

Instituto Tecnológico de Costa Rica
Área Académica de Ingeniería Mecatrónica



**Exploración de tolerancia a imprecisiones en aplicaciones ADAS
y su efecto en un sistema de control**

para optar por el título de
Ingeniero en Mecatrónica
con el grado académico de
Licenciatura

Braulio José Solano Rivas

Cartago, 16 de febrero de 2018

Declaro que el presente informe Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía he procedido a indicar las fuentes mediante las respectivas citas bibliográficas. En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.



Braulio José Solano Rivas

Cartago, 16 de febrero de 2018

Céd: 1-1579-0679

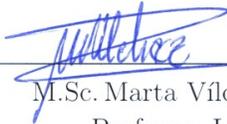
This work is licensed under a [Creative Commons](https://creativecommons.org/licenses/by-nc-sa/4.0/) “Attribution-NonCommercial-ShareAlike 4.0 International” license.



Tecnológico de Costa Rica
Área Académica de Ingeniería Mecatrónica
Proyecto de Graduación
Tribunal Evaluador

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Mecatrónica con el grado académico de Licenciatura, del Tecnológico de Costa Rica.

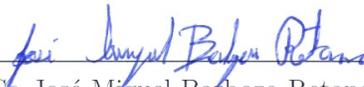
Miembros del Tribunal



M.Sc. Marta Vilchez Monge
Profesora Lectora



Dr. Ing. Juan Luis Crespo Mariño
Profesor Lector



M.Sc. José Miguel Barboza Retana
Profesor Asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por Área Académica de Ingeniería Mecatrónica.

Cartago, 16 de febrero de 2018

A mis padres y mi abuelita Zoraida

Agradecimientos

Este proyecto de graduación es la culminación de una de las etapas más significativas de mi vida. En la cual tuve el apoyo de mucha gente desde el inicio de mis estudios universitarios.

A lo largo de este tiempo, realizando trabajos o estudiando y trasnochando con los mejores compañeros, estos llegaron a ser hermanos. Hablo de Juan José Padilla, Steven Jimenez, Alonso Obando, Esteban Gonzales, Oscar Acuña, Mario Rios, Manfred Quirós y Luis Acuña.

Si algo nunca me faltó fue el apoyo de mi familia. Mis padres, Rosa Rivas y Walter Solano, los mejores profesionales que conozco. Gracias a los valores que me enseñaron he llegado tan lejos como mis sueños me lo han permitido. Los que siempre cuando tengo un proyecto en mente son los primeros en alentarme a hacerlo y no solo eso, si no terminarlo y de la mejor manera. A mi hermana y su familia por darme motivos cada vez más fuertes para ser mejor profesional.

A Marcela Amador no sé como retribuirle todo la compañía durante mi estancia en Alemania. Una hermana que me puso la vida en el camino.

A Lucía Arce le agradezco por ser quien me levanto cada vez que sentía que no podía más y por enseñarme que no es necesario estar cerca para sentirse acompañado.

A Jorge Castro por confiar en mí para la realización de este proyecto y a José Miguel Barboza por su seguimiento.

Por último, a Arys Carrasquilla, por ese voto de confianza al apoyarme durante toda mi estancia en la universidad.

Braulio José Solano Rivas

Cartago, 16 de febrero de 2018

Resumen

Los vehículos que día a día son conducidos en las calles poseen Sistemas Avanzados de Asistencia al Conductor (Advanced Driver Assistance Systems, ADAS). Estas aplicaciones ADAS le brindan apoyo al conductor para facilitarle la tarea de conducir e inclusive pueda desentenderse de algunas tareas. Este desarrollo tecnológico en la industria automotriz tiene como meta llegar a la conducción autónoma de vehículos en un 100%. El reto que se planteó para el desarrollo de este proyecto fue desarrollar 3 de estas aplicaciones que involucren Visión por Computadora (Computer Vision, CV) y comprobar su tolerancia a imprecisiones mediante la implementación Computación Aproximada (Approximate Computing, AC), lo cual aunado a un medio de pruebas donde permite observar su efecto en un sistema de control y comprobar así contra el funcionamiento “exacto” de las mismas aplicaciones.

Palabras clave: Visión por Computadora, ADAS, Aproximada

Abstract

The day-to-day vehicles that are driven on the streets have Advanced Driver Assistance Systems (ADAS). These ADAS applications provide support to the driver to facilitate the task of driving and even can get off tasks. This technological development in the automotive industry aims to reach the autonomous driving of vehicles by 100%. The challenge for the development of this project is to develop 3 of these applications involving Computer Vision (CV) and test their tolerance to errors through Approximate Computing (AC) implementation, this coupled with a test medium where its effect will be seen in a system of Control and will be checked against the “exact” operation of the same applications.

Keywords: Computer Vision, ADAS, Aproximate

Índice general

Índice de figuras	v
Índice de tablas	ix
Lista de abreviaciones	xi
1 Introducción	1
1.1 Entorno del proyecto	1
1.2 Definición del problema	3
1.3 Enfoque de la solución	4
1.4 Diagrama Causa-Efecto	4
1.5 Objetivos	5
1.5.1 Objetivo General	5
1.5.2 Objetivos Específicos	5
1.6 Metodología	5
1.7 Estructura del documento	6
2 Marco Teórico	7
2.1 Generalidades de ADAS	7
2.1.1 Niveles de Autonomía en Vehículos	7
2.1.2 Sensores en ADAS	9
2.2 Taxonomía de ADAS	11
2.2.1 Estabilidad de Conducción	12
2.2.2 Control Longitudinal	12
2.2.3 Control Lateral	14
2.2.4 Apoyo en Cabina	15
2.2.5 Apoyo al Estacionar	16
2.2.6 Luz y Vista	16
2.3 Tabla Resumen de ADAS	17
2.4 Visión por Computadora	19
2.5 Control Automático	19
2.5.1 Tipos de Controladores	19
2.6 Computación Aproximada	20
2.6.1 Sumadores aproximados	21

2.6.2	Métricas de calidad de imágenes	21
3	Control basado en aplicaciones ADAS	23
3.1	Aplicaciones seleccionadas	23
3.1.1	ADAS involucradas en Dirección	24
3.1.2	ADAS involucradas en Velocidad	24
3.2	Caso de estudio 1: Dirección	25
3.2.1	Visión Artificial Dirección	
	LDW	25
3.2.2	Sistema de Control de Dirección	30
3.3	Caso de estudio : Velocidad	32
3.3.1	Visión Artificial Velocidad	
	PPS	32
3.3.2	Visión Artificial Velocidad	
	TSR	35
3.3.3	Sistema de Control de Velocidad	38
4	Exploración de tolerancia a imprecisiones	43
4.1	Descripción del capítulo	43
4.2	Exploración de potenciales aproximaciones	43
4.2.1	Conversión de color a escala de grises	44
4.2.2	Difuminado aproximado	53
4.2.3	Combinación de funciones aproximadas	58
4.3	Implicaciones en sistemas de control	64
4.3.1	Dirección	64
4.3.2	Velocidad	65
5	Estudio Económico	67
6	Conclusiones y Trabajo futuro	71
6.1	Conclusiones	71
6.2	Trabajo futuro	72
	Bibliografía	73
A	Modelado Sistema de Dirección	79
B	Implementación en Matlab de Control de Dirección	83
C	Conjuntos de datos para entrenamiento de SVM	89
D	Modelado Sistema de Velocidad	93
E	Implementación en Matlab de Control de Velocidad	95
I	OpenCV	101

I.1	IPM	101
I.2	Funciones básicas	102
I.2.1	Conversión Color	102
I.2.2	Filtrado	102
I.2.3	Detección de bordes	102
I.3	HAAR Cascade	104
II	Machine Learning	107
II.1	Definition	107
II.2	Decision Trees	107
II.3	Neural Networks	108
II.4	Support Vector Machine	109
III	Seguro Computadora	113

Índice de figuras

1.1	Cantidad de fallecidos en sitio por año según tipo de usuario	1
1.2	Ranking mundial sobre la tasa mortalidad en carretera	2
1.3	Diagrama Causa-Efecto	4
2.1	Representación gráfica de niveles	9
2.2	Alcance de sensores	11
3.1	Diagrama de Flujo LDW	25
3.2	Etapas del procesamiento	26
3.3	Transformación de imagen original a vista Ojo de pájaro	27
3.4	Pasos de preprocesamiento LDW	27
3.5	Definición de límites y ángulo	28
3.6	Resultados LDW	28
3.7	Curvas LDW superpuestas	29
3.8	Sistema de control para la corrección de dirección.	30
3.9	Tiempos Controlador PID	31
3.10	Gráfico Ángulo vrs tiempo para vehículo de 1200kg	32
3.11	Diagrama de Flujo PPS	33
3.12	Gráfico Altura ROI vrs Distancia	34
3.13	Detección de acuerdo a tamaño de ROI	34
3.14	Diagrama de Flujo TSR	35
3.15	Desglose de etapas de SVM	36
3.16	Reconocimiento señal de 30km/h	37
3.17	Reconocimiento señal de 50km/h	37
3.18	Reconocimiento señal de 70km/h	38
3.19	Reconocimiento señal de límites de velocidad de los tres casos en estudio	38
3.20	Controlador Velocidad Diseñado	39
3.21	Ajuste de valores PID	41
3.22	Gráfico Velocidad vrs tiempo para vehículo de 1200kg	41
4.1	Resultados conversión de color a escala de grises en LDW	45
4.2	Resultados PSNR de conversión de color a escala de grises en LDW	46
4.3	Resultados SSIM de conversión de color a escala de grises en LDW.	46
4.4	Gráfica Curvas Exacta y Aproximadas en LDW.	47
4.5	Resultados conversión de color a escala de grises en PPS	48

4.6	Resultados PSNR de conversión de color a escala de grises en PPS	49
4.7	Resultados SSIM de conversión de color a escala de grises en PPS	49
4.8	Resultados conversión de color a escala de grises en PPS	50
4.9	Resultados conversión de color a escala de grises en TSR	51
4.10	Resultados PSNR de conversión de color a escala de grises en TSR	52
4.11	Resultados SSIM de conversión de color a escala de grises en TSR	52
4.12	Resultados SSIM de conversión de color a escala de grises en TSR	53
4.13	Resultados difuminado en LDW	54
4.14	Resultados PSNR de difuminado en LDW	54
4.15	Resultados SSIM de difuminado en LDW	55
4.16	Gráfica Curvas Exacta y Aproximadas en LDW	55
4.17	Resultados difuminado en PPS	56
4.18	Resultados PSNR de difuminado en PPS	56
4.19	Resultados SSIM de difuminado en PPS	57
4.20	Resultados difuminado en PPS	57
4.21	Resultados Combinación en LDW	59
4.22	Resultados PSNR de combinación de funciones en LDW	60
4.23	Resultados SSIM de combinación de funciones en LDW	60
4.24	Gráfica Curvas Exacta y Aproximadas Combinadas en LDW	61
4.25	Resultados Combinación en PPS	62
4.26	Resultados PSNR de combinación de funciones en PPS	62
4.27	Resultados SSIM de combinación de funciones en PPS	63
4.28	Resultados combinaciones en PPS	63
4.29	Gráfico Aproximado Ángulo vrs tiempo para vehículo de 1200kg	65
4.30	Ajuste de valores PID	66
A.1	Ángulos Ackerman	79
A.2	Modelado de vehículo de 4 ruedas a <i>Bicycle Model</i>	80
B.1	Modelado en Simulink de Control de Velocidad.	83
B.2	Respuesta controlador P en dirección.	84
B.3	Curva controlador P en dirección.	84
B.4	Respuesta controlador I en dirección.	85
B.5	Curva controlador I en dirección.	85
B.6	Respuesta controlador PI en dirección.	86
B.7	Curva controlador PI en dirección.	86
B.8	Respuesta controlador PID en dirección.	87
B.9	Curva controlador PID en dirección.	87
C.1	Muestras de conjuntos de datos entrenamiento de señal de 30km/h.	89
C.2	Muestras de conjuntos de datos entrenamiento de señal de 50km/h.	90
C.3	Muestras de conjuntos de datos entrenamiento de señal de 70km/h.	91
D.1	Diagrama de Cuerpo libre vehículo en movimiento	93

E.1	Modelado en Simulink de Control de Velocidad.	95
E.2	Respuesta controlador P en velocidad.	96
E.3	Curva controlador P en velocidad.	96
E.4	Respuesta controlador I en velocidad.	97
E.5	Curva controlador I en velocidad.	97
E.6	Respuesta controlador PI en velocidad.	98
E.7	Curva controlador PI en velocidad.	98
E.8	Respuesta controlador PD en velocidad.	99
E.9	Curva controlador PD en velocidad.	99
E.10	Respuesta controlador PID en dirección.	100
E.11	Curva controlador PID en dirección.	100
I.1	Transformación imagen a vista Ojo de pájaro	101
I.2	Bloques clasificadores básicos	104
I.3	Clasificadores básicos en imagen	104
II.1	Visualización de Desicion Tree	107
II.2	Muestra de Red Neuronal	108
II.3	Vectores de soporte	109
II.4	Comparación entre distintos tipo de kernel	111
III.1	Cotización INS	113

Índice de tablas

2.1	Tabla Resumen	18
3.1	Respuestas transitorias en sistema de control de dirección según tipo de controlador	31
3.2	Respuestas transitorias en sistema de control de velocidad según tipo de controlador	40
4.1	Porcentaje de disminución de calidad	61
5.1	Gastos incurridos para la realización del proyecto	68
A.1	Valores numéricos de vehículo	81
A.2	Nomenclatura	82
D.1	Nomenclatura	94

Lista de abreviaciones

Abreviaciones

AC	Computación Aproximada
AD	Conducción Autónoma
ADAS	Sistemas Avanzados de Asistencia al Conductor
CV	Visión por Computadora
KIT	Instituto Tecnológico de Karlsruhe
ML	Aprendizaje de Máquina
NN	Redes Neuronales
ROI	Región de Interés
SVM	Máquina de Vectores de Soporte
TEC	Tecnológico de Costa Rica

Capítulo 1

Introducción

1.1 Entorno del proyecto

La automatización en la conducción de automóviles no solo está inspirada en crear vehículos inteligentes, si no en lograr con esa autonomía una disminución significativa de las cifras elevadas que existen en relación a las muertes y accidentes en carretera. En el caso particular de Costa Rica las cifras son alarmantes. En el 2015, los accidentes automovilísticos fueron la tercera causa de muertes en el país [1]. Según los datos de la Policía de Tránsito de Costa Rica, para el año 2016 se contabilizaron 448 muertes por accidentes de tránsito, lo cual constituye un 12.6% más que en el 2015 y 26.2% más que en el 2014; más preocupante aún es saber que el recuento que hace la Policía de Tránsito solamente incluye las muertes en el sitio, sin contabilizar quienes tuvieron un accidente, sin embargo, fallecieron posteriormente en los centros médicos. En la Figura 1.1 se muestran los datos de defunciones según distintos usuarios de las calles y carreteras.

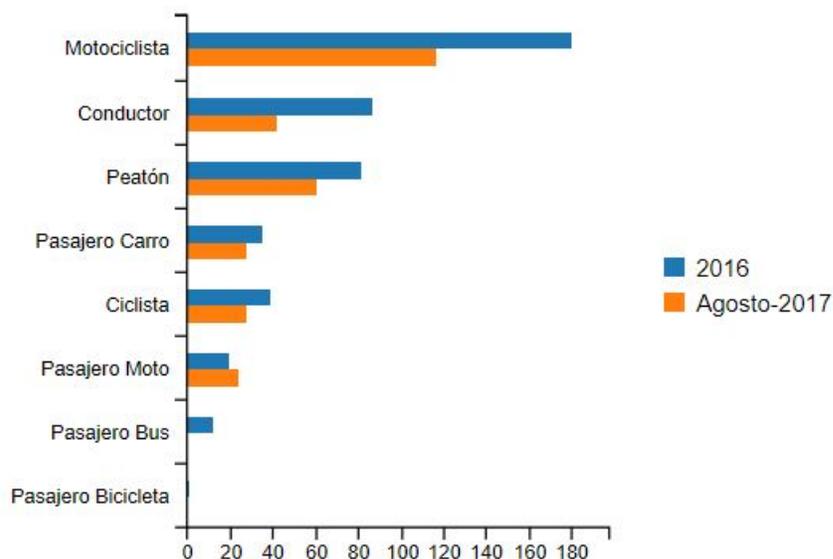


Figura 1.1: Cantidad de fallecidos en sitio por año según tipo de usuario [2].

A partir de esta información, el Consejo de Seguridad Vial de Costa Rica indica que los conductores de vehículos, los peatones y los pasajeros de carro son los que más se ven afectados en los accidentes en carretera, superados únicamente por accidentes en los que se ven involucrados motociclistas [2]. La suma de los 3 tipos de usuarios mencionados anteriormente es mayor a la cantidad de fallecidos en sitio bajo la definición de motociclistas. De ahí la necesidad de implementar sistemas que protejan directamente esos usuarios que se ven directamente afectados.

Las colisiones entre vehículos son el tipo de accidentes que cobran más vidas, con una cifra cuantificada en 213 muertes, lo cual representa un 47% del total de muertes del 2016. Otras causas como atropellos dejaron 70 fallecimientos; autos que se salieron de la vía, 70; vuelcos, 32; choques contra objetos fijos, 31; derrapes, 14; atropellos en los que el responsable se dio a la fuga, 14; y 4 conductores que se quedaron dormidos [3]. En los casos anteriores el exceso de velocidad afectó negativamente la reacción del conductor.

Costa Rica, a nivel mundial, se encuentra en la mitad de un ranking que incluye 128 países acerca de la mortalidad en carretera, con una alarmante cifra de 13.9 por cada 100000 habitantes [4], como se muestra en la Figura 1.2.

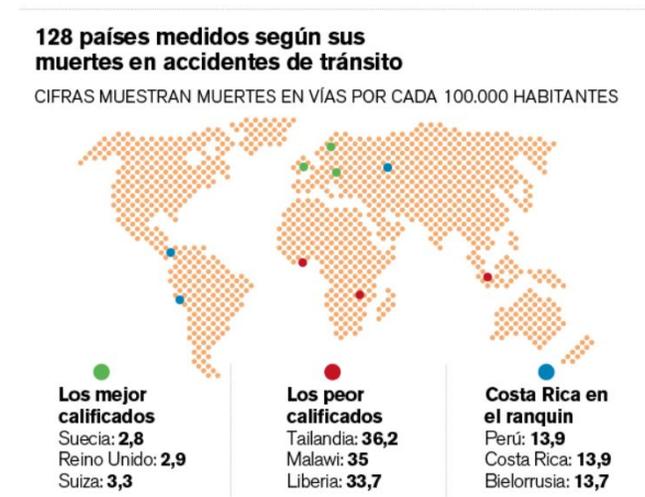


Figura 1.2: Ranking mundial sobre la tasa mortalidad en carretera [4].

La conducción autónoma es la meta que se ha trazado la industria automotriz, no solo para hacer la conducción más sencilla para el usuario si no también para convertirla en una forma más segura de movilización. Es posible lograr lo anterior mediante el uso de los Sistemas Avanzados de Asistencia al Conductor (*Advanced Driver Assistance Systems, ADAS*) basados en Visión por Computadora (*Computer Vision, CV*) y la Computación Aproximada (*Approximate Computing, AC*). Estas son tendencias que al trabajarse en forma conjunta presentan un panorama de investigación prometedor para generar contribuciones que permitan reducir los decesos en carretera.

Actualmente, tanto en la industria como en la academia, existe interés por desarrollar aplicaciones que permitan a los automóviles ser capaces de entender mejor el contexto en el que se desplazan y así lograr asistir al conductor en el proceso de conducir. Las aplicaciones conocidas como ADAS abarcan sistemas como asistencia para estacionamiento, control de la velocidad crucero, detección y reconocimiento de señales de tránsito, detección de peatones, alerta de salida de carril, entre otros. En muchas de estas aplicaciones, la visión por computadora es una pieza clave para que estos sistemas de asistencia cuenten con entradas de datos de los cuales se puede obtener información variada y abundante. Sin embargo, debido a la cantidad de información que se procesa, así como las restricciones de tiempo disponible para la generación de los resultados, hacen que estas aplicaciones requieran una gran capacidad de procesamiento computacional.

Por su parte, en los años recientes ha surgido la AC como un paradigma de diseño que participa en la búsqueda de sistemas computacionales energéticamente eficientes. Esta técnica de diseño busca reducir el esfuerzo computacional requerido (en términos de área requerida, tiempo de ejecución y consumo de potencia) reduciendo la exactitud y/o precisión de ciertos cálculos computacionales, lo cual significa que los resultados presentan errores debido a que ciertas computaciones se realizan de una forma imprecisa a propósito, ya sea reduciendo la complejidad de los circuitos, disminuyendo la tensión de operación u omitiendo ciertas computaciones [5].

1.2 Definición del problema

Generalidades

El proyecto que se plantea surge de la necesidad de responder a las interrogantes:

- a) ¿Qué tanta tolerancia a imprecisiones poseen las aplicaciones ADAS?
- b) ¿Cuál sería el efecto en un sistema de control?

Mudando de una computación exacta a una aproximada en este tipo de aplicaciones, se permitirá manipular la información de una manera diferente requiriendo un menor esfuerzo computacional. La calidad de salida se verá afectada y por esto se desea explorar la tolerancia a imprecisiones que poseen las aplicaciones ADAS y su efecto en un sistema de control. Las aproximaciones se introducirán en estas aplicaciones: a) modificando los algoritmos existentes, y/o b) mediante modelos que representen el comportamiento de hardware aproximado. Empleando las salidas obtenidas se podría valorar la funcionalidad de estas aplicaciones bajo este funcionamiento diferenciado.

Síntesis del problema

El proyecto consiste en explorar la tolerancia a imprecisiones en aplicaciones ADAS sin que se afecte significativamente la respuesta en un sistema de control teórico que se propone.

1.3 Enfoque de la solución

Se partirá de las siguientes premisas:

- a) Las aplicaciones ADAS involucran manejo de información del entorno.
- b) El procesamiento de imágenes y video proporcionan información del entorno.

Con base en lo anterior, los **Sistemas de Visión Artificial** serán los encargados de la adquisición de la información y el **Procesamiento de Señales** del manejo de la misma. Estas son áreas de conocimiento fundamentales para la investigación a realizar. Una vez que la información es obtenida del medio y debidamente procesada mediante aplicaciones ADAS se necesitará conocer su efecto en un sistema de **Control**. Para ello es necesario generar un **Modelo** del sistema de velocidad y dirección de un vehículo, esto mediante el **Análisis de Sistemas**. Apoyado en las áreas Mecatrónicas mencionadas anteriormente, se tendrá una visión completa del problema con el fin de generar una solución al mismo.

1.4 Diagrama Causa-Efecto

Para una comprensión adecuada del problema se planteó el siguiente diagrama Causa-Efecto que se presenta en la Figura 1.3. En el que se muestran las variables que justifican las causas que afectan directamente el problema a tratar.

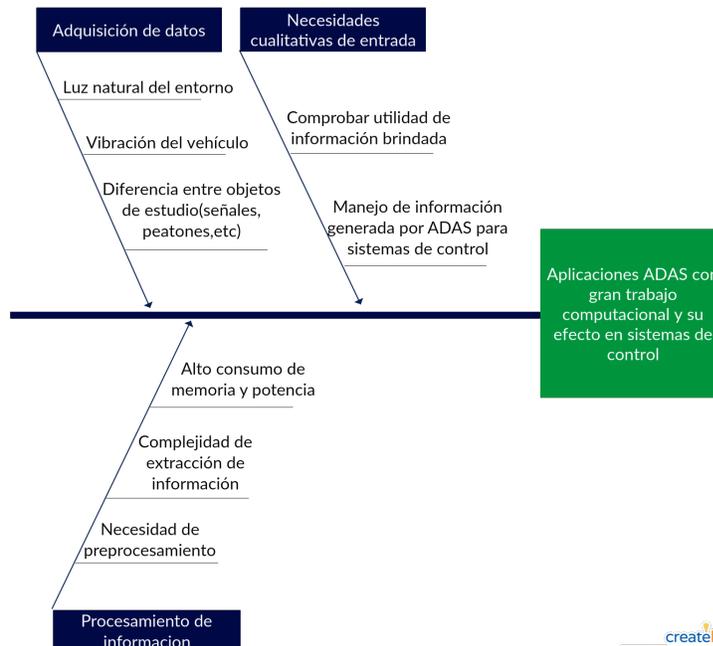


Figura 1.3: Diagrama Causa-Efecto

1.5 Objetivos

1.5.1 Objetivo General

Analizar la tolerancia a imprecisiones en aplicaciones ADAS y su efecto en un sistema de control.

1.5.2 Objetivos Específicos

- Identificar las aplicaciones ADAS con las cuales se realice la exploración.
- Desarrollar las aplicaciones ADAS en las cuales se procesen imágenes y/o videos.
- Diseñar y validar el funcionamiento de un sistema de control automático para la dirección y velocidad de un vehículo bajo un modelado analítico del sistema.
- Evaluar potenciales fuentes de aproximación en estas aplicaciones ADAS.

1.6 Metodología

Se presenta el plan de acción para la resolución del problema bajo cuatro etapas:

1. Investigar en publicaciones científicas y el mercado la variedad de ADAS existentes para redactar una taxonomía donde se muestren las mismas bajo distintos escenarios e identificar un conjunto de tres aplicaciones basadas en procesamiento de imágenes y video.
2. Implementar las aplicaciones ADAS seleccionadas con el apoyo de códigos fuente ya existentes y/o programados desde cero para validar su funcionamiento “exacto”. Esto realizado en lenguaje c/c++ con el uso de librerías de OpenCV las cuales brindan funciones adicionales de Visión por Computadora (*Computer Vision*, CV).
3. Diseñar y validar el funcionamiento de un sistema de control automático para la dirección y velocidad del vehículo. Este control debe reaccionar a la información que reciba de los sistemas ADAS en desarrollo. Además, su diseño y verificación de funcionamiento se realizará en la herramienta de Simulink de Matlab poner el símbolo de marca registrada correspondiente.
4. Explorar un conjunto de potenciales fuentes de aproximación, las cuales se determinarán a partir de las características mismas de las aplicaciones y observar así su efecto en el sistema de control propuesto.

1.7 Estructura del documento

El presente documento consta de 6 capítulos donde en el Capítulo 1 se brinda un primer vistazo de lo que consisten las ADAS y la AC, además de un planteamiento de lo que se va a realizar en la exploración. El Capítulo 2 consiste en la realización de un estudio de las ADAS que existen actualmente y su clasificación por familias y sensores utilizados. La realización de programas en c/c++ con la ayuda de openCV son presentados en el Capítulo 3 al igual que la prueba de estos en un sistema de control simulado en Matlab. El concepto de computación aproximada se introduce en el Capítulo 4 y se realiza el estudio con resultados del Capítulo 3 y 4. Un estudio económico del proyecto se desarrolla en el Capítulo 5 y en el Capítulo 6 se presentan conclusiones y recomendaciones obtenidas luego del análisis de los Capítulos 3 y 4. Al final del documento se muestran los apéndices y anexos que se consideren oportunos de agregar.

Capítulo 2

Marco Teórico

2.1 Generalidades de ADAS

2.1.1 Niveles de Autonomía en Vehículos

La Sociedad de Ingenieros Automotrices (*Society of Automotive Engineers*, SAE) en la norma J3016 establece una serie de definiciones y clasificaciones con el fin de que se tenga una terminología en común entre industria, academia y todos los involucrados en lo relacionado a conducción autónoma. En dicha norma se expone, una clasificación por niveles para la autonomía al conducir, partiendo del nivel cero donde se posee autonomía de conducción nula hasta el nivel 5 que asegura que el vehículo puede ser conducido por sí mismo sin necesidad de un conductor. Esta clasificación mencionada ayuda a mantener a todos los sectores involucrados alineados, reduciendo confusiones sobre el nivel de autonomía que posee un objeto en estudio. [6].

Nivel 0. Automatización Inexistente

El nivel 0 viene dado por cero asistencia y cero autonomía de manejo. Toda la dinámica de conducción es ejecutada por un conductor humano. En este nivel no existen los modos de manejo ni el control de aumento o descenso de velocidad.

Nivel 1. Asistencia al Conductor

El nivel 1 es apoyado por modos de conducción, asistencia de aceleración y desaceleración. Se le da al conductor libertad de pies por lo que se le conoce a este nivel como “Feet-off” o pies libres.

Nivel 2. Automatización Parcial

El vehículo posee varios sensores para obtener datos del entorno y se cuenta con un sistema de control de velocidad que permite variedad de modos de conducción. Por lo anterior, en este nivel se le da libertad al conductor de manos y debido a ello se le conoce como “Hands-off” o manos libres.

Nivel 3. Automatización Condicional

El sistema se encarga de lo que concierne la conducción pero solicita intervención del chofer en ocasiones. Se le conoce como “Eyes-off” u ojos libres pues le da la libertad al conductor de realizar tareas en las cuales necesite su visión sin afectar la conducción del vehículo.

Nivel 4. Automatización Elevada

No es necesaria la intervención del usuario ya que el vehículo cuenta con las herramientas de hardware y software necesarias para ser conducido por sí solo y la presencia del conductor es únicamente necesaria para casos excepcionales. A este nivel debido a lo mencionado se le conoce como “Brain-off” o Mente libre.

Nivel 5. Automatización Total

Para esta instancia el vehículo toma en cuenta entorno y todos los aspectos de la dinámica de manejo. Por ello no es necesario el conductor humano.

La Sociedad Internacional de Ingenieros Automotrices plantea que existen, más allá de los distintos niveles, una división general que comprende 2 grupos que abarcan 3 niveles de autonomía cada uno [6].

- El conductor humano controla el ambiente de conducción
En el cual los niveles 0, 1 y 2 se ven representados.
- Sistema de conducción automatizado controla el ambiente de conducción
Los niveles 3, 4 y 5 forman parte de esta división.

En la Figura 2.1 se muestra de manera gráfica los niveles de automatización así como las 2 grandes áreas según el encargado de las tareas de conducción.

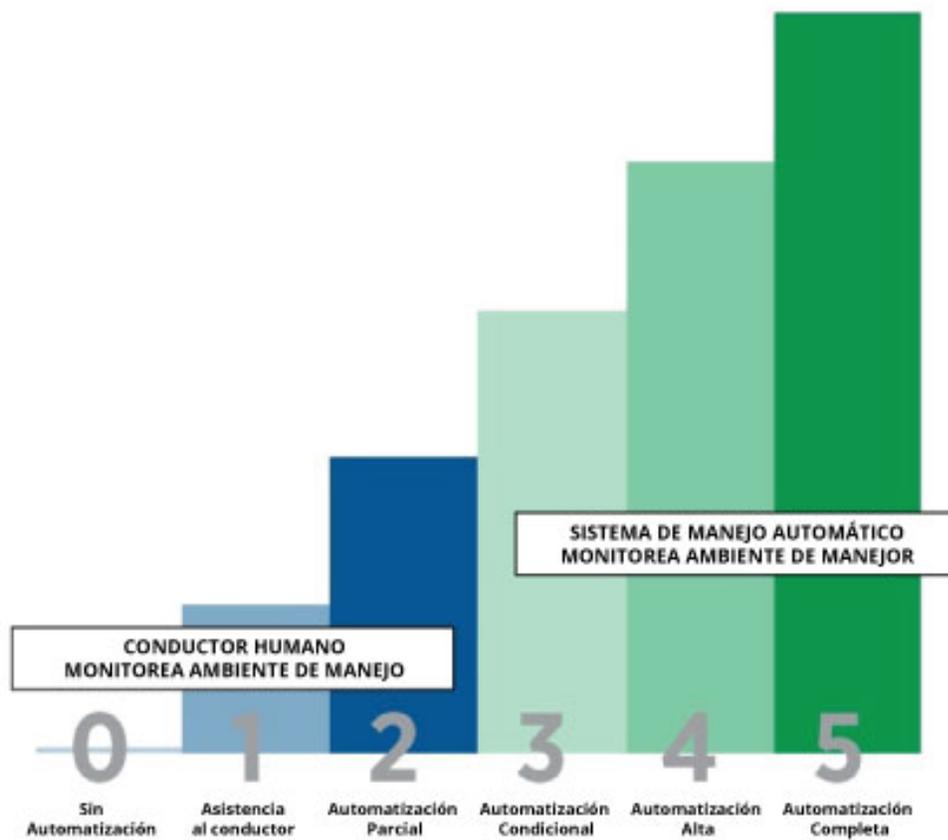


Figura 2.1: Representación gráfica de niveles [7].

2.1.2 Sensores en ADAS

Las aplicaciones ADAS requieren mediciones de variables físicas del entorno que rodea el vehículo. Esta información viene dada por distintos sensores que se describen a continuación.

Cámaras Monoculares

Las cámaras monoscópicas o monoculares tienen un solo sensor de cámara para captura de video, estas capturas requieren ser procesadas y analizadas posteriormente. Los Sistemas Monoculares también se conocen como “Single-Eyed System” y permiten la obtención de imágenes únicamente en 2 dimensiones [8]. Estos sensores son recomendados en [8] para:

- Líneas
- Peatones
- Señales de tránsito

Cámaras Stereo

Un sistema con dos cámaras, separadas una de otra se llama Sistema de Visión Stereo. El costo de este tipo de sensor es 1.5 veces el de un sensor monocular. Los sensores deben estar un mínimo 25-30cm separados entre sí para obtener imágenes en 3 dimensiones [8].

Estos sensores son recomendados en [8] para:

- Detección de objetos
- Cálculo de distancia

Cámaras Infrarrojas

Existen aplicaciones de visión artificial que requieren soluciones más allá del espectro visible debido a las características de emisión de los objetos o de la aplicación a evaluar. Por tanto, existen cámaras que permiten detectar señales que se encuentran en el espectro de radiación infrarroja, desde el rango visible hasta el de microondas [9].

Dentro de los sensores que utilizan las cámaras infrarrojas, existen 2 clasificaciones definidas en [10]: Infrarrojo cercano (Near-infrared, NIR) e infrarrojo lejano (Far-infrared, FIR). Por otro lado, los sistemas NIR tienen un mejor desempeño en la detección, siendo utilizados para ubicar peatones u otros objetos del entorno del vehículo. BMW, Mercedes-Benz y Honda utilizan este tipo de sensores en sus sistemas de visión nocturna.

Ultrasonido

Un sensor ultrasónico emite ondas ultrasónicas en el aire y detecta las ondas reflejadas de un objeto. Hay muchos usos para los sensores ultrasónicos, por ejemplo: en sistemas de alarma de la intrusión, los accesos automáticos y los sensores de retroceso para los automóviles. Acompañado por el rápido desarrollo de la tecnología de procesamiento de información [11].

LIDAR

Los Sistemas de Detección y Alcance de Luz (*Light Detection And Ranging*, LIDAR) utilizan el mismo principio que el radar y se desarrollan para una amplia gama de aplicaciones de localización, rango y generación de perfiles. Este sistema consiste en un láser capaz de transmitir la luz (pulsada o continua) sobre la gama de interés requerida con un receptor de alta velocidad y de poco ruido para el análisis de la señal reflejada. La luz transmitida interactúa y es reflejada por el objetivo. Una porción de la luz emitida se refleja de acuerdo a las características del entorno, lo cual permite determinar localización y propiedades del entorno mismo [12].

Alcance de sensores

En la Figura 2.2 se muestra de manera gráfica y descriptiva el alcance de los distintos tipos de sensores que se utilizan para las distintas aplicaciones ADAS.

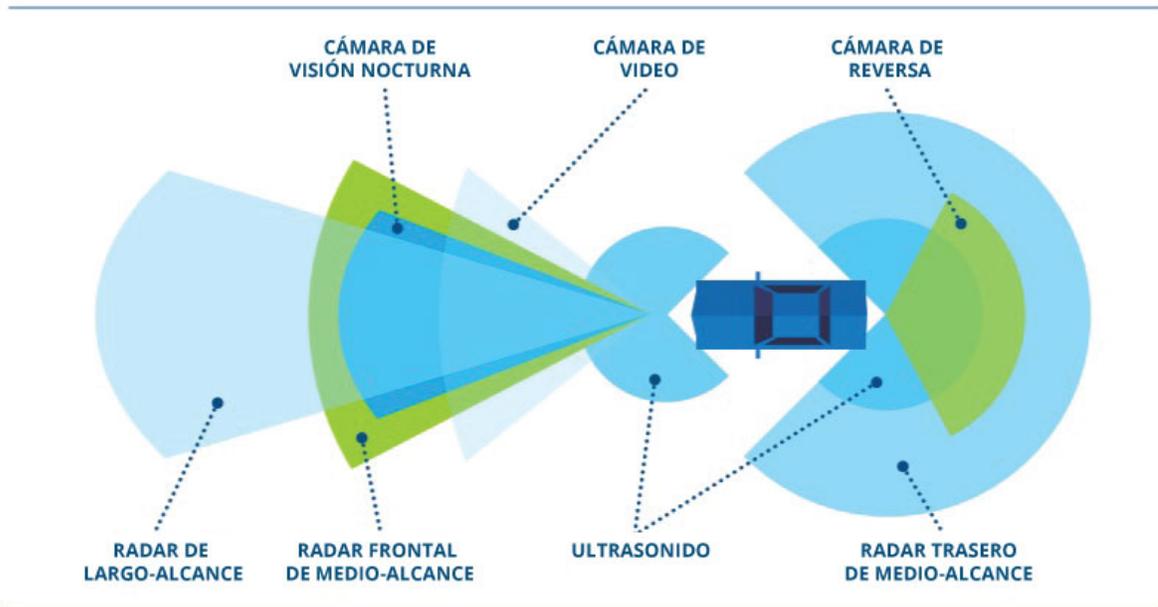


Figura 2.2: Alcance de sensores [7].

2.2 Taxonomía de ADAS

Intel propone en [7] una clasificación para las aplicaciones ADAS que se pueden encontrar en los distintos modelos de vehículos que hay en el mercado con el fin de facilitar su comprensión y estudio.

La clasificación propuesta hace una división en 6 grupos:

- Estabilidad de Conducción
- Control Lateral
- Control Longitudinal
- Apoyo al Estacionar
- Apoyo en Cabina
- Luz y Vista

Con la finalidad de profundizar el tema y conocer las principales aplicaciones ADAS dentro de las familias mencionadas anteriormente, para cada una de estas se presenta: su nombre, su acrónimo, una breve descripción y finalmente los sensores que se utilizan para cumplir con su función.

2.2.1 Estabilidad de Conducción

En esta primera división se presentan las ADAS que le permiten al conductor conducir sin preocuparse de que el vehículo vaya a tener inconvenientes movilizándose sobre distintos tipos de terreno o que se pierda el control al momento de frenado.

Control de Tracción (*Traction Control System, TCS*)

El TCS tiene como tarea prevenir la pérdida de adherencia de las ruedas, de forma que éstas no patinen cuando el conductor excede la aceleración del vehículo o cuando el suelo presenta secciones resbalosas. El sistema actúa con el fin de reducir el par de giro y así recuperar la adherencia entre neumático y suelo, realizando una de las siguientes acciones. Para lograr aumentar la adherencia primero retarda o suprime la chispa a uno o más cilindros, luego reduce la inyección de combustible a uno o más cilindros, para finalmente frenar la rueda.

Control Electrónico de Estabilidad (*Electronic Stability Control, ESC*)

Este sistema utiliza para su control un frenado automático de las ruedas individuales para evitar el cambio abrupto (derrapando) o no lo suficientemente rápido (trabándose). El ESC no puede aumentar la tracción disponible, pero maximiza la posibilidad de mantener el vehículo bajo control y en la carretera durante maniobras extremas [13]. Esta ADAS trabaja tan rápido que los conductores no perciben las necesidades de corrección de dirección. Si los conductores se frenan porque la curva es más o menos aguda de lo previsto, el sistema sigue siendo capaz de generar frenado desigual si es necesario para corregir el rumbo. Utiliza sensores de velocidad, ángulo y acelerómetros.

2.2.2 Control Longitudinal

El Control Longitudinal le brinda al conductor herramientas para que la conducción en carreteras sea más sencilla y segura.

Control Automático de Crucero (*Automatic Cruise Control, ACC*)

Esta es la versión avanzada de la función actual de control velocidad crucero. Hoy en día, el control de velocidad crucero acepta un valor ajustado de velocidad desde el controlador y gestiona el sistema de control del vehículo para mantener la velocidad de destino. Puede tomar la información adicional de la detección de señales de tránsito. Sin embargo, el control tradicional de velocidad crucero no permite que el vehículo ajuste la velocidad de forma dinámica según la distancia relativa. El ACC permite un ahorro de combustible al mantener un manejo a velocidad constante y le permite adicionalmente al conductor

descansar sus pies [14]. También llamados Asistente Inteligente de Velocidad (*Intelligent Speed Assistance*, ISA), estos comparan la velocidad del vehículo con el límite de velocidad fijado en mapas digitales o señales de límite de velocidad detectados. Así, estos sistemas restringen la velocidad del vehículo según el límite de velocidad vigente [15]. Esta aplicación requiere de cámaras monoscópicas, sensores de velocidad y sistemas LIDAR.

Reconocimiento de Señales de Tránsito (*Traffic Sign Recognition*, TSR)

Consiste principalmente en la detección de señales de tránsito y se basan principalmente en la detección de color y forma. Existen muchas investigaciones sobre la detección e identificación por color, siendo una de ellas la detección de bordes [16]. Para el TSR se utilizan cámaras monoculares para obtener la información del entorno.

Reconocimiento de Luces de Tránsito (*Traffic Signal Lamp Recognition*, SLR)

El SLR se utiliza para la detección de luces que brindan información en la navegación en carretera, como por ejemplo, semáforos, luces de paso o de emergencia. Son sistemas críticos debido a la rapidez de detección y la precisión que se logra. Además, la luz solar no es constante a lo largo del día, por lo que cambia el entorno de medición y entorpece la toma de datos [17]. EL SLR utiliza cámaras monoscópicas para la medición del entorno.

Freno Automático de Emergencia (*Automatic Emergency Brake*, AEB)

Este sistema es uno de los más importantes de los ADAS y de este dependen muchos otros como: Control de Alerta de Peatones, Sistema de Evasión de Colisiones, Control Crucero Adaptativo, etc. Lo que realiza es enviar la señal a los frenos para que estos se accionen automáticamente según las circunstancias para evitar colisiones [18]. Los sensores utilizados son cámaras, sensores de velocidad y sistemas LIDAR.

Control de Alerta de Peatones (*Pedestrian Protection System*, PPS)

Lo que se plantea con esta aplicación es detectar transeúntes sobre la ruta del vehículo mediante CV. Esto para evitar accidentes de tránsito en los que se atente directamente la integridad de un eventual peatón. Una vez detectado se trabaja en conjunto con otras ADAS como CAS o AEB para poder cumplir su objetivo [19]. Para la detección de peatones se utilizan las cámaras, sistemas de ultrasonido, LIDAR o infrarrojos.

Alerta Pre-Colisión (*Pre-Collision Warning, PCW*)

Los sistemas que pueden detectar y alertar al conductor acerca de accidentes inminentes se conocen como sistemas pre-colisión. Estos sistemas tensan los cinturones de seguridad y realiza un frenado paulatino. En primer lugar activa una alerta visual o sonora y cierra las ventanas, luego activa un frenado suave para captar la atención del conductor, seguidamente aplica una acción de frenado que provoque mayor reducción de velocidad, para finalmente frenar con todo poder [14]. Estos sistemas surgieron por primera vez a mediados de la primera década del siglo XXI por fabricantes de vehículos como lo son Toyota y Honda. El PCW utiliza sistemas LIDAR y cámaras para obtener información que le permita detectar y actuar una posibilidad de colisión.

Sistema de Evasión de Colisiones (*Collision Avoidance System, CAS*)

El sistema anti-colisión por frenado es apropiado a velocidades bajas del vehículo y el de evasión de colisión por dirección es apropiado a velocidades elevadas. Los sistemas CAS requieren de sensores frontales de igual forma que el sistema de control de cruceo adaptativo. Si el diferencial de velocidad entre el vehículo y cualquier objeto frente a él es elevado, entonces el sistema puede ser capaz de realizar una serie de acciones con el fin de evitar una colisión. Los sistemas más simples para la evasión en colisión emiten una advertencia que hará al conductor enterarse de la situación. Luego el sistema aleja el vehículo del obstáculo por medio de la dirección en coordinación con el sistema de frenado [20]. Los CAS utilizan sistemas de radar, LIDAR y cámaras frontales para determinar la situación en la navegación en carretera.

2.2.3 Control Lateral

El Control Lateral le provee al conductor las herramientas necesarias para realizar cambios de carril de manera segura.

Detección de Punto Ciego (*Blind Spot Detection, BSD*)

Los puntos ciegos son áreas fuera de un vehículo que el conductor es incapaz de ver. Estos pueden ser causados por los pilares de la ventana, reposa-cabezas, pasajeros y otros objetos. A distancias incluso moderadas, un punto ciego causado por un pilar puede oscurecer objetos grandes como los coches y las personas. Otro tipo de punto ciego vehicular existe en el espacio entre la visión periférica del conductor y el área reflejada por los espejos retrovisores. Este tipo de punto ciego puede impedir la observación de vehículos enteros, lo que convierte esta maniobra en una acción peligrosa en el manejo en carreteras. El BSD genera una alerta según sea la proximidad del objeto al punto ciego detectado [14]. Los sensores que generalmente se utilizan son cámaras y radares que permitan obtener una visión periférica de 360 grados del vehículo.

Alerta de Salida del Carril (*Lane Departure Warning, LDW*)

Los sistemas de alerta de salida de carril son un grupo de tecnologías de seguridad que están diseñados principalmente para mantener la pista visual de las marcas de carriles para identificar cualquier desviación del carril y con esto evitar la incidencia en los accidentes en carreteras y autopistas. Al advertir al conductor o incluso tomar acciones correctivas automáticas, estos sistemas son capaces de prevenir muchas colisiones y accidentes [21]. Las cámaras son los sensores utilizados para este fin.

Asistente de Cambio de Carril (*Lane Change Assistance, LCA*)

Los cambios de carril son maniobras estresantes para los conductores, particularmente durante flujos de alta velocidad del tráfico. El LCA tiene como objetivo ayudar al conductor en la maniobra de cambio de carril. Así, el LCA toma en cuenta las dinámicas y características de cada sistema vehículo/conductor para ejecutar las maniobras de cambio de carril de forma eficaz y aceptada por el conductor y que al mismo tiempo sea una acción segura [22]. Para tal fin, el LCA utiliza la información del BSD y el LDW.

Un ADAS que tenga en cuenta las dinámicas y características de cada sistema individual de vehículo/conductor durante las maniobras de cambio de carriles es eficaz y aceptable para los conductores sin sacrificar la seguridad. Las entradas de esta aplicación son los datos obtenidos por BSD y LDW.

2.2.4 Apoyo en Cabina

El apoyo en cabina le brinda al conductor facilidades tanto de localización de lugares a donde desee ir y posición actual, así como de alerta ante cansancio del mismo.

Sistema de Navegación (*Navigation System, NS*)

Permite conocer la posición del vehículo y el lugar de destino. A partir de ello, genera rutas para que el conductor se movilice de un punto inicial a un punto final con todas las indicaciones necesarias para realizarlo de manera exitosa. El GPS es la entrada que le permite al sistema conocer la localización del vehículo de manera precisa.

Monitoreo del Estado del Conductor (*Driver State Monitoring, DSM*)

Los sistemas de alerta de los conductores están estrechamente relacionados con los sistemas de alerta de salida de carril. Mientras que los sistemas de advertencia de salida de carril están diseñados para prevenir la desviación bajo cualquier circunstancia, los sistemas de alerta de los conductores están específicamente orientados a identificar signos de fatiga del conductor. En [23] se expone otro sistema que va un paso más allá, monitorizando los

ojos y la cara del conductor para ver si hay signos de somnolencia. Si el sistema determina que el conductor tiene problemas para permanecer despierto, esto con la ayuda de una cámara en el tablero del vehículo, puede tomar medidas correctivas en la conducción para evitar evasiones de carril o similares.

2.2.5 Apoyo al Estacionar

La tarea de estacionar en ocasiones no es sencilla pero estas aplicaciones le brindan apoyos al conductor para realizar esta tarea de manera fácil, rápida y precisa.

Cámara Visión Trasera (*Rear View Camera, RVC*)

Brinda apoyo al conductor al momento de conducir en reversa, principalmente cuando se estaciona en reversa. Este sistema puede trabajar con cámaras monoculares, stereos o una composición de las anteriores para generar una visión 360 grados del vehículo y observar todo el alrededor. Este sistema contempla funciones básicas de transmisión en tiempo real de lo que la cámara de retroceso graba. Este sistema no interviene ni genera alarmas [24].

Asistencia Inteligente de Parqueo (*Intelligent Parking Assistance, IPA*)

El sistema de asistencia para el estacionamiento ayuda al conductor a dirigir y revisar la parte trasera para asegurarse de que el área esté despejada al hacer retroceder el vehículo. Cuando un conductor pone un vehículo en reversa para entrar en un espacio de estacionamiento, el sistema de asistencia de estacionamiento muestra dos tipos de información. En primer lugar una imagen tomada por un sensor de imagen colocado en la parte trasera del vehículo, que toma puntos de vista que no pueden ser vistos por el conductor. En segundo lugar se le despliega al conductor una imagen superpuesta a la imagen original indicándole una curva con la dirección que debe de tomar [25]. El ultrasonido y las cámaras son los que brindan la información necesaria para que el IPA informe al conductor.

2.2.6 Luz y Vista

La Luz es factor que afecta de manera positiva y negativa la conducción de vehículos y la CV. Positivo pues permite tener la capacidad de visualizar lo que se encuentra frente al vehículo en todo momento. Por el contrario, la ausencia de luz no permite visualizar objetos, carriles, señales que estén las calles, etc. Lo anterior aplica tanto para sensores de visión así como para ojos humanos. Debido a ello, el conjunto de ADAS que se presentan en esta sección acusan a la necesidad de darle al conductor una adecuada visualización del entorno del vehículo.

Visión Nocturna (*Night Vision, NV*)

La detección de los peatones y los animales presenta una dificultad adicional en entornos nocturnos debido a la oscuridad. La capacidad de detectar estos debe ser el principal criterio de rendimiento, inclusive cuando se enfrenta a los faros de vehículos que se aproximan de frente en sentido contrario. El sistema infrarrojo lejano ha demostrado ser superior al sistema infrarrojo cercano en términos de distancia peatonal para la detección [25] por lo que se utilizan sensores FIR.

Sistema Adaptativo de Luces Frontales (*Adaptive Frontlight System, AFS*)

El Sistema Adaptativo de Luces Frontales es una parte del sistema de seguridad activo de un vehículo de gama media-alta en adelante, proporcionando una visión optimizada al conductor durante la noche y otras condiciones de la mala vista de la carretera adaptando el ángulo y la intensidad de la linterna. Para definir el ángulo y la intensidad de la linterna se utiliza la velocidad del vehículo, el ángulo del volante y las condiciones atmosféricas [26].

Sistema de Sensado de Lluvia (*Rain Sensor System, RRS*)

El sensor de lluvia está instalado en el parabrisas en una sección del mismo con una tasa de transmitancia determinada para permitir el paso de la intensidad especificada de la radiación. La sección de emisión de la unidad del sensor de lluvia emite rayos infrarrojos contra el parabrisas, donde al llover detecta la cantidad de gotas de lluvia al recibir rayos reflejados mediante un fotodiodo. La sección de detección en el parabrisas se encuentra justo por encima del punto central entre el LED y el fotodiodo [27]. Los fotodiodos son los que se encargan de detectar si entre el emisor y el receptor hay suficiente paso de luz. Si debido a la refracción generada por las gotas el receptor no percibe el haz emitido por el emisor las escobillas realizan su función de manera automática.

2.3 Tabla Resumen de ADAS

En la sección 2.2 se presentó la recopilación realizada acerca de las ADAS existentes hasta el 2017. De esta forma, en la Tabla 2.1 se muestra un resumen de la información expuesta en dicha sección. En esta tabla se muestra la familia a la que pertenece, el nivel de autonomía que aporta así como los sensores que utiliza. Importante mencionar que a pesar de que hay varias ADAS con nivel de autonomía 0, no significa que no sean importantes. Muchas de ellas combinadas entre si, o bien combinadas con otras que tengan niveles superiores permiten fortalecer, a nivel de seguridad, diversos sistemas de conducción. Es decir, que por si solos no proveen autonomía en la conducción pero si colaboran en lograr dicho objetivo en conjunto con otras ADAS.

Tabla 2.1: Tabla Resumen

Familia	ADAS	Abreviación	Nivel de Autonomía	Sensores
Estabilidad de Conducción	Control de Tracción	TCS	1	Velocidad
Estabilidad de Conducción	Control Electrónico de Estabilidad	ESC	0	Acelerómetro-Ángulo-Velocidad
Control Longitudinal	Control Automático de Crucero	ACC	1,2,3,4,5	Cámara-Velocidad-LiDAR
Control Longitudinal	Reconocimiento de Señales de Tránsito	TSR	0	Cámara
Control Longitudinal	Reconocimiento de Luces de Tránsito	SLR	0	Cámara
Control Longitudinal	Freno Automático de Emergencia	AEB	1,3,4,5	Velocidad-Cámara
Control Longitudinal	Control de Alerta de Peatones	PPS	1,3,4,5	Cámara
Control Longitudinal	Alerta Pre-Colisión	PCW	1	LiDAR-Cámara
Control Longitudinal	Sistema de Evasión de Colisiones	CAS	2,3	Cámara-Radar-LiDAR
Control Lateral	Detección de Punto Ciego	BSD	0	Cámara-Radar
Control Lateral	Alerta de Salida del Carril	LDW	0	Cámara
Control Lateral	Asistente de Cambio de Carril	LCA	2,3,4,5	Cámara-Radar-LiDAR
Apoyo en Cabina	Sistema de Navegación	NS	0	GPS
Apoyo en Cabina	Monitoreo del Estado del Conductor	DSM	0	Cámara
Apoyo al Estacionar	Cámara Visión Trasera	RVC	0	Cámara
Apoyo al Estacionar	Asistencia Inteligente de Parqueo	IPA	2,3,4,5	Cámara-Ultrasonido
Luz y Vista	Visión Nocturna	NV	0	FIR
Luz y Vista	Sistema Adaptativo de Luces Frontales	AFS	0	Cámara-Velocidad
Luz y Vista	Sistema de Sensado de Lluvia	RSS	0	Fotodiodo

2.4 Visión por Computadora

Visión por Computadora (CV), también llamada Visión Artificial, es la transformación de datos de una cámara fotográfica fija o video a una nueva representación o acción en función de la información de la misma. Todas estas transformaciones se hacen para lograr algún objetivo en particular. Los datos de entrada pueden incluir información contextual como “el vehículo está en la carretera” o “el láser indica que un objeto está a 1 metro de distancia”. La decisión podría ser “hay una persona en esta escena” o “hay 14 peatones adelante”. Una nueva representación puede significar convertir una imagen de color en una imagen en escala de grises o quitar el movimiento de la cámara de una secuencia de imágenes. ¿Qué tan difícil puede ser encontrar un coche cuando se mira fijamente en una imagen? El cerebro humano divide la señal de la visión en muchos canales por los que fluyen diferentes tipos de información en su cerebro. Éste tiene un sistema de atención que identifica partes importantes de una imagen para examinar mientras se reprime el análisis de otras áreas. [28].

2.5 Control Automático

Brindar una visión general de lo que el control automático significa, es fundamental para el adecuado entendimiento del proyecto y para ello se presentan una serie de conceptos básicos recopilados [29], los cuales se exponen a continuación:

La **variable controlada** es: “la cantidad o condición que se mide y controla” [29].

Controlar es: “medir el valor de la variable controlada del sistema y aplicar la variable manipulada al sistema para corregir o limitar la desviación del valor medido respecto del valor deseado. En el estudio de la ingeniería de control, es necesario definir términos adicionales que se precisan para describir los sistemas de control” [29].

Una **planta** es: “una parte de un equipo, tal vez un conjunto de los elementos de una máquina que funcionan juntos, y cuyo objetivo es efectuar una operación particular. Se llama planta a cualquier objeto físico que se va a controlar (como un dispositivo mecánico, un horno de calefacción, un reactor químico o un automóvil)” [29].

2.5.1 Tipos de Controladores

La industria automotriz utiliza en sus sistemas principalmente controladores PID. Sin embargo, es importante mencionar que existen otros controladores.

Controlador P

La acción de control proporcional, da una salida del controlador que es proporcional al error y puede controlar cualquier planta estable, pero posee desempeño limitado y error de estado estacionario [30].

Controlador I

Da una salida del controlador que es proporcional al error acumulado, lo que implica que es un modo de controlar lento pero con error de estado estacionario igual a cero [30].

Controlador PI

Con un control proporcional, es necesario que exista error para tener una acción de control distinta de cero. Con acción integral, un error pequeño positivo siempre nos dará una acción de control creciente, y si fuera negativo la señal de control será decreciente. Este razonamiento sencillo muestra que el error en régimen permanente será siempre cero. Muchos controladores industriales tienen solo acción PI [30].

Controlador PD

Esta acción tiene carácter de previsión, lo que hace más rápida la acción de control, aunque tiene la desventaja importante que amplifica las señales de ruido y puede provocar saturación en el actuador. La acción de control derivativa nunca se utiliza por sí sola, debido a que solo es eficaz durante periodos transitorios. Cuando una acción de control derivativa se agrega a un controlador proporcional, permite obtener un controlador de alta sensibilidad, es decir que responde a la velocidad del cambio del error y produce una corrección significativa antes de que la magnitud del error se vuelva elevada [30].

Controlador PID

Esta acción combinada reúne las ventajas de cada una de las tres acciones de control individuales [30].

2.6 Computación Aproximada

La Computación Aproximada (*Approximate Computing*, AC), ha atraído la atención tanto de la academia como de la industria debido a lo prometedor de sus beneficios computacionales [5]. La AC se basa en la capacidad de muchos sistemas y aplicaciones para tolerar alguna pérdida de calidad en el resultado calculado. Al relajar la necesidad de operaciones completamente precisas o completamente deterministas, las técnicas de computación aproximadas permiten una eficiencia energética sustancialmente mejorada [31].

La AC puede ser implementada en aplicaciones que no precisen de funcionamiento perfecto, acepten múltiples salidas y trabajen con ADAS inexactas debido al entorno, como por ejemplo los sensores [32].

Algunas aplicaciones en las que la AC se puede utilizar según [32, 33] son:

- Procesamiento imágenes y video
- Compresión de audio WAV a MP3
- Señales percibidas por sensores
- Aprendizaje de máquina
- Detección de objetos
- Juegos
- Simulaciones

2.6.1 Sumadores aproximados

Los sumadores son ampliamente utilizados para realizar sumas binarias de dos números. En [34] se propone una clasificación de estos según su desempeño en Sumadores de Alto Rendimiento (*High Performance Adder*, HPA) y consumo de potencia en Sumadores de Baja Potencia (*Low Power Adder*, LPA). En esta exploración se utilizó un sumador de cada tipo para evaluar los resultados para ambos casos. Los sumadores seleccionados se describen a continuación.

Sumador GeAr

El Sumador Genérico de Precisión Configurable (*Generic Accuracy Configurable Adder*, GeAr) es un sumador tipo HPA. Este se caracteriza por dividir las sumas en subsumas, con el fin de aligerar y agilizar el proceso [35].

Sumador LOA

El Sumador de Parte Baja OR (*Low-part-OR Adder*, LOA) es un sumador tipo LPA. Se caracteriza por implementar compuertas OR en paralelo para realizar la suma simplificada. Por lo que al ser un método computacionalmente sencillo traslada este beneficio a la aplicación en la que se utilice. [36]

2.6.2 Métricas de calidad de imágenes

La explotación de aceptación de errores en aplicaciones ADAS requiere de métricas que permitan comparar el desempeño de las ADAS desarrolladas bajo funcionamiento exacto en relación a su funcionamiento aproximado. Esto se logra mediante la sustitución de funciones propias de la biblioteca de OpenCV por funciones que cumplan las mismas tareas pero bajo distintos escenarios computacionales que se exponen en este capítulo.

PSNR

La Relación Señal a Ruido de Pico (*Peak Signal-to-Noise Ratio*, PSNR) calcula el error de reconstrucción cuadrático medio después de la eliminación del ruido [37]. Esta medida es una estimación de la calidad de la imagen reconstruida en comparación con la imagen original. Un valor pequeño de PSNR implica altas diferencias numéricas entre las imágenes [38]. Por lo tanto, un valor alto implica una alta calidad en la imagen de muestra en relación a la original. La unidad de medida de este indicador es el decibelio (dB).

SSIM

Esta métrica mide la Similitud Estructural (*Structural SIMilarity*, SSIM) entre la imagen de muestra y la imagen original [37]. El SSIM modela cualquier distorsión de la imagen como una combinación de factores tales como: la pérdida de correlación, la distorsión de luminancia y la distorsión de contraste [38]. Requiere que las dos imágenes que se comparan estén alineadas y escaladas correctamente para que puedan compararse punto por punto. Los valores resultantes a esta comparación están en el rango de [0,1], siendo 1 el valor máximo de calidad de la imagen comparada respecto a la original.

Capítulo 3

Control basado en aplicaciones ADAS

La investigación realizada en el capítulo anterior permitió conocer la amplia gama de ADAS que existen en la actualidad. Para alcanzar la conducción autónoma es necesario controlar dos variables: dirección y velocidad. Esto intrínsecamente plantea dos casos de estudio, uno para cada una de estas variables. Cada estudio conlleva una etapa de programación con la ayuda de librerías de Visión por Computadora (Computer Vision, CV) y otra etapa de análisis de la información que las ADAS genera mediante un sistema de control. Se utilizaron herramientas de CV las cuales permitieron la obtención de información de los videos cuadro por cuadro. En el Anexo I.1 se describen algunas de las funciones de OpenCV utilizadas.

El contenido de este capítulo consiste en observar el funcionamiento exacto de aplicaciones ADAS de acuerdo a la metodología presentada en los artículos científicos estudiados. Por lo cual, se realizó un análisis de las tres aplicaciones (TSR, LDW y PPS) basadas en CV. Posteriormente se realizaron pruebas en un sistema de control para así obtener una referencia ante la cual comparar los resultados en el siguiente caso de estudio presentado en el capítulo 4 correspondiente a Computación Aproximada.

3.1 Aplicaciones seleccionadas

La selección de las ADAS se hizo luego de realizar la taxonomía. Esto debido a que luego de esa recopilación se contó con las bases teóricas para elegir. Ante la cantidad de ADAS recopiladas se precisó de un criterio de selección para determinar cuales se iban a utilizar para la exploración. En este caso se definió la siniestralidad de los hechos reflejados en la sección 1.1 como los criterios de selección, entre ellos: exceso de velocidad, invasión de carril y atropellos a peatones. Estas son de las principales causas de muertes que se pueden atacar directamente mediante las siguientes correspondencias realizadas entre: las causas descritas y las ADAS que intervienen.

- Invasión de carril → Alerta de Salida del Carril (LDW)
- Atropello de peatones → Control de Alerta de Peatones (PPS)
- Exceso de velocidad → Reconocimiento de Señales de Tránsito (TSR)

Las tres aplicaciones escogidas son claves para alcanzar el nivel 5 de autonomía en la conducción de vehículos. En consecuencia, se realizó una división del trabajo en dos casos según las aplicaciones seleccionadas:

- Estudio de Dirección
 - LDW
- Estudio de Velocidad
 - PPS
 - TSR

3.1.1 ADAS involucradas en Dirección

Alerta de Salida del Carril (LDW)

La detección de carril es fundamental si se desea llegar al nivel 5 de autonomía, esto le brindaría al usuario la posibilidad de reducir el nivel de atención a la dirección del vehículo, dado que el mismo sería capaz de mantenerse en su propio carril sin invadir los carriles vecinos, de esta manera se disminuirían los accidentes por invasión de carril, ya sea en el mismo sentido como en sentido contrario de conducción, haciendo las calles lugares más seguros.

3.1.2 ADAS involucradas en Velocidad

Control de Alerta de Peatones (PPS)

Si bien los peatones no son usuarios principales de las carreteras, pueden invadir el área de las mismas. Ya sea por cruzar la carretera o transitar por el espaldón, entre otras cosas, es importante tomar en cuenta su presencia en las carreteras. Es viable e incluso pertinente tomar medidas ante la posible aparición de estos en las vías de tránsito, para así reducir las estadísticas de atropellos a peatones.

Reconocimiento de Señales de Tránsito (TSR)

Las señales de tránsito son indispensables para brindar al conductor las instrucciones necesarias para la delimitación de su accionar en las carreteras. Este sistema conjunto contiene señalamientos de velocidades permitidas, avisos, restricciones, indicaciones de lugares, etc. En este caso particular se enfocó en el análisis de señales de velocidad permitida, de forma que sea posible conocer el límite permitido de la misma según la señalización.

3.2 Caso de estudio 1: Dirección

3.2.1 Visión Artificial Dirección LDW

El primer paso en el desarrollo de esta ADAS es crear una guía de qué es lo que se va a programar. Esta guía debe tener un orden lógico y contemplar todos los pasos necesarios para la resolución del problema. En la Figura 3.1 se muestra un diagrama de flujo para la realización del LDW. La entrada al sistema es un video el cual se procesa cuadro por cuadro para trazar una línea central resultante del carril y a su vez con esta obtener el ángulo que debe girar el carro.

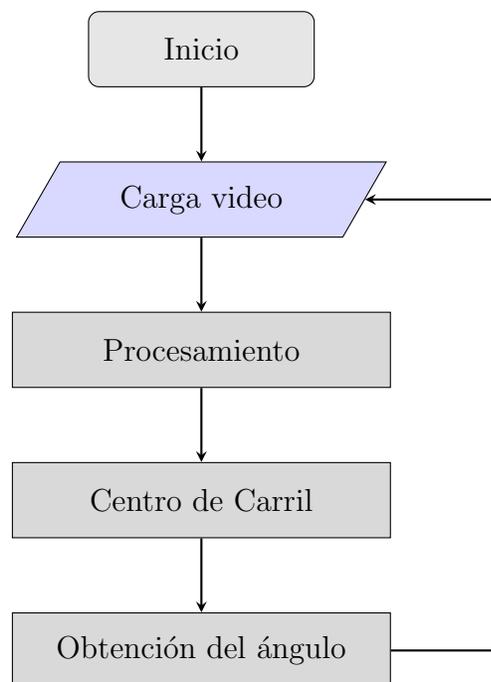


Figura 3.1: Diagrama de Flujo LDW

La etapa de Procesamiento consiste de 3 sub-etapas. Transformación, Preprocesamiento y Segmentación, como se aprecia en la Figura 3.2. En la etapa de transformación, las imágenes se analizan cuadro por cuadro a partir del video, con el fin de aplicar una conversión requerida para las etapas posteriores. Dicha transformación se obtiene mediante el Mapeo de Perspectiva Inversa (*Inverse Perspective Mapping*, IPM), con este se obtiene la llamada Vista Ojo de Pájaro (*Bird-Eye*), la cual consiste en brindar una vista distinta del video. Esto se explica a mayor detalle más adelante y también en el Anexo I.1. En la siguiente etapa, se realiza un procesamiento de la imagen con el fin de realizar una descomposición de la misma que permita en la etapa de segmentación siguiente trazar las líneas laterales de los carriles.

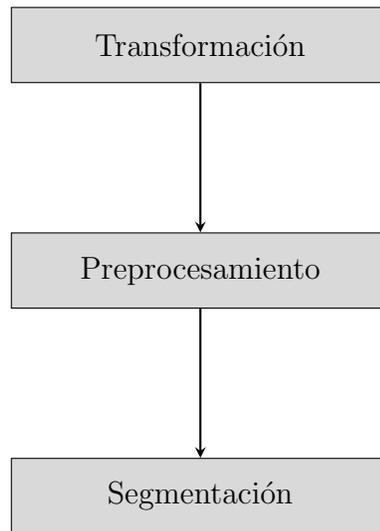


Figura 3.2: Etapas del procesamiento

La transformación Vista Ojo de Pájaro requiere seleccionar en primer lugar los puntos que delimitan la región de Interés (*Region of Interest*, ROI) que se desea transformar, es decir, el área frente al vehículo donde se pueda detectar las líneas del carril correspondiente. Al unir estos puntos se obtiene de forma un trapecio que delimita el carril por el que se maneja. El paso inmediatamente siguiente es la transformación de trapecio a rectángulo la cual permite al sistema obtener información clara de lo que se encuentra frente al vehículo desde una perspectiva completamente nueva como se describe en el Anexo I.2.

Esta perspectiva permite obtener una vista superior del área que se encuentra frente al vehículo. Lo obtenido al realizar la transformación en la LDW es lo que se presenta en la Figura 3.3. Se muestra con líneas finas amarillas la ROI en la Figura 3.3.a. la cual es la zona que se quiere estudiar, de esta manera solo se procesa dicha región y no la totalidad de la imagen. La vista de pájaro se genera con base a los 4 puntos escogidos para el trazado de las líneas finas amarillas, esta brinda un panorama más claro mostrado en la Figura 3.3.b.

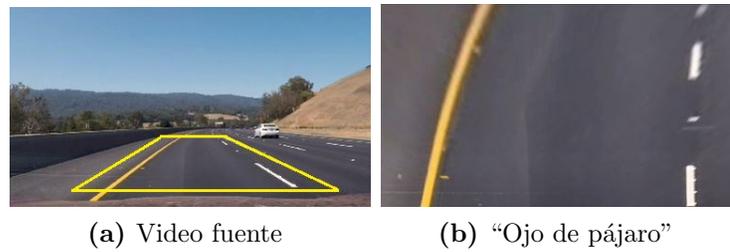


Figura 3.3: Transformación de imagen original a vista Ojo de pájaro

Luego de la obtención de la vista mediante IPM la preparación de las imágenes consiste a su vez de 3 pasos antes de permitir la obtención de información del estado del vehículo respecto a la carretera:

- Conversión a escala de grises.

La conversión de color a escala de grises nos permite trabajar con 1 solo canal de información y no los 3 originales con los que cuentan la imagen fuente RGB como se describe en el Anexo I.2.1. y se muestra en la Figura 3.4.a.

- Difuminado de la imagen.

El Difuminado permite eliminar ruido de la imagen previamente convertida a escala de grises eliminando discontinuidades descrito en el Anexo I.2.2. y se muestra su resultado en la Figura 3.4.b.

- Detección de bordes.

Los bordes se necesitan para fijar los límites de los líneas laterales del carril por el que el vehículo debe desplazarse. Su funcionamiento está descrito en el Anexo I.2.3. y se muestra su resultado en la Figura 3.4.c.

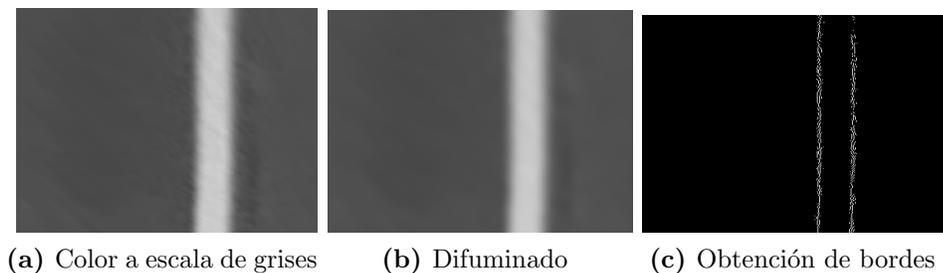


Figura 3.4: Pasos de preprocesamiento LDW

Al realizar el difuminado de la imagen y la extracción de bordes se obtiene lo que se observa en la Figura 3.5.a. Una matriz del mismo tamaño que la Vista Ojo de Pájaro pero a diferencia de esta, solo se presentan los bordes extraídos mediante la función Canny de OpenCV. Estos puntos brindan a su vez la información requerida para la etapa siguiente,

la segmentación. Se separan en puntos izquierdos y puntos derechos y se ordenan de menor a mayor de acuerdo a sus coordenadas en el eje vertical. Cada conjunto de puntos (izquierdo y derecho) cuenta con infinidad de puntos por lo que se procede a obtener un muestreo de 6 puntos de cada conjunto para trazar así una línea por conjunto. Estas 2 líneas representan los límites del carril. Luego de obtener la media entre cada dato homólogo de cada conjunto se logró dibujar una línea central resultante del carril por el que el vehículo es manejado. Estas 3 líneas que se mencionan se pueden apreciar en la Figura 3.5.b. En la Figura 3.5.c por su parte se muestra de donde es que se obtiene el ángulo que se busca. Este es obtenido analizando 3 puntos: punto A que corresponde al punto medio de la parte superior de la imagen el cual funciona como referencia y punto fijo para obtener el ángulo, punto B el cual es el punto mínimo de línea de centro y punto C que corresponde al punto máximo de línea de centro respecto a su coordenada vertical. El ángulo resultante se señala con una flecha blanca. Este es el ángulo que el vehículo precisa virar.

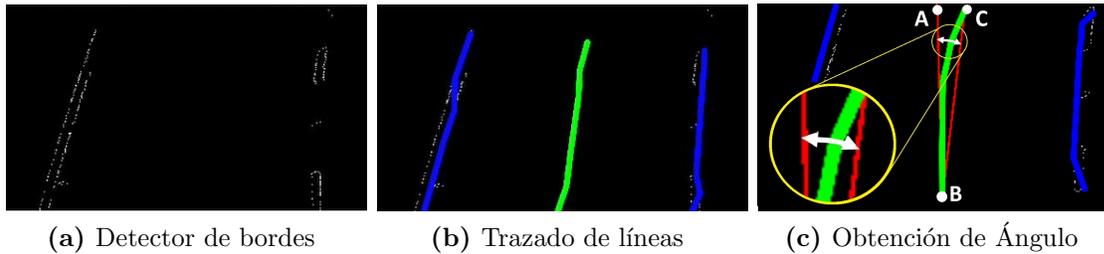


Figura 3.5: Definición de límites y ángulo

Se realiza la transformación inversa de la vista Ojo de pájaro y se despliegan en pantalla 2 valores: el ángulo de giro del vehículo y el estado del vehículo en el carril como se muestra en la Figura 3.6. Los ángulos, cuadro a cuadro se van guardaron en una hoja de cálculo en formato *csv*.

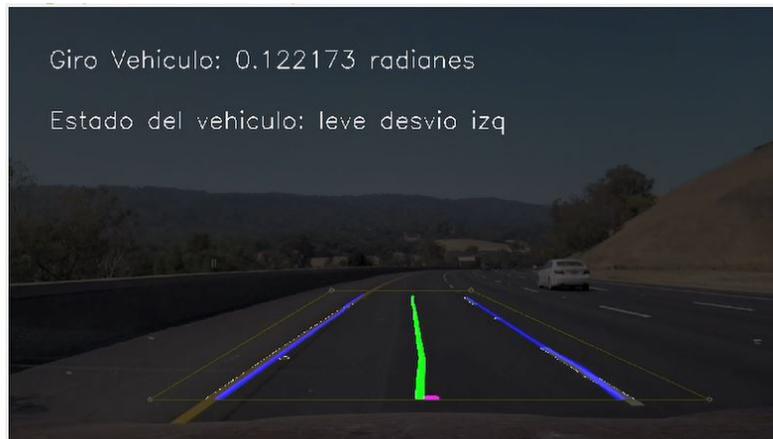


Figura 3.6: Resultados LDW

El estado del vehículo que se menciona anteriormente consiste en que tan cerca o lejos está el vehículo del centro del carril. Esto únicamente a manera de alerta para al conductor cuando el sistema no esté trabajando en modo automático. Ya que en modo automático se espera que el sistema de control lo mantenga siempre sobre la línea de centro.

Luego de generar el método de obtención de giro, se aplicó la LDW a un video compuesto por 440 cuadros, del cual se obtuvo el ángulo de giro para cada uno. Los datos obtenidos se grafican en la Figura 3.7, donde la curva naranja muestra dichos resultados. Tal y como se observa, los resultados muestran una variación aleatorio que denota una componente de ruido en los mismos. Con el fin de eliminar dicho ruido, se aplica el método de Media Móvil (*Moving Average*, MA) de 100 de manera recursiva [39]. Así, en la misma Figura 3.7 en la curva color azul se observa la curva filtrada. Estos datos filtrados se utilizaron por el sistema de control del caso de estudio correspondiente a dirección.

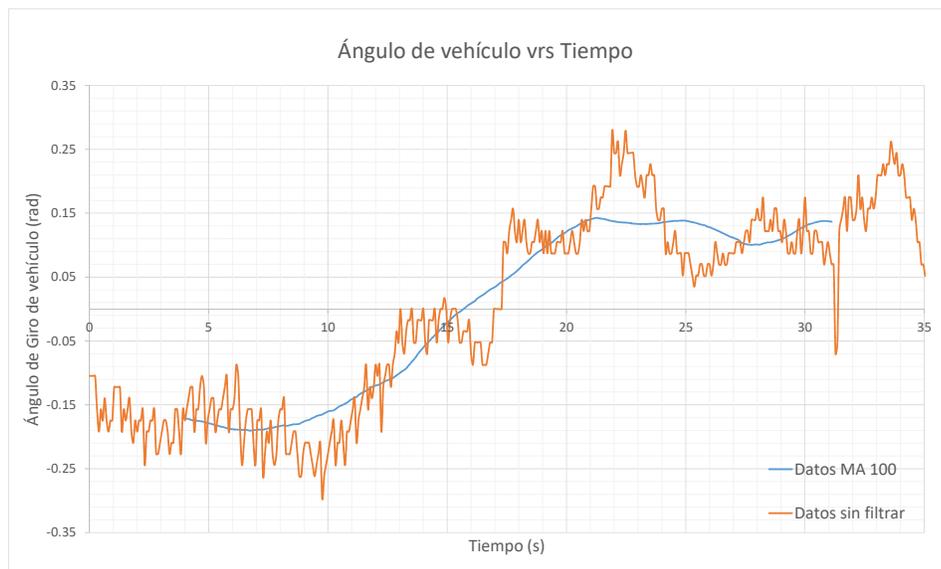


Figura 3.7: Curvas LDW superpuestas

3.2.2 Sistema de Control de Dirección

El sistema mecánico de dirección de un vehículo de cuatro ruedas se puede modelar tomando la idea del “Modelo de Bicicleta” [40]. Este propone un modelo simplificado el cual se explica a detalle en el Apéndice A. El modelado que se consigue al realizar el estudio de fuerzas brinda como resultado la función de transferencia mostrada en la ecuación 3.1. Dicha ecuación corresponde a la planta del sistema de control que en este caso es el vehículo.

$$\frac{R(s)}{\Delta(s)} = \frac{6.45 \cdot s}{s + 0.1789} \quad (3.1)$$

En dicha función de transferencia $\Delta(s)$ corresponde al ángulo equivalente que el sistema de dirección debe aplicar. Por su parte, el $R(s)$ es el ángulo que el vehículo debe rotar sobre si mismo para satisfacer las necesidades de desplazamiento en carretera, esto se llama guiñada (*yaw* en inglés).

De manera general, el sistema de control que se diseñó fue el representado en la Figura 3.8 donde la $R_{visión}$ corresponde al ángulo que el sistema de visión obtuvo luego del procesamiento del video de prueba y el R_{salida} es el ángulo resultante luego de la etapa de control. Cada uno de los bloques representa las componentes del sistema: Controlador, Planta y Mediciones. Este mismo fue implementado en Matlab con la herramienta Simulink, tal y como se muestra en el Apéndice B. Recibiendo los datos del ángulo detectado provistos por el sistema de visión observados anteriormente en la Figura 3.6.

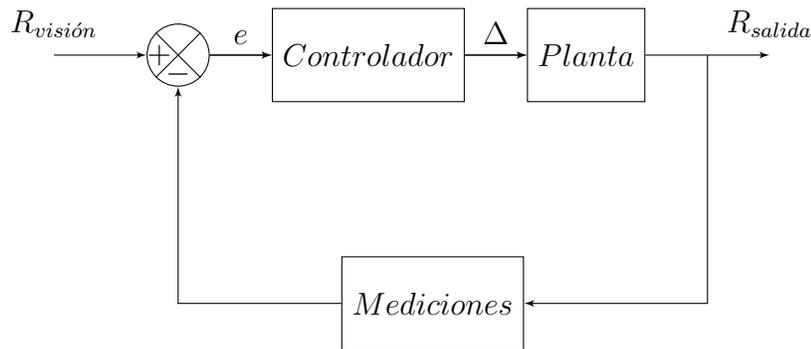


Figura 3.8: Sistema de control para la corrección de dirección.

El diseño del controlador adecuado para el sistema de dirección requirió la selección de parámetros de diseño. Para ello se realizó una búsqueda en investigaciones previas de ADAS, sin embargo, no se encontró respaldo bibliográfico de estas respuestas transitorias y se procedió a establecer estos parámetros para contar con un medio de prueba para esta exploración. El sobre-impulso máximo (M_p), el tiempo de estabilización al 98% (t_s) y el valor de error de estado estacionario (E_{ss}) fueron los tres parámetros determinados con valores menores o iguales a 10% en M_p , menores o iguales a 1,5s en t_s e igual a 0 para E_{ss} . Para obtener dichos valores fue necesario agregar un integrador en cascada al

controlador. Esto debido a que la función de transferencia es de Tipo 0 y se necesitaba elevarla a Tipo 1 para conseguir eliminar el error de estado estacionario (E_{ss}). Ya que si se pretende que el E_{ss} sea cero el sistema debe de ser de Tipo 1 o superior [29].

El estudio se realizó con los controladores que se muestran en la Tabla 3.1 y se obtuvieron los resultados que se observan en la misma. A la derecha de cada dato se muestra en rojo si no cumplió con las condiciones de diseño y en verde si sí cumplió.

Tabla 3.1: Respuestas transitorias en sistema de control de dirección según tipo de controlador

Controlador	M_p	t_s	E_{ss}	Figura
P	0%	1.5s	0	Apéndice B Figuras B.2 y B.3
I	9.41%	40s	0	Apéndice B Figuras B.4 y B.5
PI	3.24%	3.5s	0	Apéndice B Figuras B.6 y B.7
PID	10%	1.4s	0	Apéndice B Figuras B.8 y B.9

A pesar de que el controlador P y PID cumplen con los 3 parámetros se seleccionó un controlador PID debido a que es más robusto ante perturbaciones debido a que cuenta tanto con los beneficios que brinda el I y el D. Este le permitió al controlador contar con un t_s al 98% de 1.4 segundos, M_p igual a 10% y E_{ss} igual a cero. Estos valores se obtuvieron al estudiar la respuesta del sistema ante un escalón unitario. En la Figura 3.9 se muestra la curva de respuesta del sistema ante esa entrada.

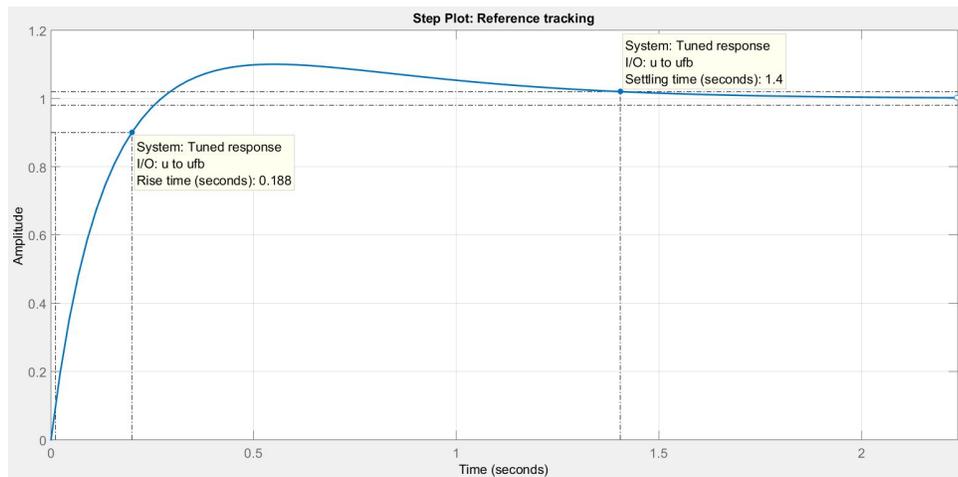


Figura 3.9: Tiempos Controlador PID

Finalmente para el caso de estudio de velocidad se probó el controlador diseñado y la curva que se observa en la Figura 3.10 muestra el resultado superponiendo los datos de entrada representados en rojo contra los datos de salida representados en azul.

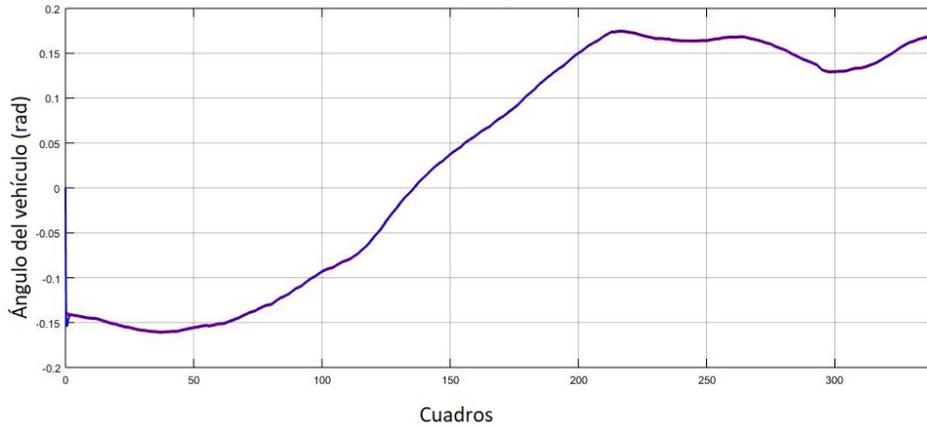


Figura 3.10: Gráfico Ángulo vrs tiempo para vehículo de 1200kg

3.3 Caso de estudio : Velocidad

En este segundo caso de estudio para controlar la velocidad se involucran las ADAS: TSR y PPS. La TSR brinda los datos de entrada al Control Crucero Adaptativo (*Adaptive Cruise Control, ACC*). De esta forma el ACC regula la velocidad según las señales detectadas y la PPS detiene el vehículo en el caso de que se detecte un peatón.

3.3.1 Visión Artificial Velocidad PPS

La Figura 3.11 muestra el diagrama de flujo, el cual es la guía de la metodología para resolver la detección de peatones PPS. En este se toma un video realizado en un vehículo en movimiento, como paso inicial. Del video, se realiza un procesamiento de cada cuadro con el fin de detectar la presencia de un peatón. En el momento que se detecta uno se procede a la obtención del tamaño del peatón, del cual se infieren tres estados: no hay peatón, hay peatón cerca o hay peatón peligrosamente cerca. En el momento que el tamaño de la ROI detectada corresponda al definido como estado de cerca, se define una señal de peatón cerca que se le muestra al usuario.

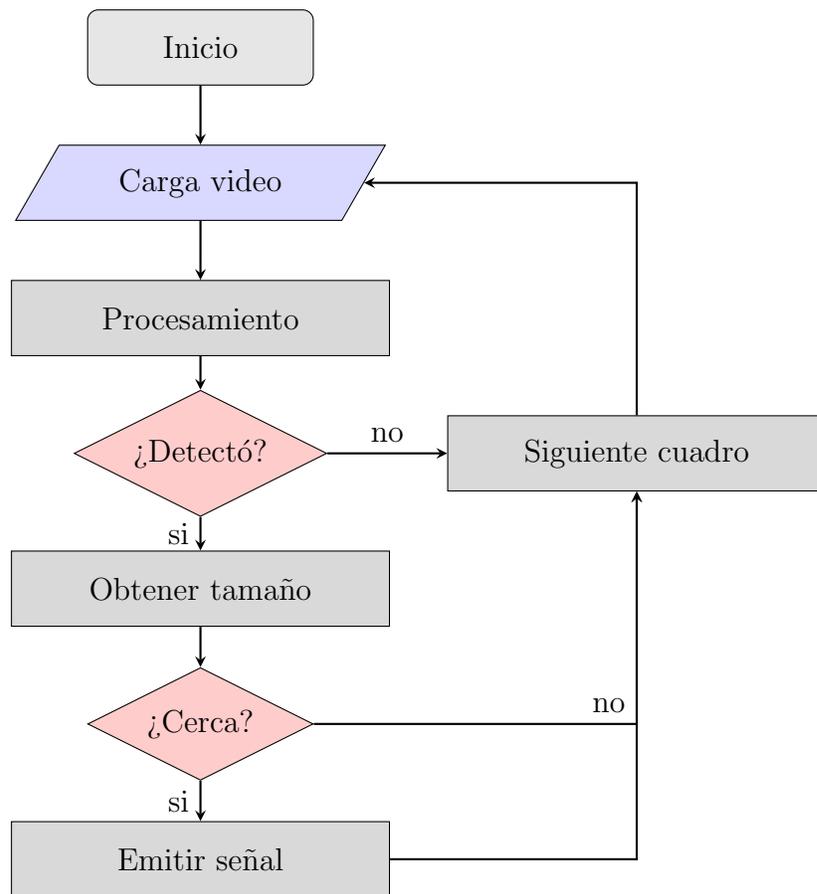


Figura 3.11: Diagrama de Flujo PPS

El procesamiento para la detección de peatones PPS consistió de dos pasos:

- Conversión a escala de grises.

La conversión de color a escala de grises nos permite trabajar con un solo canal de información y no los tres originales con los que cuentan la imagen fuente RGB como se describe en el Anexo I.2.1.

- Difuminado de la imagen.

El Difuminado permite eliminar ruido de la imagen previamente convertida a escala de grises eliminando discontinuidades descrito en el Anexo I.2.2.

Para lograr la detección de peatones se utilizó un clasificador conocido como HAAR Cascades el cual permite reconocer objetos según la detección de patrones configurados previamente, tal y como se describe en el Anexo I.3. Para este caso se seleccionó la detección de cuerpo entero el cual se detecta y enmarca en una ROI cuando es detectado.

Según la altura de la región detectada, así se define la distancia del peatón con respecto al vehículo. En la Figura 3.12 se representa gráficamente la curva generada donde a mayor altura de ROI, menor es la distancia entre el vehículo y el peatón.

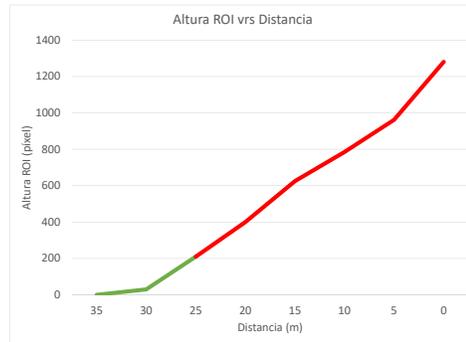


Figura 3.12: Gráfico Altura ROI vs Distancia

- Altura de ROI igual a cero, significa que no hay peatón cerca, emite un “1”.
- Altura de ROI menor a 210 píxeles, significa que hay peatones cerca pero no hay peligro, emite un “1”. Se muestra con el segmento verde de la curva de la Figura 3.12.
- Altura de ROI mayor a 210 píxeles, significa que hay peatones peligrosamente cerca, emite un “0”. Se muestra con el segmento rojo de la curva de la Figura 3.12.

De acuerdo a lo anterior en las Figura 3.13 se muestran 4 vistas de los casos a los que responde el sistema. Si se percibe que hay peatón pero a una distancia distinta a la definida como cerca, la ROI se demarca en verde para que el conductor esté atento (Figura 3.13.a-b). Si por el contrario, el peatón está “peligrosamente” cerca la ROI se torna roja alertando (Figura 3.13.c-d).



Figura 3.13: Detección de acuerdo a tamaño de ROI

3.3.2 Visión Artificial Velocidad TSR

El detector de señales de tránsito es la segunda ADAS que entra en acción en este caso de estudio número 2. Mediante algoritmos de cascada, descritos en el Anexo I.3, detecta la zona a analizar en el video fuente. Esta ROI se analiza mediante una Máquina de Vectores de Soporte (*Support Vector Machine*, SVM) la cual clasifica la señal dentro de tres clases. La SVM es una herramienta de Aprendizaje de Máquina (*Machine Learning*, ML) que se analiza con más profundidad en el Anexo II.4. Este proceso de detección se muestra en el diagrama de flujo de la Figura 3.14.

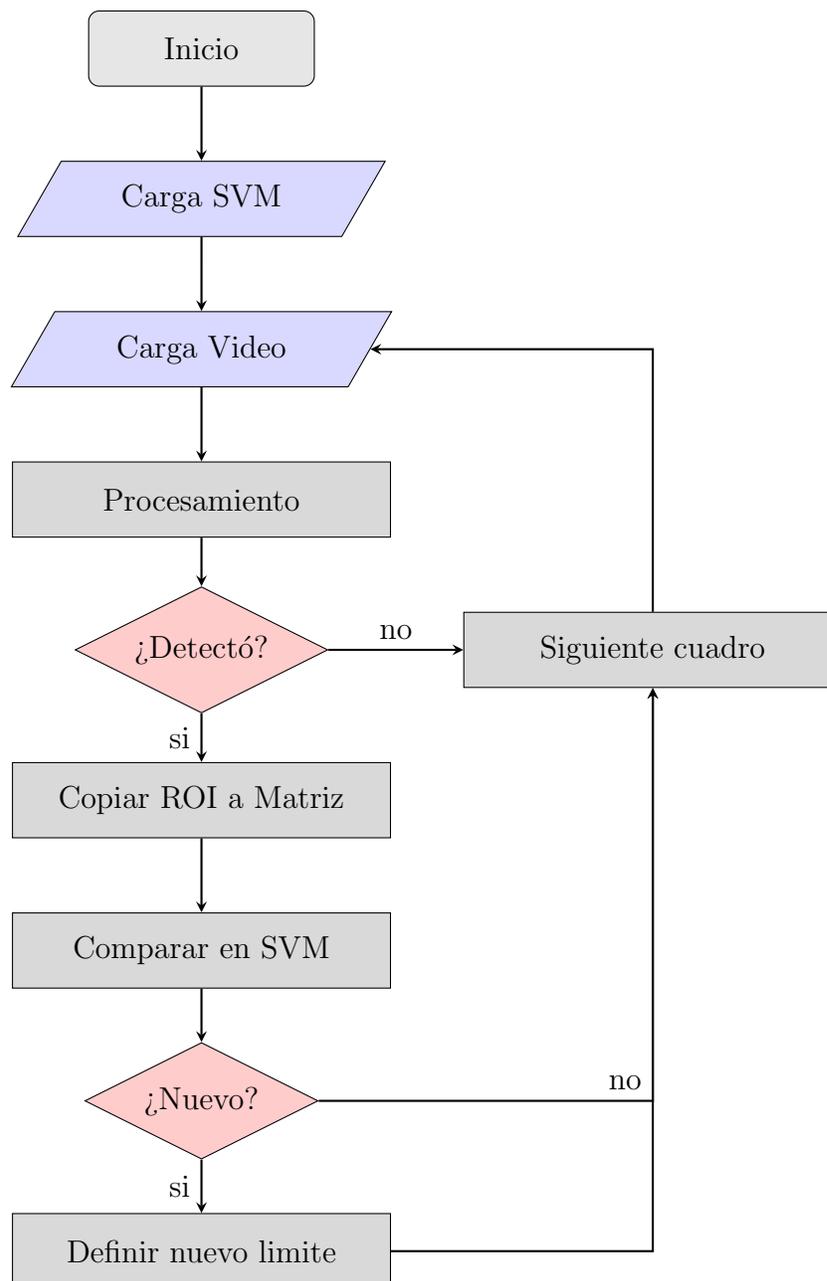


Figura 3.14: Diagrama de Flujo TSR

Implementar una SVM consta de tres etapas como se muestra en la Figura 3.15. Primero se configura la máquina con parámetros para delimitar el funcionamiento de la misma. Se definen las clases que necesita tener la SVM para desempeñar su función de clasificador. Para luego entrenar el sistema con un banco de imágenes según clase. En este caso se seleccionaron 3 clases:

- Clase “0” agrupa las muestras de señales de 30km/h
- Clase “1” agrupa las muestras de señales de 50km/h
- Clase “2” agrupa las muestras de señales de 70km/h

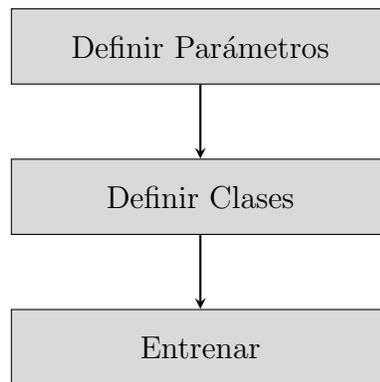


Figura 3.15: Desglose de etapas de SVM

Cada una de estas clases se entrenó con conjuntos de 500 muestras. Es decir, la SVM está entrenada con 1500 datos en total. En el Apéndice C se muestran algunas de las imágenes usadas para el entrenamiento de cada clase. Este entrenamiento se realizó con imágenes tomadas para este proyecto y otras recopiladas para el “Concurso de reconocimiento de señales de tráfico alemanas” (*German Traffic Sign Recognition Benchmark*, GTSRB) [41] organizado en el 2011 por el Instituto para la Neuro-Informática (*Institut für Neuroinformatik*, INI). Estos tres conjuntos de imágenes de las señales a estudiar (30km/h, 50km/h, 70km/h) utilizados para el entrenamiento cuentan con muestras capturadas en distintos escenarios: día, noche, autopista, ciudad entre otros.

El paso inmediatamente siguiente al entrenamiento de la SVM es guardar el entrenamiento. Una vez realizado lo anterior, la SVM se puede utilizar. Para hacer uso de la SVM entrenada se debe inicializar la misma cargando el archivo con el entrenamiento e invocando a la predicción para reconocer la ROI detectada por el HAAR Cascade. Lo comentado anteriormente acerca del funcionamiento y parámetros de SVM se explican a detalle en el Anexo II.4.

La etapa de procesamiento de las imágenes consistió en convertir los cuadros de video de color a escala de grises. Esta conversión permite trabajar con un solo canal de información y no los 3 originales con los que cuentan la imagen fuente RGB y la detección de señales

involucró un clasificador *HAAR Cascades*. Se utilizó el clasificador de HAAR de señales de velocidad europeas.

El video utilizado en la TSR consistió en un recorrido de 43 segundos por las calles y carreteras de Karlsruhe donde se aprecian señales de límites de velocidad en la vía. En el se presentaban señales de las 3 clases entrenadas. En las Figuras 3.16, 3.17 y 3.18 se muestran los resultados donde según la señal que se detectó en ellas se aprecia que:

- Se señala la ROI por un marco verde.
- Se muestra en pantalla un mensaje afirmando textualmente el límite de velocidad más una imagen del mismo.
- Se escribe en un tabla externa al programa de formato .csv el dato numérico del límite de velocidad que detectó.



Figura 3.16: Reconocimiento señal de 30km/h



Figura 3.17: Reconocimiento señal de 50km/h



Figura 3.18: Reconocimiento señal de 70km/h

En cada uno de los 3 ejemplos mostrados se detectó la señal correspondiente de forma correcta. El HAAR Cascade estaba configurado para que, dentro de toda la imagen procedente del video, detectara las señales que estuvieran en un intervalo de tamaño entre 30X30 y 100X100 píxeles. En la Figura 3.19 se presentan seis ROI que el TSR detectó según señal detectada y que fungieron como datos de entrada para la predicción de la SVM. Las muestras (a) y (b) corresponden a lo detectado en el primer escenario donde existía una señal de 30km/h. Las muestras (c) y (d) presentan la detección de la señal de 50km/h. Finalmente las muestras (e) y (f) enseñan lo detectado cuando el sistema se enfrentó a una señalización de 70km/h.

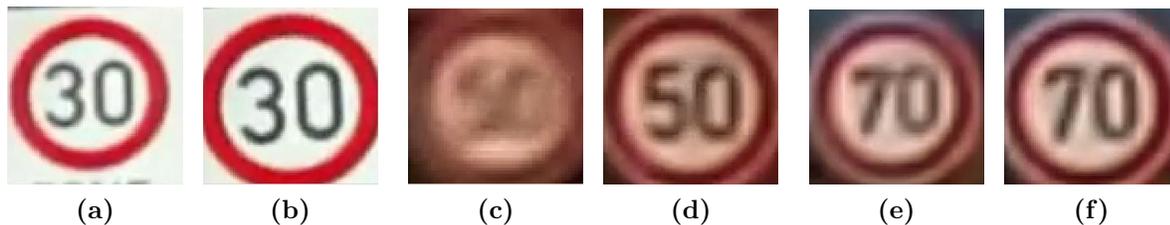


Figura 3.19: Reconocimiento señal de límites de velocidad de los tres casos en estudio

3.3.3 Sistema de Control de Velocidad

Para el primer caso de estudio se modeló el sistema de dirección de un vehículo. En esta ocasión se modela su sistema de velocidad. La obtención del modelo de velocidad para un vehículo se muestra en la sección del Apéndice D. En este, se consigue a partir de la segunda ley de Newton determinar la función de transferencia que modela el comportamiento del sistema de velocidad de un vehículo, misma que se muestra en la ecuación 3.2. Dicha ecuación corresponde a la planta del sistema de control de velocidad.

$$\frac{V(s)}{U(s)} = \frac{1}{m \cdot s + b} \quad (3.2)$$

En dicha función de transferencia $V(s)$ corresponde a la velocidad longitudinal del vehículo y $U(s)$ representa a la fuerza que requiere el vehículo para desplazarse a dicha velocidad. La Figura 3.20 muestra el esquema del sistema mencionado anteriormente, donde se visualizan dos señales de entrada en el mismo. Ambas entradas entran a un multiplicador donde definen la señal referencia del controlador. La primer señal PPS define la detección o no de un peatón en carretera. Cuando un peatón es detectado esta señal genera un “0” el cual produce que la señal de referencia de velocidad sea 0 km/h, de forma que el vehículo comienza a detenerse de forma automática en caso contrario genera un “1” y la velocidad de referencia (V_r) es igual a la V_{TSR} , esta segunda señal de entrada del sistema es la que informa al sistema de control la detección de los límites de velocidad 30, 50 ó 70km/h según sea la velocidad detectada. En un sistema con una planta física entrarían en acción ADAS de estabilización y control de tracción, sin embargo, para los alcances de esta exploración únicamente se maneja bajo simulación. Utilizando la herramienta de Matlab, Simulink, se hizo una implementación del sistema de control con el fin de realizar simulaciones del mismo y verificar el funcionamiento ante distintos cambios en las señales de entrada. Los detalles de la implementación del sistema están en el Apéndice E de este documento.

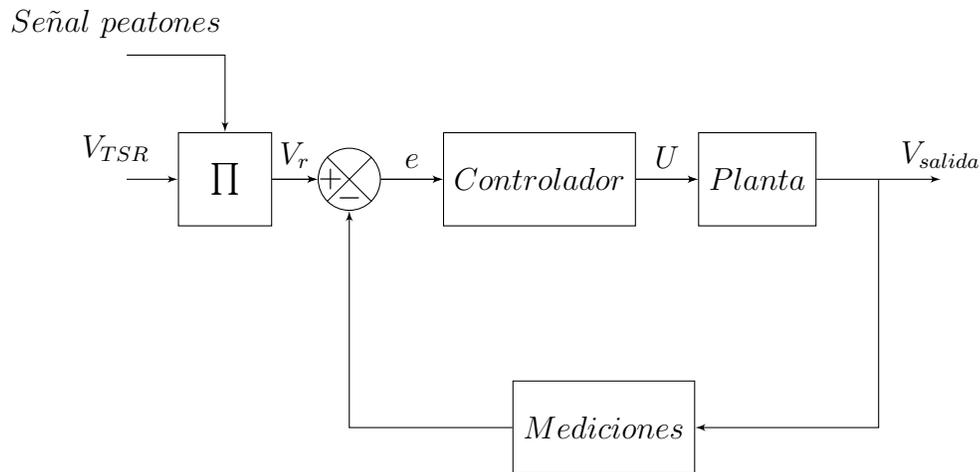


Figura 3.20: Controlador Velocidad Diseñado

Al igual que en el caso de estudio de dirección, el diseño del controlador adecuado para el sistema de velocidad requirió la selección de parámetros de diseño del mismo, sin embargo, no se encontró respaldo bibliográfico de estas respuestas transitorias y se procedió a establecer estos parámetros para contar con un medio de prueba para esta exploración. El sobre-impulso máximo (M_p), el tiempo de estabilización al 98% (t_s) y el valor de error de estado estacionario (E_{ss}) fueron los tres parámetros determinados con valores menores o iguales a 1% en M_p e igual a 0 para E_{ss} . El parámetro t_s para este caso de estudio se determinó obteniendo los tiempos de frenado según velocidad. Los límites de velocidad en los que el control de velocidad entra en acción para este caso son 30km/h, 50km/h y 70km/h y el PPS tiene que ser capaz de detener el vehículo por ello la definición de la t_s es crucial. La ecuación 3.3 define la distancia en función de la velocidad inicial, velocidad inicial, y tiempo.

Gracias a ella se pueden obtener los tiempos de frenado, es decir, que la velocidad final (v_f) sea cero para cada una de las tres velocidades de estudio donde t_{30} es igual al tiempo de frenado con velocidad inicial (v_i) de 30km/h, t_{50} con velocidad de 50km/h y t_{70} con velocidad de 70km/h. Sabiendo que, en la Figura 3.12 se muestra que la PPS detecta peatones a 25 metros de distancia (d_f) y contando con velocidad final e inicial se obtuvo el tiempo necesario para detenerse y no impactar el peatón para cada caso.

$$d_f = \frac{v_i + v_f}{2} \cdot t \quad (3.3)$$

Aplicando la ecuación anterior a cada velocidad inicial de las 3 posibles se obtuvo:

- $t_{30} = 6s$
- $t_{50} = 3.6s$
- $t_{70} = 2.57s$

De los 3 valores anteriores se tomó el de t_{50} como parámetro de diseño debido a que el sistema PPS trabaja a velocidades bajas por ejemplo en entornos urbanos y cubre el tiempo de t_{30} . En el caso de t_{70} si bien el vehículo le faltarían 1.03segundos para detenerse, el impacto sería a una velocidad menor.

El estudio se realizó con los controladores que se muestran en la Tabla 3.2 y se obtuvieron los resultados que se presentan en la misma. A la derecha de cada dato se muestra en rojo si no se cumplió la condición y en verde si sí se cumplió.

Tabla 3.2: Respuestas transitorias en sistema de control de velocidad según tipo de controlador

Controlador	M_p	t_s	E_{ss}	Figura
P	0%	1.5s	$\neq 0$	Apéndice E Figuras E.2 y E.3
I	58%	1.17e04s	0	Apéndice E Figuras E.4 y E.5
PI	1.51%	3.6s	0	Apéndice E Figuras E.6 y E.7
PD	0%	3.6s	$\neq 0$	Apéndice E Figuras E.8 y E.9
PID	0.684%	3.6s	0	Apéndice E Figuras E.10 y E.11

De los 5 controladores estudiados se seleccionó el PID por ser el único que cumple con los 3 parámetros de diseño planteados. Estos valores se obtuvieron al estudiar la respuesta del sistema ante un escalón unitario. En la Figura 3.21 se muestra la curva de respuesta al escalón unitario.

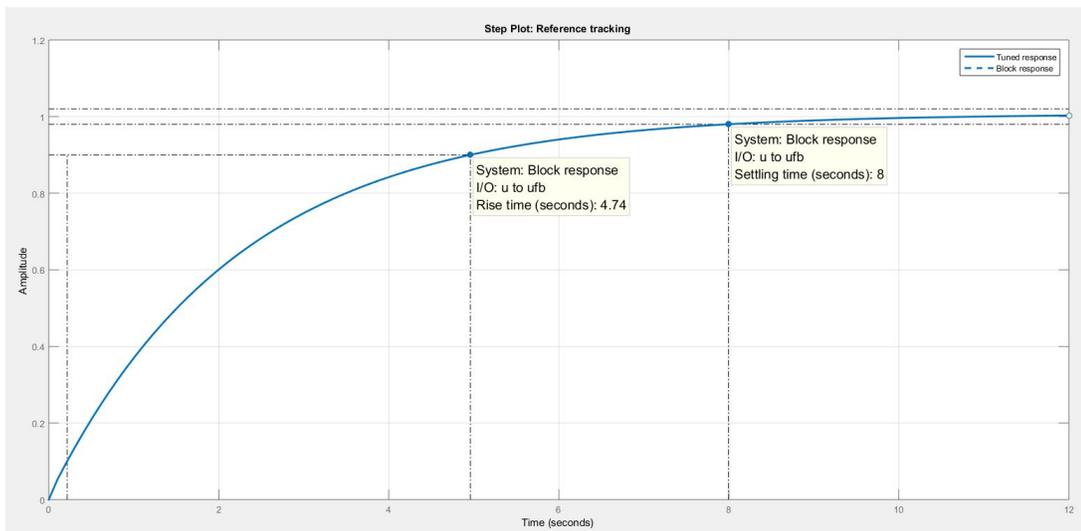


Figura 3.21: Ajuste de valores PID

Para el caso de estudio de velocidad se probó el controlador diseñado y la curva que se observa en la Figura 3.22 es el resultado superponiendo los datos de entrada representados en rojo contra los datos de salida representados en azul.

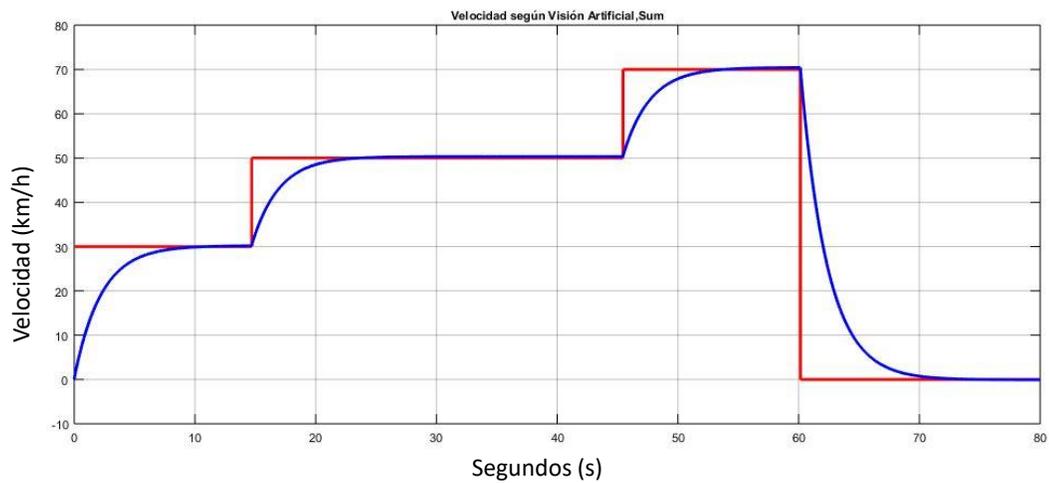


Figura 3.22: Gráfico Velocidad vrs tiempo para vehículo de 1200kg

Capítulo 4

Exploración de tolerancia a imprecisiones

4.1 Descripción del capítulo

En este capítulo se presenta el funcionamiento de las aplicaciones ADAS desarrolladas en el capítulo 3 bajo una implementación basada en computación aproximada. Se procede a realizar sustituciones de funciones a las tres aplicaciones seleccionadas (LDW, PPS y TSR). Lo anterior con el fin de comparar los resultados obtenidos y luego probarlas en los sistemas de control diseñados para cada caso de estudio.

4.2 Exploración de potenciales aproximaciones

Para los alcances de este trabajo, se definieron como las potenciales funciones para computación aproximada aquellas involucradas en el preprocesamiento de las imágenes. Dentro de las funciones de OpenCV utilizadas se definieron dos en particular: la conversión de color a escala de grises y el difuminado. Para cada una de estas dos funciones se implementó un conjunto de versiones aproximadas para ser empleadas en distintos escenarios.

Parámetros para validación

Se utilizaron los sumadores aproximados LOA y GeAr (descritos en la sección 2.6.1) en las funciones aproximadas de conversión de color a escala de grises y difuminado. Por su parte, la validación de los resultados se realizó mediante el análisis de las métricas de calidad: PSNR y SSIM (explicadas en la sección 2.6.2). De acuerdo con [42], el valor de PSNR debe ser mayor a **20 dB** para que se considere aceptable y en el caso del índice de SSIM su valor debe sobrepasar el **0.95** según [43, 44].

4.2.1 Conversión de color a escala de grises

La conversión de una imagen a color a escala de grises implica convertir la imagen original compuesta por tres canales (matriz escala de azules "B, matriz escala de rojos (R) y matriz escala de verdes (G)) a una compuesta únicamente por una matriz de grises. La primera impresión al considerar una imagen equivalente en escala de grises de una imagen a color es asignarle el mismo peso a cada una de los canales de colores como se muestra en la ecuación (4.1):

$$pixel_{resultante} = 0.333B + 0.333G + 0.333R \quad (4.1)$$

Sin embargo, en el procesamiento de imágenes para CV se le asignan pesos distintos dependiendo el color para obtener una imagen en escala de grises lo más acertada posible según la percepción de colores del ojo humano. La visión humana es más perceptible al color verde, seguido de rojo y en tercer lugar el azul. El modelo LUMA propone una asignación de pesos a estos en [45,46], los cuales corresponden a:

$$pixel_{resultante} = 0.114B + 0.587G + 0.299R \quad (4.2)$$

Funciones aproximadas

Aplicando lo establecido por el modelo y las bases teóricas se procede a construir las variantes de funciones aproximadas de la siguiente manera:

1. Se reemplazan multiplicaciones por corrimientos y se suman valores R, G y B para obtener su equivalente según el método LUMA. (Aproximación Color 1)
2. Se reemplazan multiplicaciones por corrimientos y se suman valores R, G y B mediante sumador aproximado con el método GeAr [47] para obtener su equivalente según el método LUMA. (Aproximación Color 2)
3. Se reemplazan multiplicaciones por corrimientos y se suman valores R, G y B mediante sumador aproximado con el método GeAr configurado de manera distinta al anterior [47] para obtener su equivalente según el método LUMA. (Aproximación Color 3)
4. Se reemplazan multiplicaciones por corrimientos y se suman valores R, G y B mediante sumador aproximado con el método LOA [48] para obtener su equivalente según el método LUMA. (Aproximación Color 4)
5. Se reemplazan multiplicaciones por corrimientos y se suman valores R, G y B mediante sumador aproximado con el método LOA configurado de manera distinta al anterior [48] para obtener su equivalente según el método LUMA. (Aproximación Color 5)

Conversión aproximada de color a escala de grises para LDW

La conversión de color a escala de grises fue utilizada en el LDW haciendo uso de cada una de las funciones aproximadas con las que se contaba. En la Figura 4.1 se muestra el mismo cuadro de video siendo preprocesado por cada una de las funciones aproximadas para la conversión de color a escala de grises definidas anteriormente.

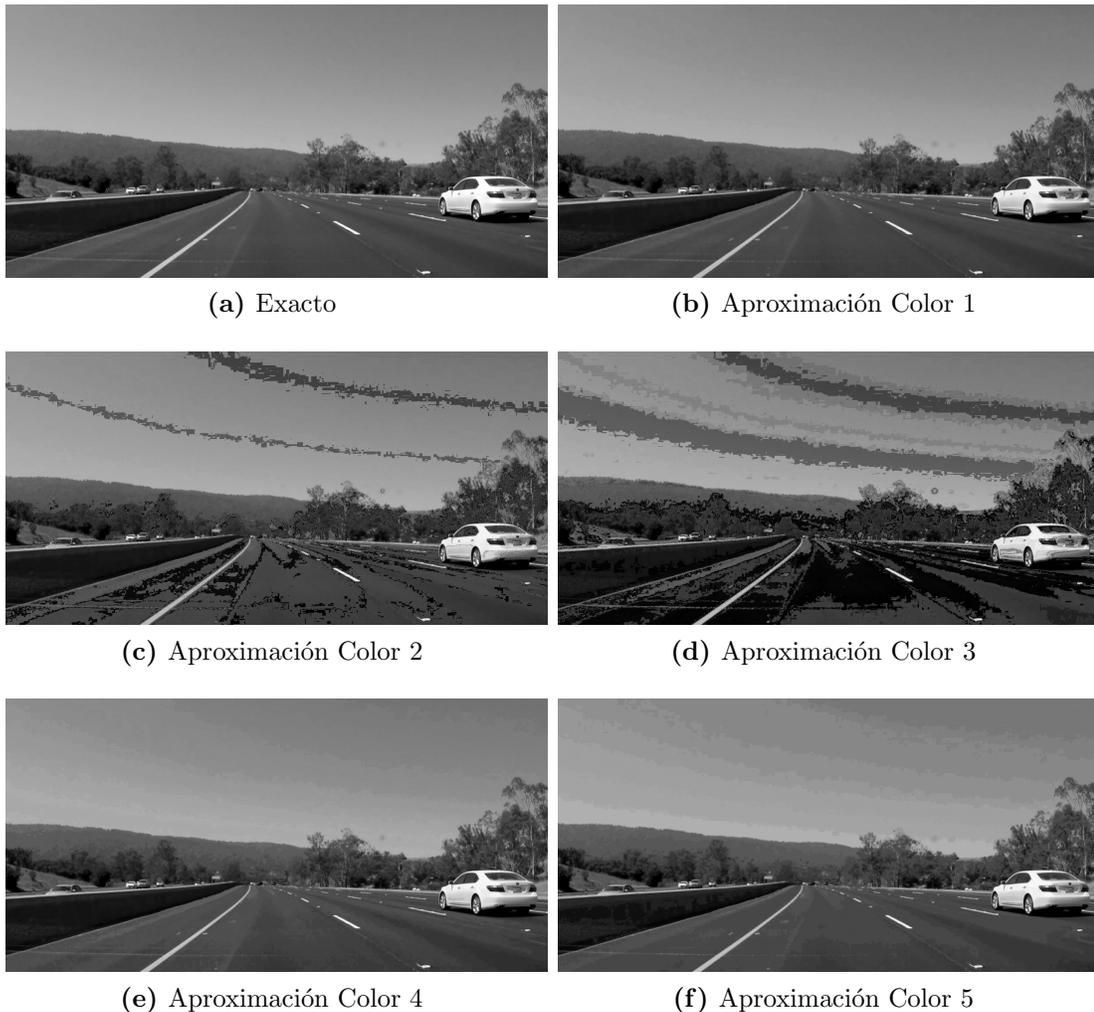


Figura 4.1: Resultados conversión de color a escala de grises en LDW

Al poner en marcha las corridas de LDW bajo las funciones aproximadas de conversión de color a escala de grises se obtuvieron resultados de PSNR de los cuales es posible inferir que a mayor cantidad de dB menor cantidad de ruido se presenta por cuadro de video, por lo que las aproximaciones 1 (33.64 dB), 4 (32.34 dB) y 5 (32.25 dB) son las que muestran un mejor comportamiento; a diferencia de las aproximaciones 2 (22.99 dB) y 3 (16.62 dB). La aproximación 2 a pesar de contar con un valor menor de PSNR que las aproximaciones 1, 4 y 5, continua siendo un valor aceptable de acuerdo con los parámetros de validación, únicamente la aproximación 3 que no cumple. Lo anterior se muestra en la gráfica de la Figura 4.2 mostrando la media de la agrupación de los datos con sus respectivos rangos.

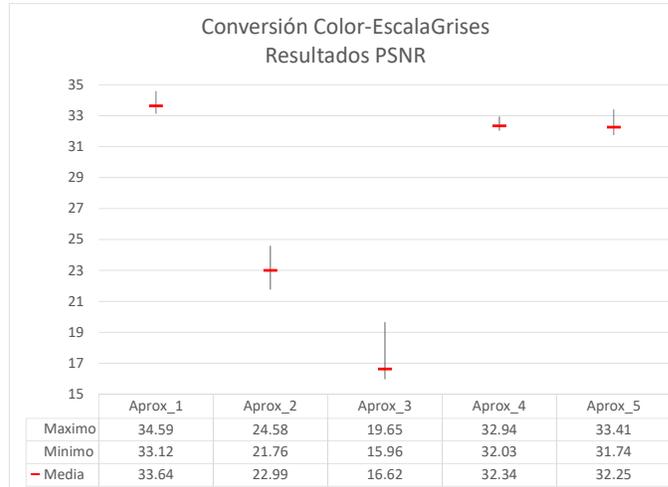


Figura 4.2: Resultados PSNR de conversión de color a escala de grises en LDW.

Bajo el mismo procedimiento se obtuvieron resultados de SSIM de los cuales es posible demostrar que a mayor cercanía a “1” mayor es la similitud estructural que se presenta por cuadro de video, por lo que las aproximaciones 1 (0.9950), 4 (0.9789) y 5 (0.9622) son las que muestran un comportamiento por encima de 0.95 en el valor del índice SSIM; a diferencia de las aproximaciones 2 (0.8402) y 3 (0.6184). Lo anterior se muestra en la gráfica de la Figura 4.3 mostrando la media de la agrupación de los datos con sus respectivos rangos.

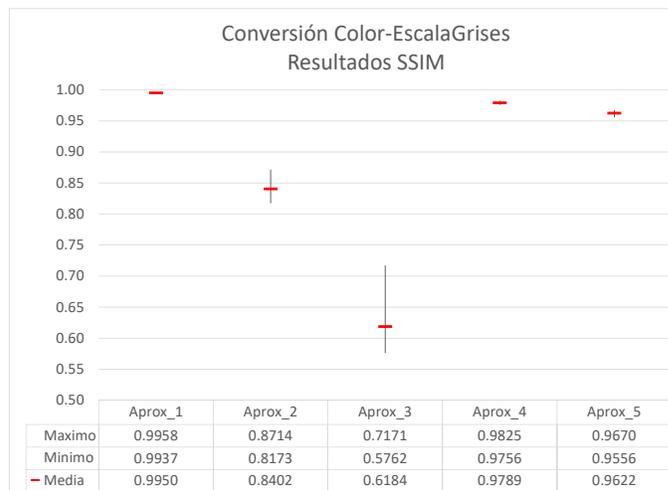


Figura 4.3: Resultados SSIM de conversión de color a escala de grises en LDW.

A cada conjunto de datos obtenidos durante la prueba de funcionamiento aproximado, se les aplicó el filtro media móvil de 100. De esta manera eliminar el ruido presente y permitir una mejor apreciación de la tendencia de las curvas en estudio. Estas se pueden observar en la Figura 4.4. A pesar de la degradación a la que se ven expuestas las imágenes, las curvas aproximadas son iguales a la de la versión exacta provista por la función de OpenCV. Por la razón anterior es que inclusive estando las seis superpuestas cada una a la aproximación anterior solo se observa el resultado de la aproximación 5.

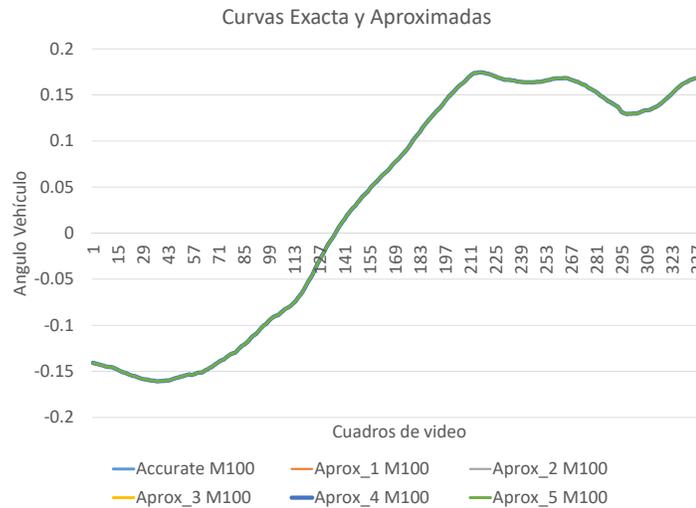


Figura 4.4: Gráfica Curvas Exacta y Aproximadas en LDW.

La conversión de color a escala de grises de manera aproximada, si bien altera la calidad de las imágenes tal y como se apreció en las Figuras 4.1, 4.2 y 4.3, no afectó al momento de la obtención de la información necesaria para la construcción de las curvas presentes en la Figura 4.4. Esto implica que esa degradación causada por cada una de las funciones producto de su procesamiento aproximado no alteró el resultado final.

Conversión aproximada de color a escala de grises para PPS

La conversión de color a escala de grises fue utilizada de igual manera en el PPS haciendo uso de cada una de las funciones aproximadas con las que se contaba. En la Figura 4.5 se muestra el mismo cuadro de video siendo preprocesado por cada una de las funciones aproximadas para la conversión de color a escala de grises.

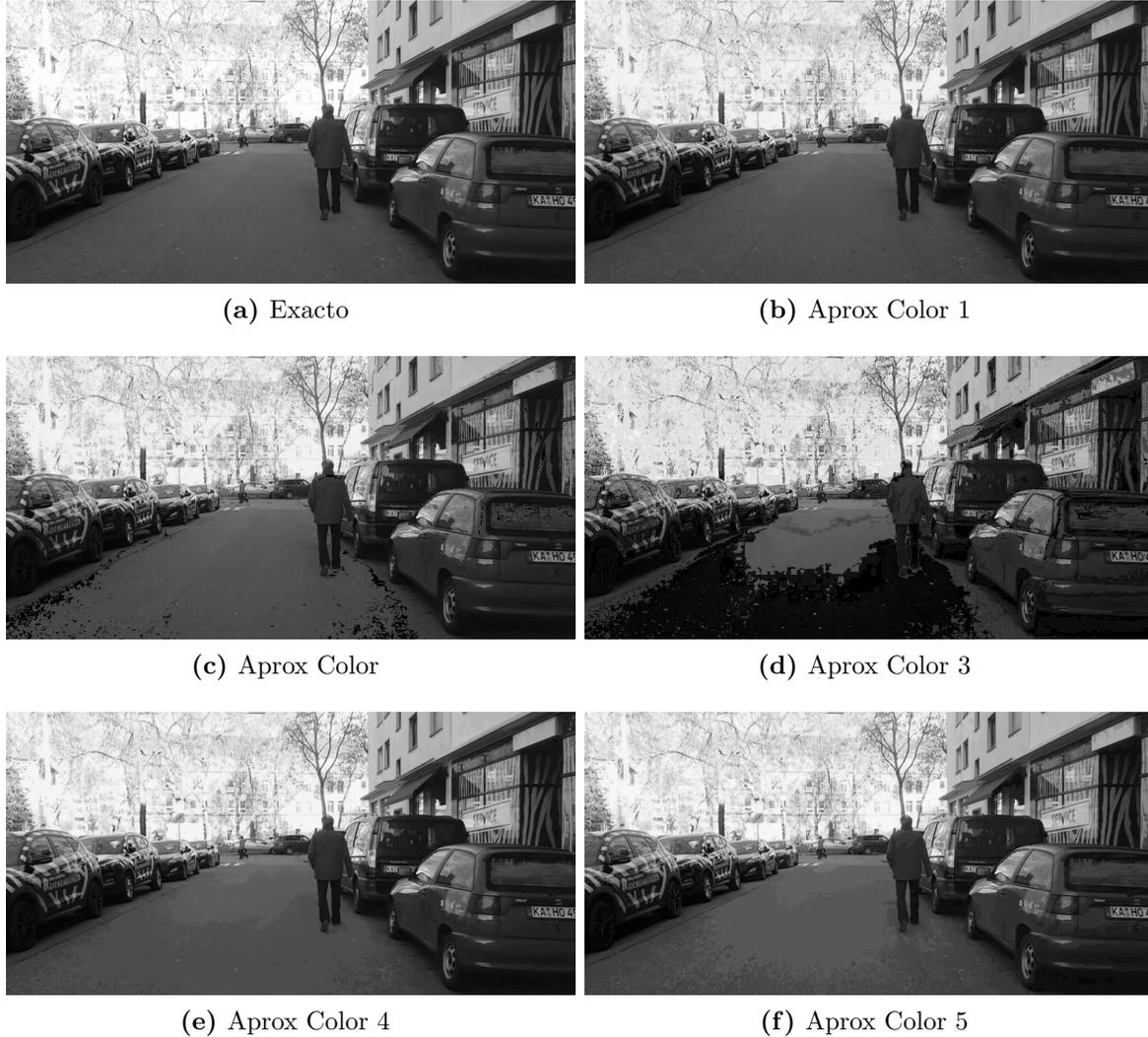


Figura 4.5: Resultados conversión de color a escala de grises en PPS

Al poner en marcha las corridas de PPS bajo las funciones aproximadas de conversión de color a escala de grises se obtuvieron resultados de PSNR de los cuales es posible inferir que a mayor cantidad de dB menor cantidad de ruido se presenta por cuadro de video, por lo que las aproximaciones 1 (32.26 dB), 2 (24.97 dB), 4 (33.82 dB) y 5 (31.40 dB) son las que muestran un comportamiento acorde al parámetro de validación de PSNR; a diferencia de la aproximación 3 (18.49 dB). Lo anterior se muestra en la gráfica de la Figura 4.6 mostrando la media de la agrupación de los datos con sus respectivos rangos.

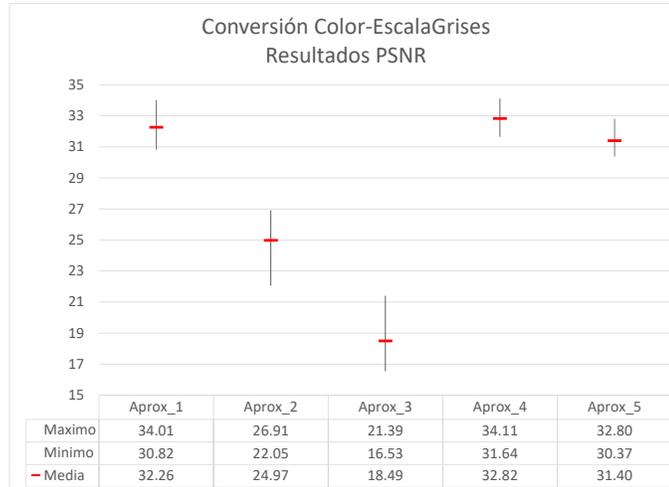


Figura 4.6: Resultados PSNR de conversión de color a escala de grises en PPS

De igual manera se obtuvieron resultados de SSIM de los cuales es posible demostrar que a mayor cercanía a “1” mayor es la similitud estructural que se presenta por cuadro de video, por lo que las aproximaciones 1 (0.9933), 4 (0.9785) y 5 (0.9529) son las que muestran un comportamiento aceptable; a diferencia de las aproximaciones 2 (0.8685) y 3 (0.6446). Lo anterior se muestra en la gráfica de la Figura 4.7 mostrando la media de la agrupación de los datos con sus respectivos rangos.

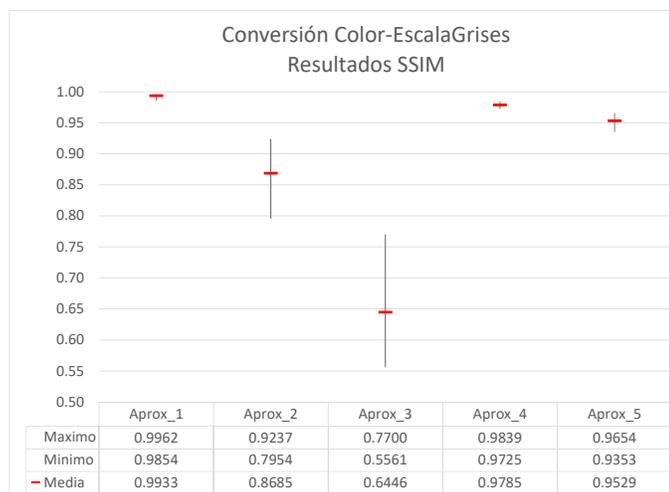


Figura 4.7: Resultados SSIM de conversión de color a escala de grises en PPS

A manera comparativa se muestran en la Figura 4.6 los resultados de PPS según la función de conversión de color a escala de grises utilizada.

En las gráficas de la Figura 4.8 se presentan los tres posibles estados expuestos en la sección 3.4.1:

- Altura de ROI igual a cero, significa que no hay peatón cerca, emite un “1”.
- Altura de ROI menor a 210 píxeles, significa que hay peatones cerca pero no hay peligro, emite un “1”.
- Altura de ROI mayor a 210 píxeles, significa que hay peatones peligrosamente cerca, emite un “0”.

Para el caso (a) el cambio de estado se da en el cuadro 88 y 155. Estos 2 datos anteriores son la base para la comparación. Los cuadros en los que se produce el cambio de estado en el caso (b) son 91 y 156, caso (c) son 88 y 154, caso (e) son 91 y 155 y caso (f) son 91 y 156. En el caso particular de caso (d) solo hay cambio de “2” a “1” en 112 pero no hay cambio a estado “0”.

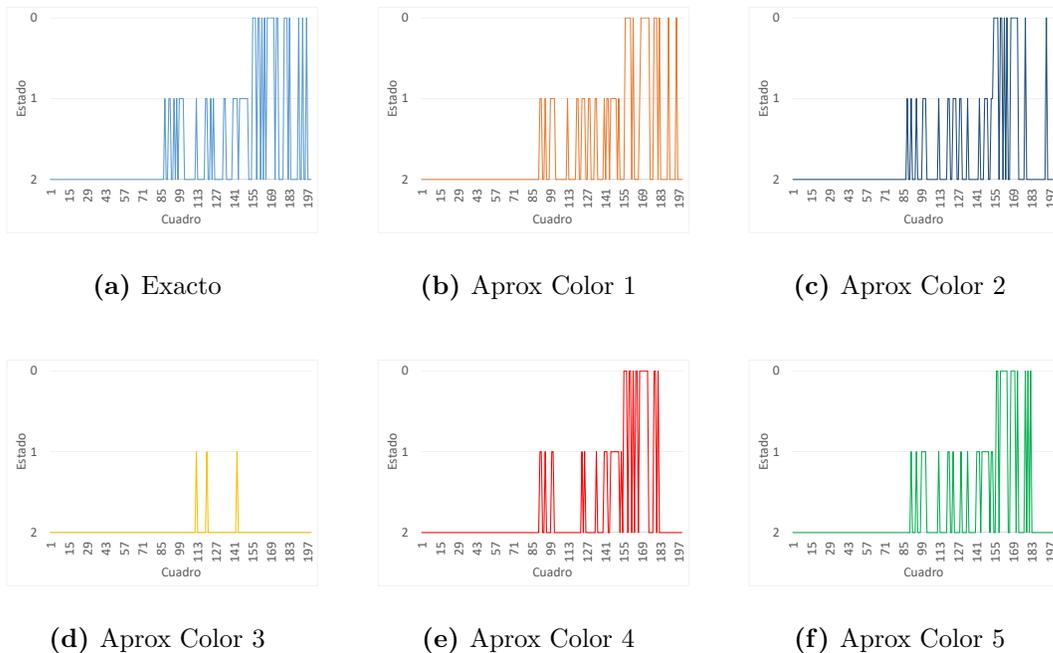


Figura 4.8: Resultados conversión de color a escala de grises en PPS

Exceptuando el caso (d), correspondiente a la aproximación 3, las demás aproximaciones obtuvieron resultados en los que la detección del peatón (estado) difería del exacto un máximo de tres cuadros. Tomando en consideración que el video se reproduce y procesa a 25 fps, esta diferencia corresponde a 0.12 segundos respecto al exacto, siendo un retardo con el cual se puede manejar la detección del peatón para el sistema de velocidad del vehículo.

Conversión aproximada de color a escala de grises para TSR

La conversión de color a escala de grises fue utilizada de igual manera en el TSR haciendo uso de cada una de las funciones aproximadas con las que se contaba. En la Figura 4.9 se muestra el mismo cuadro de video siendo preprocesado por cada una de las funciones aproximadas para la conversión de color a escala de grises.

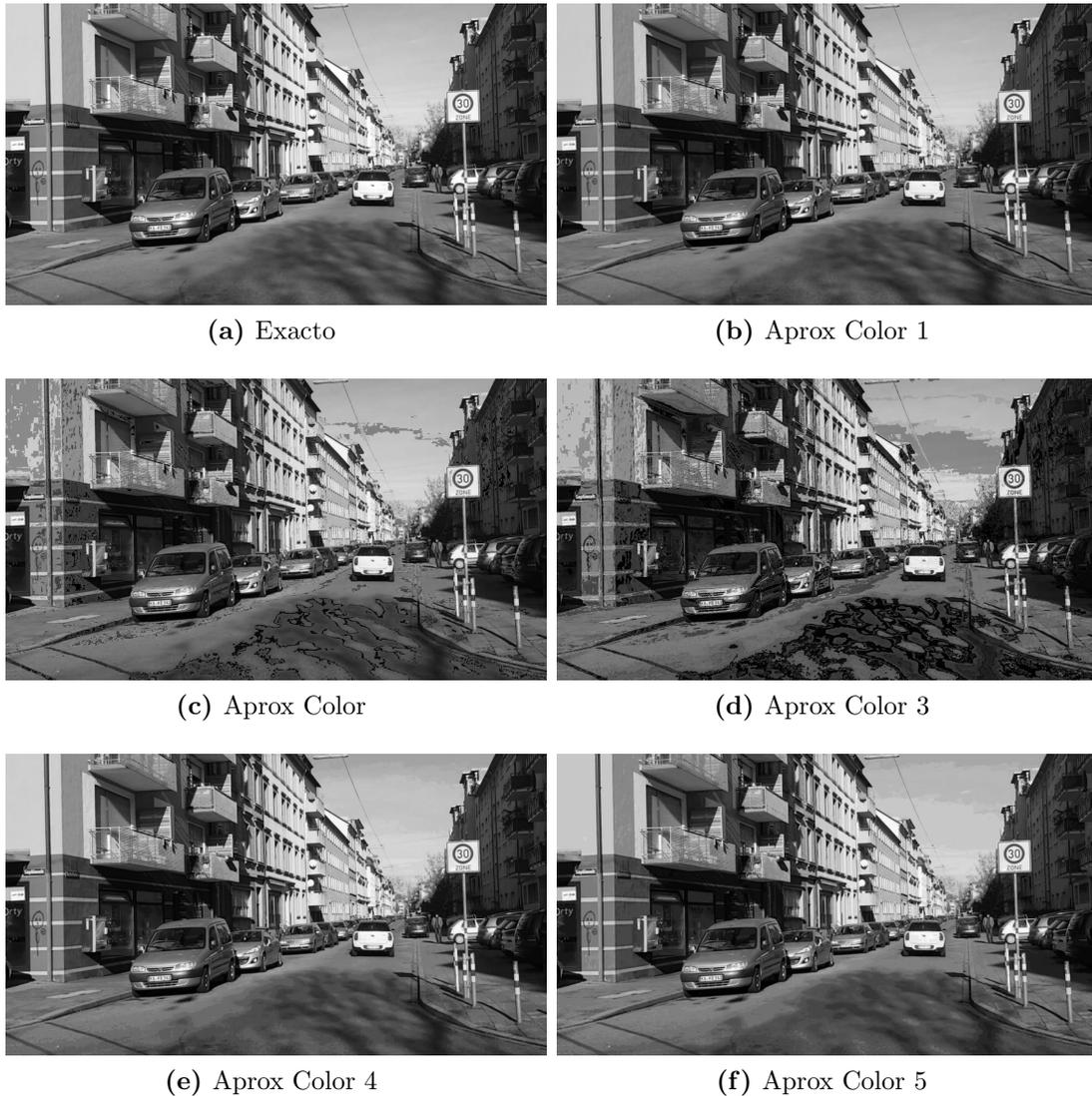


Figura 4.9: Resultados conversión de color a escala de grises en TSR

Al poner en marcha las corridas de TSR bajo las funciones aproximadas de conversión de color a escala de grises se obtuvieron resultados de PSNR de los cuales es posible inferir que a mayor cantidad de dB menor cantidad de ruido se presenta por cuadro de video, por lo que las aproximaciones 1 (33.00 dB), 2 (25.52 dB), 4 (33.11 dB) y 5 (31.83 dB) son las que muestran un comportamiento acorde; a diferencia de la aproximación 3 (19.04 dB). Lo anterior se muestra en la gráfica de la Figura 4.10 mostrando la media de la agrupación de los datos con sus respectivos rangos.

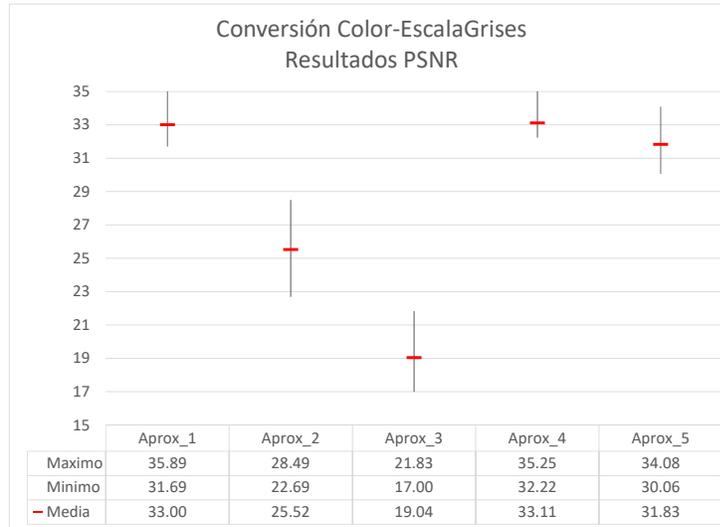


Figura 4.10: Resultados PSNR de conversión de color a escala de grises en TSR

De igual manera se obtuvieron resultados de SSIM de los cuales es posible demostrar que a mayor cercanía a “1” mayor es la similitud estructural que se presenta por cuadro de video, por lo que las aproximaciones 1 (0.9900), 4 (0.9732) y 5 (0.9483) son las que muestran un comportamiento adecuado; a diferencia de las aproximaciones 2 (0.8885) y 3 (0.7060). Lo anterior se muestra en la gráfica de la Figura 4.11 mostrando la media de la agrupación de los datos con sus respectivos rangos.

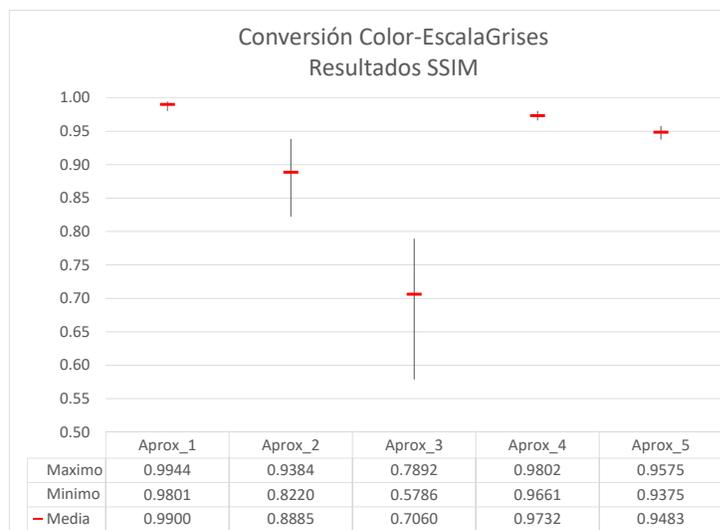


Figura 4.11: Resultados SSIM de conversión de color a escala de grises en TSR

La Figura 4.12 brinda una perspectiva clara de como se ven afectados los resultados de las predicciones realizadas por la SVM en el programa en discusión. Las aproximaciones que consiguen percibir las mismas señales en los mismos cuadros son la aproximación 1 y 5, las otras aproximaciones luego de realizar la conversión en escala de grises con un método aproximado produjeron que la SVM no detectara de forma acertada los límites de velocidad presentes en los cuadros en análisis.

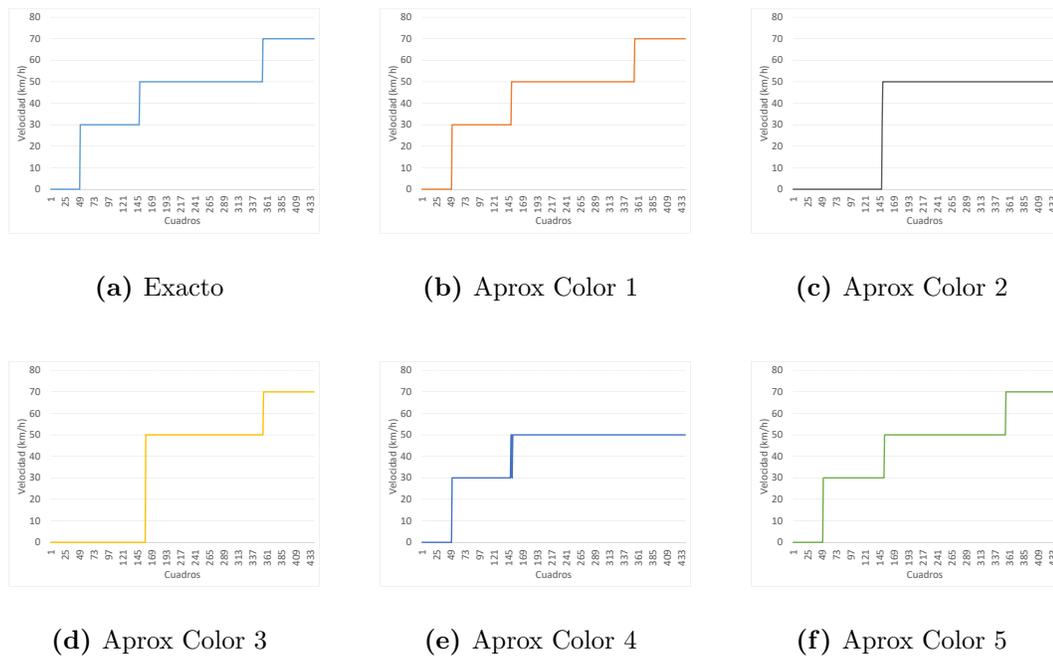


Figura 4.12: Resultados SSIM de conversión de color a escala de grises en TSR

4.2.2 Difuminado aproximado

El difuminado es una acción que se le realiza a las imágenes para suavizar, desenfocar, eliminar ruido, etc. En el procesamiento de imágenes, el difuminado consiste en reducir el contraste entre píxeles de un área determinada. Si bien OpenCV tiene sus funciones para realizar esta tarea, se realizó la exploración con 3 funciones aproximadas que realizan la misma función y así comprobar su desempeño.

1. Se realiza un proceso simplificado de difuminado. (Aprox Difuminado 1)
2. Se basa en el simplificado pero se suman valores de la matriz mediante sumador aproximado con el método LOA [48]. (Aprox Difuminado 2)
3. Se basa en el simplificado pero se suman valores de la matriz mediante sumador aproximado con el método LOA configurado de manera distinta al anterior [48]. (Aprox Difuminado 3)

Difuminado aproximado para LDW

El difuminado fue utilizado en el LDW haciendo uso de cada una de las funciones aproximadas con las que se contaba, definida anteriormente. En la Figura 4.13 se muestra el mismo cuadro de video siendo preprocesado por cada una de las funciones aproximadas para el difuminado aproximado.

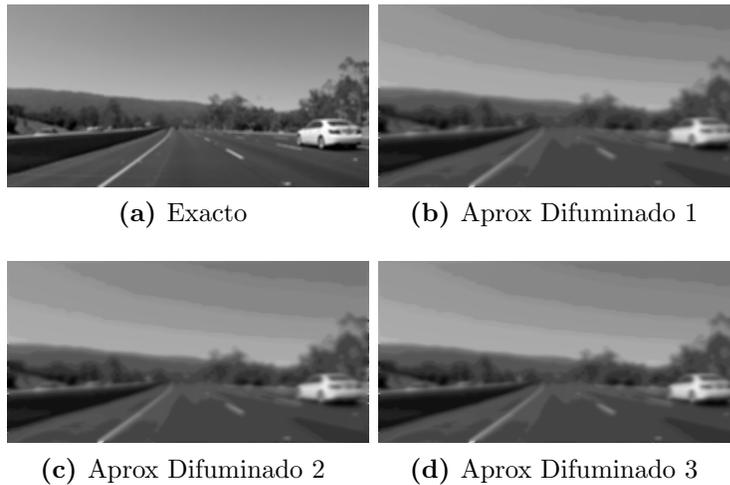


Figura 4.13: Resultados difuminado en LDW

Al poner en marcha las corridas de LDW bajo las funciones aproximadas de difuminado se obtuvieron resultados de PSNR de los cuales es posible inferir que a mayor cantidad de dB menor cantidad de ruido se presenta por cuadro de video, por lo que las aproximaciones presentan un comportamiento acorde al parámetro de validez según el valor de PSNR. Lo anterior se muestra en la gráfica de la Figura 4.14 mostrando la media de la agrupación de los datos con sus respectivos rangos.

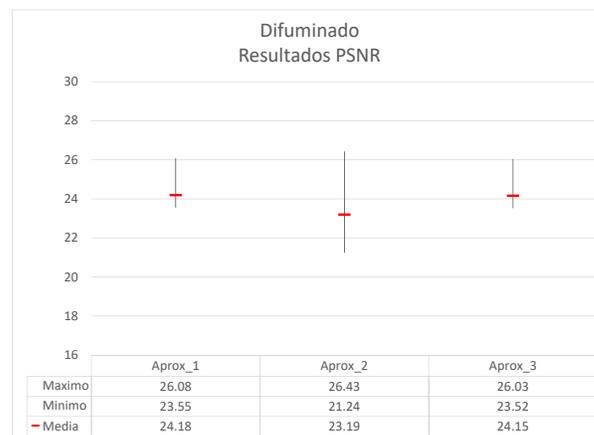


Figura 4.14: Resultados PSNR de difuminado en LDW

De igual manera se obtuvieron resultados de SSIM de los cuales es posible demostrar que a mayor cercanía a “1” mayor es la similitud estructural que se presenta por cuadro de video, las aproximaciones 1 (0.9673) y 2 (0.9633) cuentan con valores superiores al 0.95 indicado como mínimo a diferencia del difuminado aproximado 3 (0.9442). Lo anterior se muestra en la gráfica de la Figura 4.15 mostrando la media de la agrupación de los datos con sus respectivos rangos.

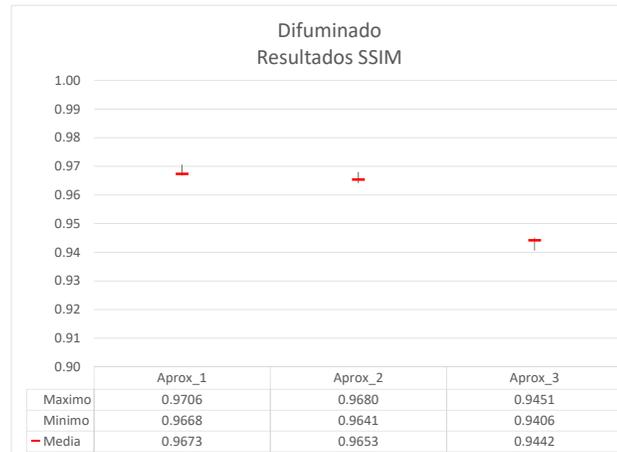


Figura 4.15: Resultados SSIM de difuminado en LDW

Luego de aplicar difuminado bajo funciones aproximadas se utilizaron para la detección LDW, resultando las curvas que se muestran en la Figura 4.16. A pesar de la degradación que experimentan las imágenes, las curvas aproximadas son similares en su tendencia, iguales a la de la versión exacta provista por la función de OpenCV.

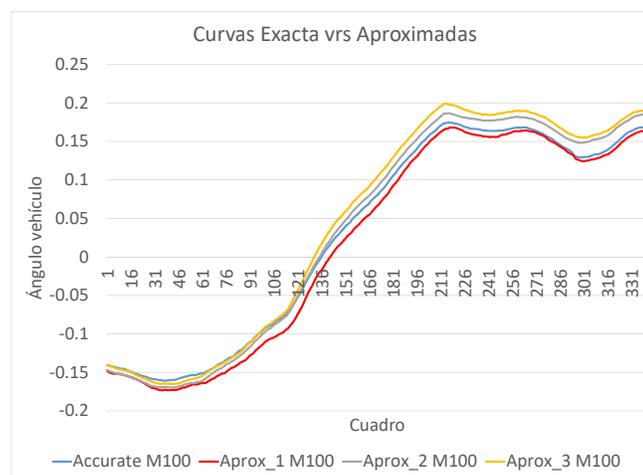


Figura 4.16: Gráfica Curvas Exacta y Aproximadas en LDW

Difuminado aproximado para PPS

El difuminado fue utilizado de igual manera en el PPS haciendo uso de cada una de las funciones aproximadas con las que se contaba. En la Figura 4.17 se muestra el mismo cuadro de video siendo preprocesado por cada una de las funciones aproximadas. Previo a este difuminado aproximado la imagen fue convertida de color a escala de grises mediante el método exacto.



Figura 4.17: Resultados difuminado en PPS

Al poner en marcha las corridas de PPS bajo las funciones aproximadas de difuminado se obtuvieron resultados de PSNR de los cuales es posible inferir que a mayor cantidad de dB menor cantidad de ruido se presenta por cuadro de video; las aproximaciones 1 (23.21 dB), 2 (23.19 dB) y 3 (23.17 dB) muestran un comportamiento similar entre ellas y en todos los casos mayor al parámetro de 20 dB de PSNR estipulado anteriormente. Lo anterior se muestra en la gráfica de la Figura 4.18 mostrando la media de la agrupación de los datos con sus respectivos rangos.

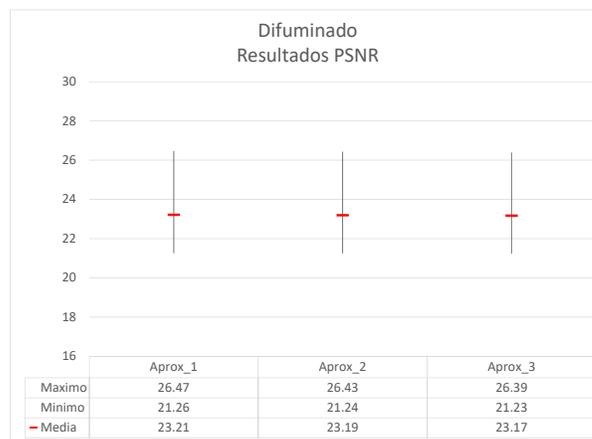


Figura 4.18: Resultados PSNR de difuminado en PPS

De igual manera se obtuvieron resultados de SSIM de los cuales es posible demostrar que a mayor cercanía a “1” mayor es la similitud estructural que se presenta por cuadro de video, las aproximaciones 1 (0.9678) 2 (0.9577) y 3 (0.9490) muestran un comportamiento cercano a “1”, sin embargo, la 3 es menor a 0.95. Lo anterior se muestra en la gráfica de la Figura 4.19 mostrando la media de la agrupación de los datos con sus respectivos rangos.

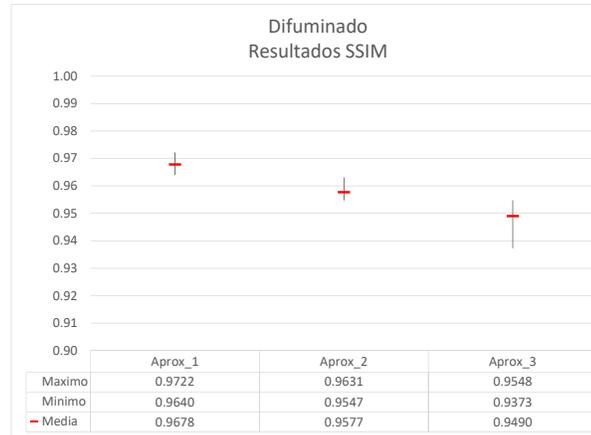


Figura 4.19: Resultados SSIM de difuminado en PPS

Luego de utilizar las funciones aproximadas para realizar el difuminado de los cuadros estos se usaron para definir el comportamiento del PPS con el fin de determinar la tolerancia de este ADAS. De esta forma, en la Figura 4.20 se muestran los resultados obtenidos según cada uno de los tres métodos aproximados en evaluación.

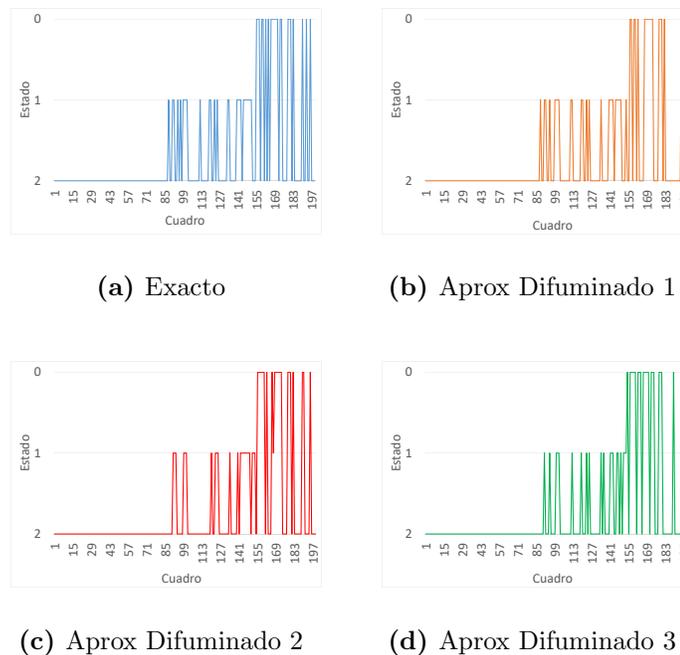


Figura 4.20: Resultados difuminado en PPS

Para el caso (a) el cambio de estado se da en el cuadro 88 y 155. Estos 2 datos anteriores son la base para la comparación con los obtenidos bajo difuminado aproximado. Los cuadros en los que se produce el cambio de estado en el caso (b) son 89 y 157, caso (c) son 92 y 156, y caso (d) son 92 y 155.

Los tres casos aproximados obtuvieron resultados en los que la detección del cambio de estado se retarda en un máximo de 2 cuadros respecto al exacto en cuanto a la detección del cambio de estado de “1” a “0”. Tomando en cuenta que el video se reproduce y procesa a 25 fps. Esta diferencia corresponde a 0.08 segundos respecto al exacto lo cual hace que estas aproximaciones se comporten similar al exacto.

4.2.3 Combinación de funciones aproximadas

En esta sección se realizaron pruebas de las ADAS seleccionadas bajo el uso de ambas funciones aproximadas simultáneamente, conversión de color a escala de grises y difuminado. Las combinaciones realizadas se fundamentan en los parámetros definidos al inicio de la sección 4.2, los cuales señalaban que los resultados aproximados en cada función eran validos sí y solo sí los valores eran mayores o iguales a 20 dB en PSNR y 0.95 en el índice de SSIM. Según esto se realizaron las siguientes combinaciones:

- Combinación 1
 - Conversión color a escala de grises aproximación 1.
 - Difuminado aproximación 1.
- Combinación 2
 - Conversión color a escala de grises aproximación 1.
 - Difuminado aproximación 2.
- Combinación 3
 - Conversión color a escala de grises aproximación 4.
 - Difuminado aproximación 1.
- Combinación 4
 - Conversión color a escala de grises aproximación 4.
 - Difuminado aproximación 2.
- Combinación 5
 - Conversión color a escala de grises aproximación 5.
 - Difuminado aproximación 1.

- Combinación 6
 - Conversión color a escala de grises aproximación 5.
 - Difuminado aproximación 2.

Este estudio de combinaciones de funciones aproximadas se realizó únicamente a las ADAS PPS y LDW. Estas dos cuentan en su código con ambas fuentes de aproximaciones (conversión de color a escala de grises y difuminado). En el caso particular de TSR únicamente requirió de conversión de color a escala de grises.

LDW

La combinación de funciones fue realizada para los 6 casos mencionados al inicio de esta sección. En la Figura 4.21 se muestra el mismo cuadro de video siendo preprocesado por cada una de las combinaciones de funciones aproximadas.

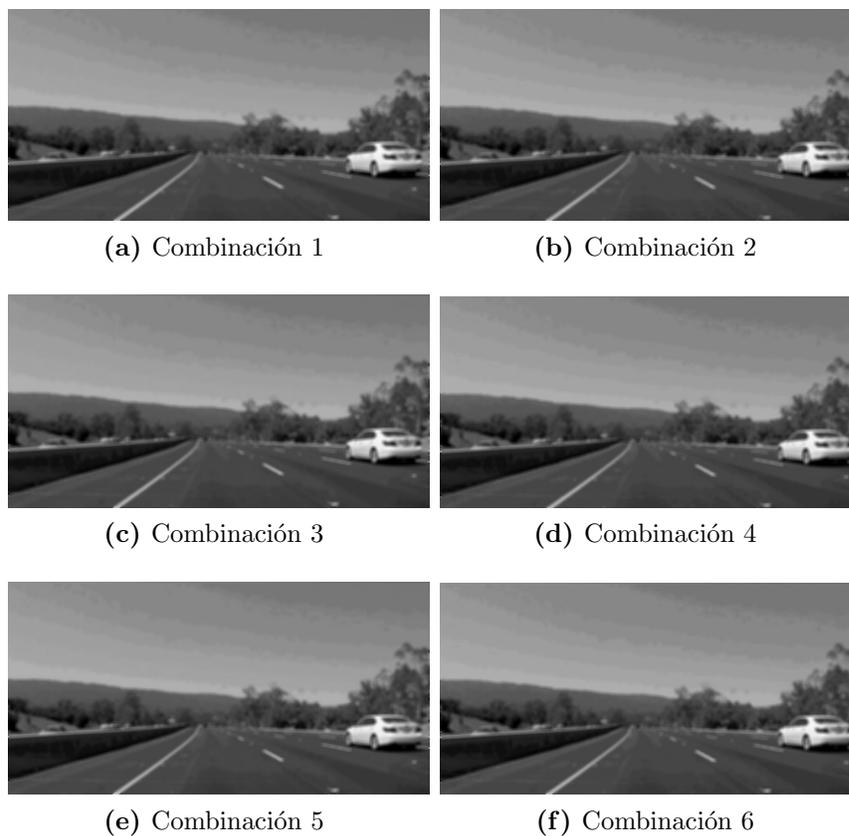


Figura 4.21: Resultados Combinación en LDW

Al poner en marcha las corridas de LDW bajo las combinaciones de funciones aproximadas se obtuvieron resultados de PSNR donde los 6 casos presentan valores mayores a 20 dB lo que indica que son válidos. Lo anterior se muestra en la gráfica de la Figura 4.22.

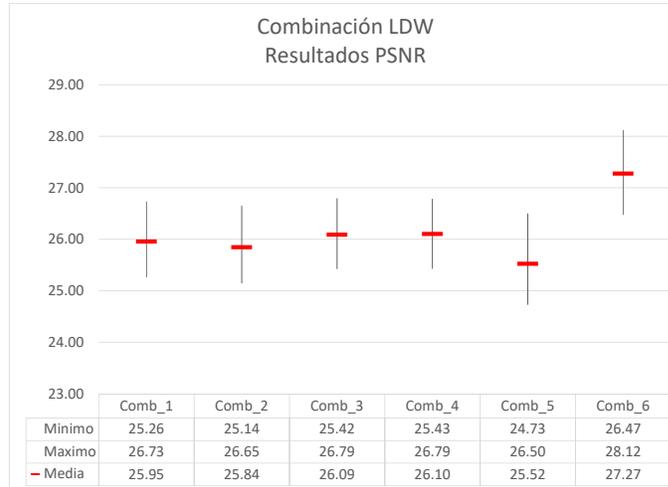


Figura 4.22: Resultados PSNR de combinación de funciones en LDW

De igual manera se obtuvieron resultados de SSIM donde los 6 casos combinados presentan valores medios por encima de 0.95 como se muestra en la gráfica de la Figura 4.23.

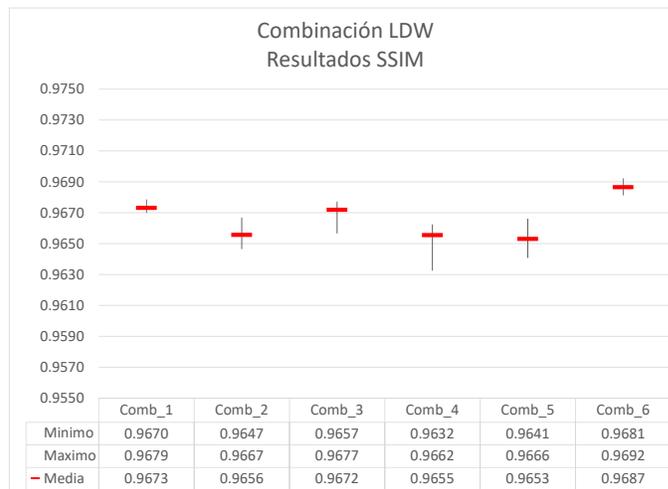


Figura 4.23: Resultados SSIM de combinación de funciones en LDW

En la Figura 4.24 se muestra gráficamente las curvas de ángulo de vehículo para las combinaciones de funciones aproximadas. Se observa un comportamiento similar entre todas las curvas de acuerdo a la tendencia que se obtuvo bajo funcionamiento exacto. Estas tendencias se asemejan entre sí debido a sus altos índices de SSIM y PSNR como se presentó en las Figuras 4.22 y 4.23.

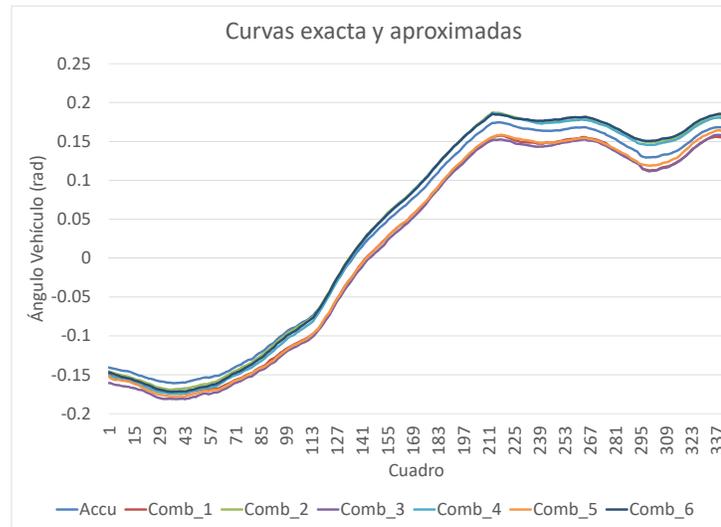


Figura 4.24: Gráfica Curvas Exacta y Aproximadas Combinadas en LDW

La pérdida de calidad es una característica propia de la AC la cual para este tipo de aplicaciones ADAS no se encuentra documentada por fabricantes ni universidades por lo que se establece un máximo de 10% de aceptación a errores. Mediante los datos generados por la LDW se obtuvo el valor promedio de la diferencia entre los valores de los ángulos para cada una de las combinaciones realizadas respecto al valor exacto del ángulo detectado. Estos resultados se presentan en la Tabla 4.1.

Tabla 4.1: Porcentaje de disminución de calidad

Comb 1	Comb 2	Comb 3	Comb 4	Comb 5	Comb 6
10.74%	5.66%	10.76%	5.97%	3.91%	5.72%

De acuerdo con lo mencionado anteriormente, la combinación 1 (10.74%) y 3 (10.76%) no cumplen con el criterio de aceptación de pérdida de calidad y las combinaciones 2 (5.66%), 4 (5.97%), 5 (3.91%) y 6 (5.72%) si brindan resultados favorables para la investigación realizada.

PPS

Al igual que en la LDW, la PPS cuenta con combinaciones de funciones y sus resultados se pueden apreciar gráficamente en la Figura 4.25 donde se muestra el mismo cuadro de video siendo preprocesado por cada una de las combinaciones realizadas.



Figura 4.25: Resultados Combinación en PPS

Luego de poner en marcha las corridas de PPS bajo las combinaciones de funciones aproximadas se obtuvieron resultados de PSNR donde los 6 casos presentan valores mayores a 20 dB lo que indica que son válidos. Lo anterior se muestra en la gráfica de la Figura 4.26.

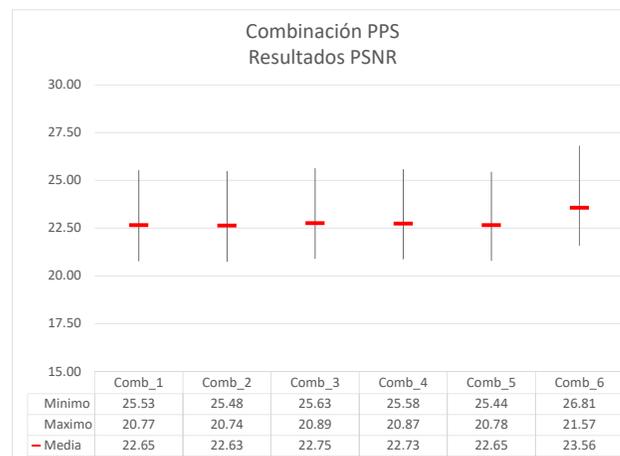


Figura 4.26: Resultados PSNR de combinación de funciones en PPS

De la misma manera que en el caso de LDW, se obtuvieron resultados de SSIM donde los 6 casos combinados presentan valores medios por encima de 0.95 como se muestra en la gráfica de la Figura 4.27.

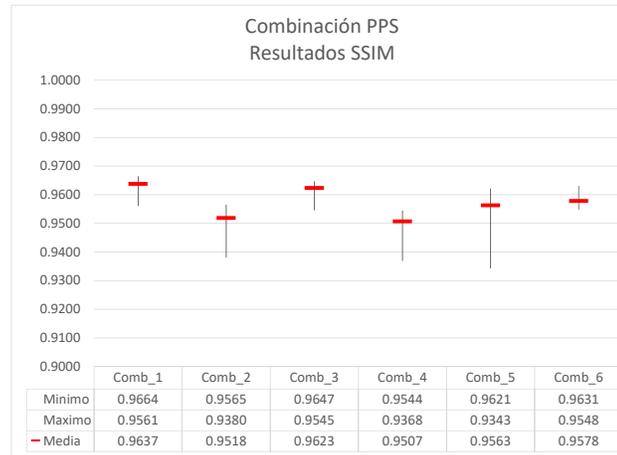


Figura 4.27: Resultados SSIM de combinación de funciones en PPS

En la Figura 4.28 se muestran comparativamente los resultados de PPS según la combinación de funciones utilizadas. De los mismos, se tiene que los instantes de cambio de estado son fundamentales para el presente estudio.

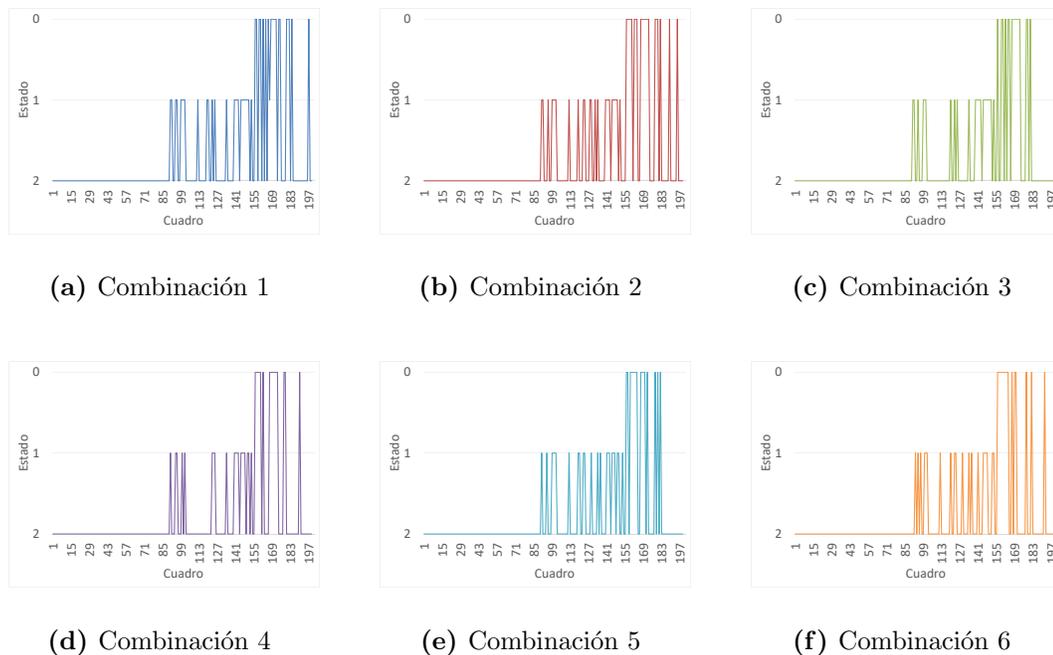


Figura 4.28: Resultados combinaciones en PPS

En las gráficas se presentan los tres posibles estados para cada caso. Para 5 de las 6 combinaciones el cambio de estado de “2” a “1” se efectuó en el mismo cuadro, únicamente en la combinación 6 hubo un retarde de 2 cuadros en la detección inicial del peatón, esto representa un retraso en la reacción del sistema de 0.08 segundos lo cual independientemente de la velocidad instantánea no afecta en el funcionamiento del control de velocidad debido a que este primer cambio de estado no genera una señal para el control correspondiente. En lo que concierne al cambio de estado de “1” a “0” las 6 combinaciones detectaron el peatón en el mismo cuadro. Esto quiere decir que al no presentar un retardo en la detección del peatón las 6 combinaciones de funciones aproximadas generan el cambio de estado “1” a “0” de la misma manera que la versión exacta, por ende la degradación producida en el preprocesamiento si bien su afectación se ve reflejada en los valores de PSNR y SSIM, no afecta la calidad de los resultados obtenidos donde su retraso es de 0 segundos, su retardo en el frenado es de 0 metros y el error es de 0%.

4.3 Implicaciones en sistemas de control

Una vez realizado el estudio de combinaciones se procedió a observar el comportamiento de estas en los sistemas de control respectivos a cada caso de estudio (Dirección y Velocidad). Este análisis se realizó con el fin de comprobar el efecto de las aproximaciones realizadas en etapas anteriores en los sistemas de control presentados en los Apéndice B y E.

4.3.1 Dirección

Haciendo uso de datos a partir de ADAS utilizando funciones aproximadas se realizaron pruebas en el sistemas de dirección propuesto anteriormente para computación exacta. De las 6 combinaciones se seleccionó la combinación 5 de LDW, la cual utiliza:

- Conversión color a escala de grises aproximación 5.
- Difuminado aproximación 1.

Esta combinación fue seleccionada debido a que fue la combinación con el porcentaje de diferencia entre la aproximación y versión exacta más bajo de los 6 casos estudiados. La simulación se realizó con los datos de la combinación escogida y se obtuvo la curva presente en la Figura 4.29. Esto bajo los mismos valores de masa y coeficiente de fricción que en el Capítulo 3.

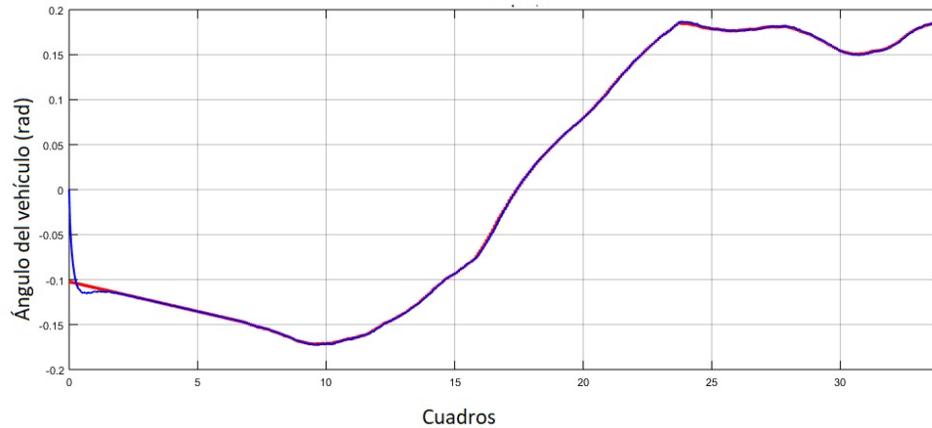


Figura 4.29: Gráfico Aproximado Ángulo vrs tiempo para vehículo de 1200kg

La respuesta del controlador no se ve afectada por el cambio de entrada del sistema de control. Esto quiere decir que el diseño del controlador se mantiene sin necesidad de hacerle modificaciones en los valores P, I y D debido a que las entradas aproximadas al sistema de control son similares a las entradas exactas así que si las entradas son buenas el controlador funcionará de la misma manera.

4.3.2 Velocidad

En el caso del estudio aproximado de velocidad se seleccionaron:

1. PPS

- Combinación 5
 - Conversión color a escala de grises aproximación 5.
 - Difuminado aproximado 1.

2. TSR

- Conversión color a escala de grises aproximación 1.

Realizando la simulación con los datos de la combinación escogida para PPS y la versión de TSR se obtuvo la curva presente en la Figura 4.30. Esto bajo los mismos parámetros presentes en el modelado del Apéndice D.

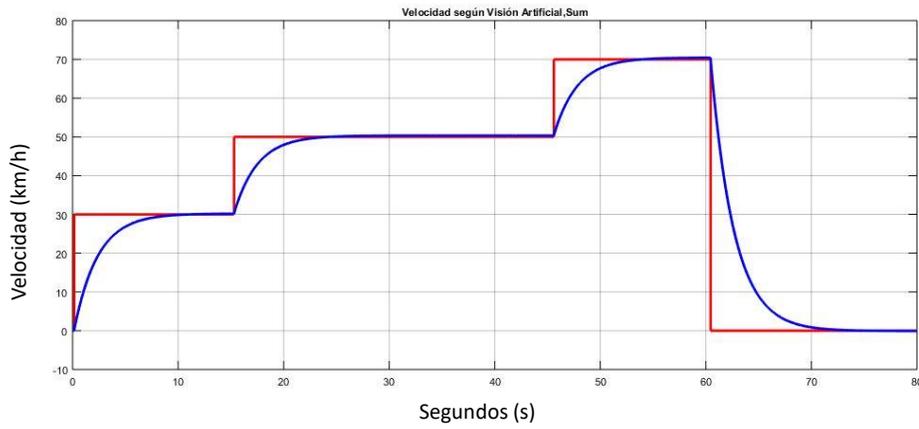


Figura 4.30: Ajuste de valores PID

Para este caso, la respuesta del controlador no se ve afectada por el cambio de entrada del sistema de control. Esto quiere decir que el diseño del controlador se mantiene sin necesidad de hacerle modificaciones en los valores P, I y D al igual que el caso anterior debido a que los retrasos de detección según cuadros de video no afectan significativamente los resultados.

Capítulo 5

Estudio Económico

La realización de este proyecto requirió de la utilización de equipo y licencias así como una serie de gastos adicionales con el fin de generar los resultados para la exploración realizada. La descripción de estos se enumera de mayor a menor según su monto:

1. Horas Ingeniero

Se calculó el costo de ingeniería tomando como base el salario de licenciado universitario utilizando como fuente el Ministerio de Trabajo de Costa Rica (MTSS) según el listado de salarios mínimos para el primer semestre del 2018 [49].

2. Sostenimiento en Alemania

Tomando en cuenta transporte público, gastos de alimentación y alquiler por seis meses se obtuvo el monto de Sostenimiento en Alemania.

3. Tiquetes aéreos

Se voló con la aerolínea Condor en vuelo directo desde San José, Costa Rica hasta Frankfurt, Alemania [50].

4. Licencia Simulink

Se requirió el uso de la herramienta Simulink para el desarrollo de los controladores. La licencia se vende por plazos de 1 año y se utilizó por 6 meses entonces se agrega el monto correspondiente al periodo citado [51].

5. Licencia Matlab

Se requirió el uso del software Matlab para el desarrollo de los controladores. La licencia al igual que la de Simulink se vende por plazos de 1 año y se utilizó por 6 meses por tanto se agrega el monto correspondiente [52].

6. Seguro Médico Internacional

El seguro médico internacional fue adquirido con la empresa Mawist, el cuál se contrató por los 6 meses de duración del proyecto [53].

7. Equipo computacional

Se hizo uso de la computadora de uso personal a lo largo de los 6 meses de duración de la investigación. Por lo que se cuantifica únicamente el uso durante ese periodo. Este se calculó tomando en cuenta el costo del equipo [54] dividido entre 24 meses de vida útil del mismo según el artículo de la reportera Siegel del diario The Guardian [55] y este se multiplicó por los 6 meses de uso.

8. Cámara Monoscópica para obtención de videos fuente

Para la toma de los videos se requirió del uso de una cámara monoscópica. [56].

9. Visado alemán

Los trámites del visado alemán tuvieron un costo de 60 euros.

10. Seguro de equipo

Se aseguró el equipo de computo ante daños y robo con el Instituto Nacional de Seguros (INS). La cotización se presenta en el Anexo III.

11. Horas Técnico para generación de videos fuente

Finalmente, la toma de los videos necesarios para probar el funcionamiento del sistema se contabiliza como horas técnico según el listado de salarios mínimos del MTSS [49].

Desglose de gastos incurridos

En la Tabla 5.1 que se presenta a continuación los montos en colones de cada uno de los rubros tomados en cuenta para este estudio económico.

Tabla 5.1: Gastos incurridos para la realización del proyecto

Descripción	Monto en colones
Ingeniero	3 275 360
Estancia Alemania	3 000 000
Tiquetes aéreos	658 323
Licencia Simulink	401 130
Licencia Matlab	265 537
Seguro médico	245 190
Computadora	212 500
Cámara Monoscópica	50 000
Visado	41 615
Grabación Videos	16 880
Seguro de equipo	15 552
Costo Directo del Proyecto	8 182 087

El monto final de la Tabla 5.1 brinda el costo directo del proyecto que se realizó, el cual es de **8 226 535** colones. Sin embargo, se deben tomar en consideración los imprevistos y la utilidad, a lo que se tomó un 5% y un 10% respectivamente del costo directo. Por tanto, se presenta el costo total del proyecto en **9 409 400.05** colones (nueve millones cuatrocientos nueve mil cuatrocientos colones con cinco céntimos).

Capítulo 6

Conclusiones y Trabajo futuro

6.1 Conclusiones

El análisis de tolerancia a errores en aplicaciones ADAS y su efecto en un sistema de control fue llevado a cabo con éxito y los resultados obtenidos lo demuestran.

El trabajo presenta una investigación de los distintos tipos de ADAS existentes y a partir de esta información el trabajo presenta una taxonomía de la cual se escogieron 3 ADAS debido a que atacaban directamente el problema planteado, las cuales fueron: LDW, TSR y PPS. Esta taxonomía se realizó separando las ADAS en conjuntos según tareas realizadas.

Se realizó para cada ADAS una implementación en c++ mediante el uso de bibliotecas de openCV y mediante sistemas de control, con plantas adecuadamente modeladas, se contó con los medios de pruebas para las señales de detección de estas ADAS.

Las ADAS implementadas contaban con fuentes de aproximación en los preprocesamientos de conversión de color a escala de grises y difuminado que realizaban antes de proceder a extraer la información necesaria para su correspondiente funcionamiento. La exploración de tolerancia a errores se realizó para 23 escenarios en total (15 de conversiones de color a escala de grises y 6 de difuminado).

Las combinaciones de funciones aproximadas se realizaron únicamente para LDW y PPS debido a que contaban con ambas fuentes de aproximación y se eligieron 6 de 15 combinaciones posibles por ADAS basados en parámetros de validación establecidos. La TSR por su parte, solo contaba con la conversión de color a escala de grises por lo que no se pudo combinar con otras potenciales fuentes de aproximaciones.

La TSR presentó un comportamiento igual al exacto en las aproximaciones 1 y 5 donde las salidas realizan los cambios de velocidad según detección 100% igual al exacto. Las aproximaciones 2, 3 y 4 no detectaron los cambios de señal de velocidad adecuadamente. La LDW demostró ser compatible a los métodos aproximados para ambas funciones por separado, donde obtuvo resultados iguales en conversión aproximada de color a

escala de grises y levemente distintos en difuminado aproximado. Al combinar funciones aproximadas se obtuvieron resultados que según los parámetros de validación fueron favorables para las combinaciones 2, 4, 5 y 6 donde su pérdida de calidad respecto al exacto fue 5.66%, 5.97%, 3.91% y 5.72% respectivamente. La PPS presentó resultados aceptables en las combinaciones realizadas donde, en comparación al exacto, los cambios de estado si bien se vieron retardados 2 cuadros del estado “2” a “1”, del “1” a “0” se realizó el cambio de igual manera del exacto sin presentar ningún retardo, por lo que la diferencia entre la versión de combinaciones aproximadas y exacta es 0%.

Los resultados anteriormente mencionados comprueban que haciendo uso de herramientas de computación aproximada se le permite a las distintas ADAS manipular la información de manera distinta obteniendo resultados positivos mediante el uso de sumadores aproximados. Estos como se menciona en la sección 4.2, brindan beneficios de rapidez, simplicidad y ligereza en los cálculos, por ende menor esfuerzo computacional.

A nivel de control, se seleccionó el caso de dirección la combinación de LDW con funciones aproximadas 5, donde el controlador funcionó de igual manera ante esta entrada aproximada. En lo que concierne al estudio de velocidad, para la TSR se seleccionó la aproximación 1 y para la PPS se seleccionó la combinación de funciones aproximadas 5. A nivel de control esta tuvo el mismo comportamiento que bajo funcionamiento exacto.

6.2 Trabajo futuro

El trabajo actual establece una base para que se pueda posteriormente continuar la exploración de tolerancia a errores en aplicaciones ADAS en hardware. Al comprobar que las fuentes de aproximación seleccionadas demostraron ser compatibles con este paradigma de computación en el preprocesamiento de estas ADAS, la búsqueda de otras fuentes de aproximación en etapas de procesamiento es uno de los siguientes pasos.

Si bien los alcances de este trabajo permitieron la exploración de tolerancia y su efecto en un ambiente simulado, esto da pie para una segunda etapa donde se pretende contar con un vehículo a escala y así realizar un modelado de una planta física para comprobar su funcionamiento en tiempo real. Para esto se debe mudar a un sistema real con sistemas embebidos.

Finalmente, se debe cuantificar la disminución de área y consumo de recursos computacionales, así como tiempos de ejecución y potencia.

Bibliografía

- [1] I. Rodríguez, “Accidentes viales toman tercer lugar como causa de muerte en costa rica,” *La Nación*, 2016. [Online]. Available: http://www.nacion.com/vivir/bienestar/Accidentes-viales-tercer-causas-muerte_0_1595240471.html
- [2] COSEVI, “Cantidad de fallecidos en sitio por año según tipo de usuario,” *COSEVI*, 2017. [Online]. Available: <http://datosabiertos.csv.go.cr/dashboards/19669/fallecidos/1>
- [3] D. Bosque, “Accidentes de tránsito dejaron 448 muertes en 2016,” *La Nación*, 2017. [Online]. Available: http://www.nacion.com/nacional/transportes/Accidentes-transito-dejaron-muertes_0_1607439269.html
- [4] S. Chinchilla, “Costa Rica entra en emergencia por epidemia de muertes en vías,” *La Nación*, 2017. [Online]. Available: http://www.nacion.com/nacional/transportes/Pais-entra-emergencia-epidemia-muertes_0_1650434973.html
- [5] “Approximate Computing: A Survey,” *IEEE Design and Test*, vol. 33, no. 1, pp. 8–22, 2016.
- [6] SAE, “Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles,” 2016.
- [7] Intel, “Advanced Driver Assistant System: Theats, Requirements and Security Solutions,” *White Paper*, 2016.
- [8] A. Dubey, “Stereo vision — Facing the challenges and seeing the opportunities for ADAS applications,” 2016.
- [9] I. Alava, “Cámaras infrarrojas para investigación y desarrollo,” *FLIR Systems*, 2010.
- [10] J.-E. Källhammer, “Night vision: Requirements and possible roadmap for FIR and NIR systems,” *Proceedings of SPIE*, no. table 1, pp. 61 980F–61 980F–11, 2006. [Online]. Available: <http://link.aip.org/link/?PSISDG/6198/61980F/1>
- [11] A. Manual, “Application Manual,” *Ultrasonic Sensor*, pp. 1–17, 2008.
- [12] J. Ha, B. Jeon, W. Jun, J. Lee, and H. Jeong, “An improved 2D cost aggregation method for advanced driver assistance systems,” *IEEE Intelligent Vehicles Symposium, Proceedings*, vol. 2015-Augus, no. Iv, pp. 89–94, 2015.

- [13] D. Hospach, S. Mueller, O. Bringmann, J. Gerlach, and W. Rosenstiel, "Simulation and evaluation of sensor characteristics in vision based advanced driver assistance systems," *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*, pp. 2610–2615, 2014.
- [14] R. Thakur, "Scanning LIDAR in advanced driver assistance systems and beyond," *IEEE Consumer Electronics Magazine*, vol. 5, no. 3, pp. 48–54, 2016.
- [15] Swov, "SWOV Fact sheet," *October*, vol. 3, no. October, pp. 1–6, 2008.
- [16] H.-j. Wang, N. Ren, W.-q. Zhang, T.-t. Zhao, C. Lun-feng, Z. Rong-xue, and T. Feng, "Research on Unmanned Vehicle Traffic Signal Recognition Technology," *2010 International Conference on Intelligent System Design and Engineering Application*, pp. 298–301, 2010. [Online]. Available: <http://ieeexplore.ieee.org/document/5743426/>
- [17] W. Wang, S. Sun, M. Jiang, Y. Yan, and X. Chen, "Traffic lights detection and recognition based on multi-feature fusion," *Multimedia Tools and Applications*, vol. 76, no. 13, pp. 14829–14846, 2017. [Online]. Available: <http://link.springer.com/10.1007/s11042-016-4051-5>
- [18] S. Hosseini, N. Murgovski, G. R. De Campos, and J. Sjoberg, "Adaptive forward collision warning algorithm for automotive applications," *Proceedings of the American Control Conference*, vol. 2016-July, pp. 5982–5987, 2016.
- [19] Y.-C. Kuo, C.-M. Fu, C.-T. Tsai, C.-C. Lin, and G.-H. Chang, "Pedestrian Collision Warning of Advanced Driver Assistance Systems," *2016 International Symposium on Computer, Consumer and Control (IS3C)*, pp. 740–743, 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7545299/>
- [20] J. B. Pinto Neto, L. C. Gomes, E. M. Castanho, M. M. Elias Campista, L. M. Henrique K Costa, and P. M. Cezar Ribeiro, "An Algorithm for Error Correction in Forward Collision Warning Application," *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 10–14, 2016.
- [21] O. Otaegui and G. Velez, "Embedding vision-based advanced driver assistance systems: a survey," *IET Intelligent Transport Systems*, pp. 1–29, 2016. [Online]. Available: <http://digital-library.theiet.org/content/journals/10.1049/iet-its.2016.0026>
- [22] V. Butakov, "Personalized driver/vehicle lane change models for adas," 2015.
- [23] G. Vélez, O. Otaegui, J. D. Ortega, M. Nieto, and A. Cortés, "On creating vision-based advanced driver assistance systems," *IET Intelligent Transport Systems*, vol. 9, no. 1, pp. 59–66, 2015. [Online]. Available: <http://digital-library.theiet.org/content/journals/10.1049/iet-its.2013.0167>

- [24] S. Park, D. Kim, S. Han, M. J. Kim, and J. Paik, "Monocular fisheye lens model-based distance estimation for forward collision warning systems," *IEEE Vehicular Technology Conference*, 2017.
- [25] "Parking assist system," vol. 1, no. 19, 2010.
- [26] T. INSTRUMENTS and R. Ma, "Automotive Adaptive Front-lighting System Reference Design," no. July, pp. 1–42, 2013.
- [27] R. S. Wiper, "Automatic Sensing," vol. 6, pp. 3–9, 2004.
- [28] G. Bradski and A. Kaehler, *Projection and 3D Vision*, 2008.
- [29] K. Ogata, *Ingeniería de Control Moderna*, 2013, vol. 53, no. 9.
- [30] V. Mazzone, "Controladores PID," p. 12, 2012. [Online]. Available: <http://www.eng.newcastle.edu.au/~jhb519/teaching/caut1/Apuntes/PID.pdf>
- [31] J. Han, "Approximate computing: An emerging paradigm for energy-efficient design," *IEEE*, 2013.
- [32] L. Ceze, "Disciplined approximate computing: From language to hardware, and beyond," 2017.
- [33] J. Park, "Axcgames: Towards crowdsourcing quality target determination in approximate computing," 2016.
- [34] D. Hernandez, "Desarrollo de una herramienta para la generacion de circuitos aritmeticos aproximados," 2017.
- [35] J. Castro, "Compiler-driven error analysis for designing approximate accelerators," 2016.
- [36] J. Hu, "A new approximate adder with low relative error and correct sign calculation," 2013.
- [37] A. Yusra, "Comparison of image quality assessment:psnr, hvs, ssim, uiqu," *International Journal of Scientific Engineering Research*, 2012. [Online]. Available: <https://www.ijser.org/researchpaper/Comparison-of-Image-Quality-Assessment-PSNR-HVS-SSIM-UIQI.pdf>
- [38] A. Horé, "Image quality metrics: Psnr vs. ssim," pp. 2366–2369, 08 2010.
- [39] J. Proakis, "Tratamiento Digital de Señales," *Pearson*, p. 102, 2007.
- [40] D. Takács, "Contact patch memory of tyres leading to lateral vibrations of four-wheeled vehicles," p. 4, 2012.

- [41] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, “Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition,” *Neural Networks*, no. 0, pp.–, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0893608012000457>
- [42] I. E. T. . Dissertations, “Performance evaluation factors of stereo image compression, chapter 7,” 2016.
- [43] Imatest, “Documentation ssim: Structural similarity index,” 2017.
- [44] B. Bing, *3D and HD Broadband Video Networking*, 2010.
- [45] O. S. C. Vision, “Color conversions,” 2017. [Online]. Available: https://docs.opencv.org/3.1.0/de/d25/imgproc_color_conversions.html
- [46] M. Keith, *Video Demystified: A Handbook for the Digital Engineer*, 2005.
- [47] M. Shafique, W. Ahmad, R. Hafiz, and J. Henkel, “A low latency generic accuracy configurable adder,” pp. 1–6, June 2015.
- [48] H. Jiang, “A comparative review and evaluation of approximate adders,” 2017. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.702.1823&rep=rep1&type=pdf>
- [49] MTSS, “Salarios mínimos 2018,” 2018. [Online]. Available: <http://www.mtss.go.cr/temas-laborales/salarios/lista-salarios.html>
- [50] Condor, “Booking,” 2018. [Online]. Available: <https://www.condor.com/es/index.jsp>
- [51] Mathworks, “Pricing and licensing,” 2017. [Online]. Available: <https://la.mathworks.com/pricing-licensing.html?prodcode=SL&intendeduse=undefined>
- [52] —, “Pricing and licensing,” 2017. [Online]. Available: <https://la.mathworks.com/pricing-licensing.html?prodcode=ML&intendeduse=undefined>
- [53] Mawista, “Seguro médico,” 2018. [Online]. Available: <https://www.mawista.com/>
- [54] H. P. Enterprise, “Hp store,” 2018. [Online]. Available: <http://store.hp.com/us/en/SearchDisplay?client=&searchTerm=envy&search=&charset=utf-8&storeId=10151&catalogId=10051&langId=-1&beginIndex=0&pageSize=12>
- [55] L. Siegle, “What is the lifespan of a laptop?” *The Guardian*, 2013. [Online]. Available: <https://www.theguardian.com/environment/2013/jan/13/lifespan-laptop-pc-planned-obsolescence>
- [56] Amazon, “Dash cam,” 2018. [Online]. Available: https://www.amazon.com/s/ref=nb_sb_noss_2?url=search-alias%3Daps&field-keywords=dash+cam&rh=i%3Aaps%2Ck%3Adash+cam

- [57] E. Velasco, “Tema: El Sistema de Dirección,” *Universitas Miguel Hernandez*, 2013. [Online]. Available: <http://umh1796.edu.umh.es/wp-content/uploads/sites/272/2013/02/sistema-de-direccion-texto1.pdf>
- [58] R. Corrado, “El Sistema de Dirección,” *Dto de Automotores*, 2017. [Online]. Available: <http://escueladeltrabajo.net/direccion.pdf>
- [59] R. Jazar, “Steering Dynamics. In: Vehicle Dynamics: Theory and Application,” p. 390, 2008.
- [60] R. Rajamani, “Lateral Vehicle Dynamics,” *Vehicle Dynamics and Control*, 2006.
- [61] C. Tutorials, “Cruise Control: System Modeling,” *Control Tutorials for Matlab Simulink*, 2017. [Online]. Available: <http://ctms.engin.umich.edu/CTMS/index.php?example=CruiseControl§ion=SystemModeling>
- [62] M. Oliveira, “Multimodal inverse perspective mapping,” *ScienceDirect*, 2014.
- [63] O. dev team, “Color conversions,” 2017. [Online]. Available: https://docs.opencv.org/3.1.0/de/d25/imgproc_color_conversions.html
- [64] OpenCV, “Image filtering,” 2017. [Online]. Available: <https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html?highlight=blur#blur>
- [65] O. dev team, “Feature detection,” 2017. [Online]. Available: https://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html?highlight=canny#canny
- [66] OpenCV, “Cascade classification,” 2017. [Online]. Available: https://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html
- [67] O. dev team, “Machine learning,” 2017. [Online]. Available: <https://docs.opencv.org/3.0-beta/modules/ml/doc/ml.html>
- [68] OpenCV, “Decision tree,” 2017. [Online]. Available: https://docs.opencv.org/3.0-beta/modules/ml/doc/decision_trees.html
- [69] O. dev team, “Neural network,” 2017. [Online]. Available: https://docs.opencv.org/3.0-beta/modules/ml/doc/neural_networks.html
- [70] OpenCV, “Support vector machine,” 2017. [Online]. Available: https://docs.opencv.org/3.0-beta/modules/ml/doc/support_vector_machines.html

Apéndice A

Modelado Sistema de Dirección

Para modelar un Sistema de Dirección de un vehículo se debe comprender el principio de Ackerman. En la Figura A.1 se muestra que el ángulo de viraje de la rueda que se presenta del lado interior (δ_i) a centro del radio de giro (O) es mayor que el ángulo exterior (δ_o). Esto se debe a que en una curva cada una de las ruedas delanteras describe una circunferencia de distinto radio. Como el radio que describe en la curva la rueda interior es menor que el de la exterior, si girasen igual las dos ruedas la interna o se trabaría, o sería arrastrada en las curvas [57]. Para evitar esto, Ackerman, a principios de siglo estudiando los sistemas de dirección en los carruajes, estableció que era necesario que la prolongación de los brazos de dirección se corten a la altura del eje trasero. De esta forma la rueda interior de la curva girará más que la exterior y no será arrastrada. A la diferencia entre los valores de los dos ángulos cuando la dirección se encuentra girada a tope, es a lo que se denomina efecto Ackerman o ángulo de viraje [58].

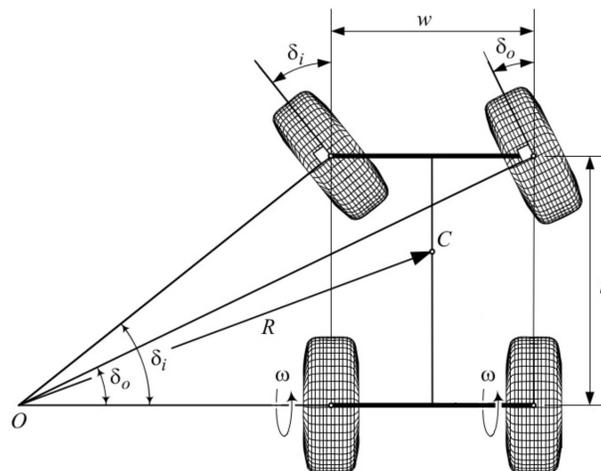


Figura A.1: Ángulos Ackerman [59].

Un vehículo de cuatro ruedas se puede modelar bajo el *Bicycle Model* (Modelo Bicicleta). Este es un modelo simplificado de vehículo. Como se observó en la Figura A.1 los ángulos

de viraje son distintos en las ruedas delanteras y para este modelado se toma un ángulo de viraje equivalente (δ). En la Figura A.2 se presenta el Diagrama de Cuerpo Libre de un vehículo simplificado.

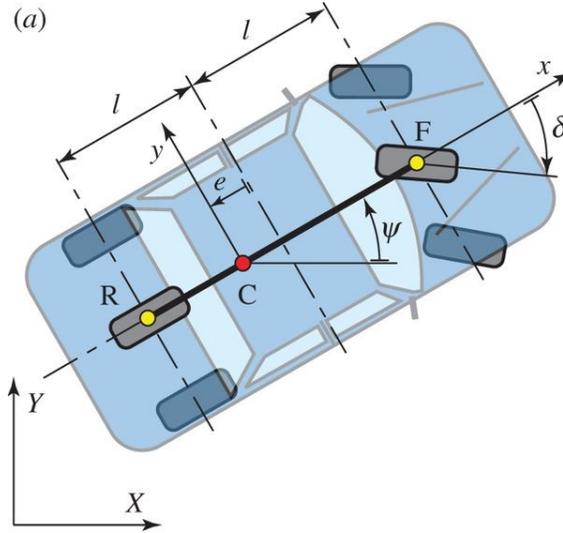


Figura A.2: Modelado de vehículo de 4 ruedas a *Bicycle Model* [40].

Aplicar sumatoria de momentos igual a cero es el primer paso del modelado.

$$\dot{r} \cdot I_z = l_f \cdot F_{yf} - l_r \cdot F_{yr} \quad (\text{A.1})$$

Los valores de las fuerzas requeridas están dadas por [60] de la siguiente manera:

$$F_{yf} = C_{\alpha f} \cdot \alpha_f \quad (\text{A.2})$$

$$F_{yr} = C_{\alpha r} \cdot \alpha_r \quad (\text{A.3})$$

Por su parte [60] define los valores de α de la siguiente manera:

$$\alpha_f = \delta - \beta - \frac{l_f \cdot r}{V_x} \quad (\text{A.4})$$

$$\alpha_r = -\beta + \frac{l_r \cdot r}{V_x} \quad (\text{A.5})$$

Sustituyendo las ecuaciones (A.4) y (A.5) en (A.2) y (A.3) respectivamente y estas a su vez en (A.1) se obtiene (A.6) que es el viraje del vehículo(r).

$$\frac{\partial r}{\partial t} = \frac{l_f \cdot C_{\alpha f}}{I_z} \left(\delta - \beta - \frac{l_f \cdot r}{V_x} \right) - \frac{l_r \cdot C_{\alpha r}}{I_z} \left(-\beta + \frac{l_r \cdot r}{V_x} \right) \quad (\text{A.6})$$

Tabla A.1: Valores numéricos de vehículo

Variable	Valor Numérico	Unidad
l_f	1.3	m
l_r	1.3	m
V_x	16.66	$\frac{m}{s}$
$C_{\alpha f}$	80000	$\frac{N}{rad}$
$C_{\alpha r}$	80000	$\frac{N}{rad}$
I_z	2900	$kg \cdot m^2$

Tomando los siguientes valores que se presentan en la Tabla A.1 que corresponden a un vehículo tipo *Sedan* se obtiene (A.7).

$$\frac{\partial r}{\partial t} = 35.86(\delta - \beta - 0.078 \cdot r) - 35.86(-\beta + 0.078 \cdot r) \quad (A.7)$$

En (A.8) y (A.9) se realizan despejes para obtener finalmente (A.10) donde se cuenta únicamente con 2 variables: viraje del vehículo(r) y viraje equivalente de las ruedas (δ).

$$\frac{\partial r}{\partial t} = 35.86 \cdot \delta - 35.86 \cdot \beta - 2.798 \cdot r + 35.86 \cdot \beta - 2.798 \cdot r \quad (A.8)$$

$$\frac{\partial r}{\partial t} = 35.86 \cdot \delta - 5.596 \cdot r \quad (A.9)$$

$$\frac{\partial r}{\partial t} + 5.596 \cdot r = 35.86 \cdot \delta \quad (A.10)$$

Para obtener la función de transferencia de la planta se realiza la Transformada de *Laplace* a (A.10) y se obtiene (A.11).

$$R(s) + 5.596 \cdot s \cdot R(s) = 35.86 \cdot s \cdot \Delta(s) \quad (A.11)$$

$$R(s) (1 + 5.596 \cdot s) = 35.86 \cdot s \cdot \Delta(s) \quad (A.12)$$

$$\frac{R(s)}{\Delta(s)} = \frac{35.86}{5.596} \cdot \frac{s}{(s + 0.1789)} \quad (A.13)$$

Finalmente, luego del despeje presentado en (A.12) y (A.13) se obtiene la Función de Transferencia de la planta (A.14) requerida para el sistema de control.

$$\frac{R(s)}{\Delta(s)} = 6.45 \cdot \frac{s}{s + 0.1789} \quad (A.14)$$

Tabla A.2: Nomenclatura

Variable		Unidad
l_f	distancia centro de gravedad a eje frontal	m
l_r	distancia centro de gravedad a eje trasero	m
V_x	velocidad del vehículo	$\frac{m}{s}$
α_f	ángulo de deslizamiento ruedas frontales	rad
α_r	ángulo de deslizamiento ruedas traseras	rad
F_{yf}	fuerza lateral en las ruedas frontales	N
F_{yr}	fuerza lateral en las ruedas traseras	N
$C_{\alpha f}$	rigidez en las curvas de neumáticos frontales	$\frac{N}{rad}$
$C_{\alpha r}$	rigidez en las curvas del neumático traseros	$\frac{N}{rad}$
I_z	momento de inercia del vehículo	$kg \cdot m^2$
δ	ángulo de giro equivalente de las ruedas	rad
δ_i	ángulo de giro equivalente de las rueda interna a la curva	rad
δ_o	ángulo de giro equivalente de las ruedas externa a la curva	rad
r	guiñada del vehículo	rad
β	ángulo de deslizamiento en el vehículo (centro de gravedad)	rad

Apéndice B

Implementación en Matlab de Control de Dirección

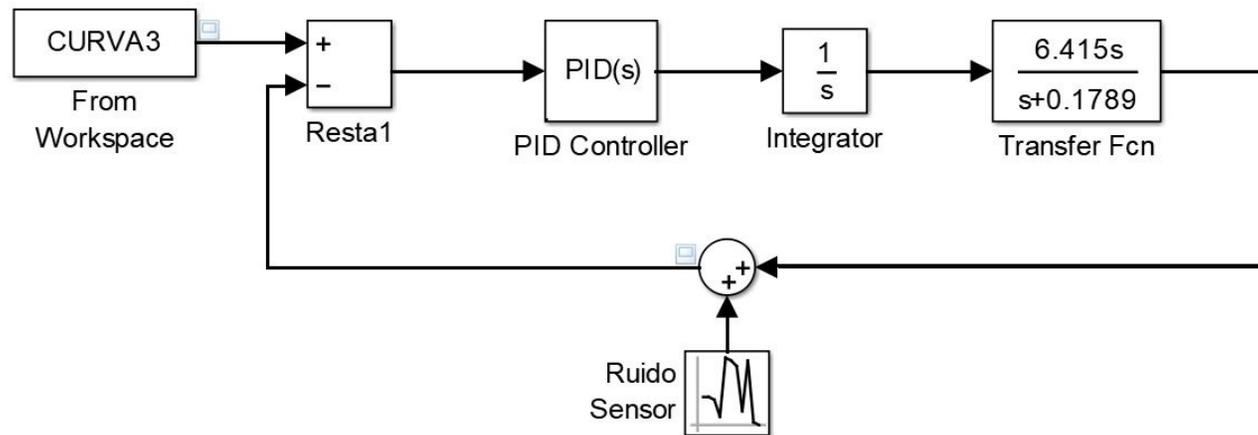


Figura B.1: Modelado en Simulink de Control de Velocidad.

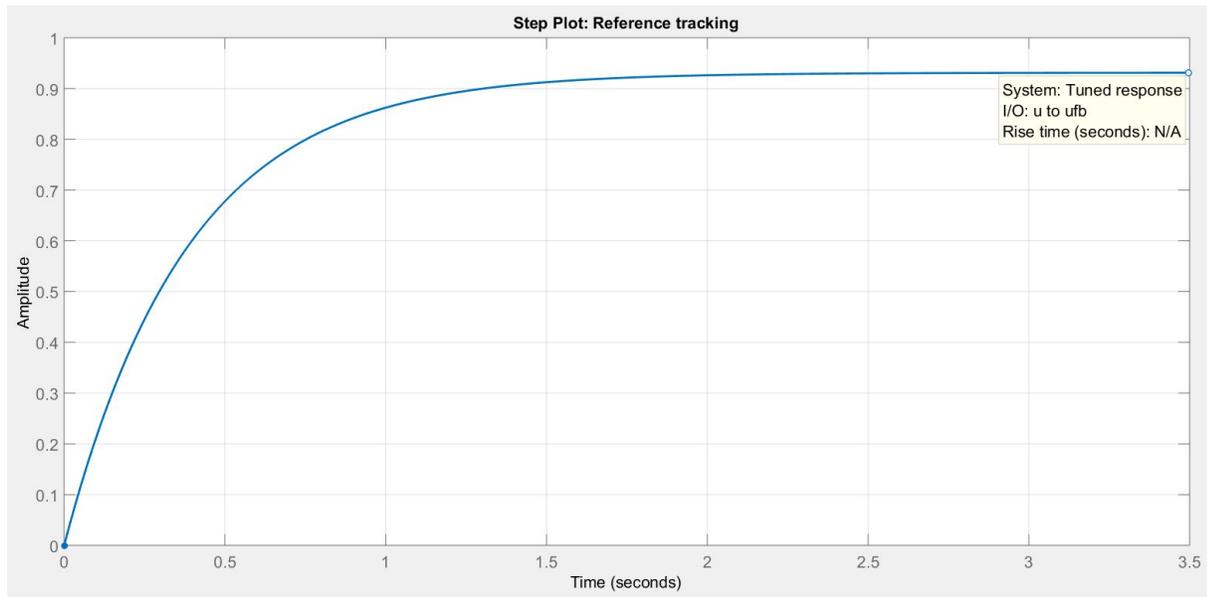


Figura B.2: Respuesta controlador P en dirección.

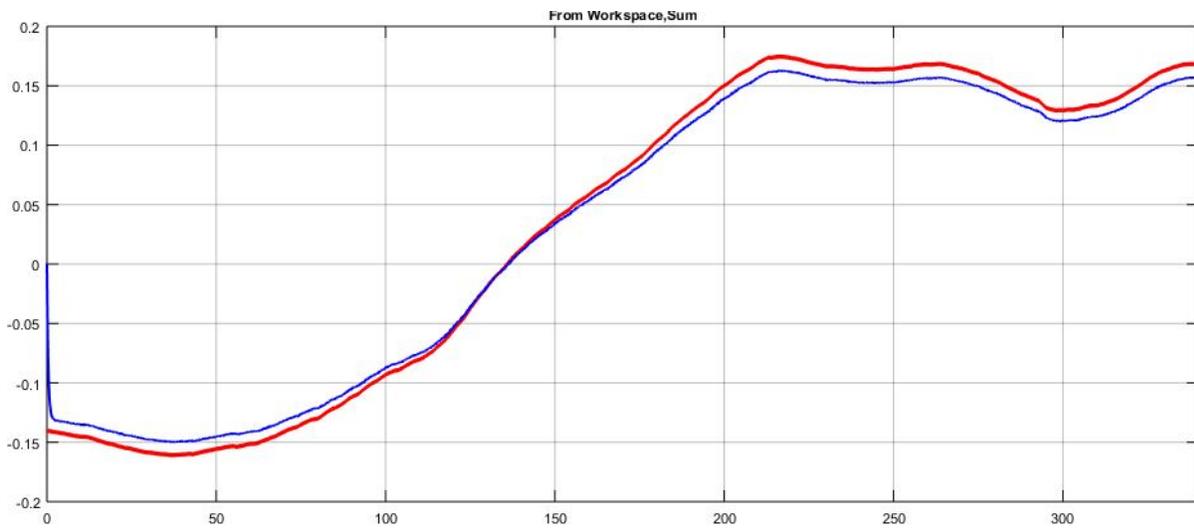


Figura B.3: Curva controlador P en dirección.

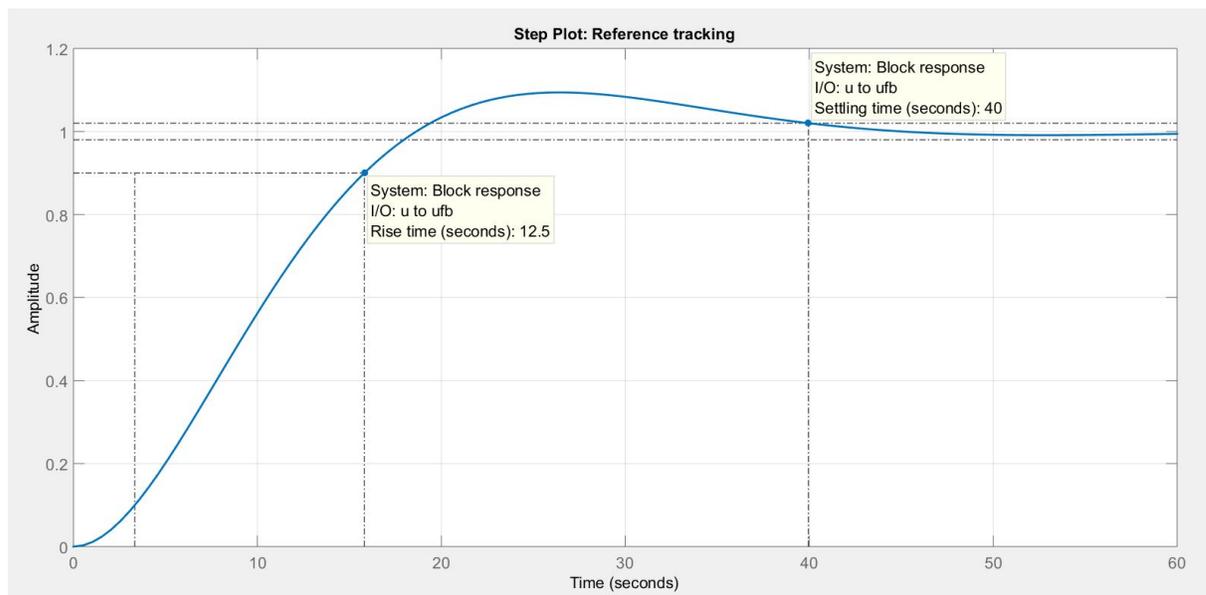


Figura B.4: Respuesta controlador I en dirección.

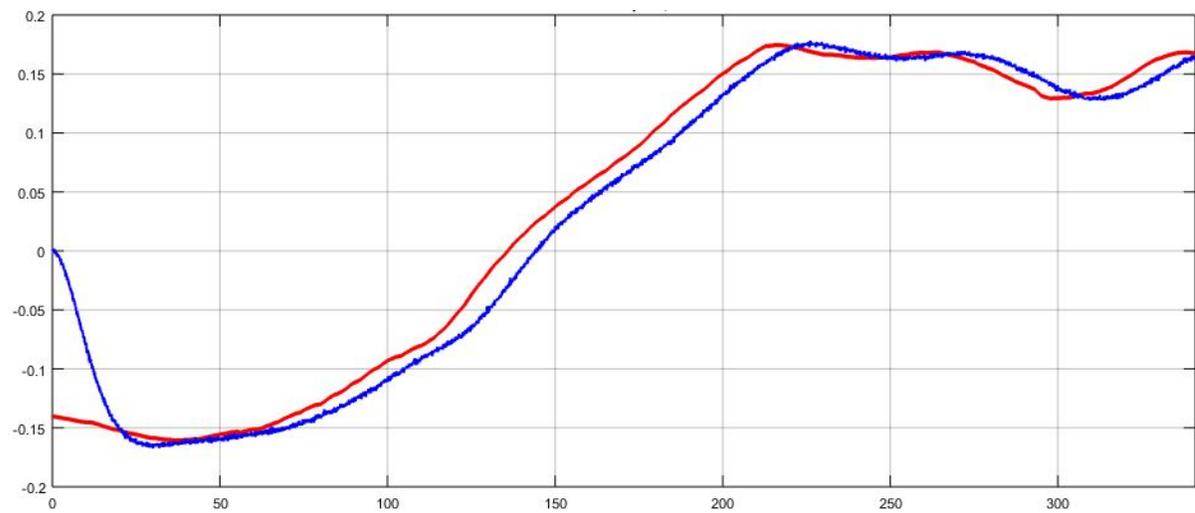


Figura B.5: Curva controlador I en dirección.

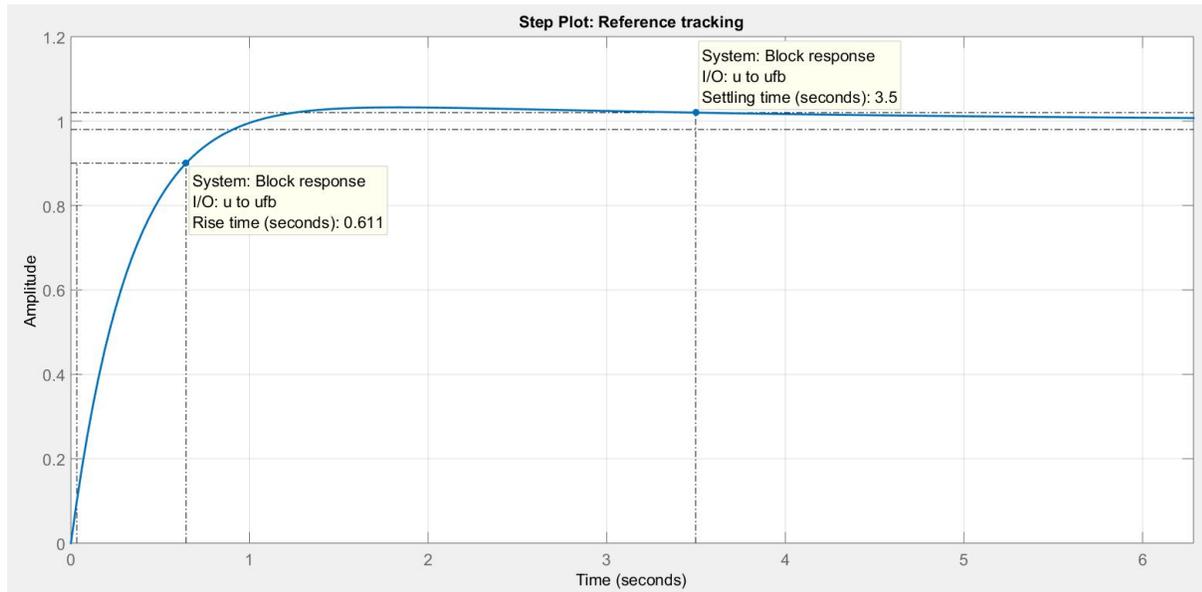


Figura B.6: Respuesta controlador PI en dirección.

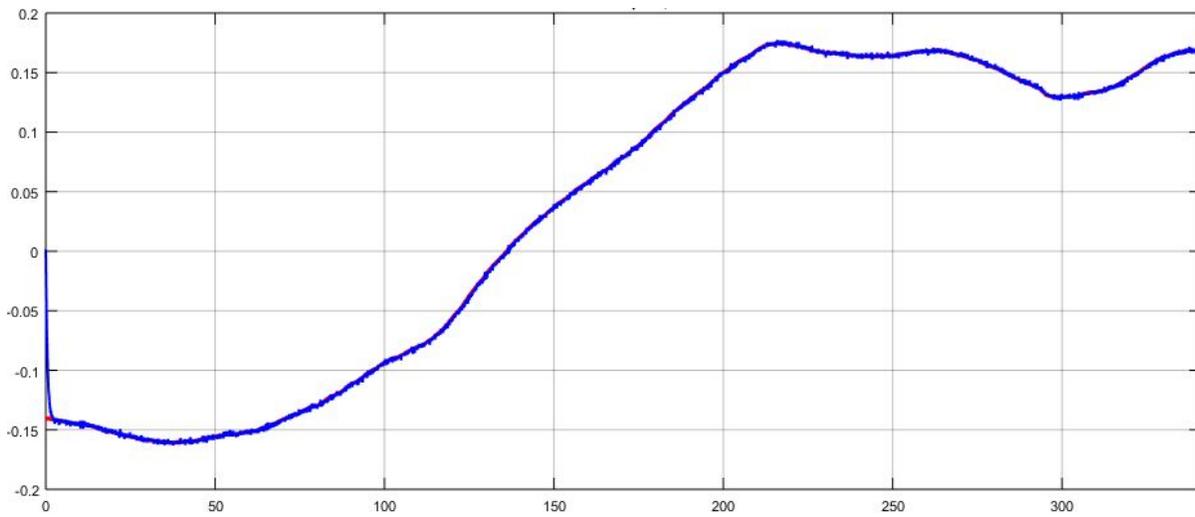


Figura B.7: Curva controlador PI en dirección.

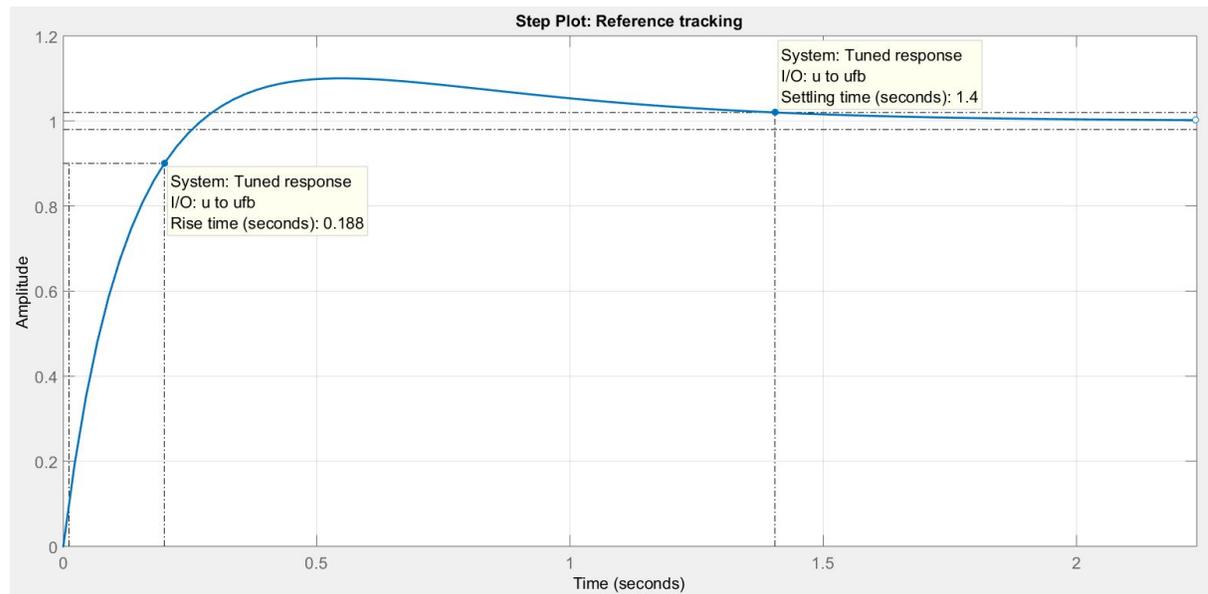


Figura B.8: Respuesta controlador PID en dirección.

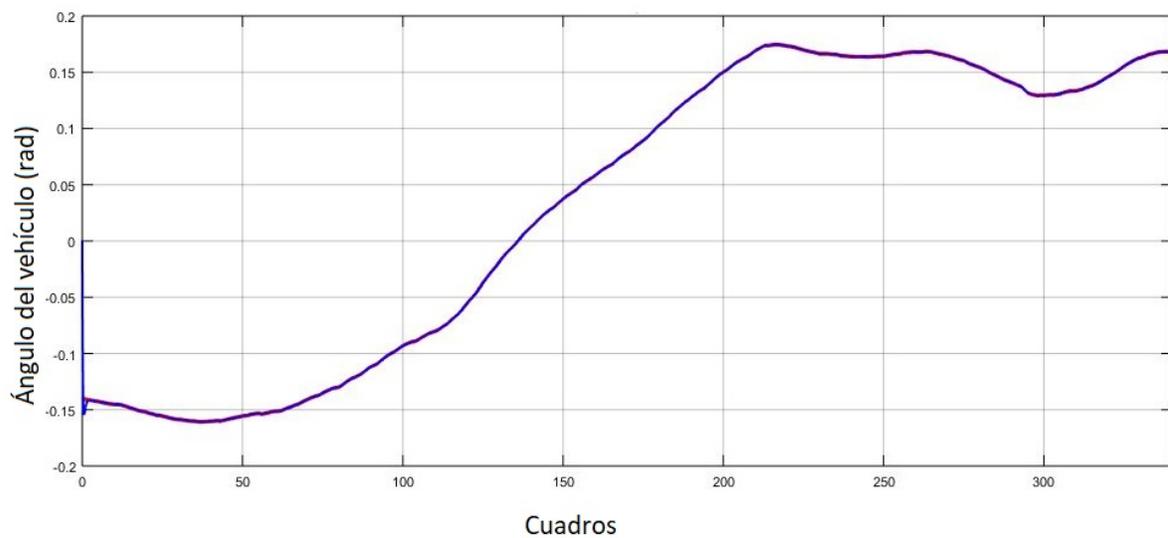


Figura B.9: Curva controlador PID en dirección.

Apéndice C

Conjuntos de datos para entrenamiento de SVM



Figura C.1: Muestras de conjuntos de datos entrenamiento de señal de 30km/h.



Figura C.2: Muestras de conjuntos de datos entrenamiento de señal de 50km/h.



Figura C.3: Muestras de conjuntos de datos entrenamiento de señal de 70km/h.

Apéndice D

Modelado Sistema de Velocidad

Para el diseño del sistema de control se partió de entender que variable se quería controlar, que constantes afectan el sistema y se encontró que la velocidad es lo que se desea controlar y que se ve afectada por la masa y aceleración, así como por la fricción que la superficie ejerce sobre las ruedas del vehículo. La segunda Ley de Newton permite relacionar todo eso como se muestra en la ecuación (D.1).

$$F = m \cdot a \quad (D.1)$$

Se analiza el diagrama de cuerpo libre presenta en la Figura D.1 donde la m representa la masa del vehículo, la b el coeficiente de fricción dinámica del suelo, u la Fuerza hacia delante del vehículo y v la velocidad.

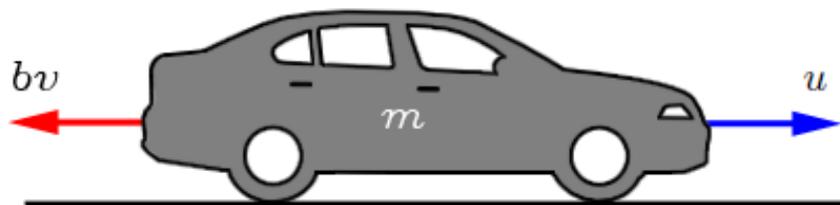


Figura D.1: Diagrama de Cuerpo libre vehículo en movimiento [61].

En base a lo anterior se realiza la sumatoria de fuerzas, ecuación (D.2) y su correspondiente despeje en la ecuación (D.3).

$$u - b \cdot v = m \cdot a \quad (D.2)$$

$$u = m \cdot \frac{\partial v}{\partial t} + b \cdot v \quad (D.3)$$

Realizando transformada de *Laplace* a (D.3) se obtiene (D.4).

$$U(s) = m \cdot s \cdot V(s) + b \cdot V(s) \quad (\text{D.4})$$

En (D.5) se muestra un paso intermedio antes de obtener el despeje deseado.

$$U(s) = V(s) \cdot (m \cdot s + b) \quad (\text{D.5})$$

Para finalmente obtener la función de transferencia del sistema (D.6).

$$\frac{V(s)}{U(s)} = \frac{1}{m \cdot s + b} \quad (\text{D.6})$$

Tabla D.1: Nomenclatura

Variable		Unidad
b	coeficiente de fricción ruedas-carretera	N/A
v	velocidad del vehículo	$\frac{m}{s}$
m	masa del vehículo	kg
u	fuerza de empuje del vehículo	N
a	aceleración del vehículo	$\frac{m}{s^2}$

Apéndice E

Implementación en Matlab de Control de Velocidad

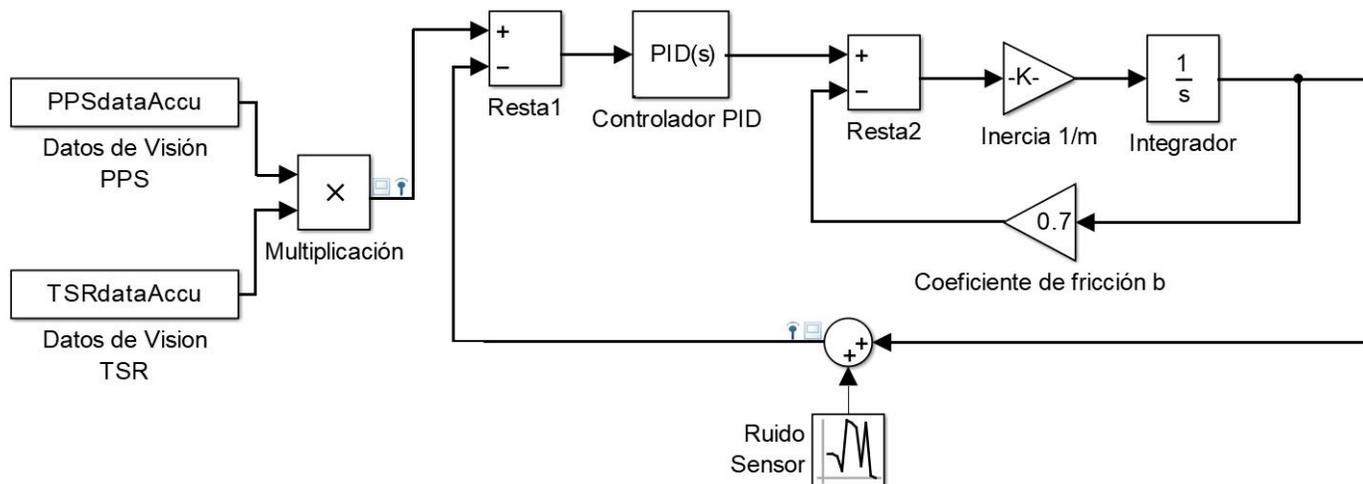


Figura E.1: Modelado en Simulink de Control de Velocidad.

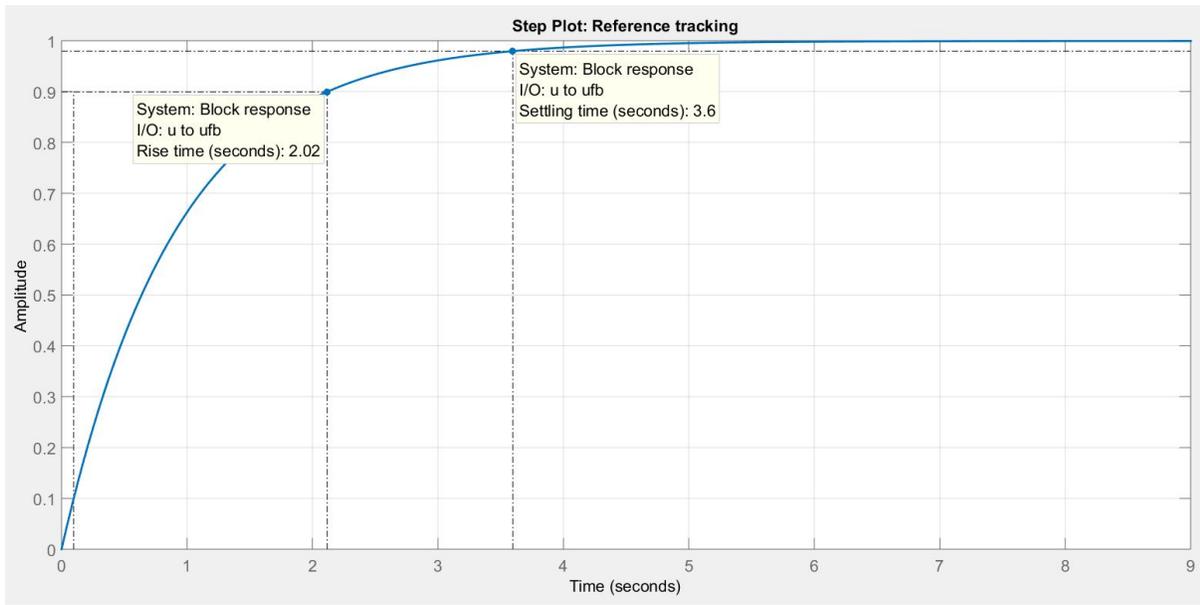


Figura E.2: Respuesta controlador P en velocidad.

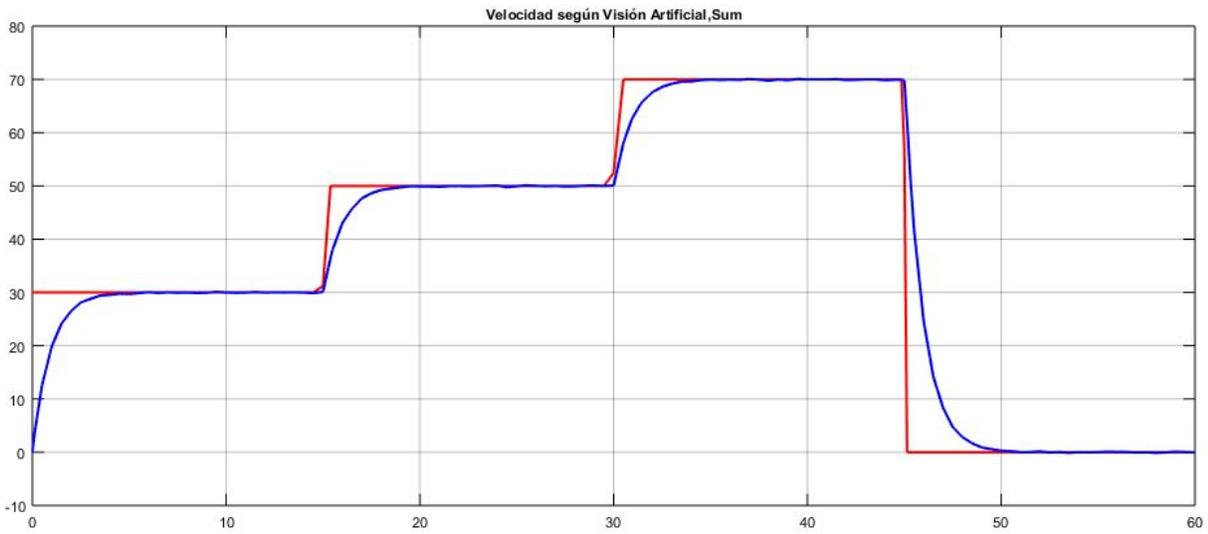


Figura E.3: Curva controlador P en velocidad.

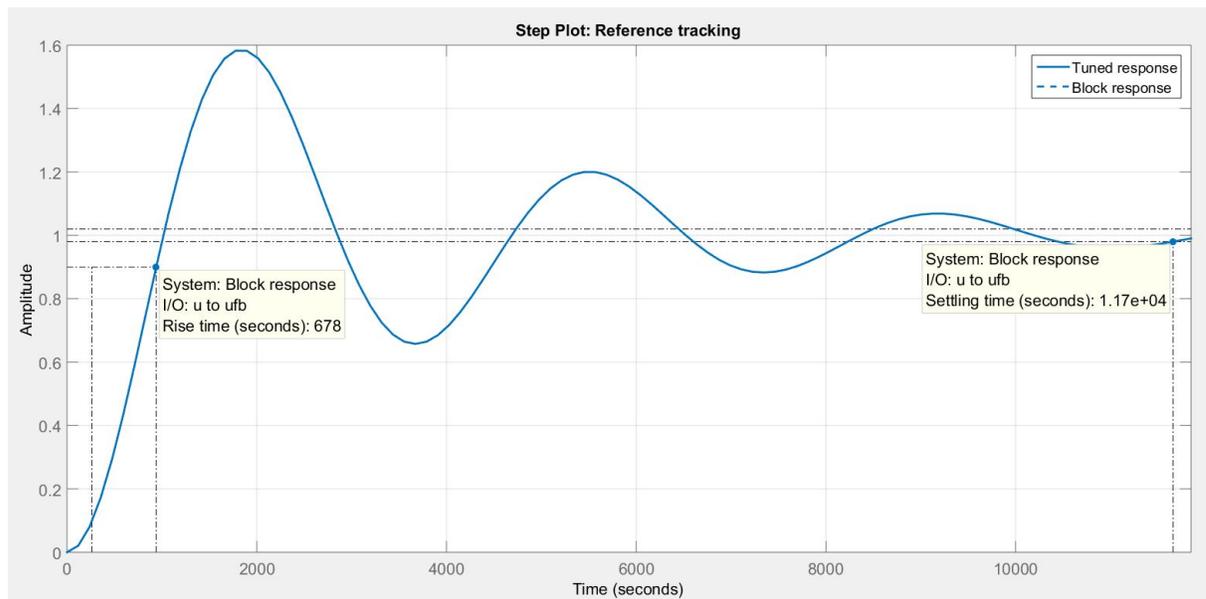


Figura E.4: Respuesta controlador I en velocidad.

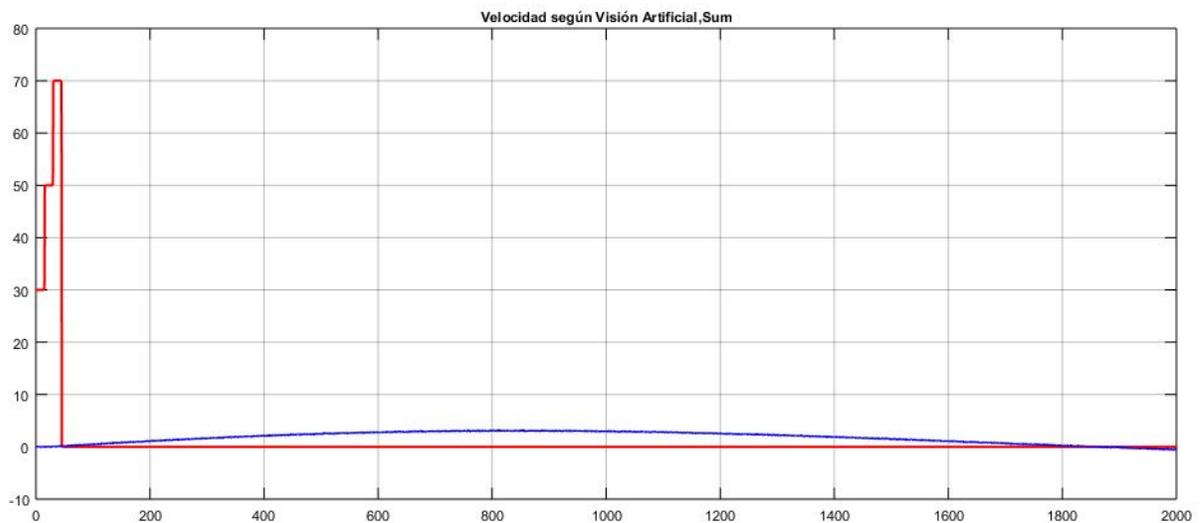


Figura E.5: Curva controlador I en velocidad.

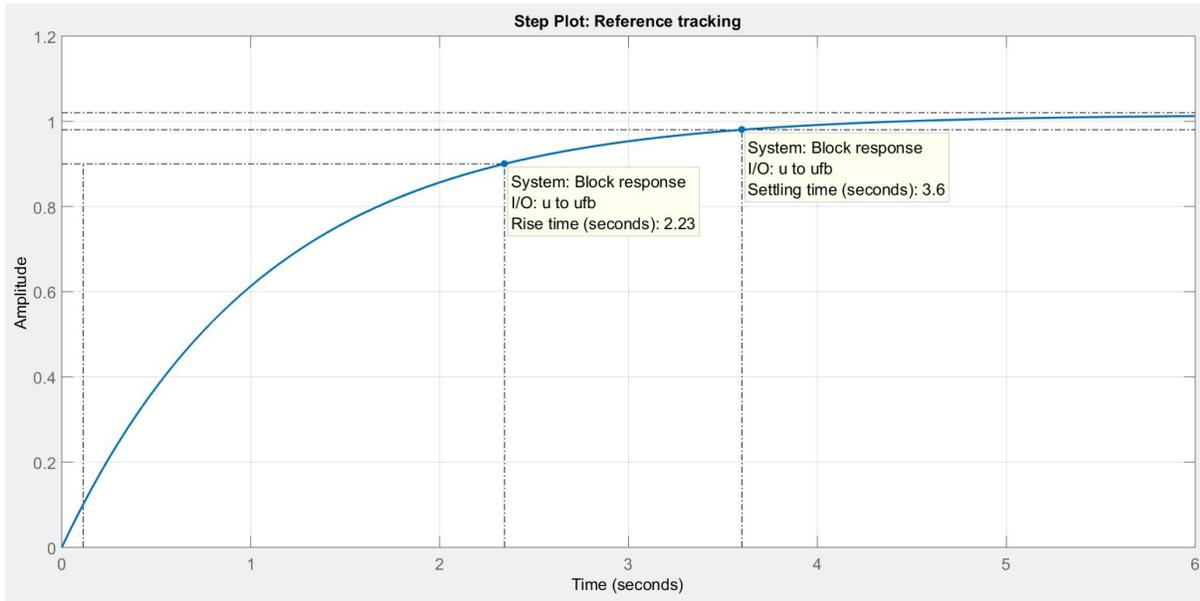


Figura E.6: Respuesta controlador PI en velocidad.

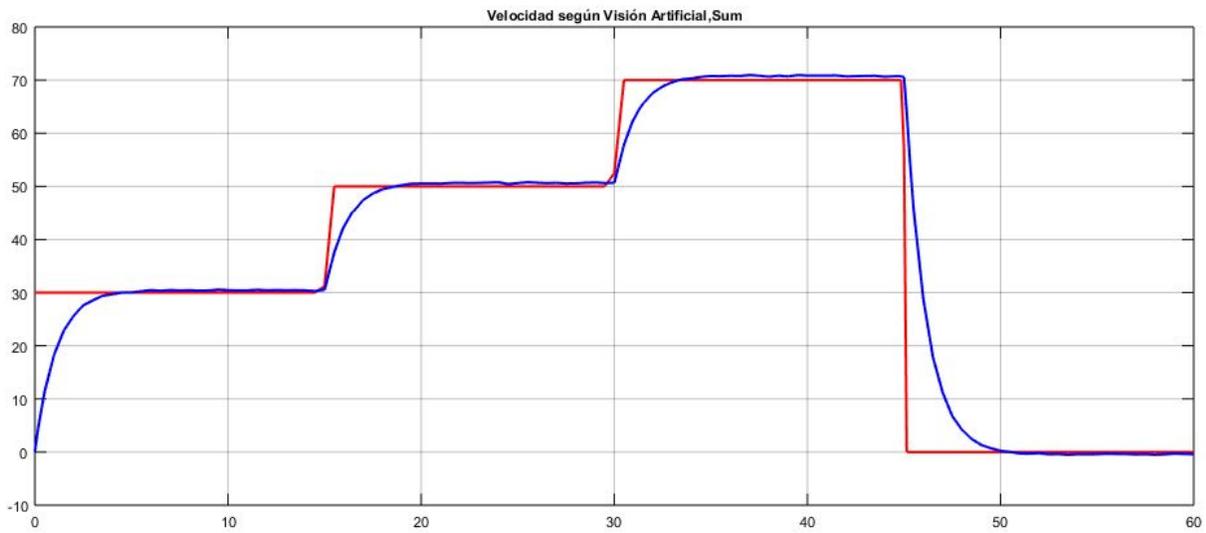


Figura E.7: Curva controlador PI en velocidad.

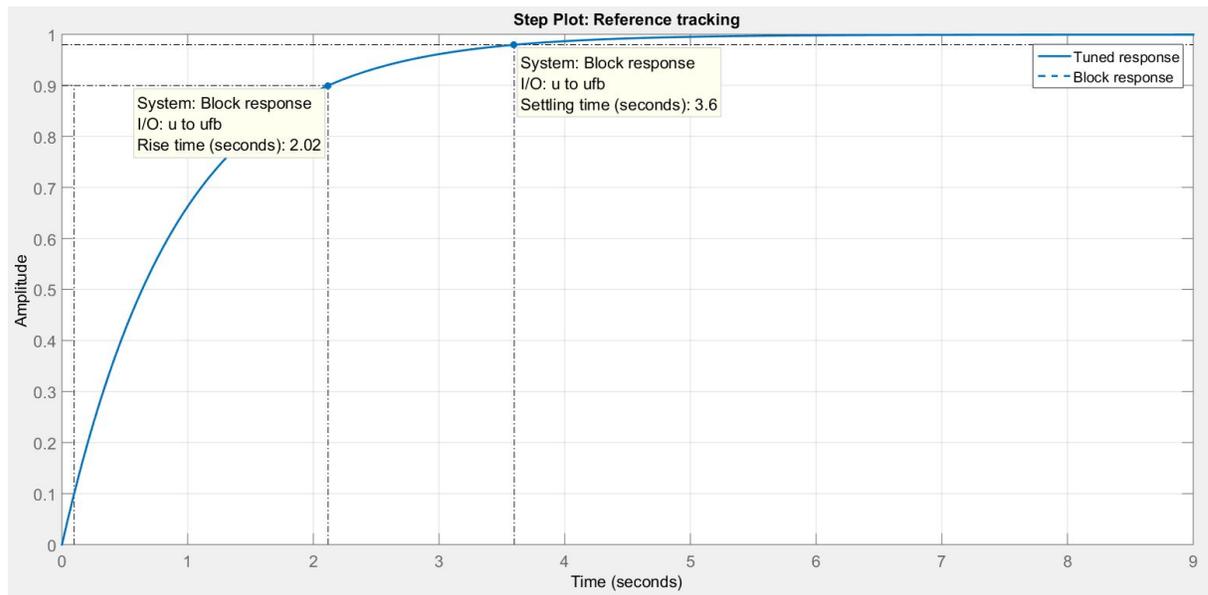


Figura E.8: Respuesta controlador PD en velocidad.

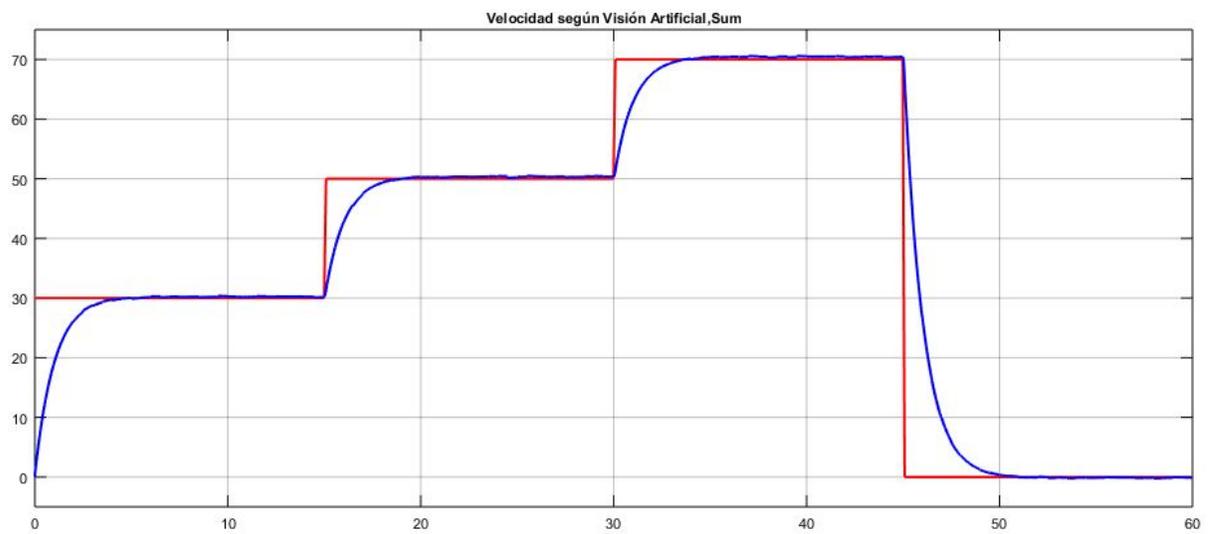


Figura E.9: Curva controlador PD en velocidad.

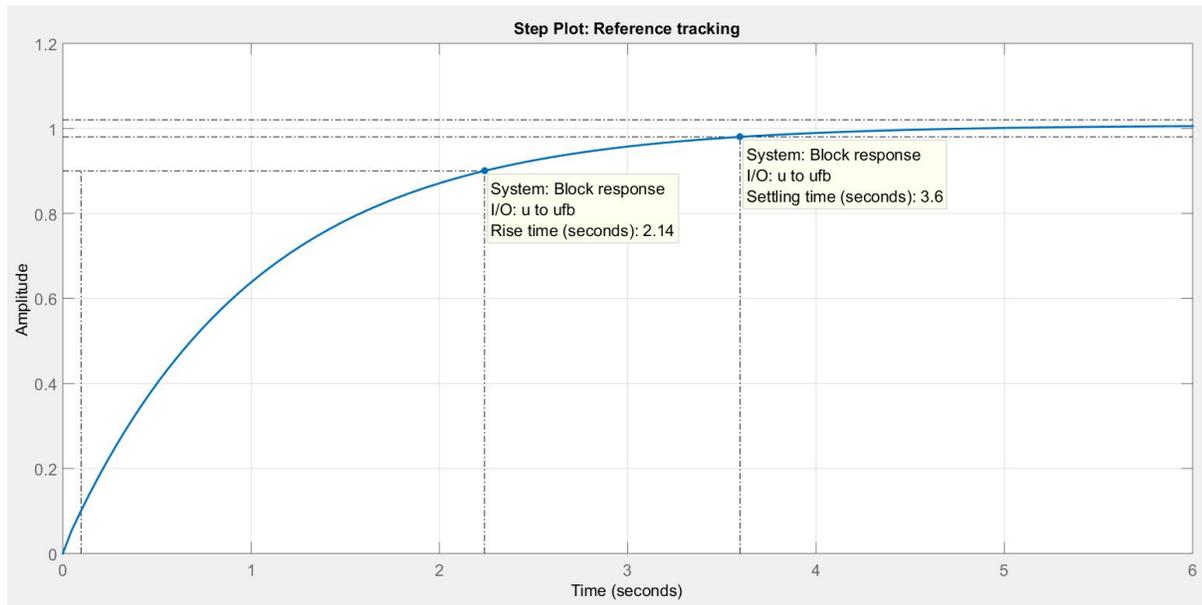


Figura E.10: Respuesta controlador PID en dirección.

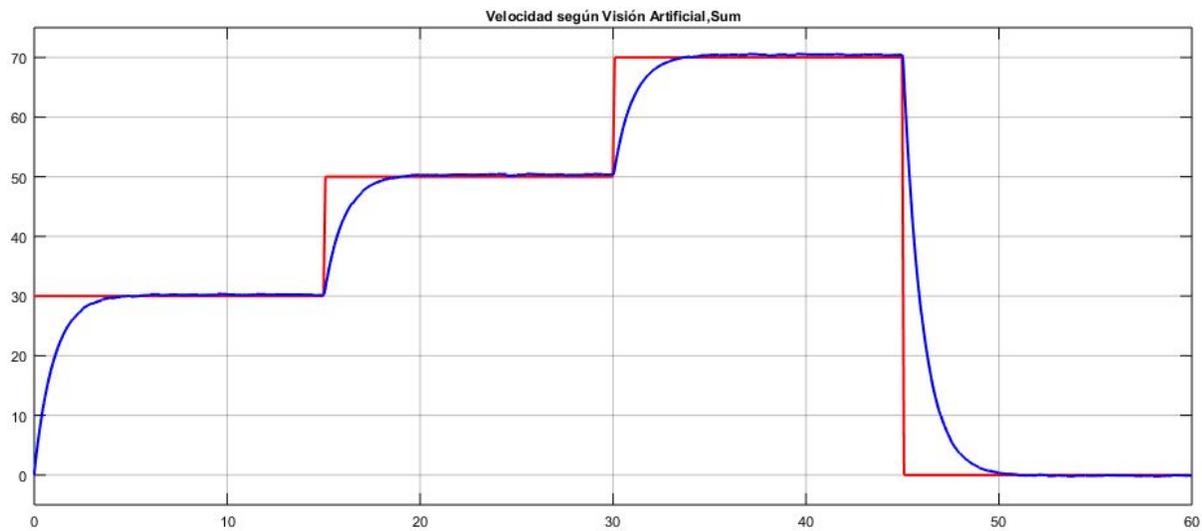


Figura E.11: Curva controlador PID en dirección.

Anexos I

OpenCV

I.1 IPM [62]

The method consists of mapping images to a new coordinate system where perspective effects are removed. The removal of perspective associated effects facilitates road and obstacle detection and also assists in free space estimation. There is, however, a significant limitation in the inverse perspective mapping: the presence of obstacles on the road disrupts the effectiveness of the mapping. The current paper proposes a robust solution based on the use of multimodal sensor fusion. Data from a laser range finder is fused with images from the cameras, so that the mapping is not computed in the regions where obstacles are present. As shown in the results, this considerably improves the effectiveness of the algorithm and reduces computation time when compared with the classical inverse perspective mapping. Furthermore, the proposed approach is also able to cope with several cameras with different lenses or image resolutions, as well as dynamic viewpoints.

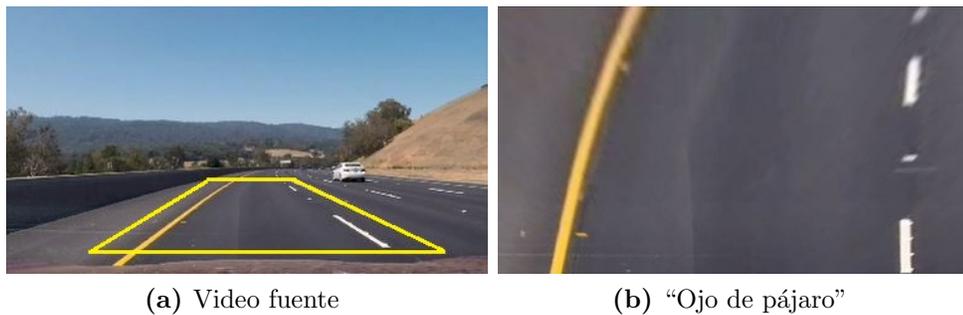


Figura I.1: Transformación imagen a vista Ojo de pájaro

I.2 Funciones básicas

I.2.1 Conversión Color [63]

C++: `void cvtColor(InputArray src, OutputArray dst, int code);`

Parameters:

- `src` – input image
- `dst` – output image of the same size and type as `src`.
- `code` – color space conversion code (`cv::COLOR_RGB2GRAY`, `cv::COLOR_GRAY2RGB`)

I.2.2 Filtrado [64]

C++: `void blur(InputArray src, OutputArray dst, Size ksize, Point anchor=Point(-1,-1), int borderType=BORDER_DEFAULT)`

Parameters:

- `src` – input image
- `dst` – output image of the same size and type as `src`.
- `ksize` – blurring kernel size.
- `anchor` – anchor point; default value `Point(-1,-1)` means that the anchor is at the kernel center.
- `borderType` – border mode used to extrapolate pixels outside of the image.

I.2.3 Detección de bordes [65]

C++: `void Canny(InputArray src, OutputArray edges, double threshold1, double threshold2, int apertureSize=3, bool L2gradient=false)`

- `src` – single-channel 8-bit input image.
- `edges` – output edge map; it has the same size and type as image.
- `threshold1` – first threshold for the hysteresis procedure.
- `threshold2` – second threshold for the hysteresis procedure.
- `apertureSize` – aperture size for the Sobel() operator.

- `L2gradient` – a flag, indicating whether a more accurate L₂ norm $=\sqrt{(dI/dx)^2 + (dI/dy)^2}$ should be used to calculate the image gradient magnitude (`L2gradient=true`), or whether the default L₁ norm $=|dI/dx| + |dI/dy|$ is enough (`L2gradient=false`).

I.3 HAAR Cascades [66]

The object detector described below has been initially proposed by Paul Viola [Viola01] and improved by Rainer Lienhart [Lienhart02].

First, a classifier (namely a cascade of boosted classifiers working with haar-like features) is trained with a few hundred sample views of a particular object (i.e., a face or a car), called positive examples, that are scaled to the same size (say, 20x20), and negative examples - arbitrary images of the same size.

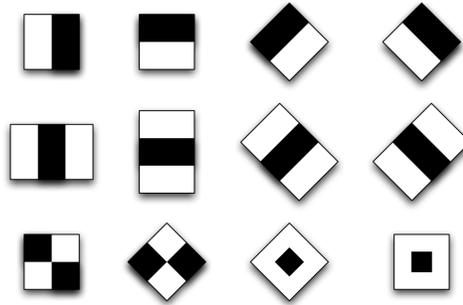


Figura I.2: Bloques clasificadores básicos [66]

After a classifier is trained, it can be applied to a region of interest (of the same size as used during the training) in an input image. The classifier outputs a “1” if the region is likely to show the object (i.e., face/car), and “0” otherwise. To search for the object in the whole image one can move the search window across the image and check every location using the classifier. The classifier is designed so that it can be easily “resized” in order to be able to find the objects of interest at different sizes, which is more efficient than resizing the image itself. So, to find an object of an unknown size in the image the scan procedure should be done several times at different scales.

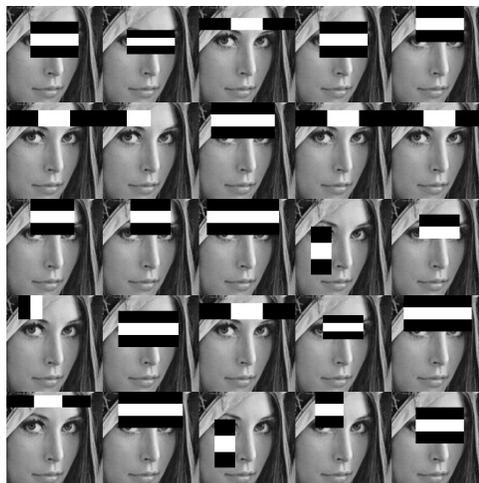


Figura I.3: Clasificadores básicos en imagen [66]

The word “cascade” in the classifier name means that the resultant classifier consists of several simpler classifiers (stages) that are applied subsequently to a region of interest until at some stage the candidate is rejected or all the stages are passed. The word “boosted” means that the classifiers at every stage of the cascade are complex themselves and they are built out of basic classifiers using one of four different boosting techniques (weighted voting). Currently Discrete Adaboost, Real Adaboost, Gentle Adaboost and Logitboost are supported. The basic classifiers are decision-tree classifiers with at least 2 leaves. Haar-like features are the input to the basic classifiers, and are calculated as described below. The current algorithm uses the following Haar-like features:

```
C++: void CascadeClassifier::detectMultiScale(const Mat& image, vector< Rect >&
objects, double scaleFactor=1.1, int minNeighbors=3, int flags=0, Size minSize=Size(),
Size maxSize=Size())
```

- cascade – Haar classifier cascade.
- image – Matrix of the type CV_8U containing an image where objects are detected.
- objects – Vector of rectangles where each rectangle contains the detected object.
- scaleFactor – Parameter specifying how much the image size is reduced at each image scale.
- minNeighbors – Parameter specifying how many neighbors each candidate rectangle should have to retain it
- flags – Parameter with the same meaning for an old cascade as in the function cvHaarDetectObjects. It is not used for a new cascade
- minSize – Minimum possible object size. Objects smaller than that are ignored.
- maxSize – Maximum possible object size. Objects larger than that are ignored.

Anexos II

Machine Learning

II.1 Definition

The Machine Learning Library (MLL) is a set of classes and functions for statistical classification, regression, and clustering of data. Most of the classification and regression algorithms are implemented as C++ classes. As the algorithms have different sets of features (like an ability to handle missing measurements or categorical input variables), there is a little common ground between the classes. This common ground is defined by the class CvStatModel that all the other ML classes are derived from [67].

II.2 Decision Trees [68]

A decision tree is a binary tree (tree where each non-leaf node has two child nodes). It can be used either for classification or for regression. For classification, each tree leaf is marked with a class label; multiple leaves may have the same label. For regression, a constant is also assigned to each tree leaf, so the approximation function is piecewise constant.

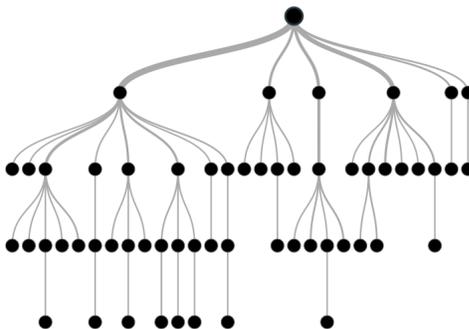


Figura II.1: Visualización de Desicion Tree [68]

II.3 Neural Networks [69]

ML implements feed-forward artificial neural networks or, more particularly, multi-layer perceptrons (MLP), the most commonly used type of neural networks. MLP consists of the input layer, output layer, and one or more hidden layers. Each layer of MLP includes one or more neurons directionally linked with the neurons from the previous and the next layer. The example below represents a 3-layer perceptron with three inputs, two outputs, and the hidden layer including five neurons

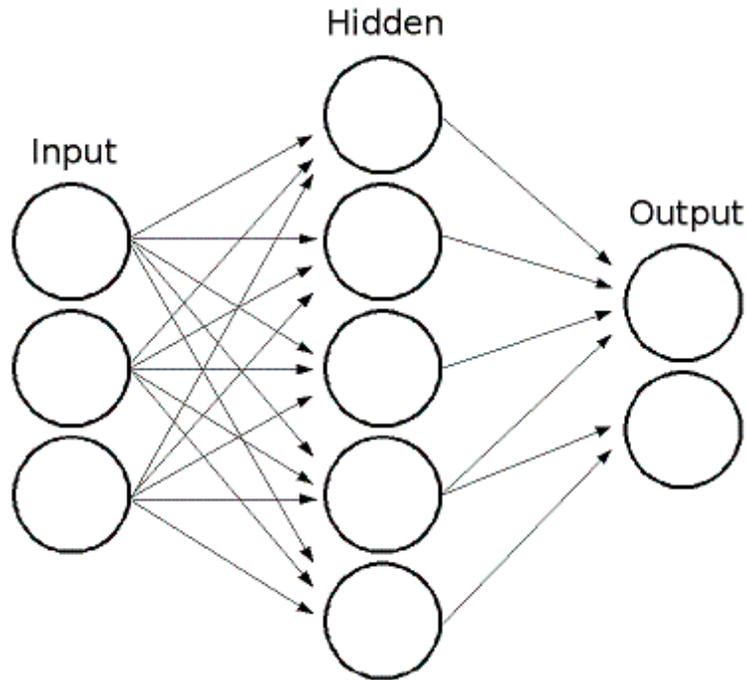


Figura II.2: Muestra de Red Neuronal [69]

II.4 Support Vector Machine [70]

Originally, support vector machines (SVM) was a technique for building an optimal binary (2-class) classifier. Later the technique was extended to regression and clustering problems. SVM is a partial case of kernel-based methods. It maps feature vectors into a higher-dimensional space using a kernel function and builds an optimal linear discriminating function in this space or an optimal hyper-plane that fits into the training data. In case of SVM, the kernel is not defined explicitly. Instead, a distance between any 2 points in the hyper-space needs to be defined.

The solution is optimal, which means that the margin between the separating hyper-plane and the nearest feature vectors from both classes (in case of 2-class classifier) is maximal. The feature vectors that are the closest to the hyper-plane are called support vectors, which means that the position of other vectors does not affect the hyper-plane (the decision function).

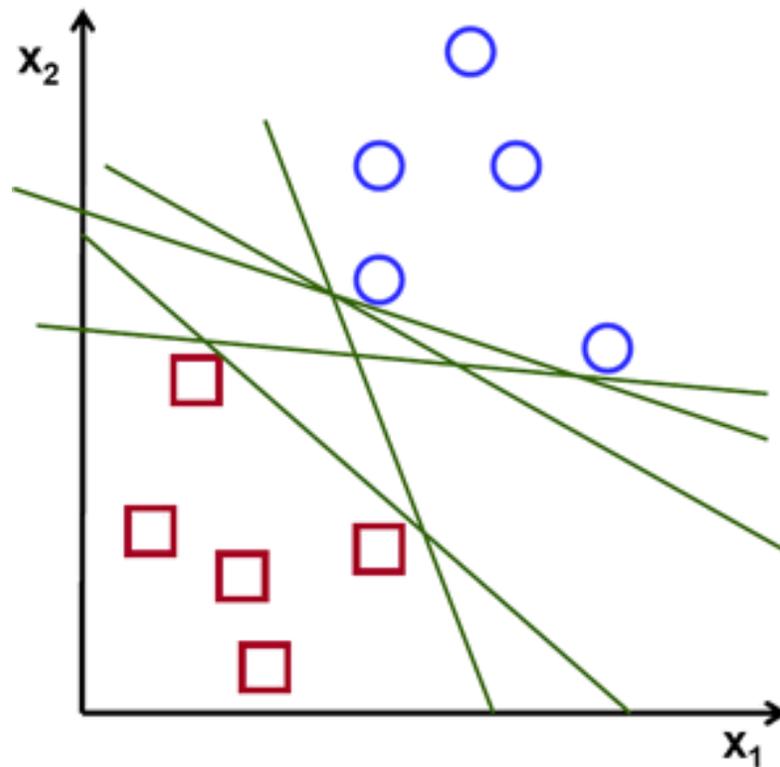


Figura II.3: Vectores de soporte [70]

C++: SVM::Params::Params(int svmType, int kernelType, double degree, double gamma, double coef0, double Cvalue, double nu, double p, const Mat& classWeights, TermCriteria termCrit)

- svmType – Type of a SVM formulation. Possible values are:
 - SVM::C_SVC C-Support Vector Classification. n-class classification ($n \geq 2$), allows imperfect separation of classes with penalty multiplier C for outliers.
 - SVM::NU_SVC ν -Support Vector Classification. n-class classification with possible imperfect separation. Parameter ν (in the range 0...1, the larger the value, the smoother the decision boundary) is used instead of C.
 - SVM::ONE_CLASS Distribution Estimation (One-class SVM). All the training data are from the same class, SVM builds a boundary that separates the class from the rest of the feature space.
 - SVM::EPS_SVR ϵ -Support Vector Regression. The distance between feature vectors from the training set and the fitting hyper-plane must be less than p. For outliers the penalty multiplier C is used.
 - SVM::NU_SVR ν -Support Vector Regression. ν is used instead of p.
- kernelType – Type of a SVM kernel. Possible values are:
 - SVM::LINEAR Linear kernel. No mapping is done, linear discrimination (or regression) is done in the original feature space. It is the fastest option.
 - SVM::POLY Polynomial kernel
 - SVM::RBF Radial basis function (RBF), a good choice in most cases.
 - SVM::SIGMOID Sigmoid kernel.
 - SVM::CHI2 Exponential Chi2 kernel, similar to the RBF kernel.
 - SVM::INTER Histogram intersection kernel. A fast kernel.
- degree – Parameter degree of a kernel function (POLY).
- gamma – Parameter γ of a kernel function (POLY / RBF / SIGMOID / CHI2).
- coef0 – Parameter coef0 of a kernel function (POLY / SIGMOID).
- Cvalue – Parameter C of a SVM optimization problem (C_SVC / EPS_SVR / NU_SVR).
- nu – Parameter ν of a SVM optimization problem (NU_SVC / ONE_CLASS / NU_SVR).
- p – Parameter ϵ of a SVM optimization problem (EPS_SVR).

- `classWeights` – Optional weights in the C_SVC problem , assigned to particular classes. They are multiplied by C so the parameter C of class #i becomes `classWeights(i) C`. Thus these weights affect the misclassification penalty for different classes. The larger weight, the larger penalty on misclassification of data from the corresponding class.
- `termCrit` – Termination criteria of the iterative SVM training procedure which solves a partial case of constrained quadratic optimization problem. You can specify tolerance and/or the maximum number of iterations.

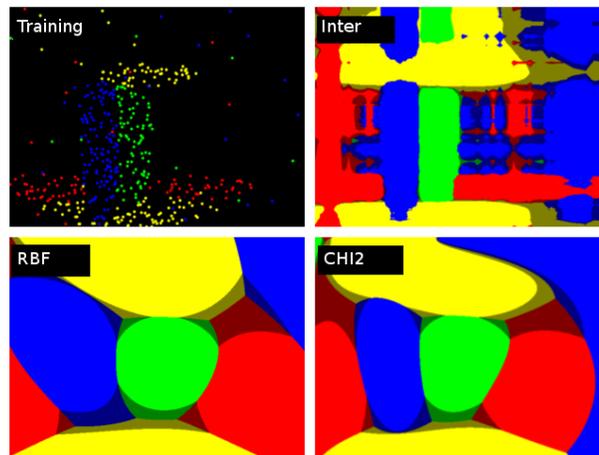


Figura II.4: Comparación entre distintos tipo de kernel [70]

I trained it:

- `mySVM.train(trainingDataMat, labelsMat, Mat(), Mat(), params);`

Saved it:

- `mySVM.save("classifier.xml")`

I'm loading it like this:

- `CvSVM mySVM1`
- `mySVM1.load("C: classifier.xml")`
- `mySVM1.predict(testingDataMat0,result0)`

Anexos III

Seguro Computadora



Informe de Cotización

Equipo Electrónico Riesgo Nombrado

Numero de cotización: I-EQERN1546ZCYAAWC
Nombre del Intermediario: INS
Fecha de cotización: 23/01/2018
Moneda: Colones
Tipo de tarifa: No Cooperativas

Cobertura	Monto asegurado	Prima
E. Equipo móvil y/o portátil	¢ 850,000.00	¢ 25,487.00
Prima anual (I.V.I)		¢ 28,800.00
Prima semestral (I.V.I)		¢ 15,552.00

GRACIAS POR COTIZAR NUESTROS SEGUROS

Este documento únicamente constituye una cotización de seguro, la cual puede presentar alguna modificación de acuerdo con el análisis del riesgo que se realice. Cuenta con una vigencia de 30 días naturales contados a partir de la fecha en que se genera la cotización.

La documentación contractual y la nota técnica que integran este producto, están registrados ante la Superintendencia General de Seguros de conformidad con lo dispuesto por el artículo 29, inciso d) de la Ley Reguladora del Mercado de Seguros Ley 8653, bajo el registro: G07-45-A01-086 de fecha 12 de enero del 2010

Instituto Nacional de Seguros Sede INS San José
Tel: 2287-6000 Fax: 2243-9900 www.ins-cr.com

Los seguros del INS tienen la garantía del Estado

Figura III.1: Cotización INS

