



UNIVERSIDAD DE LA FRONTERA
FACULTAD DE INGENIERÍA Y CIENCIAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

“Diseño de plataforma de captura de datos en apoyo del control
de tráfico vehicular”

LUIS SEBASTIÁN LÓPEZ LLANTÉN

2018



**UNIVERSIDAD DE LA FRONTERA
FACULTAD DE INGENIERÍA Y CIENCIAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA**

**“Diseño de plataforma de captura de datos en apoyo del control
de tráfico vehicular”**

**TRABAJO PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL TELEMÁTICO**

Profesor Guía: Mg. Alejandro Navarro Cifuentes

LUIS SEBASTIÁN LÓPEZ LLANTÉN

2018

DISEÑO DE PLATAFORMA DE CAPTURA DE DATOS EN APOYO DEL CONTROL DE TRÁFICO VEHICULAR

Luis Sebastián López Llantén

COMISIÓN EXAMINADORA

Mg. Alejandro Navarro Cifuentes
Profesor Guía

Dr. Fernando Huenupan Quinan
Académico Evaluador

Dr. Ricardo Gacitua Bustos
Académico Evaluador

Nota trabajo escrito :

Nota examen :

Nota final :

...Servus Mariae nunquam peribit

Agradecimientos

A todos los que han hecho de mi quien soy. A mi madre y Dookito, a mis querídisimos padrinos Galicia y Marcelo, a mis suegros, a mis amigos, a mis profesores de disciplinas y de la vida... en especial a todos aquellos que ya no están. Sepan que fueron un aliciente constante a continuar un camino que en muchas ocasiones pareció no tener objetivo. Guardo en mi corazón todos esos momentos en los que, en tiempos de tormenta, estuvieron ahí.

Finalmente a mi amada esposa. Con seguridad nada de esto hubiera ocurrido sin tu constante apoyo. Gracias por estar siempre a mi lado, en mis locuras, en las buenas y especialmente en las malas. Gracias, infinitas gracias.

llanten@buhó:~\$

Agosto, 2018.
Temuco, Chile.

Resumen

Este trabajo se desarrolla en el área de las tecnologías smartcity como apoyo específicamente a la etapa de captura de datos. Para esto se plantea diseñar una nueva plataforma para contar vehículos y registrar su velocidad. La plataforma se compone de tres partes. La primera parte consta de sensores. Estos permiten medir la distancia y la velocidad. La segunda etapa consiste en un Arduino que es una plataforma de prototipos electrónica de código abierto (open-source). Como tercera etapa se encuentra un servidor que en este trabajo se limitó a recibir los datos por el puerto serial del arduino y procesar los datos para discriminar adecuadamente si se trata de un vehículo o no. La etapa de discriminación se realiza aplicando algunos criterios usando una mezcla de los datos capturados. Este proceso de validación se realizó usando Python.

Como primer paso se realizó un estudio de las características físicas de construcción de las intersecciones. Estas establecen desde aspectos físicos de la calle y de señalética, hasta aspectos de las instalaciones eléctricas que permiten el funcionamiento de semáforos y otros equipos. Esto permitió definir posibles escenarios que establecerían los requerimientos mínimos del resto de los componentes. Posterior a esto se prosiguió con la selección de componentes a utilizar, comenzando por los sensores. Estos debían cumplir características adicionales como bajo coste, simplicidad operativa y bajo mantenimiento. En el caso del servidor no se contempló una selección de hardware específico ya que se consideró que este cumple un rol genérico.

Para este trabajo solo se implementó un arduino con un kit de sensores en la configuración de escenario más simple representada por una calzada y un semáforo de poste simple.

Teniendo los sensores elegidos, se realizó la programación del arduino mediante el IDE Arduino usando el puerto USB. Posterior a la implementación de Arduino y sus sensores, se procedió a tomar datos para poder definir los criterios de validación de los datos. Esto consistió en la analítica de los datos como comportamiento de curvas entre otros. Finalmente se programó un script en Python que implementó los algoritmos anteriormente definidos.

Como resultado se obtuvo una favorable respuesta del sensor de distancia escogido. Este presentaba un excepcional comportamiento ya que incluso al estar fuera de rango este no interfería de manera significativa a la etapa de validación. Complementariamente a este estaba un sensor doppler que permitía determinar la velocidad. Lamentablemente este sensor no logró cumplir las expectativas técnicas. El principal motivo fue su alto nivel de ruido en la señal que, a pesar de usar filtros y amplificadores no fue posible obtener un comportamiento adecuado que permitiera validar datos.

Como conclusión se puede decir que existe un gran potencial en la plataforma propuesta pero es necesario investigar más sobre los sensores doppler y sus características. La plataforma tiene una alta probabilidad de funcionar a gran escala como parte en un sistema más complejo que permita explotar la información capturada en un conjunto de intersecciones que, con seguridad, presentarán un patrón relevante para la toma de decisiones a gran escala.

Palabras Claves: semáforos; Arduino; smartcity;

Abstract

This work is developed in the area of smartcity technologies as a support specifically for the data capture stage. For this it is proposed to design a new data capture platform to count vehicles and record their speed. The platform consists of three parts. The first part consists of sensors. These allow you to measure distance and speed. The second stage consists of an Arduino that is an open-source electronic prototype platform. As a third stage is a server that in this work was limited to receive data through the serial port of the arduino and process the data to discriminate adequately whether it is a vehicle or not. The discrimination stage is carried out by applying some criteria using a mixture of the captured data. This validation process was done using Python.

As a first step, a study was made of the physical characteristics of the construction of the intersections. These establish from physical aspects of the street and signage, to aspects of electrical installations that allow the operation of traffic lights and other equipment. This allowed us to define possible scenarios that would establish the minimum requirements of the rest of the components. After this, the selection of components to be used continued, beginning with the sensors. These had to fulfill additional characteristics like low cost, operational simplicity and low maintenance. In the case of the server, a specific hardware selection was not contemplated since it was considered that it fulfills a generic role.

For this work, only one arduino was implemented with a sensor kit in the simplest scenario configuration represented by a road and a single pole traffic light.

Having chosen the sensors, the programming of the arduino was done by the Arduino IDE using the USB port. After the implementation of Arduino and its sensors, we proceeded to capture data to be able to define the validation criteria of the data. This consisted in the analysis of the data as curve behavior among others. Finally, a Python script was programmed that implemented the previously defined algorithms.

As a result, a favorable response of the chosen distance sensor was obtained. This presented an exceptional behavior because even out of range it didn't significantly interfere with the

validation stage. In addition to this was a Doppler sensor that allowed to determine the speed. Unfortunately this sensor failed to meet technical expectations. The main reason was high noise level in the signal that, even using filters and amplifiers, it was'n'ot possible to obtain an adequate behavior that would adequately validate data.

In conclusion it can be said that there is great potential in the proposed platform but it's necessary to investigate more about the doppler sensors and their characteristics. The platform has a high probability of operating on a large scale as part of a more complex system that allows exploiting the information captured in a set of intersections that will surely present a relevant pattern for large-scale decision making.

Keywords: traffic lights; Arduino; smartcity;

Índice de Contenidos

Agradecimientos	VI
Resumen	VII
Abstract	IX
Índice de Tablas	XII
Índice de Figuras	XIV
Capítulo I Introducción	1
Capítulo II Estado del arte	3
2.1 Descripción	3
2.1.1 Detección por procesamiento de video	4
2.1.2 Detección neumática	5
2.1.3 Sensor piezoeléctrico	5
2.1.4 Circuito inductivo	6
2.1.5 Sensor magnético	7
2.1.6 Detector acústico	7
2.1.7 Detector infrarrojo	8
2.1.8 Microondas y radar	8
2.1.9 Radio frecuencia	9
2.1.10 GPS	9
2.1.11 Láser	10
2.1.12 Registro de datos	10
2.1.13 Transmisión de datos	11
Capítulo III Diseño e implementación del prototipo	12
3.1 Alcance del proyecto	12
3.2 Escenarios	12
3.2.1 Dos pistas	13
3.2.2 Tres pistas	13
3.2.3 Caso de estudio	14
3.3 Materiales	15
3.3.1 Arduino	15
3.3.2 Lenguaje Arduino	17
3.3.3 Arduino IDE	17

3.3.4	Range finder	17
3.3.5	Sensor	18
3.4	Métodos	20
Capítulo IV	Resultados y Discusión	22
4.1	Montaje experimental	22
4.1.1	Arduino y sensor LIDAR	23
4.1.2	Arduino y sensor doppler	27
Capítulo V	Conclusiones	32
5.1	Conclusiones generales	32
5.1.1	Aportes relevantes	33
5.1.2	Trabajo futuro	33
Nomenclatura		34
Bibliografía		35
Anexo I	Modelo matemático	37
A.1	Sensor LIDAR	37
A.2	Sensor Doppler	38
Anexo II	Programación	39
B.1	Arduino y sensor LIDAR	39
B.1.1	Captura de distancia en Arduino	39
B.1.2	Filtro en Matlab	40
B.1.3	Detección en Matlab	42
B.2	Sensor Doppler	45
B.2.1	Captura de velocidad en Arduino	45

Índice de Tablas

3.1	Especificaciones técnicas Arduino Mega 2560.	16
3.2	Especificaciones técnicas TfMini.	18

Índice de Figuras

2.1	Detección por procesamiento de imagen.	4
2.2	Sistema neumático de doble tubo.	5
2.3	Sistema de detección piezoeléctrico.	6
2.4	Sistema de circuito inductivo.	6
2.5	Sistema de detección magnético.	7
2.6	Sensor acústico en carretera.	8
2.7	Patrón infrarrojo de una locomotora.	8
2.8	Efecto Doppler de detección de velocidad como radar.	9
2.9	Sistema RFID.	9
2.10	Detección por patrón láser.	10
3.1	Concepto de diseño.	13
3.2	Implementación para dos pistas.	13
3.3	Implementación para tres pistas.	14
3.4	Implementación para dos pistas y un semáforo.	14
3.5	Diagrama lógico de propuesta técnica.	15
3.6	Arduino Mega 2560.	15
3.7	Arduino IDE.	17
3.8	Sensor range finder modelo TfMini.	18
3.9	Sensor doppler HB100.	19
3.10	Patrón radiación polar horizontal y vertical.	19
3.11	Amplificador operacional.	20
3.12	Diagrama de algoritmo de clasificación.	21
4.1	Implementación de desarrollo.	22
4.2	Implementación para toma de muestras con vehículos.	23
4.3	Gráfica para 5 vehículos.	24
4.4	Gráfica para 5 vehículos con identificador de zonas.	24
4.5	Gráfica variación de pendiente.	25
4.6	Gráfica variación de pendiente filtrada para 5 vehículos con detección.	25
4.7	Gráfica para 5 vehículos con detección.	26
4.8	Implementación para toma de muestras con personas.	26
4.9	Gráfica análisis para 1, 2 y 3 personas.	27
4.10	Captura de pantalla del puerto serial del sensor aislado.	28
4.11	Generador de señales.	28
4.12	Arduino capturando frecuencias del generador de señales.	29
4.13	Comparación respuesta del amplificador operacional y frecuencia del generador de señales.	29

4.14 Comparación de señal del sensor doppler original y posterior al amplificador operacional.	30
4.15 Diagrama lógico de propuesta técnica con mejora.	30
4.16 Señal del sensor doppler original y posterior amplificación.	31
4.17 Comparación de señal del sensor doppler original y posterior al amplificador operacional incluyendo mejora.	31
A.1 Principio LIDAR.	37

Capítulo I

INTRODUCCIÓN

Capítulo I

Introducción

En la actualidad la congestión vehicular es un problema real que afecta a muchas ciudades en el mundo y las ciudades en Chile no son la excepción. Esto como consecuencia trae varios efectos como lo son el aumento del uso de combustibles, en teoría, también puede causar un aumento de los costes de combustible, aumento de la polución, el desgaste de los vehículos como consecuencia de la ralentización en el tráfico y la frecuencia de aceleración y frenado, lo que hace más frecuentes que se produzca reparaciones y reemplazos, prolongación de los tiempos de desplazamiento, accidentes, stress de los pasajeros entre otros tantos efectos no tan visibles. Varios de estos aspectos ya son estudiados por distintas disciplinas en búsqueda de medidas mitigadoras o definitivamente erradicadoras. Esto a llevado a ingenieros y científicos a crear modelos matemáticos etc. con tal de crear una solución.

En la actualidad Temuco solo cuenta con un plan paleativo que solo afecta al transporte público (Araucanianoticias, 2017a) (Araucanianoticias, 2017b), y parcialmente a los vehículos particulares (MTT, 2018) que en los últimos quince años se ha septuplicado con Temuco como la cuarta ciudad con más vehículos en Chile con una tasa anual del 8% (Díaz, 2016). Es importante mencionar que este plan ha sido propuesto y llevado acabo por el mismo gremio (DiarioTiempo21, 2016).

El objetivo general de este trabajo es diseñar un dispositivo basado en la tecnología Arduino, sensores LIDAR y doppler que registren el tránsito vehicular para colaborar en la masificación de la incorporación de más Scoots para agilizar el tránsito y reducir la congestión.

Los objetivos específicos son:

- Seleccionar sensores que permitan registrar velocidad y distancia según las condiciones operativas determinadas por la documentación de la UOCT

- Desarrollar un dispositivo de captura, procesamiento y transmisión de datos de los sensores
- Desarrollar algoritmo de clasificación de datos para el conteo de vehículos basado en la data capturada

Capítulo II

ESTADO DEL ARTE

Capítulo II

Estado del arte

En este capítulo se revisarán las tecnologías más usadas en la actualidad para la detección exponiendo las distintas técnicas que explota cada una de ellas. Como cierre se mencionan algunos aspectos relativos al almacenamiento y transmisión de datos.

2.1 Descripción

El estado del arte actual es bastante amplio, existen muchos métodos de sensado de tráfico (Windmill, 2018), desde el procesamiento de imagen, el uso de dispositivos inteligentes como los teléfonos celulares, vehículos inteligentes, hasta el uso de sensores de algún tipo en la vía pública como pueden ser de presión o magnético. Estos entregan a su vez datos que permiten determinar la velocidad de desplazamiento, la densidad vehicular circulante, y en algunos casos, identificar el vehículo o al ocupante mediante su matrícula o su teléfono móvil. Estos sistemas alimentan una plataforma distribuida que permite tomar decisiones como lo son el desvío de calles, cambio en el tiempo de los semáforos, informar a los usuarios el estado del tránsito para que tomen vías alternativas e incluso proveer información estadística de comportamiento ciudadano para el diseño de planificación urbana.

Los sistemas que solamente registran información cuentan como mínimo con las siguientes etapas:

- Detección
- Transmisión de datos

La etapa de detección es tal vez la más compleja de todas. Esta etapa busca detectar un

vehículo por medio del uso de algún transductor que es medido generando algún tipo de dato que posteriormente es procesado para determinar si existe o no la presencia de un vehículo. En la mayoría de los dispositivos que se presentarán más adelante esta etapa está resuelta como caja negra para el sistema.

Finalmente la transmisión de la información puede usar variadas tecnologías que permiten enviar la información recabada a algún sistema distribuido o centralizado según se requiera. Esto puede ser realizado mediante sistemas de comunicaciones cableadas o inalámbricas. Existen muchas tecnologías con distintas técnicas que tratan de solucionar esto. A continuación se presentarán los sistemas más usados en la actualidad.

2.1.1 2.1.1 Detección por procesamiento de video

La detección por procesamiento de video es una labor intensiva para los equipos que la realizan con similar precisión a la de una persona observando el video. Dado que es intensivo el uso de hardware, este también requiere de mejores prestaciones técnicas como lo son cámaras de buena resolución, capacidad nocturna, protección contra la lluvia, fuente de energía con alta capacidad, etc. además de requerir mayor mantenimiento. Todo esto, si bien la tecnología ha bajado sus costos, aún sigue siendo costoso tanto de implementar como de operar. (Meng Mao, 2016)

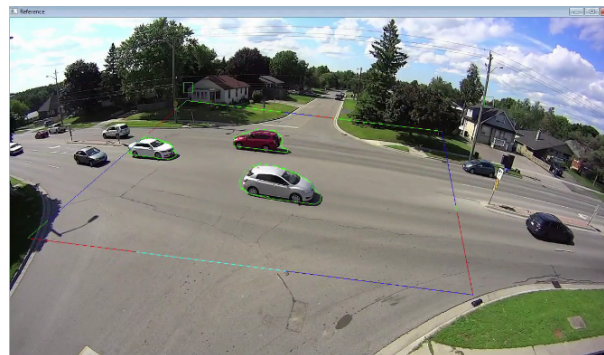


Figura 2.1: Detección por procesamiento de imagen.

Este sistema si bien es costo tiene notables ventajas como lo son poder contar vehículos en varias direcciones, detectar algunos aspectos del vehículo, registrar conteo de vehículos y no de video (generar pocos datos para su almacenamiento), requerir una sola unidad por intersección en general, entre otros.

2.1.2 2.1.2 Detección neumática

Este sistema por muchos años fue el más usado. El principio central de funcionamiento se basa en el cambio de presión de aire del tubo al recibir la presión de un vehículo, este cambio es detectado por la unidad.



Figura 2.2: Sistema neumático de doble tubo.

Este sistema puede trabajar en varias configuraciones de tubos lo que le permite tener varias características que pueden ser de utilidad como lo son la dirección. El principal inconveniente de este sistema son precisamente los tubos, ya que con el constante pisar de los vehículos, estos se deterioraban rápidamente. Es por esto mismo que el sistema requiere de un elevado mantenimiento y de un supervisor. El sistema en si es preciso para registro de bajas tasas de vehículos. Un estudio con un lapso de medición de 15 minutos arrojó que el error absoluto es cercano al 10 %. Este sistema requiere de una unidad de medición por cada tubo de sensado. (B.A. Harvey, 1993)

2.1.3 2.1.3 Sensor piezoeléctrico

Los sensores piezoeléctricos funcionan con el principio de la detección por presión. Estos convierten la energía mecánica en energía eléctrica que altera el estado del sistema. El vehículo genera una diferencia de potencial al pisar el sensor, tan pronto deja de presionar retoma su voltaje normal. Al igual que el sistema neumático, este requiere de una unidad central de medición que registra estos datos. (P. Anand Balaji, 2017)



Figura 2.3: Sistema de detección piezoeléctrico.

2.1.4 2.1.4 Circuito inductivo

Este sistema usa el principio de la inducción magnética de los metales al atravesar un campo magnético generado por un circuito en el cual circula una corriente. El sistema es capaz de detectar estas alteraciones en la corriente circulante, lo que se traduce en el conteo de un vehículo. Este dispositivo contempla la inserción de un circuito eléctrico, una o varias espiras, en el pavimento en la cual se hace circular una corriente generando un campo magnético que al ser atravesado por un metal como la carrocería de un vehículo, se produce una alteración por efectos de la inducción del metal en el campo magnético. (Zbigniew Marszalek, 2015)

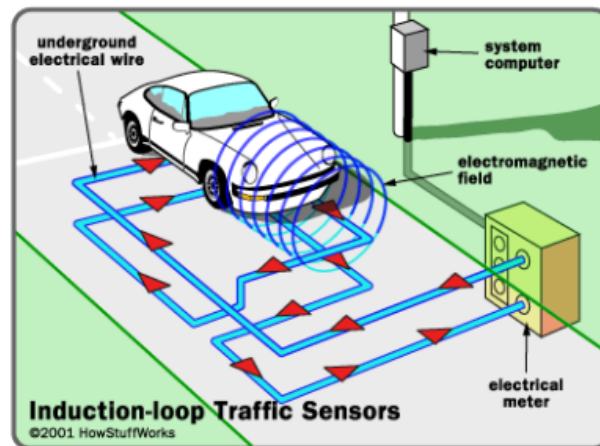


Figura 2.4: Sistema de circuito inductivo.

La principal desventaja de este sistema es su alto coste, ya que requiere de la instalación de un circuito en el pavimento por cada carril.

2.1.5 2.1.5 Sensor magnético

Este sistema funciona con un principio parecido al anterior pero con la diferencia que se mide el campo magnético de la tierra. El sensor al estar cerca de objetos con distinta densidad en el uso de metal en las distintas partes que componen el vehículo, detecta anomalías en el campo magnético, lo cual se interpreta como la presencia de un vehículo. La principal problemática se da en el caso de que los vehículos estén muy cerca entre ellos, lo cual dificultaría la discriminación de uno a otro. Esta situación sería el clásico embotellamiento de vehículos. (J. Pelegri, 2002) Adicionalmente tenemos lo costoso que es implementar estos dispositivos que, si bien usan comunicación inalámbrica, requieren de obras civiles.

La UOCT hace uso principalmente de este sistema. Este organismo dependiente del Ministerio de Transporte y Telecomunicaciones de Chile que se encarga de administrar y operar los sistemas de control de tránsito en las principales ciudades de Chile. Los sistemas de control monitorean en tiempo real la operación de la mayoría de los semáforos existentes en las ciudades mediante una red de comunicaciones que conecta cada semáforo con el sistema de control. (UOCT, 2011) Según especialistas de esta unidad, este sistema tiene un grado de asertividad superior al 90 % que usualmente ronda el 95 %.

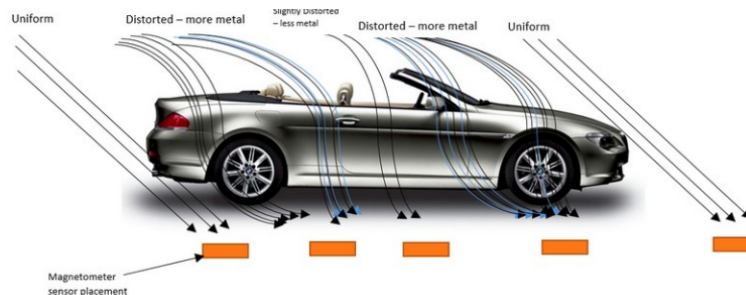


Figura 2.5: Sistema de detección magnético.

2.1.6 2.1.6 Detector acústico

Este sistema reconoce patrones acústicos característicos generados por vehículos mediante el uso de un arreglo de micrófonos. Este sistema no se puede usar en intersecciones por la sensibilidad al ruido del ambiente, en especial en embotellamientos además de ser costosos de implementar. Pueden cubrir hasta cinco carriles. La gran ventaja de este es su simplicidad técnica que incluso puede determinar la velocidad. (J.F. Forren, 1997)



Figura 2.6: Sensor acústico en carretera.

2.1.7 2.1.7 Detector infrarrojo

Este sistema se basa en la medición de la energía infrarroja irradiada por el vehículo. Los vehículos, al ser de combustión de combustibles fósiles, liberan una gran cantidad de energía infrarroja que puede ser medida.



Figura 2.7: Patrón infrarrojo de una locomotora.

La dificultad que podría tener esto es el caso de los vehículos eléctricos. Esta tecnología es muy usada en otras áreas. Estos detectores existen en varias configuraciones de implementación pero todos hacen uso del mismo principio. (Enas Oudat, 2015)

2.1.8 2.1.8 Microondas y radar

En este caso el sistema emite microondas de baja energía a una zona objetivo de la cual se analiza el patrón reflectivo. Este fenómeno comúnmente se le conoce como Doppler. Este fenómeno mide los cambios en la frecuencia de una señal al ser reflejada por una superficie en movimiento. Este patrón varía según el movimiento de los vehículos que entran, atraviesan

y salen de la zona objetivo. Una de las ventajas de este sistema es además de detectar un objetivo, permite determinar su velocidad, dirección, sentido e incluso su tamaño. Esta tecnología es muy conocida y utilizada desde hace mucho tiempo en muchas otras aplicaciones. (Lin Cheng, 2008)

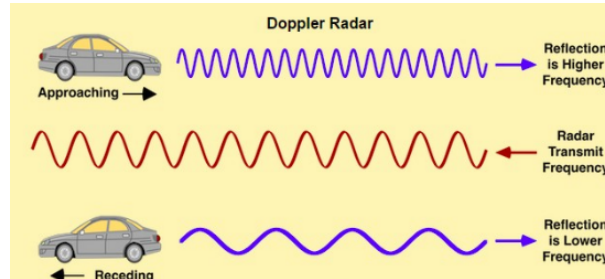


Figura 2.8: Efecto Doppler de detección de velocidad como radar.

2.1.9 Radio frecuencia

La tecnología de radio frecuencia se basa en el uso de un dispositivo identificador o RFID en el vehículo que permite determinar el cruce de un pórtico o poste mediante la recepción de una señal que transmite el pórtico o poste que es contestada por la unidad RFID. Este sistema es muy utilizado en las autopistas para el cobro. En Chile correspondería al sistema TAG. (Wang Hongjian, 2012)

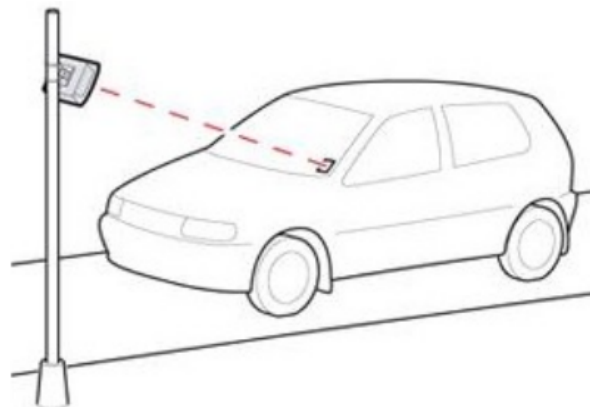


Figura 2.9: Sistema RFID.

2.1.10 GPS

El GPS, y originalmente Navstar GPS, es un sistema que permite determinar en toda la Tierra la posición de un objeto (una persona, un vehículo) con una precisión de hasta

centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión. Esta tecnología hoy viene implementada en los vehículos mediante tres formas. La más común es mediante el uso del GPS ofrecido por el teléfono móvil que viene provisto de este sistema. Este sensa algunos parámetros de desplazamiento y es capaz de determinar si se desplaza en vehículo o a pie. La segunda opción y poco a poco más común son los vehículos que están equipados con tecnología GPS de fábrica integrados en el sistema de navegación de este. Como última alternativa es la opción de instalar un GPS de abordó en un vehículo. Todas estas opciones funcionan de igual forma. (Brindle, 2013)

2.1.11 2.1.11 Láser

La detección por geometría mediante el uso de láser es otra de las alternativas. En este caso el principio se basa en la reflexión de la longitud de onda del láser proyectado sobre el vehículo que es detectada por diodos sensores de la misma longitud de onda. La principal desventaja es su compleja implementación. (Laksono Kurnianggoro, 2014)

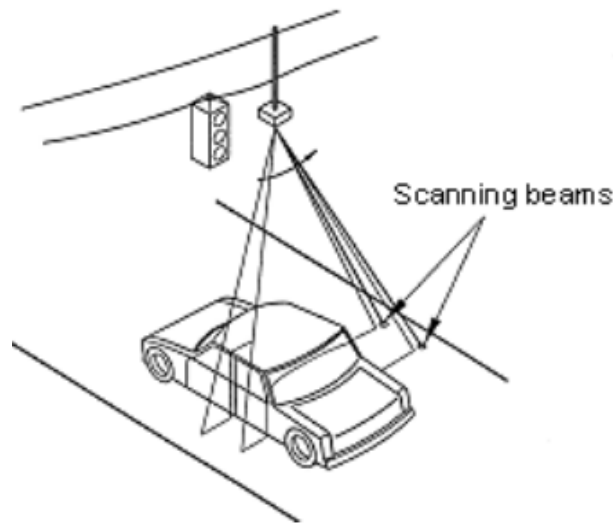


Figura 2.10: Detección por patrón láser.

2.1.12 2.1.12 Registro de datos

El registro de datos puede realizarse de forma local o remota. En el caso del registro local, esto se realiza con el objetivo de descargar cada cierto periodo de tiempo la información. Este es el caso más común, ya que es más tolerante a fallas distribuidas en el sistema de comunicación. El método de almacenamiento local no es relevante.

En el caso de los sistema de registro remoto, esto se da en situaciones donde el sistema en sí se encuentra conectado mediante cableado que permite el sensado y registro remoto en tiempo real.

2.1.13 2.1.13 Transmisión de datos

La etapa de transmisión de datos puede ser manual, por acceso físico a la unidad de almacenamiento del sistema o remota. Esta se puede realizar mediante comunicación inalámbrica, ya sea mediante comunicación celular o cualquier otra. Actualmente la más común es el uso de las tecnologías celulares que son de bajo costo tanto el servicio como de equipos además de poseer excelentes prestaciones técnicas para el acceso a la información registrada.

Capítulo III

DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO

Capítulo III

Diseño e implementación del prototipo

En este capítulo se presentan el alcance del proyecto comentando algunos posibles escenarios y el caso de estudio desarrollado. También se comenta sobre los materiales seleccionados y la metodología usada

3.1 Alcance del proyecto

En este proyecto se pretende innovar desarrollando una nueva alternativa de detección. La plataforma base de detección, almacenamiento y transmisión estarán soportados en una plataforma MCU Arduino que contarán con un hardware diseñado a medida para cumplir la funciones ya mencionadas. El objetivo de este desarrollo no es más que presentar una alternativa que sea práctica y de bajo costo de mantenimiento y operación. La confiabilidad del sistema en la detección debe encontrarse en rangos por determinar que permitan tomar decisiones. Para este proyecto se utilizará la alternativa láser en conjunto con un radar doppler para la detección de vehículos. Se debe entender que el proceso de validación de los datos capturados se realizará en el computador, no en el Arduino, esto es solo una primera aproximación a desarrollar esta tecnología.

3.2 Escenarios

La documentación de la UOCT (UOCT, 2011) comprende varias consideraciones como por ejemplo condiciones para instalar semáforos de mayor altura o con protección o de brazo. Como propuesta se pretende utilizar tecnología óptica en conjunto con radar para medición de distancia a objetos para así poder determinar su presencia o no. El concepto de implemen-

tación se ilustra en la Figura 3.1. En la figura se aprecia un rectángulo rojo que representa la unidad con los sensores. Los vectores que nacen desde esta figura representan la proyección espacial de las señales de los pares de sensores. Cada vector está compuesto por dos señales. La óptica del sensor LIDAR y la electromagnética del sensor doppler.

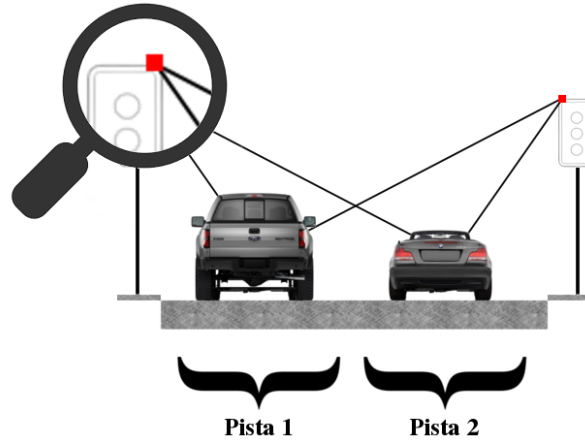


Figura 3.1: Concepto de diseño.

3.2.1 3.2.1 Dos pistas

El caso de dos pistas aparece ilustrado en la Figura 3.2. Este caso es tal vez la situación más común, es por esto que el estudio nace de esta premisa.

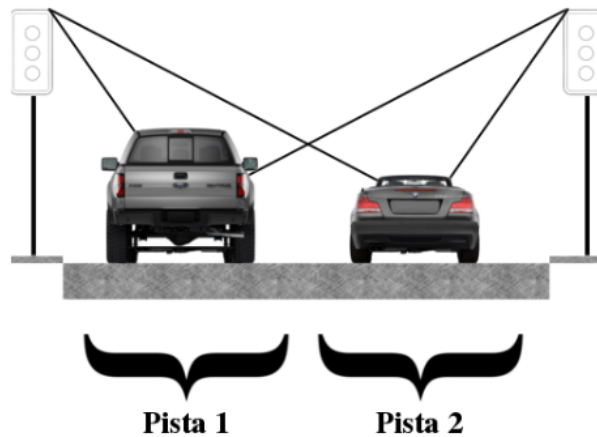


Figura 3.2: Implementación para dos pistas.

3.2.2 3.2.2 Tres pistas

El siguiente caso presenta la situación que se da para tres pistas y dos semáforos laterales.

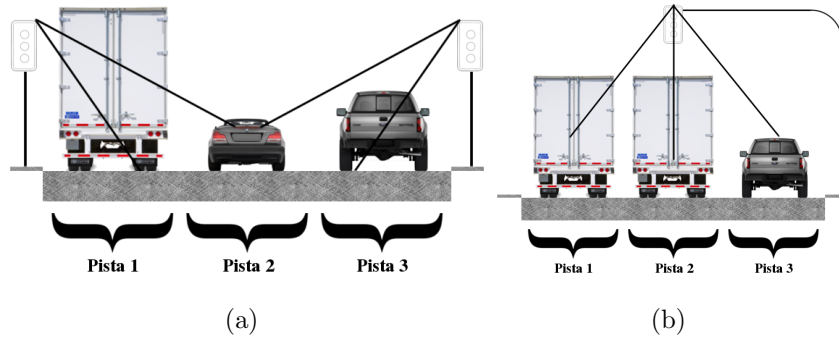


Figura 3.3: Implementación para tres pistas. En a) semáforos laterales, en b) brazo.

Como se aprecia en la Figura 3.3.a, se puede dar la posibilidad de obstrucción de la vista de uno de las unidades controladoras de los sensores. La unidad debe ser capaz de detectar esta situación. En conjunto se puede ver que si existieran dos vehículos grandes en ambos costados es imposible determinar la presencia o no de algún vehículo en la pista del medio. Esta situación se soluciona si es que se da el caso de un semáforo con brazo como aparece en la Figura 3.3.b.

Las ilustraciones han sido realizadas siguiendo el escalamiento correspondiente.

3.2.3 3.2.3 Caso de estudio

Para efectos de este trabajo, solo se considerará la situación ilustrada en la Figura 3.4. Esta configuración muestra solo un semáforo y un vector, esto con el fin de ver la factibilidad técnica de la configuración más común de dos pistas y una unidad controladora con sensores apuntando a una pista siguiente a la contigua.

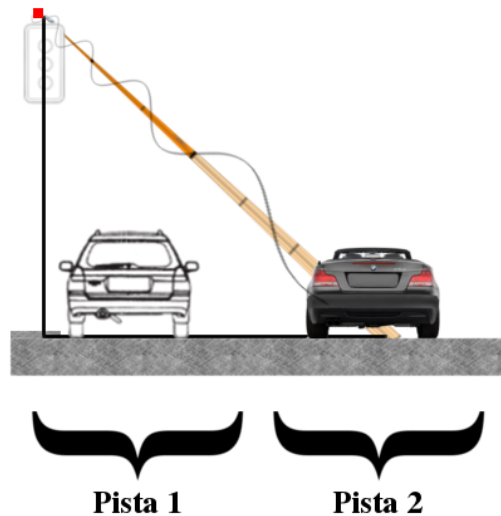


Figura 3.4: Implementación para dos pistas y un semáforo.

El siguiente diagrama de bloque muestra la interacción de los sensores.

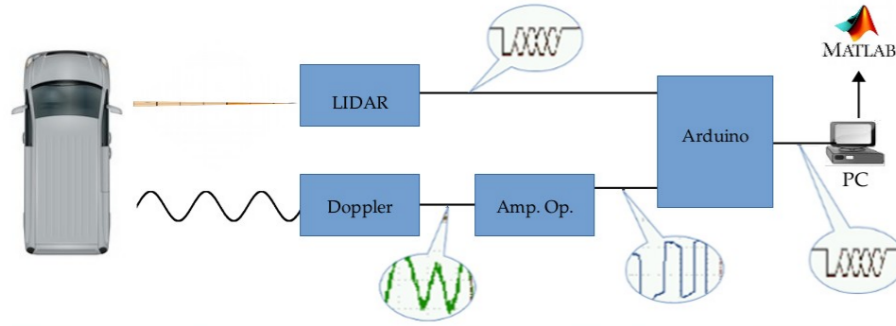


Figura 3.5: Diagrama lógico de propuesta técnica.

El sensor LIDAR entrega por protocolo UART al Arduino la distancia, esta es simplemente entregada al computador. No así en el caso del sensor doppler que debe determinar la frecuencia para determinar la velocidad que finalmente es el dato que entrega al computador. El sensor doppler entrega una señal analógica muy débil que debe ser convertida en un tren de pulsos de amplitud entre 0 a 5 voltios baja. Finalmente luego de haber determinado la velocidad, se envía mediante protocolo UART desde el Arduino al computador para hacer posterior procesamiento.

3.3 3.3 Materiales

A continuación se detallan las prestaciones de los materiales seleccionados comentando ventajas y desventajas por las cuales han sido elegidos.

3.3.1 3.3.1 Arduino

Arduino, para efectos del desarrollo se utilizará la plataforma Mega 2560.

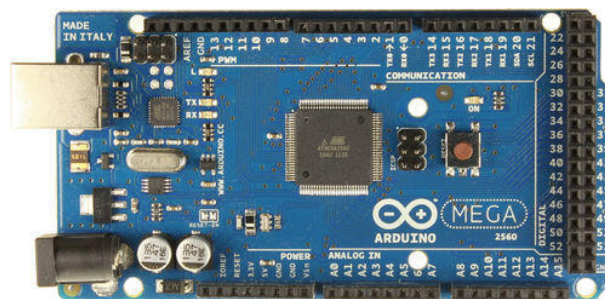


Figura 3.6: Arduino Mega 2560.

Arduino (Genuino a nivel internacional hasta octubre 2016), es una compañía de hardware libre y una comunidad tecnológica que diseña y manufactura placas de desarrollo de hardware, compuestas por microcontroladores, elementos pasivos y activos. Por otro lado las placas son programadas a través de un IDE, el cual compila el código al modelo seleccionado de placa. Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios. Toda la plataforma, incluyendo sus componentes de hardware y Software, son liberados con licencia de código abierto que permite libertad de acceso a ellos. El hardware consiste en una placa de circuito impreso con un microcontrolador, usualmente Atmel AVR, puertos digitales y analógicos de entrada/salida, los cuales pueden conectarse a placas de expansión (shields), que amplían los funcionamientos de la placa Arduino. Asimismo, posee un puerto de conexión USB desde donde se puede alimentar la placa y establecer comunicación con el computador.

Tabla 3.1: Especificaciones técnicas Arduino Mega 2560

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Esta unidad cuenta con las características especificadas en la Tabla 3.1. Este equipo entrega un piso de soporte de sensores para el desarrollo de aplicaciones complejas, pudiendo capturar varias señales, darles un procesamiento primitivo y transmitir esto mediante puerta serial u otro dispositivo como WiFi entre otros. Esta unidad cuenta con la notable ventaja de un bajo consumo energético. Voltaje inferior a 12 voltios, ya que los sistemas de

los semáforos trabajan con ese estándar. Esto permitiría una fácil integración de la fuente de alimentación desde este sistema ya dispuesto. Otra ventaja que presenta es su compacto tamaño. Para el proyecto se utilizó el Arduino Mega 2560 pero para las pruebas en exteriores se usó el Arduino Nano con procesador ATmega328P que es mucho más compacto pudiendo incluso ocupar la cuarta parte o menos de la versión Mega 2560. Como es de esperar, la versión Nano dispone de características técnicas reducidas pero suficientes para poder cumplir la labores requeridas.

3.3.2 3.3.2 Lenguaje Arduino

Como se mencionaba en la descripción del Arduino, este usa un lenguaje de programación estándar y libre que se compila para el hardware específico que implemente el modelo de Arduino o equivalente utilizado. Este lenguaje es codificado en un IDE que lo implementa. Este lenguaje tiene una codificación muy similar al lenguaje C.

3.3.3 3.3.3 Arduino IDE

Este programa IDE permite codificar la rutina operativa del Arduino además de cargarlo y hacer monitoreo de su puerto serial. Este es de uso gratuito, se adapta a muchas versiones de hardware del Arduino, además de permitir integrar librerías externas que implementen otras características como funciones que puedan explotar otras funciones de shields externos añadidos. Durante este trabajo se utilizó la versión 1.8 para Linux x64.



Figura 3.7: Arduino IDE.

3.3.4 3.3.4 Range finder

Un range finder emite una señal luminosa láser que es captada y analizada por el mismo dispositivo para así poder obtener una distancia al objetivo de impacto de la señal luminosa

emitida. La Figura 3.8 muestra el range finder elegido.



Figura 3.8: Sensor range finder modelo TfMini.

Las características técnicas de este dispositivos aparecen en la siguiente Tabla 3.2.

Tabla 3.2: Especificaciones técnicas TfMini

Operating Range (Indoor)	0.3-12[m]
Measurement accuracy	$\pm 4[\text{cm}]@ (0.3-6[\text{m}])$
Default unit of distance	cm
Range resolution	5[mm]
Receiving half angle	1.15°
Transmitting half angle	1.5°
Frequency	100 [Hz]
Wavelength	850 [nm]

Este sensor utiliza tecnología óptica para medición de distancia a objetos. Esta se basa en determinar el tiempo transcurrido entre la emisión y recepción de un as de luz modulado, conocido como el principio TOF. En el caso de usar un láser como luz emisora es que se le conoce como tecnología LIDAR. Esta unidad tiene varios niveles de visibilidad pero en este proyecto se considera el rango menor que va desde los 0.3 a 6 metros de distancia.

3.3.5 3.3.5 Sensor

Para complementar el proceso de detección es que se decidió utilizar un sensor doppler. Este en conjunto con la información entregada por el sensor LIDAR proveerán información cruzable para, mediante criterios definidos más adelante, validar o no la presencia de un vehículo. El sensor escogido es el HB100.



Figura 3.9: Sensor doppler HB100.

Esta unidad trabaja en la frecuencia de 10.587 GHz con un alcance máximo de 30 metros. Operacionalmente esta unidad envía una señal de 10.587 GHz que al rebotar en un cuerpo en movimiento y gracias al efecto doppler, la señal reflejada altera su frecuencia. Esta señal reflejada es recibida en el sensor el cual, mediante un comparador, devuelve una señal que representa la diferencia entre ambas señales. El diagrama de radiación polar horizontal y vertical aparece en la Figura 3.10.

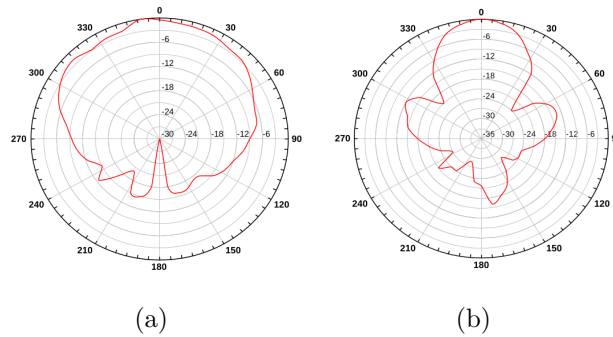


Figura 3.10: Patrón radiación polar. En a) horizontal, en b) vertical.

Para poder hacer uso de este sensor se debe usar un amplificador operacional, ya que la señal de salida es el resultado de una modulación en frecuencia y amplitud muy débiles. El amplificador operacional fue construido siguiendo un diseño propuesto por el fabricante que aparece en la Figura 3.11. Este permite amplificar y discretizar la señal prácticamente dándole una forma de señal cuadrada de tal forma de facilitar la tarea al Arduino que captura la señal de salida de poder tomar muestras y determinar su frecuencia. Al determinar su frecuencia en Hertz se puede dividir por un factor de 44 para obtener la velocidad en kilómetros por hora $[km/h]$ o por 19 para millas por hora $[mi/h]$.

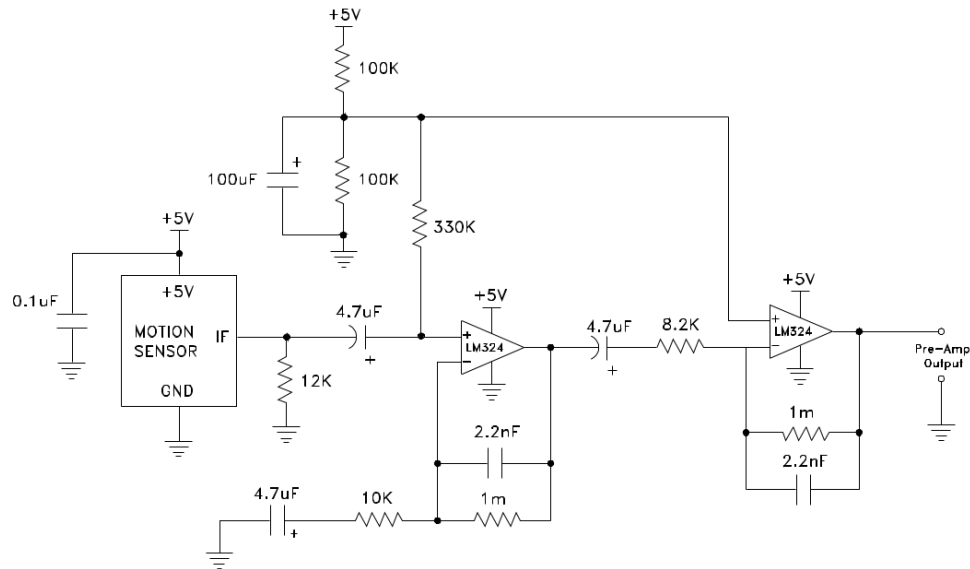


Figura 3.11: Amplificador operacional.

3.4 3.4 Métodos

Inicialmente el problema de implementación de dos sensores se subdividirá, iniciando por el sensor LIDAR.

- Implementar físicamente cada sensor
- Verificar en laboratorio el correcto funcionamiento del sensor
- Programar rutina base de captura de datos
- Capturar datos en un ambiente controlado
- Definir y programar algoritmo de clasificación según datos capturados
- Probar algoritmo de clasificación

Para el uso del sensor LIDAR, se diseñó el algoritmo que se aprecia en la Figura 3.12. Este algoritmo se aplica sobre la conversión de la gráfica de distancia obtenida por el LIDAR a la de la variación de la pendiente determinada entre muestras. En el siguiente capítulo se desarrolla la aplicación y resultados de este algoritmo junto al sensor LIDAR para el caso de vehículos y personas.

Es relevante mencionar que todos los rangos de calibración requeridos han sido obtenidos mediante el ajuste empírico. Para cada escenario e incluso según las condiciones ambientales como luz, humedad y opacidad de la superficie deben ser ajustadas.

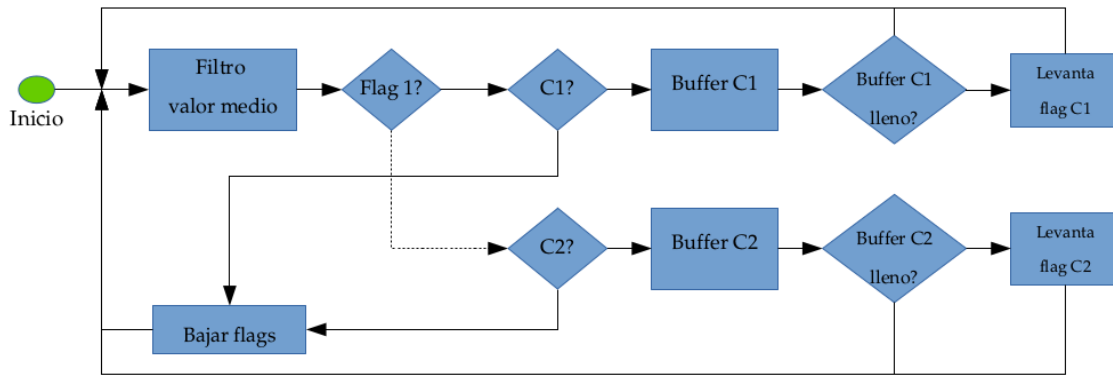


Figura 3.12: Diagrama de algoritmo de clasificación.

El detalle de los elementos del diagrama es el siguiente:

- Filtro valor medio: filtro que toma una ventana y normaliza los valores fuera de rango
- Flag 1?: bandera de cumplimiento de criterio 1
- Criterio 1: entrada del vehículo al cono de aceptación del LIDAR (zona del motor y parabrisas) según rango calibrado
- Buffer C1: score de acumulación según rango calibrado
- Buffer C1 lleno?: caja de decisión si se lleno el buffer anterior para levantar bandera
- Levanta flag C1: levanta bandera de cumplimiento de criterio 1 completo
- Criterio 2: baja variación de la distancia (zona del techo y o lateral del vehículo) según rango calibrado.
- Buffer C2: score de acumulación según rango calibrado
- Buffer C2 lleno?: caja de decisión si se lleno el buffer anterior para levantar bandera
- Levanta flag C2: levanta bandera de cumplimiento de criterio 2 completo
- Bajar flags: se encarga de bajar las banderas para descartar todo

Capítulo IV

RESULTADOS Y DISCUSIÓN

Capítulo IV

Resultados y Discusión

En este capítulo se presentará la implementación física del equipo además de un análisis de los resultados obtenidos por separado para cada sensor.

4.1 Montaje experimental

A continuación se presenta la implementación del Arduino con ambos sensores y el amplificador operacional necesario para utilizar el sensor doppler. La Figura 4.1 ilustra la implementación de desarrollo utilizada.

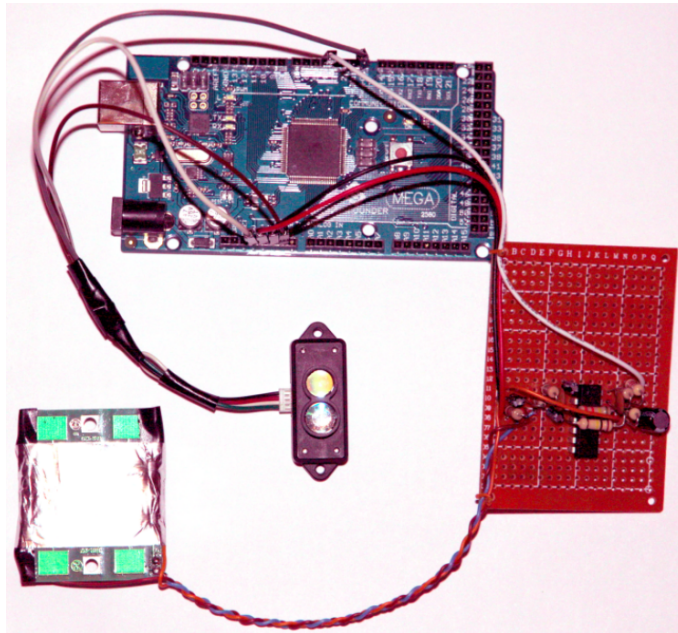


Figura 4.1: Implementación de desarrollo.

4.1.1 4.1.1 Arduino y sensor LIDAR

Luego de implementar el Arduino y el sensor LIDAR además de verificar que cumplía en el laboratorio con las especificaciones del fabricante, se procedió a capturar datos. Recordar que el escenario que se utilizó fue el que aparece en la Figura 3.4.

En los anexos se encuentran las rutinas en Matlab usadas para el filtrado y validación de los datos.

Autos

Como primera etapa se trabajó con mediciones de vehículos. Esta consistió en instalar un Arduino y el sensor en un árbol que se encuentra frente al departamento de Ingeniería Eléctrica a la misma altura donde se ubicaría según se muestra en la Figura 3.4. En la Figura 4.2 se puede ver la disposición de los dispositivos en la toma de muestras.

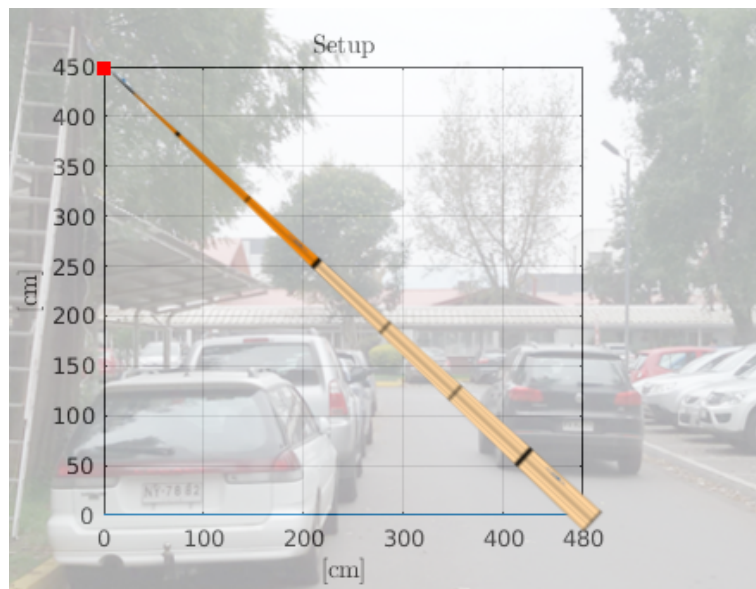


Figura 4.2: Implementación para toma de muestras con vehículos.

La gráfica de los resultados obtenidos corresponde a la Figura 4.3

Se pueden observar variaciones en las curvas planas dado que algunos vehículos pasaban a distinta distancia del sensor, más arriba o abajo y distintas longitudes por su velocidad y dimensión física. Esto se hace irrelevante para el sensor dado que la curva plana es característica para todos es prácticamente la misma.

La curva representa la distancia, esta puede oscilar desde 30 hasta aproximadamente 650 centímetros según las condiciones. Si trabaja fuera de rango el valor por defecto es -1 centímetros.

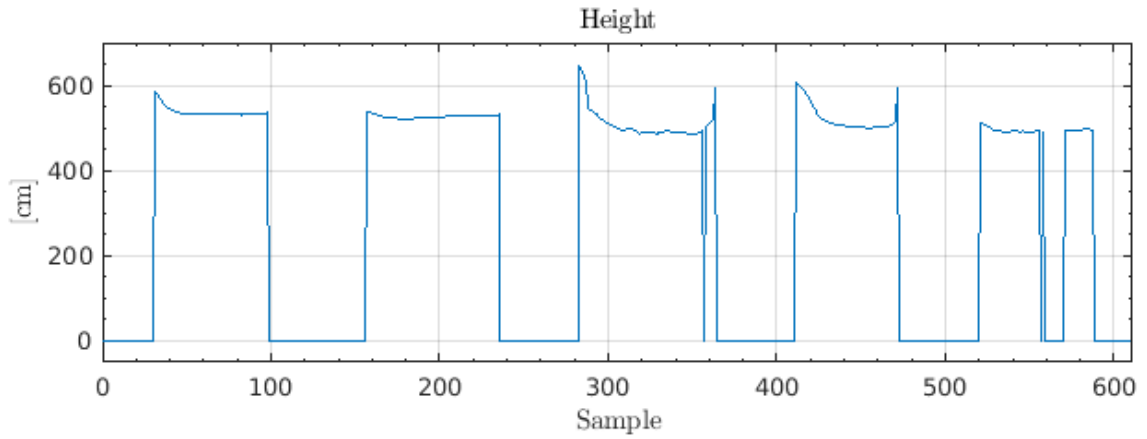


Figura 4.3: Gráfica para 5 vehículos.

A continuación se marcarán zonas en la Figura 4.3 para desarrollar el análisis de caracterización de la curva (Ver Figura 4.4) según la variación de la pendiente.

Las zonas han sido definidas de la siguiente forma.

- Zona roja: Entrada del vehículo al cono de aceptación (zona de motor).
- Zona verde: Baja variación de la distancia (zona del techo o lateral del vehículo)
- Zona rosada: Salida del vehículo del cono de aceptación. Si el vehículo no tiene maletero o zona de carga, como el caso de un city car o un furgón, la salida es más brusca y carece de esta zona.

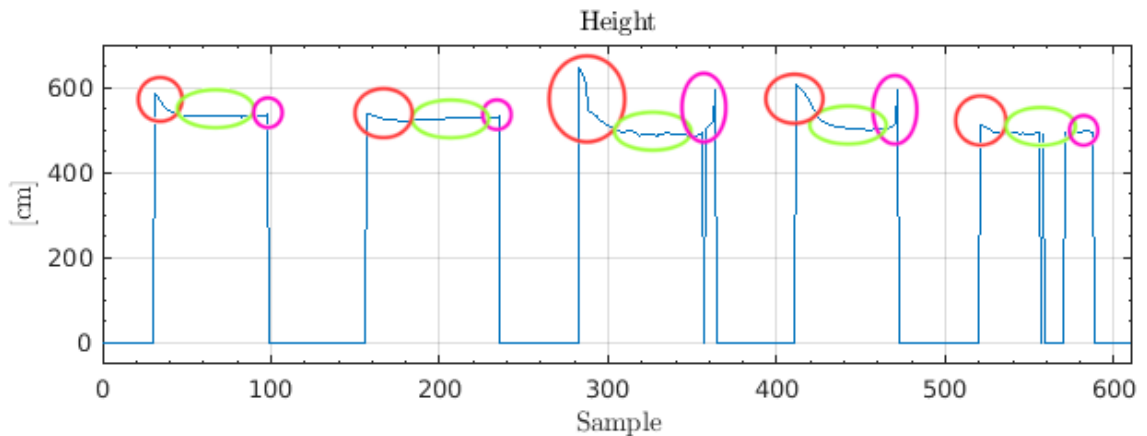


Figura 4.4: Gráfica para 5 vehículos con identificador de zonas.

Es relevante mencionar que si bien entre vehículos el sensor entra en zona fuera de rango al filtrar la señal de variación de pendiente desaparece todo efecto que pudiese tener. En el último vehículo de la muestra el sensor entra en zona fuera de rango pero poco después

recupera la señal.

En la Figura 4.5 se muestra la curva que describe el comportamiento de la pendiente original y filtrada en una ventana de cuatro muestras calculando la media cuando se encuentra fuera de un rango de $[-20, 20]$. Para el procesamiento se deben definir valores límite para acotar los criterios (Ver Anexo B B.1.3).

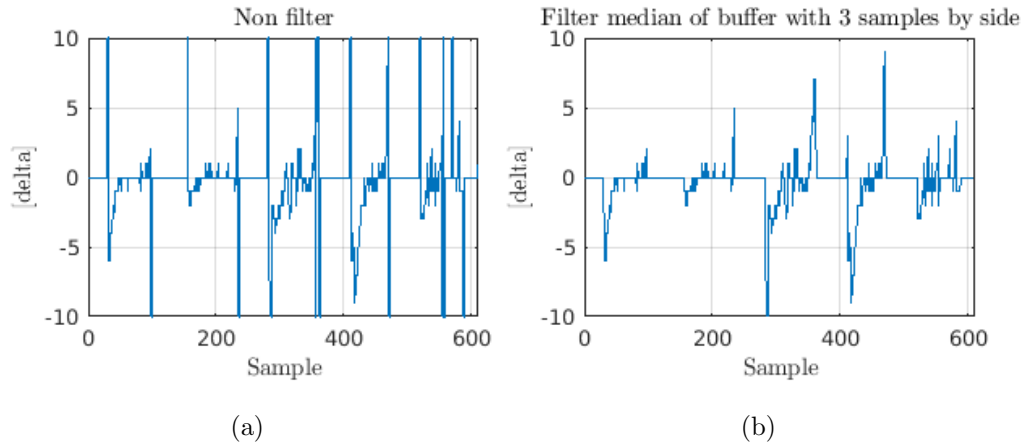


Figura 4.5: Gráfica variación de pendiente. En a) original, en b) filtrado.

La zona roja acumula la pendiente negativa durante cierta cantidad de tiempo, una vez llena, levanta una bandera para pasar a acumular pendiente estable con cierta tolerancia, la zona verde. Una vez completada se levanta la bandera. Se puede seguir acumulando zona verde pero se espera o salir de ella acumulando pendiente positiva hasta cierta cantidad o acumulando cierta cantidad nula.

Una vez aplicado los criterios de zonas descritos anteriormente y programados en Matlab el resultado de detección es el que se aprecia en la Figura 4.6. Al llegar a 2 la curva azul representa una detección positiva para un vehículo.

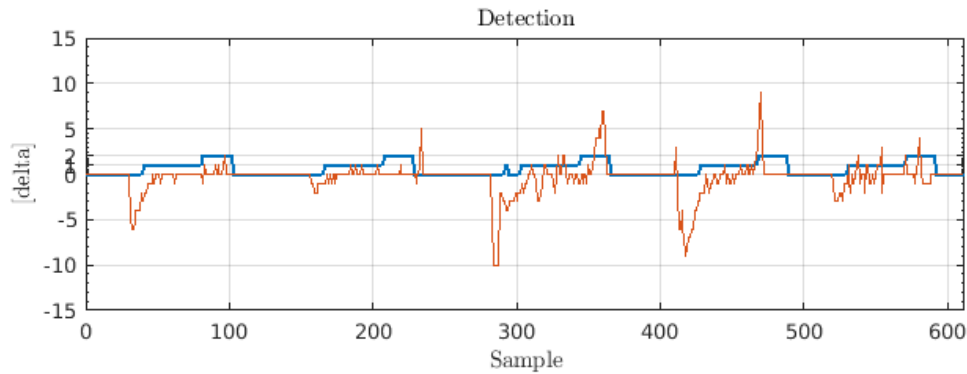


Figura 4.6: Gráfica variación de pendiente filtrada para 5 vehículos con detección.

La Figura 4.7 ilustra la variación de distancia detectada por el sensor LIDAR y la curva

de detección escalada. Al llegar a 200 la curva azul representa una detección positiva para un vehículo.

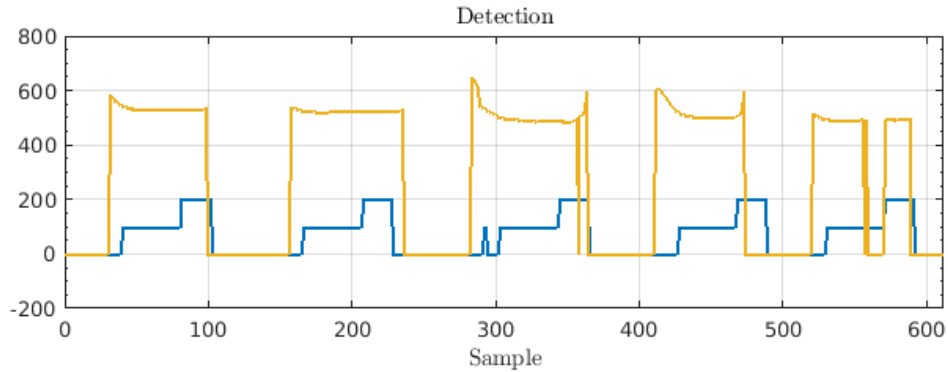


Figura 4.7: Gráfica para 5 vehículos con detección.

Personas

El escenario de toma de muestras es el que se presenta en la Figura 4.8. Haciendo uso del mismo análisis para un gráfico de una, dos y tres personas resulta la Figura 4.9.

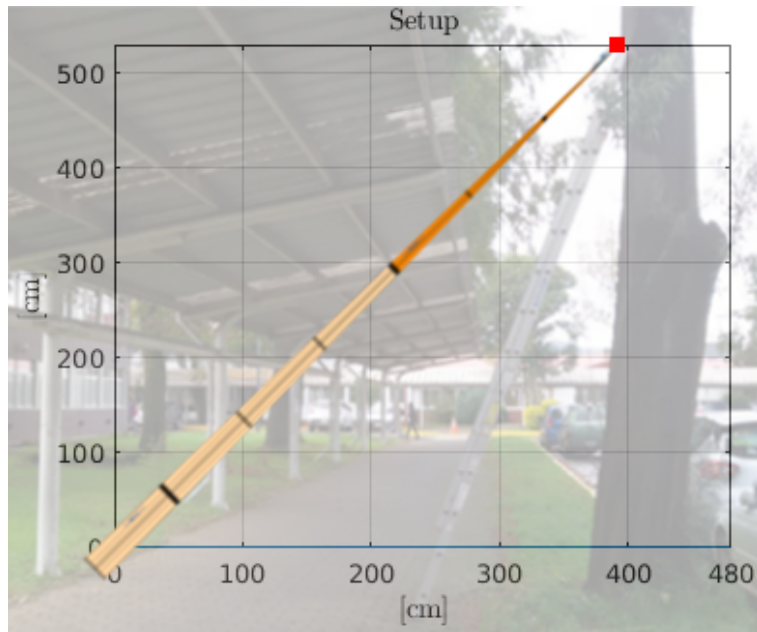


Figura 4.8: Implementación para toma de muestras con personas.

En la figura se representan las personas que caminaron frente al sensor, esto fue direccionando el sensor LIDAR a la calzada peatonal considerando las mismas dimensiones que para la calzada vehicular. Se puede apreciar que la curva de variación de pendiente es sustancialmente distinta a la de los vehículos y haciendo uso de los criterios de detección, ninguna

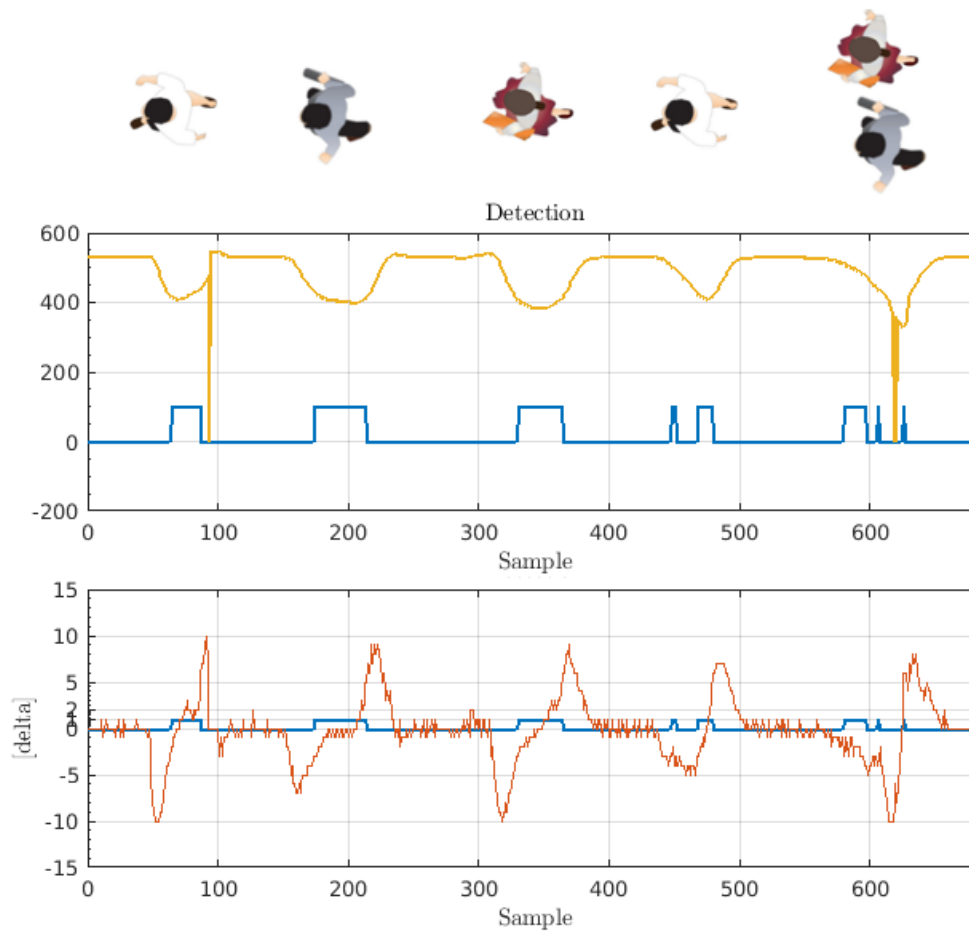


Figura 4.9: Gráfica análisis para 1, 2 y 3 personas.

persona fue detectada como un vehículo.

Otro aspecto relevante que anteriormente se mencionó y aquí se puede comprobar es la indiferencia del sistema a la ausencia de zonas de fuera de rango del sensor entre personas y durante las de muestras. Se puede finalmente concluir que no es problema tener como piso una distancia fija o estar fuera de rango.

4.1.2 4.1.2 Arduino y sensor doppler

Después de hacer la implementación como aparece en la Figura 4.1, se comprobó su funcionamiento en Arduino con códigos básicos de detección de velocidad. Se pudo determinar que no funcionaba correctamente por los valores de velocidad que entregaba (ver Figura 4.10), ya que en algunos casos los valores no eran compatibles con el ancho de banda según las especificaciones o eran datos falsos dado que el sensor estaba dispuesto en la mesa sin ningún tipo de estímulo.

```

320Hz : 7km/h
283Hz : 6km/h
2583Hz : 58km/h
237Hz : 5km/h
284Hz : 6km/h
266Hz : 6km/h
373Hz : 8km/h
258Hz : 5km/h
263Hz : 5km/h
64Hz : 1km/h
..
635Hz : 14km/h
    
```

Figura 4.10: Captura de pantalla del puerto serial del sensor aislado.

Generador de señales

Se procedió a realizar comprobación del amplificador operacional. Para esto se utilizó un generador de señales y el mismo Arduino con el mismo programa para detección de velocidad según la frecuencia. Se utilizaron tres frecuencias de una senoide, $99[Hz]$ equivalente a $2[km/h]$, $1803[Hz]$ equivalente a $40[km/h]$ y $4420[Hz]$ equivalente a $100[km/h]$. En la Figura 4.11 se aprecia el generador de señales indicando la frecuencia de salida. En la Figura 4.12 se aprecian las frecuencias capturadas por el Arduino. Se aprecia una leve diferencia que se incrementa según aumenta la frecuencia.

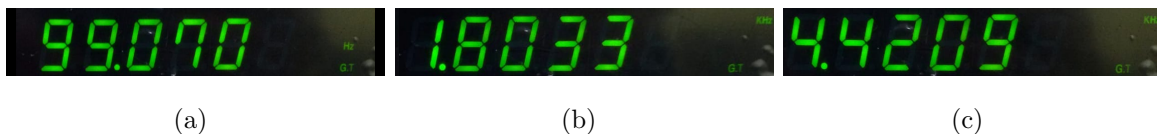


Figura 4.11: Generador de señales. En a) $99[Hz]$, en b) $1803[Hz]$, en c) $4420[Hz]$

Se puede comprobar con esto que la operación del amplificador es correcta pero deficiente. En la Figura 4.13 se ilustran la señal del generador y la respuesta del amplificador para $567[Hz]$, $1803[Hz]$ y $4518[Hz]$.

La respuesta del amplificador operacional comienza a notar serias alteraciones sobre los $500[Hz]$. Esta respuesta es analizada por el Arduino mediante la toma de muestras, esto permite determinar la frecuencia que posteriormente al dividir por 44 se obtiene la velocidad en $[km/h]$. Dado que la señal se deforma, se altera la frecuencia determinada. Es importante

99Hz : 2km/h	1801Hz : 40km/h	4424Hz : 100km/h
99Hz : 2km/h	1798Hz : 40km/h	4424Hz : 100km/h
99Hz : 2km/h	1801Hz : 40km/h	4405Hz : 100km/h
99Hz : 2km/h	1805Hz : 41km/h	4405Hz : 100km/h
99Hz : 2km/h	1801Hz : 40km/h	4405Hz : 100km/h
99Hz : 2km/h	1801Hz : 40km/h	4405Hz : 100km/h

(a)
(b)
(c)

Figura 4.12: Arduino capturando frecuencias del generador de señales. En a) 99[Hz], en b) 1803[Hz], en c) 4420[Hz]

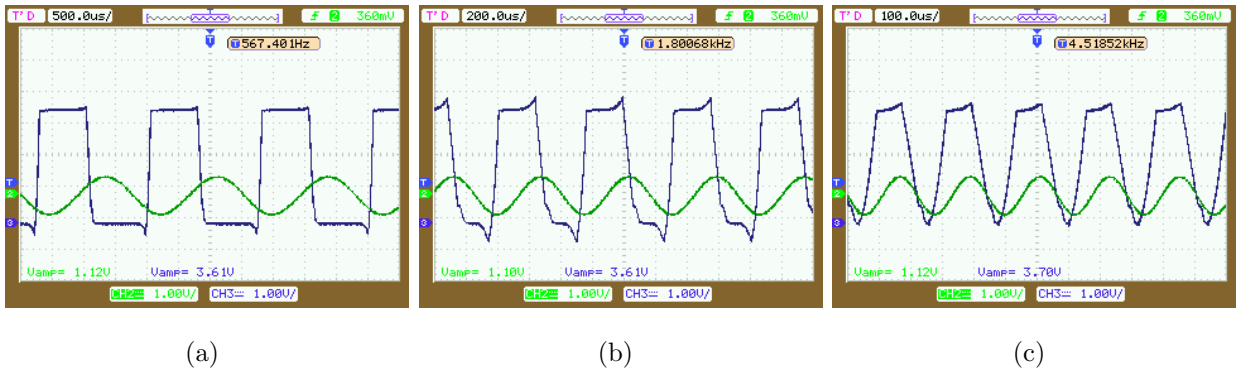


Figura 4.13: Comparación respuesta del amplificador operacional y frecuencia del generador de señales. En a) 567[Hz], en b) 1803[Hz], en c) 4518[Hz]

mencionar que la señal del generador de señales presenta una relación señal-ruido muy elevada en comparación a la del sensor doppler (ver Figura 4.14).

Mano humana agitada

En la Figura 4.14 se aprecia la respuesta del amplificador operacional y la salida del sensor frente al estímulo de la agitación con rapidez de una mano humana a unos 30[cm] de distancia por sobre el sensor. En color verde, la señal de salida del sensor entorno a los 60[mV] de amplitud y un estimado de frecuencia, tomando como referencia la señal de salida del amplificador operacional según el osciloscopio en color azul, de unos 90[Hz]. Se aprecia que la respuesta del amplificador operacional no muestra ningún efecto de la señal recibida del sensor doppler y que no hay relación entre los periodos del tiempo de cada una. Es relevante mencionar que el generador de señales, en las anteriores figuras, tenía una tensión de poco más de 1[V] por lo que la relación señal-ruido naturalmente ayuda bastante al amplificador en comparación a esta última situación.

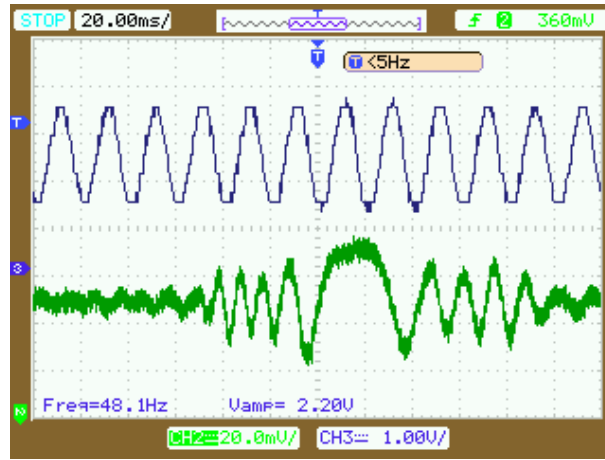


Figura 4.14: Comparación de señal del sensor doppler original y posterior al amplificador operacional.

Mejora

Como prueba se agregó una etapa prefabricada de amplificación entre el sensor doppler y el amplificador operacional. Básicamente esto consistió en amplificar la señal antes de entrar al amplificador operacional para facilitar la conversión a un tren de pulsos por parte de este.

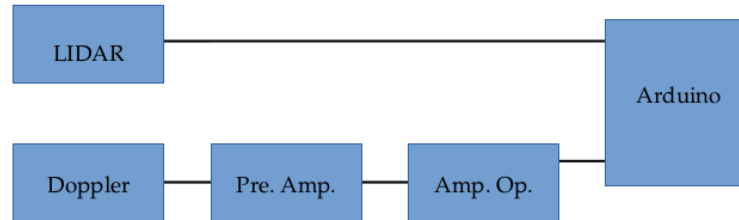


Figura 4.15: Diagrama lógico de propuesta técnica con mejora.

En la Figura 4.16 se puede ver la mejora que experimentó la señal de salida del sensor doppler previa entrada al amplificador operacional.

Para el mismo ejercicio con la mano el resultado se aprecia en la Figura 4.17. En esta figura se puede ver con claridad que hay una relación entre la frecuencia del tren de pulsos y la señal análoga que sale directamente del sensor. Si bien el tren de pulsos tiene un retardo, esta representa con bastante aproximación el periodo relevante en la señal del sensor. Según los cursores del instrumento, tenemos una frecuencia de $108[Hz]$ que correspondería aproximadamente a $2,5[Km/h]$. Adicionalmente se aprecia con claridad el momento en el que la mano cambia de dirección. Otro aspecto relevante a mencionar es la amplitud del tren de pulsos que, en el caso del uso de la etapa de pre amplificación, tiene un valor pick to pick de casi 5 voltios. Esto representaría un incremento de más del 200 % en la amplitud en la salida del amplificador operacional a la entrada del Arduino.

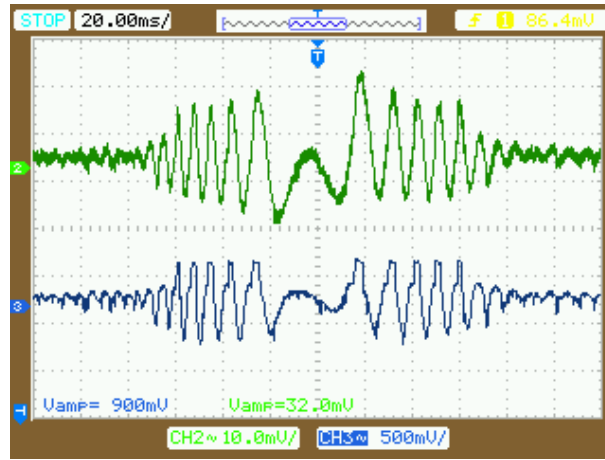


Figura 4.16: Señal del sensor doppler original y posterior amplificación.

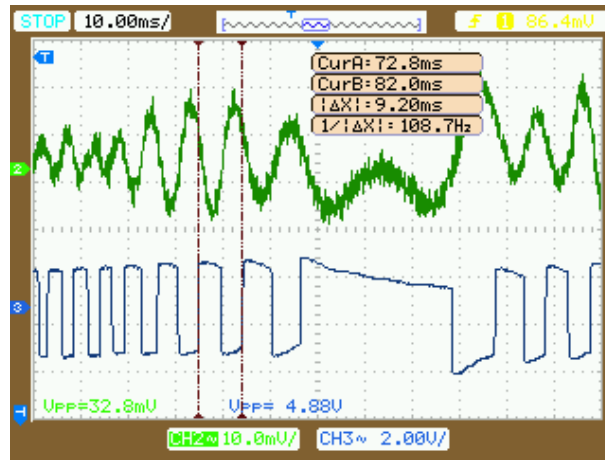


Figura 4.17: Comparación de señal del sensor doppler original y posterior al amplificador operacional incluyendo mejora.

Lamentablemente por falta de tiempo en el desarrollo de este trabajo es que no se pudo incluir esta mejora como para ser sometida a pruebas de laboratorio y posterior experimentación en un ambiente controlado como se hizo en el caso del sensor LIDAR.

Capítulo V
CONCLUSIONES

Capítulo V

Conclusiones

5.1 Conclusiones generales

- El mayor esfuerzo se realizó en conseguir y poner a prueba el concepto, ya que no se sabía si con los sensores disponibles sería viable económica y técnicamente. Parte de las pruebas se vieron demoradas por factores climáticos como la lluvia y el viento que en muchas ocasiones obligó a suspender las actividades. Mencionar también que conseguir el sensor LIDAR tomó poco menos de un mes ya que hubo que realizar una importación.
- El concepto demostró ser completamente viable tanto económica como técnicamente pero requiere profundizar el trabajo realizado con los sensores, específicamente el doppler. En particular el sensor LIDAR resultó ser más costoso que el doppler pero igual de dificultoso de conseguir. Si bien las pruebas realizadas fueron mínimas, cinco autos y seis personas, la asertividad fue del 100 %. Esto cumpliría con las exigencias establecidas por parte de la UOCT.
- El proyecto demostró ser más complejo de lo que parecía inicialmente por lo que tuvo que acotarse para ajustarse a los plazos.
- Algunos aspectos de diseño tuvieron que cambiarse durante el proyecto dado que se logró concretar una reunión con personal especializado de la UOCT que entregaron valiosa información a tener en cuenta. Aspectos prácticos, técnicos y legales como para aumentar su potencial interés comercial en el rubro.

5.1.1 5.1.1 Aportes relevantes

En este trabajo el aporte más relevante fue lo realizado con los sensores poco comunes, ya que estos hubo que adquirirlos fuera de la región incluso fuera del país como fue el caso del sensor LIDAR. Esto se tradujo en trabajar en base a la investigación y experiencias compartidas por internet por otros usuarios. Por esto mismo resultó que el sensor doppler no se lograra hacer funcionar correctamente en el plazo requerido, ya que determinar que el proceso que se requirió en un proceso de validación de los componentes para llegar a esa conclusión.

5.1.2 5.1.2 Trabajo futuro

- Desarrollar una etapa de filtraje y amplificación de mejor calidad para poder determinar si se puede usar el sensor doppler en el escenario que se definió. En caso de determinar que no cumple los requerimientos, probar otro sensor LIDAR con mejores prestaciones que provee el mismo fabricante que, además, sigue estando en el rango de precios viables.
- Sería interesante hacer pruebas más reales como una instalación de los sensores y Arduino en un semáforo a modo de comparación con los resultados obtenidos, ya que las pruebas fueron realizadas en solo una de las calles de la universidad.
- Generar registro remoto en una base de datos. El proyecto inicialmente contemplaba esto pero por plazos hubo que reducirlo solo a los sensores ya que, como se mencionó anteriormente, gran parte del esfuerzo se utilizó en evaluar su funcionamiento.
- Implementar dos pares de sensores en un Arduino. Técnicamente esto podría presentar complicaciones dado la cercanía de las señales además de las limitaciones de procesamiento y I/O del Arduino.
- Desarrollar kits de implementación para darle modularidad y así generar una base técnica escalable práctica.

Nomenclatura

Abreviación	Significado	Unidad
MTT	Ministerio de transportes y Telecomunicaciones	
GPS	Sigla en inglés de Global Positioning System	
Scoot	Árcos de control dinámico	
UOCT	Unidad Operativa de Control de Tránsito	
IDE	Sigla en inglés de Integrated Development Environment (Entorno de desarrollo integrado)	
TOF	Sigla en inglés de Time Of Flight (tiempo de vuelo)	
LIDAR	Light Detection and Ranging (luz de detección y rango)	
MCU	Sigla en inglés de MicroController unit (unidad microcontrolador)	
I/O	Sigla en inglés de input/output o I/O (entrada/-salida)	
w	Velocidad angular	rad/s
c_0	Velocidad de la luz	m/s
φ	Fase angular	rad/s
f	Frecuencia	Hz, nm
v	Velocidad	$m/s, km/h$
V	Voltaje	V, mV

Bibliografía

- Araucanianoticias. 2017a (Feb). *Intendente: renovar microbuses es 'mejorar condiciones de trabajo y de servicio a los usuarios'*.
- Araucanianoticias. 2017b (Jan). *Por primera vez más de 200 taxi-colectivos podrán renovarse gracias a programa de Gobierno*.
- B.A. Harvey, G.H. Champion, R. Deaver. 1993. Accuracy of traffic monitoring equipment field tests. *Paper*.
- Benewake. 2015. *Product mannual, TFmini - Official Product Information*.
- Brindle, Beth. 2013 (Jan). *How does Google Maps predict traffic?*
- DiarioTiempo21. 2016 (Jan). *Congestión en Temuco: Los pro y los contra de la restricción vehicular*.
- Díaz, Catalina. 2016 (Jul). *Parque autormotriz de Temuco y Padre Las Casas aumenta un 8 % cada año*.
- Electronics, Stephan. 2017 (Feb). *Making the Electronics for a 24GHz Doppler Motion Sensor*.
- Enas Oudat, Mustafa Mousa, Christian Claudel. 2015. Vehicle Detection and Classification Using Passive Infrared Sensing. *Paper*.
- J. Pelegri, J. Alberola, V. Llario. 2002. Vehicle detection and car speed monitoring system using GMR magnetic sensors. *Paper*.
- J.F. Forren, D. Jaarsma. 1997. Traffic monitoring by tire noise. *Paper*.
- Laksono Kurnianggoro, Wahyono, Danilo Cáceres Hernández. 2014. Camera and laser range finder fusion for real-time car detection. *Paper*.
- Lin Cheng, Benjamin Henty, Fan Bai. 2008. Doppler Spread and Coherence Time of Rural and Highway Vehicle-to-Vehicle Channels at 5.9 GHz. *Paper*.
- Meng Mao, Yong Zhang, Boyue Wang. 2016. Vehicle Detection through Traffic Video in Congested Traffic Flow. *Paper*.
- MTT. 2018 (Jan). *Marcha blanca para calle Portales, segunda vía exclusiva de Temuco*.

-
- P. Anand Balaji, R. Aadhivijay, P. Deepa Sharma. 2017. Hill road safety assistance using piezoelectric sensor. *Paper*.
- UOCT. 2011. *Documentación técnica*.
- Wang Hongjian, Tang Yuelin. 2012. RFID Technology Applied to Monitor Vehicle in Highway. *Paper*.
- Windmill. 2018 (March). *Vehicle Sensing: Ten Technologies to Measure Traffic*.
- Zbigniew Marszalek, Ryszard Sroka, Tadeusz Zeglen. 2015. Inductive loop for vehicle axle detection from first concepts to the system based on changes in the sensor impedance components. *Paper*.

ANEXO A

MODELO MATEMÁTICO

Anexo I

Modelo matemático

A.1 A.1 Sensor LIDAR

El principio LIDAR utilizado por el sensor es el que se muestra en la Figura A.1.

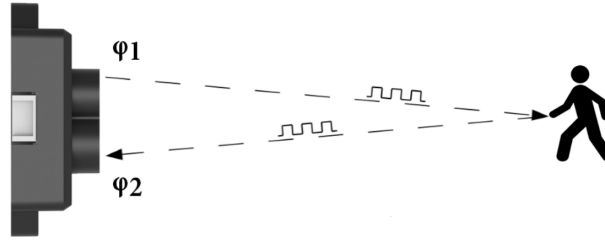


Figura A.1: Principio LIDAR.

La ecuación para determinar la distancia se constituye de:

$$c_0 \approx 299,792,458[m/s]$$

$$w = 2\pi f[rad/s]$$

$$\Delta\varphi = \varphi_2 - \varphi_1[rad]$$

Donde c_0 representa aproximadamente la velocidad de luz, w la velocidad angular de la señal emitida con frecuencia $f = 850[nm]$ y $\Delta\varphi$ como la diferencia angular de fase entre la señal recibida y la emitida. (Benewake, 2015)

La expresión final de la distancia es:

$$Distancia \approx \frac{c}{2} \frac{1}{w} \Delta\varphi[m] \tag{A.1}$$

A.2 A.2 Sensor Doppler

Para el sensor doppler se usó la ecuación de Doppler. (Electronics, 2017)

$$f_D = 2f_0 \frac{v}{c_0} \cos \alpha$$

En la ecuación f_0 es la frecuencia de emisión que para el sensor es de $24,125[GHz]$, c_0 representa aproximadamente la velocidad de luz y $\cos \alpha$ donde α es el ángulo que se forma entre el vector de movimiento del objeto y la línea recta al objeto. Para este caso el sensor solo se probó con movimiento con un ángulo alfa de cero grados aproximadamente que fue el ejercicio de la mano agitada rápidamente (Ver sección 4.1.2).

Despejando la ecuación para la variable v y reemplazando los valores aproximados de las variables se obtiene:

$$v \approx \frac{f_D}{160} [m/s] \tag{A.2}$$

$$v \approx \frac{f_D}{44} [km/h] \tag{A.3}$$

ANEXO B

PROGRAMACIÓN

Anexo II

Programación

B.1 B.1 Arduino y sensor LIDAR

B.1.1 B.1.1 Captura de distancia en Arduino

El siguiente código fue desarrollado para capturar la distancia:

```
#include<SoftwareSerial.h>
SoftwareSerial Serial1(2,3);
int dist;
int strength;
int check;
int i;
int uart[9];
const int header=0x59;
void setup(){
    Serial.begin(115200);
    Serial1.begin(115200);
}

void loop(){
    if (Serial1.available()){
        if (Serial1.read()==header){
            uart[0]=header;
            if (Serial1.read()==header){
                uart[1]=header;
```

```
    for ( i=2;i<9;i++){
        uart [ i]=Serial1.read ();
    }
    check=uart [0]+uart [1]+uart [2]+uart [3]+uart [4];
    check=check+uart [5]+uart [6]+uart [7];
    if ( uart [8]==( check&0xff)){
        dist=uart [2]+uart [3]*256;
        strength=uart [4]+uart [5]*256;
        Serial.print (" dist=");
        Serial.print ( dist );
        Serial.print ( '\t ' );
        Serial.print (" strength=");
        Serial.print ( strength );
        Serial.print ( '\n ' );
    }
}
}
}
}
```

Este código se modificó para capturar los datos sin texto y posteriormente poder procesarlo en Matlab.

B.1.2 Filtro en Matlab

A continuación se presenta el código desarrollado para filtrar los datos registrados:

```
%filter median buffer

%filter range
i=3;
rangeSize=(i*2);%only values pars
tfiltered=zeros(rangeSize/2,1);%zero for first values
upper=10;
lower=0;
```

```
%last value on range
for i=rangeSize/2+1:length(tdata)-(rangeSize/2)
    if abs(tdata(i))>upper%over limit
        tfiltered=[tfiltered;
            median(rangeVertical(tdata,i-(rangeSize/2),
                i+(rangeSize/2)))];%temporary var to pull last stable value
    else
        if abs(tdata(i))>lower%under limit
            tfiltered=[tfiltered;tdata(i)];
        else
            tfiltered=[tfiltered;0];
        end
    end
end

tfiltered=[tfiltered;zeros(rangeSize/2,1)];%add zeros to the end
x=1:length(tdata);

figure1 = figure;
subplot1 = subplot(1,2,1,'Parent',figure1);
hold(subplot1,'on');
plot(x,t(:,3),'Parent',subplot1);
xlabel('Sample','Interpreter','latex');
title('Non filter','Interpreter','latex');
ylabel('[delta]','Interpreter','latex');
xlim(subplot1,[0 length(x)]);
ylim(subplot1,[-10 10]);
box(subplot1,'on');
set(subplot1,'XGrid','on','XMinorTick','on','YGrid','on',
    'YMinorTick','on');

subplot2 = subplot(1,2,2,'Parent',figure1);
hold(subplot2,'on');
```

```
plot(x,tfiltered,'Parent',subplot2);
xlabel('Sample','Interpreter','latex');
title('Filter median of buffer with 3 samples by side',
'Interpreter','latex');
ylabel('[delta]','Interpreter','latex');
xlim(subplot2,[0 length(x)]);
box(subplot2,'on');
set(subplot2,'XGrid','on','XMinorTick','on','YGrid',
'on','YMinorTick','on');
```

Función rangeVertical

La siguiente función se implementó para retornar un sub arreglo del arreglo entrega. El código es el siguiente:

```
function [temp]=rangeVertical(t,init,last)
temp=[];
for i=init:last
    temp=[temp;t(i)];
end
```

B.1.3 Detección en Matlab

Para la detección se desarrolló el siguiente código:

```
%zones

%vars zones, adjust to calibrate
z1size=8;
z2size=40;
z3size=20;
upper=4;
lower=-4;

tz1=0;
```

```
tz2=0;
tz3=0;
tz3=0;

tdata=filtered;

flag=[0];
for i=1:length(tdata)
    temp=flag(i);
    %zone 0
    if temp==0 %building zone 1
        if tdata(i)<0
            if tz1<z1size
                tz1=tz1+1;
                flag=[flag;0];
            else
                if tdata(i)<upper && tdata(i)>lower %if enter into
                    %the zone 2 range, go to build zone 2
                    flag=[flag;1]; %go to build zone 2
                    tz1=0;tz2=0;tz3=0;
                else
                    flag=[flag;0];
                end
            end
        end
    else
        flag=[flag;0]; %restart
        tz1=0;tz2=0;tz3=0;
    end
end
%zone 1
if temp==1 %building zone 2
    if tdata(i)<upper && tdata(i)>lower
        if tz2<z2size %between range
            tz2=tz2+1;
```

```
        flag=[flag;1];
    else %full zone 2 goto zone 3
        flag=[flag;2];% full zone 2
    end
else
    flag=[flag;0];% restart
    tz1=0;tz2=0;tz3=0;
end
end
%zone 2
if temp==2 %holding zone 2 o getting out
    if tdata(i)<upper && tdata(i)>lower && tz3<z3size %on range
        flag=[flag;2];
        tz3=tz3+1;
    else
        if tz3<z3size %count 5 of 10
            flag=[flag;2];
            tz3=tz3+1;
        else
            flag=[flag;0];
            tz1=0;tz2=0;tz3=0;
        end
    end
end
end
end

t2=[tdata;zeros(length(flag)-length(tdata),1)];
x=1:length(flag);
t3=[t(:,1);zeros(1,1)];

figure1 = figure;
axes1 = axes('Parent',figure1);
hold(axes1,'on');
plot1 = plot(x,flag,x,t2,x,t3,'Parent',axes1);
```

```
set(plot1(1),'LineWidth',1.5);
set(plot1(3),'Visible','off','LineWidth',2);
xlabel('Sample','Interpreter','latex');
ylabel('[delta]','Interpreter','latex');
title('Detection','Interpreter','latex');
xlim(axes1,[0 length(x)]);
ylim(axes1,[-15 15]);
box(axes1,'on');
set(axes1,'XGrid','on','YGrid','on','YMinorTick','on','YTick',...
    [-15 -10 -5 0 1 2 5 10 15]);
```

```
flag=100*flag;%scale to view
```

```
figure1 = figure;
axes1 = axes('Parent',figure1);
hold(axes1,'on');
plot1 = plot(x,flag,x,t2,x,t3,'Parent',axes1);
set(plot1(1),'LineWidth',1.5);
set(plot1(2),'Visible','off','LineWidth',1.5);
set(plot1(3),'Visible','on','LineWidth',2);
xlabel('Sample','Interpreter','latex');
title('Detection','Interpreter','latex');
xlim(axes1,[0 length(x)]);
box(axes1,'on');
set(axes1,'XGrid','on','YGrid','on','YMinorTick','on');
```

B.2 B.2 Sensor Doppler

B.2.1 B.2.1 Captura de velocidad en Arduino

El siguiente código fue desarrollado para capturar las velocidades:

```
#define PIN_NUMBER 4
#define AVERAGE 4
unsigned int doppler_div=44;
```

```
unsigned int samples[AVERAGE];
unsigned int x;

void setup(){
    Serial.begin(115200);
    pinMode(PIN_NUMBER,INPUT);
}

void loop(){
    noInterrupts();
    pulseIn(PIN_NUMBER,HIGH);
    unsigned int pulse_length=0;
    for (x=0;x<AVERAGE;x++){
        pulse_length=pulseIn(PIN_NUMBER,HIGH);
        pulse_length+=pulseIn(PIN_NUMBER,LOW);
        samples[x]=pulse_length;
    }
    interrupts();

    // Check for consistency
    bool samples_ok=true;
    unsigned int nbPulsesTime=samples[0];
    for (x=1;x<AVERAGE;x++){
        nbPulsesTime+=samples[x];
        if ((samples[x]>samples[0]*2)|| (samples[x]<samples[0]/2)){
            samples_ok=false;
        }
    }

    if(samples_ok){
        unsigned int Ttime=nbPulsesTime/AVERAGE;
        unsigned int Freq=1000000/Ttime;
        Serial.print("\r\n");
        Serial.print(Freq);
    }
}
```

```
    Serial.print("Hz : ");  
    Serial.print(Freq/doppler_div);  
    Serial.print("km/h\r\n");  
}  
else{  
    Serial.print(".");  
}  
}
```