen de datos de entrada y una función de coste a minimizar.
- Aprendizaje reforzado, situado a medio camino entre el aprendizaje supervisado y no supervisado. En este aprendizaje la información proporcionada a la red es mínima, limitándose a indicar si la respuesta de la red es correcta o incorrecta.

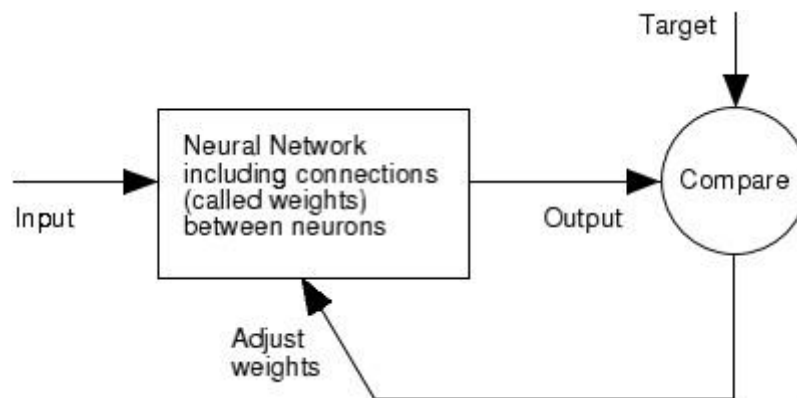


Figura 5.2: Proceso de ajuste de pesos en el aprendizaje supervisado. **Fuente:** Mathworks Online Help

APLICACIONES

Estas redes han sido y son ampliamente utilizadas en muchos campos: aeroespacial (pilotos automáticos, simulaciones de trayectorias de vuelo, detección de fallo de componentes), automoción (sistemas de guiado automático, controladores, análisis de la garantía), finanzas (lectores automáticos de cheques, solicitudes de crédito, detección de fraude, calificación de bonos), defensa (seguimiento de objetos, reconocimiento facial, eliminación de ruido), medicina (análisis de células cancerígenas, diseño de prótesis, análisis de electroencefalografía y electrocardiograma), telecomunicaciones (reconocimiento y compresión de voz, compresión de imágenes, traducción en tiempo real de un idioma), etc.



5.1 REDES PERCEPTRON MULTICAPA

El perceptrón multicapa es una red neuronal artificial compuesta por múltiples capas, lo cual le permite resolver problemas linealmente dependientes, lo cual es su principal limitación. El perceptrón multicapa puede ser local o totalmente conectado. En el primer caso cada salida de una neurona de la capa "i" es entrada de todas las neuronas de la capa "i+1", mientras que en el segundo cada neurona de la capa "i" es entrada de una serie de neuronas de la capa "i+1".

5.1.1 Creación de la red

La creación de una red perceptron multicapa en Matlab se realiza con la función `newff`, esta función devuelve una variable que representa a la red neuronal. Esta función tendrá los siguientes argumentos de entrada:

- R, matriz con un número de filas igual al número de entradas a la red y dos columnas. Esta matriz señala los valores máximos y mínimos que pueden tomar cada una de las entradas de la red.
- [Slayer1, Slayer2, Slayer3,...] es el vector que define el número de neuronas del que disponen las capas ocultas de la red y el número de neuronas de salida. El tamaño de este vector es el que permite al programa saber el número de capas que debe tener la red neuronal a crear.
- {Funciones}, es un vector con varias cadenas de caracteres en el que se indican las funciones de activación que poseerán las neuronas de una determinada capa. Siendo así, la primera cadena indica la función de activación concreta que tendrán las neuronas de la primera capa oculta de la red, la segunda cadena la de las de la segunda capa, y así sucesivamente. Este vector tendrá tantas cadenas como capas ocultas vaya a tener nuestra red neuronal. Por defecto Matlab utiliza la función tangente hiperbólica, distinguiéndose las siguientes funciones de activación:
 - Para que la función de activación sea lineal la cadena debe ser "purelin".
 - Para que la función de activación sea tangente hiperbólica debe escribir "tansig".
 - Para que la función de activación sea una función sigmoideal escribiremos "logsig".
- El siguiente argumento de entrada debe indicar el tipo de entrenamiento que hemos elegido para el entrenamiento de nuestra red. Si no se indica nada el entrenamiento por defecto empleado por Matlab es "trainlm", es decir por el método Levenber-Marquardt.

5.1.2 Simulación de la red neuronal

Para obtener las salidas de una red neuronal ante unas determinadas entradas se empleará la función `sim`. Esta función genera un vector cuyas componentes serán las salidas que se obtienen

de cada neurona de salida de la red neuronal. Esta función recibe como argumentos el nombre del objeto que representa la red en Matlab, y el vector de las entradas a la red neuronal.

Así pues si disponemos de una red “net” de la que queremos saber su respuesta ante unas entradas determinadas “x” escribiremos el siguiente código en Matlab:

$$m = \text{sim}(\text{net}, x);$$

5.1.3 Entrenamiento de la red neuronal

Tal como se ha comentado con anterioridad, el entrenamiento de la red (empleado en este proyecto) consiste en proporcionarle a la red neuronal creada una serie de entradas con sus correspondientes salidas (conocidas), para que la red vaya reajustando su salida o respuesta mediante la modificación de sus pesos y valores umbrales, de manera que se consiga así la minimización máxima posible del error de actuación de la red. Este valor será almacenado por defecto en la variable `net.performFcn`. Como expusimos anteriormente la medida por defecto en Matlab del error es el error medio cuadrático, cuya expresión es la siguiente:

$$E(n) = \frac{1}{N} \sum_{k=1}^N \sum_{n=1}^r (s_k(n) - y_k(n))^2 \quad (\text{ec 5.1})$$

Para el entrenamiento de la red en Matlab se emplea la función `train`, esta función emplea como argumentos el nombre de la red a entrenar (que hemos debido crearla con anterioridad) y los patrones, compuestos por los valores de los vectores de entradas a la red y sus salidas asociadas (todos ellos dentro de una matriz que llamaremos “m”). Así el código que debemos escribir es el siguiente:

$$[\text{net}, \text{pr}] = \text{train}(\text{net}, x, m);$$

La variable `pr` contendrá la información sobre el proceso del entrenamiento de la red y la variable `net` contendrá a la red ya entrenada, con sus correspondientes pesos y valores umbrales correctamente ajustados. Se destaca que esta función utiliza una serie de variables para definir el entrenamiento de la red, variables que pueden definirse con anterioridad, variando en función del tipo de entrenamiento al que se vaya a someter a la red, dependiendo del algoritmo de aprendizaje asociado a dicho tipo de entrenamiento.

5.1.3.1 Algoritmo de Levenberg-Marquardt

En el anterior capítulo se ha comentado como entrenar una red neuronal, Matlab ofrece gran cantidad de algoritmos en función del entrenamiento que deseemos para nuestra red neuronal.

Para este proyecto se ha usado el algoritmo de Levenber-Marquardt como método de entrenamiento de la red neuronal. Este algoritmo es una modificación del método de Newton, diseñado para acercarse en segundo orden, entrena a gran velocidad sin tener que calcular la matriz Hessiana. Requiere mayor esfuerzo computacional que otros algoritmos disponibles en Matlab pero es el más rápido y el empleado por defecto en Matlab para el entrenamiento de las redes neuronales. Dado que la función de activación tiene forma de una suma de cuadrados la matriz Hessiana podrá aproximarse por (Fuente Mathworks Online Help):

$$H = J^T J \quad (\text{ec 5.2})$$

Y el gradiente puede computarse como:

$$g = J^T e \quad (\text{ec 5.3})$$

donde J es la matriz de Jacobianos que contiene la derivada de los errores de la red primero con respecto a los pesos y el umbral y e es el vector de errores de la red. La matriz de Jacobianos es mucho más sencilla de computar mediante *backpropagation* que la matriz Hessiana.

El algoritmo de Levenberg-Marquardt hace la siguiente aproximación a la matriz Hessiana como la siguiente actualización Newton:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \quad (\text{ec 5.4})$$

Cuando el valor del escalar μ sea cero, el algoritmo se comportará como el método de Newton empleando para el entrenamiento la matriz Hessiana aproximada. Cuando el valor de μ sea elevado, actuará usando el método de descenso de gradiente. El método de Newton es más rápido y más exacto a la hora de minimizar el error, así pues el objetivo será el de cambiar hacia el método de Newton con la mayor celeridad posible. De esta manera, μ disminuirá después de cada paso exitoso (reducción en la función de activación) y aumentará en los pasos provisionales.

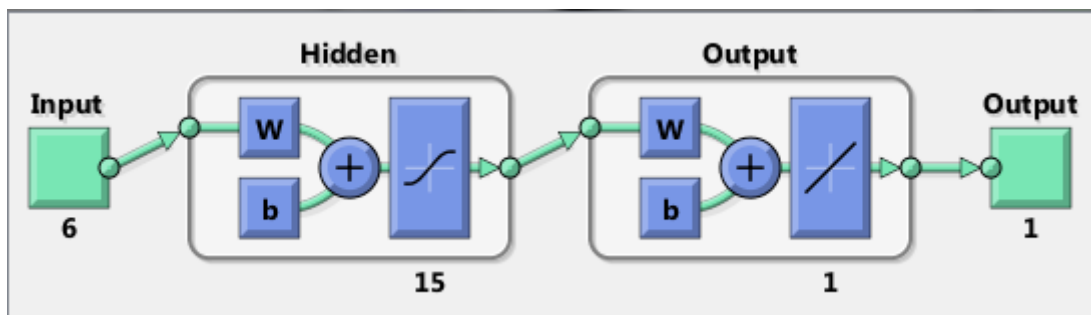


Figura 5.3: Arquitectura de la red neuronal implementada

5.2 ANÁLISIS DE LOS RESULTADOS OBTENIDOS TRAS EL ENTRENAMIENTO DE LA RED NEURONAL

Tras la creación, simulación y entrenamiento de la red, se procede a evaluar los resultados obtenidos, si estiman correctamente la transferencia de carga y el error cometido a la hora de realizar la estimación, para ello se procede a comparar los datos entregados a la red como salidas, es decir los resultados obtenidos para los valores de transferencia de carga desde Trucksim, y serán comparados con los resultados arrojados por la red neuronal tras el entrenamiento.

5.2.1 Comparativa de resultados en función del algoritmo de entrenamiento de la red neuronal

Se realizarán distintas comparativas en función de los tipos de entrenamiento (empleando distintos algoritmos), comparando el error cometido en cada caso.

5.2.1.1 Resultados objetivo vs resultados obtenidos por el algoritmo de “Levenberg-Marquardt”

En el siguiente gráfico Figura 5.8 se puede observar la comparativa entre los resultados obtenidos en las simulaciones de las maniobras en Trucksim y los obtenidos en Matab tras el entrenamiento de la red neuronal por el algoritmo de “Levenberg-Marquardt”. Será implementado en Matlab por la función “trainlm”:

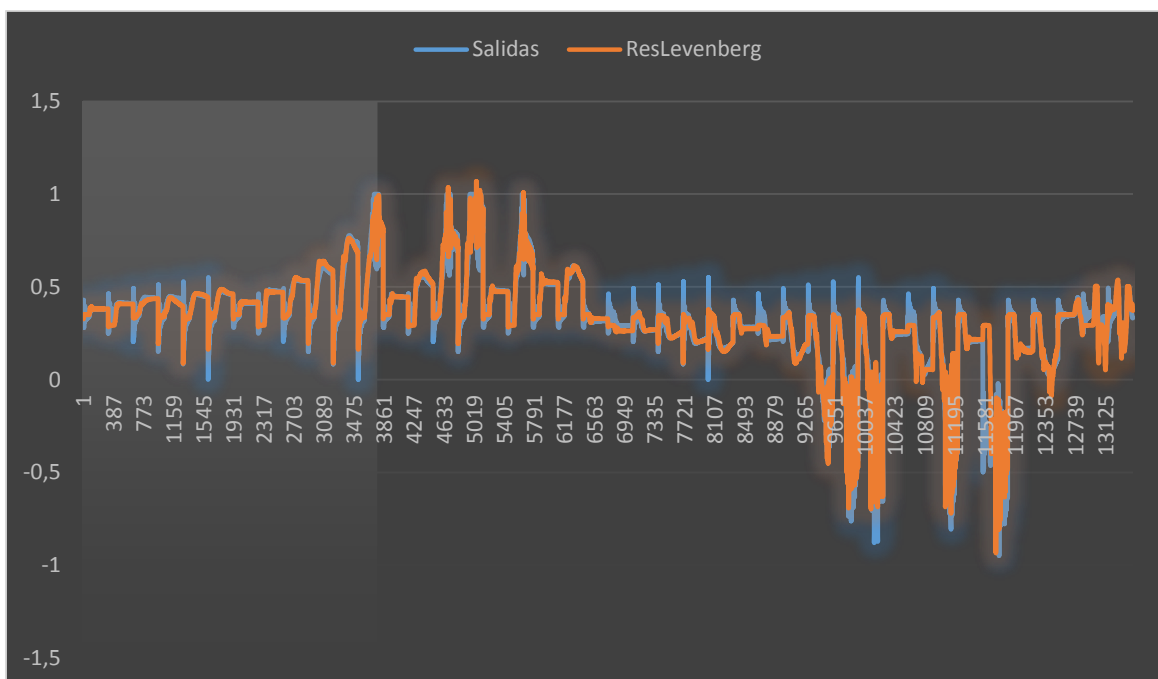


Figura 5.4: Comparativa salidas objetivo vs salidas obtenidas por algoritmo Levenber-Marquardt

El error cuadrático medio (Mean Square Error – MSE) que se ha obtenido tras el entrenamiento por el algoritmo de “Levenberg-Marquardt”, será calculado por la ecuación 11 explicada en el apartado 3.2.1:

$$mse = 0,00012261$$

5.2.1.2 Resultados objetivo vs resultados obtenidos por el algoritmo de “Regularización Bayesiana”

En el método de entrenamiento de la red por el algoritmo de Regularización Bayesiana o Automatizada, los pesos y el umbral de la red neuronal se asumen como variables al azar con distribuciones específicas. Los parámetros de regularización se relacionan con las variables desconocidas y asociadas a estas distribuciones.

En la figura adjunta a este apartado (Figura 5.9) se puede observar la comparativa entre los resultados objetivo obtenidos en Truksim y los resultados obtenidos en Matlab tras el entrenamiento de la red por el algoritmo de “Regularización Bayesiana o Automatizada (Bayesian Regularization), que será implementado por la función “trainbr”:

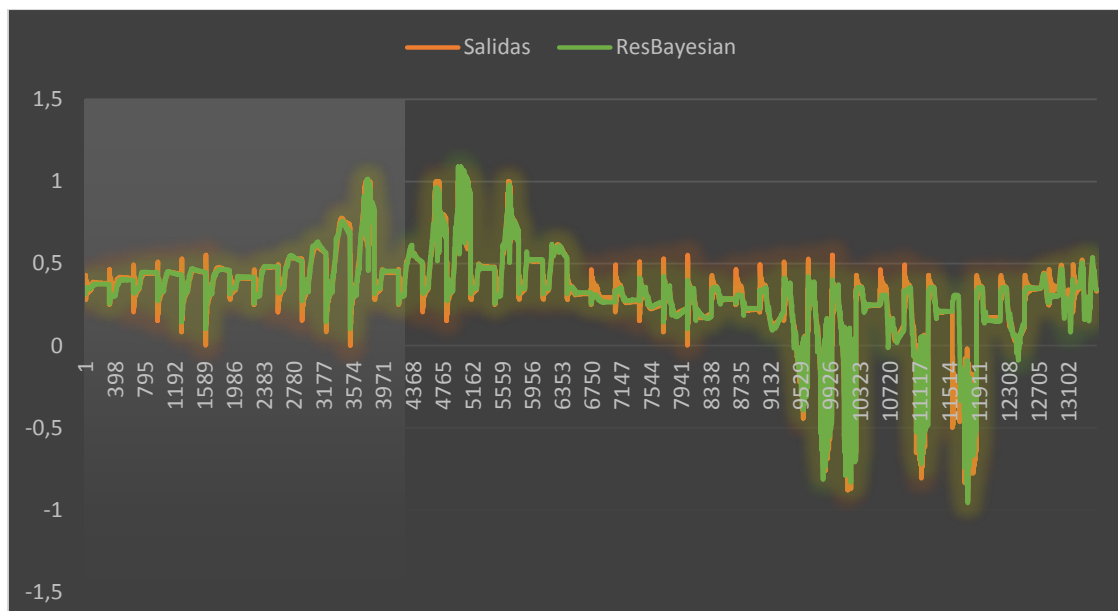


Figura 5.5: Comparativa salidas objetivo vs salidas obtenidas por algoritmo de Regularización Bayesiana o automatizada

El error cuadrático medio (Mean Square Error – MSE) que se ha obtenido tras el entrenamiento por el algoritmo de “Bayesian Regularization”, será calculado por la ecuación 11 explicada en el apartado 3.2.1:

$$mse = -0,0004188$$

5.2.1.3 Resultados objetivo vs resultados obtenidos por el algoritmo del “Gradiente conjugado escalado”

El método de entrenamiento de la red por el algoritmo del Gradiente conjugado escalado tiene como finalidad y fue diseñado para evitar la búsqueda lineal. Este algoritmo es muy complejo para ser explicado en unas líneas, pero la idea básica de este algoritmo es la de combinar la aproximación de región modelo-confianza con el gradiente conjugado.

En la figura adjunta a este apartado (Figura 5.10) se puede observar la comparativa entre los resultados objetivo obtenidos en Trucksim y los resultados obtenidos en Matlab tras el entrenamiento de la red por el algoritmo de “Gradiente Conjugado Escalado (Scaled Conjugate Gradient)” Será implementado en Matlab por la función “trainsgc”:

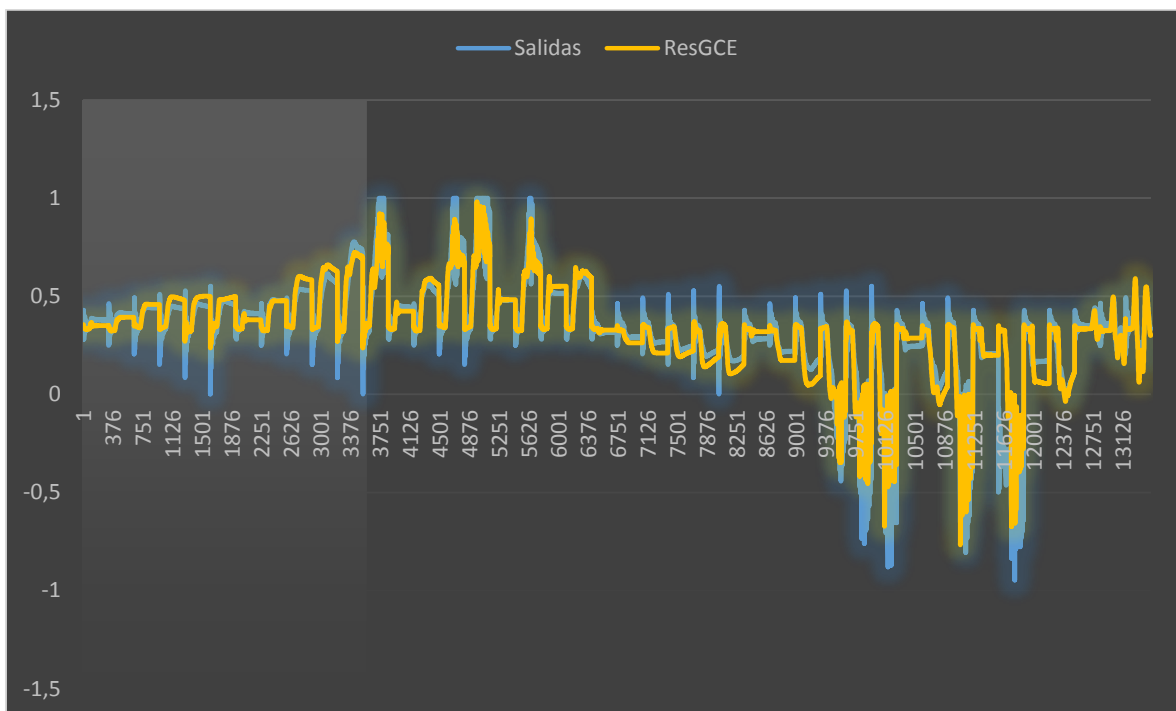


Figura 5.6: Comparativa salidas objetivo vs salidas obtenidas por algoritmo del Gradiente Conjugado Escalado

El error cuadrático medio (Mean Square Error – MSE) obtenido tras el entrenamiento por el algoritmo del “Gradiente Conjugado Escalado”, será calculado por la ecuación 11 explicada en el apartado 3.2.1:

$$mse = -0,0005516$$

A la vista de los resultados obtenidos se puede afirmar que por todos los algoritmos de entrenamiento se ajustan bien a los resultados objetivo, observando las gráficas de cada uno de los algoritmos puede observarse que el patrón descrito por los resultados es bastante similar para la red neuronal y la simulación real en todos los casos.

Comparando los tres métodos se puede observar que tanto el algoritmo de “Levenberg-Marquardt” como el del “Gradiente Conjugado Escalado” necesitan un número de iteraciones bastante reducido para ajustar correctamente la red a los resultados objetivos, simulando con estos dos algoritmos con intervalos de entre 100 y 150 iteraciones en ambos casos ajustaba correctamente la red minimizando el error medio cuadrático. Sin embargo empleando el algoritmo de “Regularización Bayesiana” necesita un número significativamente más elevado que en los otros dos algoritmos, para este algoritmo Matlab necesito un numero de iteraciones de entre 850 y 950 para ajustar correctamente la red a las exigencias impuestas, consiguiéndose además resultados muy similares a los obtenidos por los otros dos métodos, por lo cual el entrenamiento de la red es significativamente más lento en este caso.

Adicionalmente cabe destacar el hecho de que los tres tipos de algoritmos realizan procesos diferentes, el algoritmo de “Levenberg-Marquardt” es un algoritmo de aplicación rápida, ya que requiere más memoria de la computadora pero realiza el entrenamiento en menos tiempo que los otros, se detiene el entrenamiento cuando se detiene la mejora de generalización al aumentar el error cuadrático medio de las muestras de validación. Por otro lado el algoritmo de “Regularización Bayesiana” normalmente necesita más tiempo para el entrenamiento de la red, pero para generalización de conjuntos de datos difíciles, pequeños o ruidosos puede ser una buena opción. Finalmente el algoritmo del “Gradiente Conjugado Escalado” al igual que el de “Levenberg-Marquardt” requiere menos memoria y se detiene siguiendo el mismo método, al aumentar el valor del error medio cuadrático de las muestras de validación.

El algoritmo de “Levenberg-Marquardt” y el del “Gradiente Conjugado Escalado” son dos métodos de entrenamiento muy parecidos como se observa en su forma de proceder a nivel computacional, en el número de iteraciones que necesitan para un correcto ajuste de la red y en la forma en la que detienen el entrenamiento.

A pesar de realizar simulaciones con los tres algoritmos, para este proyecto como se ha citado anteriormente, a la hora de elegir uno, se elegirá el algoritmo de “Levenberg-Marquardt”, por ser el que ha ofrecido unos resultados más fiables y además ser el método de entrenamiento que menor error medio cuadrático ofrece (y el que Matlab ofrece por defecto para el entrenamiento de las redes neuronales), es decir consigue ajustar la red neuronal con más precisión a los resultados perseguidos, y por tanto para el diseño de este estimador es el método más adecuado.



Como se observa en la gráfica comparativa entre la salida objetivo y la salida obtenida tras el entrenamiento de la red por el algoritmo “Levenberg-Marquardt” (Figura 5.8), el resultado se ajusta fielmente a lo buscado en cada uno de los 13482 valores de transferencia de carga comparados. Adicionalmente se observa que el valor del error obtenido tras la simulación de la red neuronal para cada una de las salidas objetivo es muy bajo, y por tanto aceptable para los objetivos marcados previos al entrenamiento.

5.2.2 Comparativa entre los errores cometidos en el entrenamiento de la red neuronal

En la tabla que se muestra a continuación (Tabla 5.1) se observan los resultados del error cometido en el entrenamiento de la red neuronal, en función de cada maniobra y el algoritmo de entrenamiento empleado.

Tabla 5.1: Comparativa de errores cometidos en el entrenamiento de la red neuronal en función de el algoritmo empleado y la maniobra simulada

Maniobra	Ángulo Girado	Velocidad (km/h)	Algoritmo De Entrenamiento		
			Levenberg-Marquardt	Regularización Bayesiana	Gradiente conjugado escalado
J-Turn	45	40	-0,00367541	0,00562197	0,02400105
J-Turn	45	60	0,01764948	0,0268234	0,02105686
J-Turn	45	80	-0,00144018	-0,00228851	-0,00891533
J-Turn	45	100	0,00927354	0,00071638	-0,03520941
J-Turn	45	120	0,00377465	-0,00557944	-0,03015493
J-Turn	45	140	-0,00305981	0,00207129	-0,02522803
J-Turn	90	40	-0,00453911	-0,00182962	0,02510239
J-Turn	90	60	0,01633652	0,01547267	0,00882922
J-Turn	90	80	-0,00175175	0,0032724	-0,03857887
J-Turn	90	100	-0,01826806	-0,00842202	-0,04417538
J-Turn	90	120	0,01051038	0,00752947	0,01298161
J-Turn	90	140	0,00105522	0,00013095	0,03257666
J-Turn	135	40	0,00057011	-0,00385373	0,01735379
J-Turn	135	60	0,01214026	0,01023368	-0,01864566
J-Turn	135	80	-0,00128602	0,01803022	0,03020548
J-Turn	135	100	---	---	---
J-Turn	135	120	---	---	---
J-Turn	135	140	---	---	---
J-Turn	180	40	0,00521642	0,00564418	0,00046664
J-Turn	180	60	0,02550494	0,01282129	0,03997778
J-Turn	180	80	---	---	---
J-Turn	180	100	---	---	---
J-Turn	180	120	---	---	---
J-Turn	180	140	---	---	---



CAPÍTULO 5: APRENDIZAJE DE LA TRANSFERENCIA DE CARGA UTILIZANDO REDES NEURONALES EN MATLAB

J-Turn	225	40	-0,00177405	-0,0059738	-0,02344756
J-Turn	225	60	---	---	---
J-Turn	225	80	---	---	---
J-Turn	225	100	---	---	---
J-Turn	225	120	---	---	---
J-Turn	225	140	---	---	---
J-Turn	270	40	-0,00016092	0,00140977	-0,02318968
J-Turn	270	60	---	---	---
J-Turn	270	80	---	---	---
J-Turn	270	100	---	---	---
J-Turn	270	120	---	---	---
J-Turn	270	140	---	---	---
J-Turn	-45	40	0,0047911	-0,00568032	-0,00379055
J-Turn	-45	60	0,03518357	0,03313785	0,02558806
J-Turn	-45	80	0,00534247	-0,00342886	0,03843518
J-Turn	-45	100	0,00437101	-0,01238812	0,02072666
J-Turn	-45	120	0,00199378	0,00656991	0,03527627
J-Turn	-45	140	-0,00618889	0,00052631	0,03387487
J-Turn	-90	40	-0,0047206	-0,00178984	-0,02299443
J-Turn	-90	60	6,5081E-06	0,01211293	0,03559567
J-Turn	-90	80	-0,00692653	0,007044	0,05929522
J-Turn	-90	100	-0,00607802	-0,00417811	-0,01930658
J-Turn	-90	120	0,00493064	0,00680781	-0,02555413
J-Turn	-90	140	-0,00970565	-0,01401364	-0,03497386
J-Turn	-135	40	-0,00440498	-0,00095425	-0,02841633
J-Turn	-135	60	0,03939926	0,027621	0,05870695
J-Turn	-135	80	0,00137168	0,00122967	0,00834686
J-Turn	-135	100	---	---	---
J-Turn	-135	120	---	---	---
J-Turn	-135	140	---	---	---
J-Turn	-180	40	0,00595171	-0,00013212	-0,0002169
J-Turn	-180	60	-0,1936971	-0,19527774	-0,33104081
J-Turn	-180	80	---	---	---
J-Turn	-180	100	---	---	---
J-Turn	-180	120	---	---	---
J-Turn	-180	140	---	---	---
J-Turn	-225	40	0,00618401	0,01164011	0,06022014
J-Turn	-225	60	---	---	---
J-Turn	-225	80	---	---	---
J-Turn	-225	100	---	---	---
J-Turn	-225	120	---	---	---
J-Turn	-225	140	---	---	---
J-Turn	-270	40	0,00371148	0,00010886	0,00590498
J-Turn	-270	60	---	---	---
J-Turn	-270	80	---	---	---
J-Turn	-270	100	---	---	---
J-Turn	-270	120	---	---	---
J-Turn	-270	140	---	---	---



DLC	---	40	-0,00250956	-0,00124158	0,01391966
DLC	---	60	0,05238598	0,0381358	0,02283303
DLC	---	80	0,00091233	0,00384426	0,01876368
DLC	---	100	-0,00282677	0,00904273	-0,00244757
DLC	---	120	-0,01080153	-0,01824179	-0,02119634
DLC	---	140	0,02808554	0,00273387	-0,01835557

Como se observa en la tabla hay algunas maniobras sin resultados válidos, la razón de dicha situación es que al realizar la simulación de la maniobra pertinente se produjo una situación de vuelco, por lo que los resultados asociados a dichas maniobras fueron descartados para la implementación de la red neuronal.

5.2.2.1 Comparativa del error cometido durante la creación de la red neuronal en función del paquete de datos

En la tabla adjunta, Tabla 5.2, se muestran los resultados del error cometido por Matlab en la creación de la red en función de el algoritmo de entrenamiento empleado, y además en función del paquete de datos concreto simulado, que puede ser de los datos empleados del total de los datos obtenidos de Trucksim para el entrenamiento, para la validación de la red o para el test de la red neuronal.

En la red que se ha implementado en Matlab se seleccionan los datos que se dedican a cada paquete a partir del total de los datos que se entregan a la red, y de estos el 70% se emplean para el entrenamiento de la red neuronal, y del 30% restante, 15% se emplean para la validación de la red y 15% para el test de la misma.

De esta manera de los 13482 vectores de datos entregadas para la creación de la red 9438 vectores son empleados para el entrenamiento, 2022 para la validación y 2022 para el test de la red neuronal implementada.

Tabla 5.2: Comparativa errores en función algoritmo de entrenamiento y paquete de datos

Algoritmo de entrenamiento	Paquete de datos		
	Entrenamiento	Validación	Test
Levenberg-Marquardt	0,00358894	0,00474457	-0,02067899
Regularización Bayesiana	0,00355459	0,00326401	-0,02264802
Gradiente Conjugado Escalado	0,00803748	-0,00849577	-0,03269872

5.2.3 Graficas asociadas a las simulaciones en Matlab

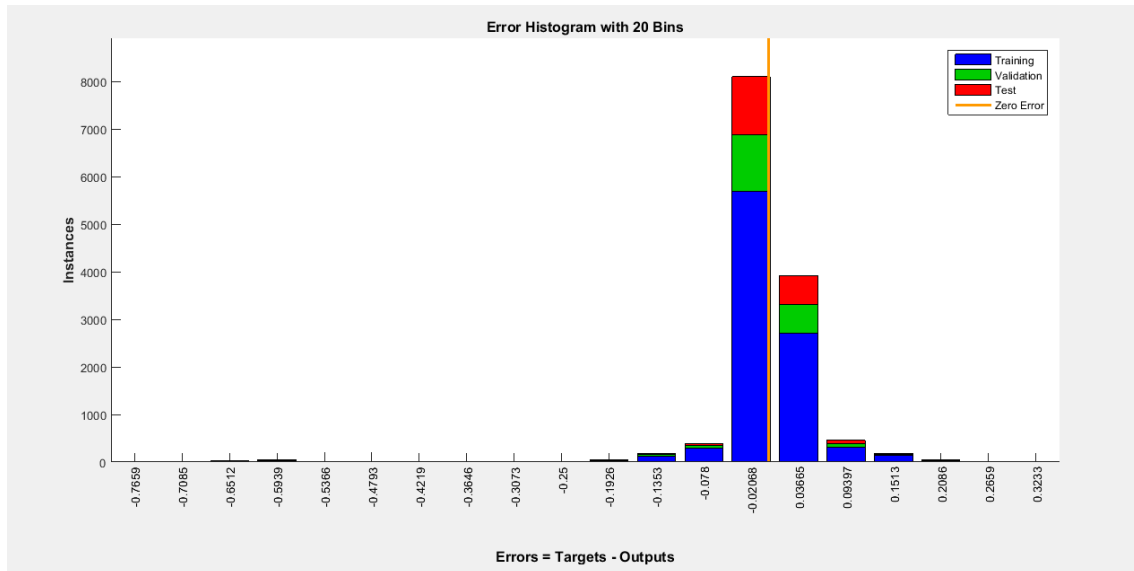


Figura 5.7: Historiograma de errores durante la simulación

En la Figura adjunta 5.4 puede observarse el historiograma de los errores acaecidos durante el entrenamiento de la red neuronal.

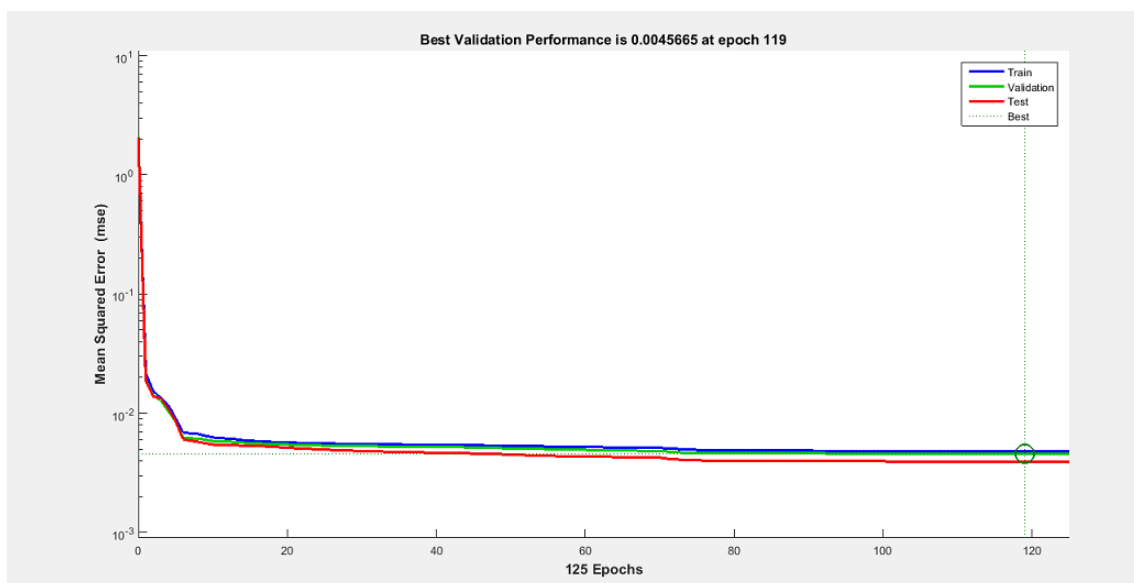


Figura 5.8: Evolución del error cuadrático medio durante la simulación

Durante el entrenamiento de la red neuronal, la red va ajustando los valores umbrales y modificando sus pesos para así conseguir la máxima minimización del error medio cuadrático, la Figura adjunta 5.5, nos muestra la evolución del valor de este durante el entrenamiento de la red hasta alcanzar el valor óptimo y dar así por finalizado el entrenamiento.

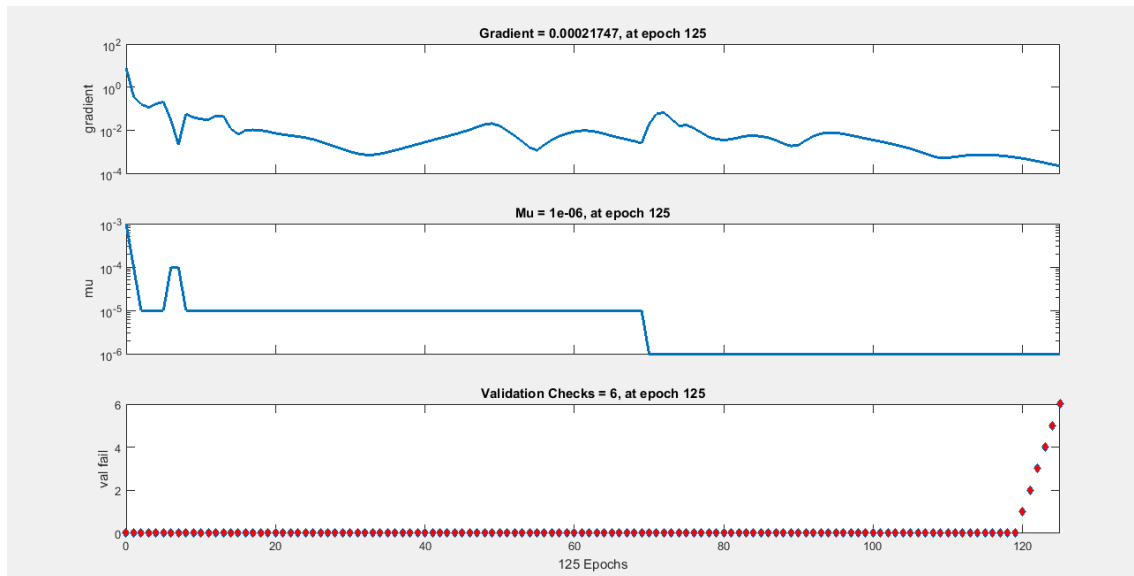


Figura 5.9: Evolución del entrenamiento durante la simulación

En la Figura 5.6 se muestra la evolución del gradiente durante el entrenamiento, y la validación del entrenamiento mediante las muestras dedicadas a la validación del total de maniobras simuladas.

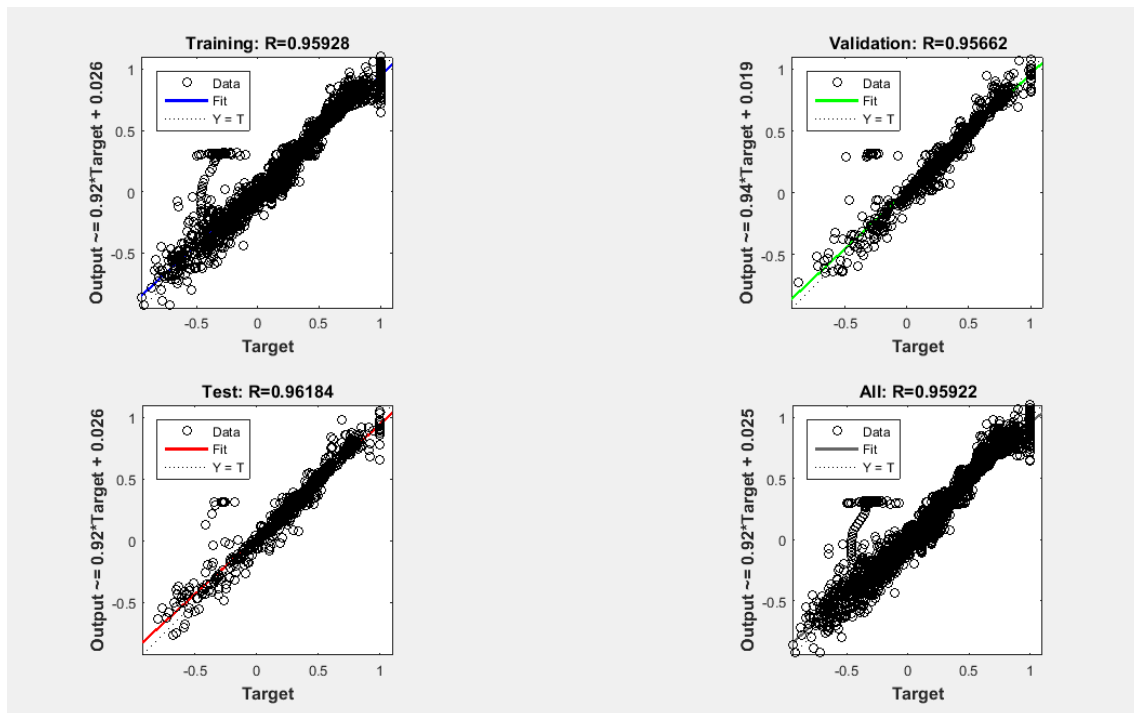


Figura 5.10: Regresión asociadas a las salidas en función de los distintos tipos de datos empleados

La Figura 5.7 muestra las regresiones lineales asociadas a los distintos paquetes de datos empleados. Regresiones para los datos empleados para el entrenamiento, para los datos empleados para el testeo de la red neuronal, para la validación de la red y finalmente una regresión lineal más asociada al total de los datos (Incluyendo entrenamiento, validación y test).



CAPÍTULO 6: CONCLUSIONES Y FUTUROS TRABAJOS

CAPÍTULO 6: CONCLUSIONES Y FUTUROS TRABAJOS

En este apartado se detallan las conclusiones que se han podido obtener de la realización de este proyecto y además se comentan los posibles trabajos o proyectos asociados, ya sea como continuación del mismo o como base o alternativa para algún estudio semejante.

6.1 Conclusiones

Este proyecto es solo el comienzo de una serie de estudios, trabajos o proyectos que podrían realizarse acerca del tema tratado o sobre temas de índole semejante.

De la parte puramente mecánica del proyecto hemos inferido lo siguiente:

- La transferencia de carga en los vehículos es un parámetro fundamental de diseño, ya que no solo afecta a la seguridad del mismo, además influye en la fiabilidad y confort de una manera muy significativa.
- Durante las realizaciones en la herramienta Trucksim hemos podido observar que, el eje en el que se produce con mayor celeridad la situación de pérdida de contacto con el firme es el trasero, por tanto la situación de vuelco siempre comienza por el eje trasero. Debido a esta circunstancia nos vemos obligados a pensar alguna medida para paliar esta tendencia, la medida más adecuada para lidiar con esto es la de sobredimensionar los amortiguadores traseros, permitiendo así que el vehículo sea capaz de circular a velocidades mayores sin que haya riesgo de vuelco.
- De la dinámica del vehículo inferimos que la masa suspendida es la que más afecta al vehículo de cara a un posible vuelco, por tanto disminuir en la medida de lo posible la masa suspendida es la mejor medida de conseguir circular a velocidades mayores de manera segura, ya que así disminuiríamos las cargas laterales que son las causantes de que el vehículo vuelque.

De la parte correspondiente a las redes neuronales y el desarrollo e implementación de la red neuronal para nuestro proyecto destacamos:

- Cabe destacar que las redes neuronales artificiales son una simplificación de las redes neuronales biológicas, es decir las redes neuronales biológicas son infinitamente más complejas que las artificiales, por lo que hay que exigir a las redes artificiales en esta medida, son una herramienta muy útil pero tienen sus limitaciones y no se les puede exigir una precisión acorde con las biológicas.
- Podemos afirmar que las redes neuronales artificiales son una manera relativamente sencilla de resolver problemas complejos, que empleando otros métodos sería mucho más complejo y no estaría al alcance de cualquiera.
- Las redes neuronales son una herramienta muy válida para la resolución de problemas complejos, a pesar de desarrollar una herramienta que se ajusta correctamente a los parámetros estipulados no significa que siempre se vayan a conseguir resultados satisfactorios, por problemas de ajuste o complejidad, o porque la elección del modelo de red neuronal no sea el más adecuado para el problema a resolver.

- Se infiere del desarrollo de la herramienta, que el éxito del modelo depende de multitud de factores como el tipo de entrenamiento, las capas ocultas empleadas, el algoritmo de entrenamiento o el número de neuronas empleadas.
- Se puede afirmar que en este caso la elección del modelo de red, entrenamiento y simulación ha sido el adecuado y el resultado ha sido muy satisfactorio y cumple con los objetivos del proyecto.
- Se puede destacar que mediante el empleo de distintos algoritmos para el entrenamiento de la red neuronal y a pesar de las diferencias de criterio de entrenamiento entre estos, los resultados obtenidos por todos los métodos de entrenamiento son bastante semejantes y además se ajustan correctamente al objetivo de la simulación.

6.2 Futuros trabajos

El campo en el que se desarrolla este proyecto es un campo con infinidad de posibilidades de mejora por lo que ya sea como punto de partida para la realización de un estudio alternativo, para ahondar más en este tema y mejorarlo o para el estudio de otro parámetro alternativo de la circulación de un vehículo automóvil podremos destacar los siguientes posibles trabajos futuros:

- Posibilidad de implementar dicho sistema en un vehículo físico, además de aumentarle el rango de maniobras y la precisión en los cálculos, con el fin de que sea un sistema activo en la seguridad de dicho automóvil.
- Emplear otro tipo de red neuronal para las simulaciones, las redes backpropagation son las más extendidas, pero para otro tipo de simulaciones podría ser más indicado emplear otro tipo de red que se ajustase más a los parámetros buscados y la realidad del modelo.
- Introducción de nuevos tipos de algoritmos de entrenamiento o aumentar la flexibilidad de los algoritmos ya existentes, así el usuario podría elegir mayor número de parámetros y así tener mayor influencia en la simulación y por tanto en el resultado final.
- Analizar el comportamiento del vehículo en condiciones adversas, tales como mala climatología como lluvia o nieve, circulación en carreteras con irregularidades o mal pavimentada.
- Extrapolar el modelo para su uso con vehículos pesados de mayor tonelaje, con remolques o autobuses.
- Empleando como base este proyecto se podría trabajar en la incorporación de medidas de seguridad en base a los resultados obtenidos.
- Realizar el proyecto con redes neuronales pero empleando una herramienta alternativa a Matlab, así se podría evaluar la calidad de la simulación en cada herramienta.
- Realización de cálculos alternativos a la transferencia de carga con las herramientas empleadas, ya que por su pluralidad ofrecen una considerable cantidad de posibilidades a la hora de estudiar diferentes parámetros en cualquier vehículo que se desee simular.
- Empleo de distintos algoritmos de entrenamiento de la red neuronal a los probados en este proyecto.
- Alternativa al cálculo con la herramienta Trucksim para el cálculo de los parámetros del vehículo necesarios para la realización de este proyecto.



- Realizar un estudio con mayor profundidad en la incorporación de elementos o sistemas para la prevención de vuelco, especialmente en vehículos pesados que como se ha explicado con anterioridad son los más propensos a esta circunstancia.



CAPÍTULO 7: BIBLIOGRAFÍA

CAPÍTULO 7: BIBLIOGRAFÍA

- [1] Highman, D.J. (2000). *“Matlab Guide”*, Siam.
- [2] *“Quick Start Guide, Version 8”*. Trucksim / Carsim.
- [3] *“Anuario estadístico de accidentes año 2013”*, Dirección General de Tráfico.
- [4] *“Matalb Help”*.
- [5] *“Real Decreto 818/2009 [9481], de 8 de mayo”*, Reglamento General de Conductores, BOE.
- [6] *“Reglamentación sobre vehículos pesados, prioritarios, especiales, de transporte de personas y mercancías y tramitación administrativa”*, Dirección General de Tráfico.
- [7] Juan Alberdi, *“Amortiguadores y suspensión, manuales de automoción”*.
- [8] Doumiati, M., Victorino, A., Charara, A. and Lechner, D.(2009) *“Lateral load transfer and normal forces estimation for vehicle safety: experimental test”*, Vehicle System Dynamics, 47: 12, 1511 – 1533.
- [9] Dr. Rajesh Rajamani, *“Lateral Vehicle Dynamics”*.
- [10] Taylor & Francis, *“Vehicle System Dynamics”*.
- [11] *“Modelos de suspensión”*, aficionadosalamecanica.net.
- [12] Trucksim online help & models, carsim.com/products/trucksim/.
- [13] *“Matalab Toolbox, Capítulo 3”*, documento de internet, Fundación Universitaria Konrad Lorenz.
- [14] R. Kamnik, F. Boettiger and K. Hunt, *“Roll dynamics and lateral load transfer estimation in articulated heavy freight vehicles”*.
- [15] Freeman, J. A., Skapura, D.M., & García – Bermejo Giner, R. *“Redes neuronales, algoritmos, aplicaciones y técnicas de programación”*.
- [16] *“Redes neuronales y sistemas borrosos”*, 3ª Edición, Bonifacio Martín del Brio y Alfredo Sanz Molina.
- [17] MATLAB User’s Guide, The MathWorks, Inc., Massachusetts, 1995.
- [18] *“Redes neuronales artificiales”*, J. R. Hilera, V. J. Martínez.
- [19] *“Fundamentos, modelos y aplicaciones”*. Ed. Rama, 1995.
- [20] S. Haykin, *“Neural networks. A comprehensive foundation”*. IEEE Press, 1994.
- [21] Load Transfer 101 – *“The First Basic Look”*
- [22] *“Weight, Mass load and Load transfer in a Vehicle”*, Manipal Institute of Technology, Dept of Mechanical and Manufacturing Engg.

- [23] “*Roll Dynamics and Lateral Load Transfer Estimation in Articulated Heavy Freight Vehicles: A Simulation Study*”, Roman Kamnik, Ken Hunt, Friedrich Boettiger.
- [24] Ray, L.R. “*Nonlinear state and tire force estimation for advanced vehicle control*”. IEEE Transactions on Control System Technology, 1995, 3(1), 117-124.
- [25] Gillespie, T.D. “*Fundamentals of Vehicle Dynamics*”, Society of Automotive Engineers, Inc. Warrendale, 1992.
- [26] Swinson, Derek B. “*Vehicle rollover*”. The physics teacher, 1995, 33 (6), 360-366.
- [27] “*Matlab & Redes Neuronales*”, Rubén Wismark Plata Cheje. Revistas Bolivianas, Revista de Información, Tecnología y Sociedad, versión impresa ISSN 1997-4044
- [28] “*Introducción a las redes neuronales aplicadas*”, documento web Universidad Carlos III de Madrid.
- [29] “*El automóvil al desnudo / Suspensiones Vehículos*”, blog de internet.
- [30] “*PED bases del Aprendizaje*”, <http://laciudadeducativa.blogspot.com.es/>. Blog de internet asociado al Grado de Pedagogía de la Universidad Nacional de Educación a Distancia, UNED.
- [31] “*Vehicle Load Transfer*”, Wm Harbin, Technical Director, BND Techsource.
- [32] “*Entrenamiento de redes neuronales basado en algoritmos evolutivos*”, Luis Federico Bertona, Universidad de Buenos Aires, Facultad de Ingeniería.
- [33] “*Simulación Numérica de la Cirugía Ortognática*”, Juan Francisco Martí Martínez, Escuela Superior de Ingenieros de la Universidad de Sevilla.
- [34] “*Redes Neuronales Artificiales Para Resolver Problemas de Programación Lineal*”, Argenis Antonio Moreno. Universidad de los Andes, Facultad de Ingeniería, Escuela de Ingeniería de Sistemas.
- [35] Tesis doctoral “*Implementación de redes neuronales sobre lógica reconfigurable*”, Magdiel Pascual García Juárez, Universidad Tecnológica de la Mixteca, División de estudios de Posgrado, México.
- [36] Fernando Mateo Jiménez. Tesis doctoral “*Redes neuronales y preprocesado de variables para modelos y sensores en bioingeniería*”, Universidad Politécnica de Valencia, Departamento de Ingeniería Electrónica.
- [37] “*Capítulo 3: La Inteligencia Artificial y las Redes Neuronales Artificiales*”. Documento de internet.
- [38] “*New method of identifying real-time Predictive Lateral load Transfer Ratio for rollover prevention systems*”, IEEE, America Control Conference, 2009. ACC '09.



[39] *“Selection of load–transfer functions for passive lateral loading of pile groups”*, M. Fraser Branby, Sarah Springman, Centre for Offshore Foundation Systems, The University of Western Australia.

[40] David J.M. Sampson & David Cebon. *“Active Roll Control of Single Unit Heavy Road Vehicles”*. Cambridge University Engineering Department, 2002.