



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA CIVIL

MICROSIMULACIÓN DE LA OPERACIÓN DE VEHÍCULOS DE EMERGENCIA

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE LA
INGENIERÍA, MENCIÓN TRANSPORTE

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL

BRUNO AMADEO STEFONI ESCUDERO

PROFESOR GUÍA:
CRISTIÁN EDUARDO CORTÉS CARRILLO

MIEMBROS DE LA COMISIÓN:
FERNANDO ORDOÑEZ PIZARRO
MAURICIO REPETTO CONTRERAS
ALEJANDRO TIRACHINI HERNÁNDEZ

SANTIAGO DE CHILE
2018

RESUMEN DE LA TESIS PARA OPTAR
AL GRADO DE MAGÍSTER EN CIENCIAS DE LA INGENIERÍA, MENCIÓN TRANSPORTE
Y AL TÍTULO DE INGENIERO CIVIL
POR: BRUNO AMADEO STEFONI ESCUDERO
FECHA: JULIO 2018
PROFESOR GUÍA: CRISTIÁN EDUARDO CORTÉS CARRILLO

MICROSIMULACIÓN DE LA OPERACIÓN DE VEHÍCULOS DE EMERGENCIA

El ruteo y modelación de la operación de vehículos de emergencia ha sido un tema de interés, aunque no muy presente en la literatura científica en el pasado. Los viajes involucrados se realizan con el propósito de atender urgencias y por ende es comprensible que se quieran realizar de la manera más rápida y eficiente posible. Para lograr esto, es necesario conocer el tiempo o costo asociado a cada calle que se recorre. Esto corresponde al desafío que actualmente presenta el sistema de ruteo que ocupa el Cuerpo de Bomberos de Santiago (CBS), dicho sistema se apoya en un algoritmo de rutas mínimas que se aplica sobre un grafo de calles de Santiago. Con el fin de estimar, corregir o validar los tiempos de viaje de dichas calles, tanto actuales como a futuro, es una buena idea el uso de un simulador microscópico de tráfico, herramienta que permite simular el tráfico real.

En este trabajo de tesis se desarrolla una API en C++ que modifica un software de simulación de forma que pueda reproducir el comportamiento anómalo que presentan vehículos comunes al enfrentarse a un vehículo de emergencia, dicho comportamiento está documentado en la literatura y se puede observar en terreno. Éste consiste en cruzar intersecciones sin tener la preferencia, usar pistas exclusivas o vías solo bus, forzar maniobras de cambio de pista a vehículos cercanos u obligar a éstos a cruzar con luz roja.

Adicionalmente, se desarrollan técnicas de minería de datos para procesar los datos GPS registrados por distintos vehículos del CBS con el fin de alimentar el microsimulador y para mejorar las estimaciones de tiempos de viaje en el sistema de ruteo que ocupa el Cuerpo de Bomberos actualmente. En paralelo, se codifica una red real del centro histórico de Santiago en el Software de Microsimulación PARAMICS para usarse a futuro en la estimación de tiempos de respuesta del CBS en la red de calles de esta zona.

Por último, se propone y aplica una metodología de estimación y calibración de parámetros para simular un viaje hacia una emergencia real. Se ocupan videos capturados por cámaras de la UOCT en distintas intersecciones como información de terreno, que se complementa con el registro de pulsos GPS y videos tomados desde el carro bomba que efectúa la expedición que se pretende simular. Como resultado, se calculan los flujos de la red codificada de manera de maximizar la bondad de ajuste, cuantificada mediante el error cuadrático de la distancia recorrida en el tiempo (entre la simulación y los pulsos GPS). De esta forma se obtiene un error porcentual promedio de 16.3 %.

Un objetivo a mediano plazo de este trabajo es producir un impacto en la evaluación de efectos sobre los tiempos de respuestas de los vehículos del Cuerpo de Bomberos bajo distintas modificaciones potenciales en la red tales como: cierre de calles, mejora de veredas e implementación de nuevas vías para buses de transporte público.

a mis padres, Jeannette y Francisco

Agradecimientos

“Si he logrado ver más lejos, ha sido porque he subido a hombros de gigantes”

En primer lugar, deseo expresar mi infinito agradecimiento hacia mi familia: papá; mamá; mis hermanos Adio y Valeria; y mis abuelos Elba y Guillermo. Gracias por criarme, apoyarme y acompañarme en todo momento. Aprovecho de mencionar también: a mis tíos Quique, Paty, Adio y Victoria; y mis primos Martina, Atio y Maximiliano quienes también me han acompañado constantemente a través de los años.

Adicionalmente, quiero dar mi reconocimiento hacia las persona que ayudaron a llevar a cabo este trabajo de tesis. Fue indispensable el apoyo técnico de Raúl Espinoza, el apoyo administrativo de Mónica Chávez, la ayuda de los voluntarios del cuerpo de Bomberos de Santiago que me facilitaron sus datos y se reunieron conmigo, a las personas de la UOCT que tuvieron excelente disposición a la hora de compartir sus datos para este trabajo. Además, me gustaría expresar mi gratitud por todo su apoyo y enseñanzas a mis profesores, amigos y compañeros que conocí a lo largo de la carrera y en mi época escolar.

Agradezco a mi profesor guía Cristián Cortes, por acompañarme en este extenso proceso de formación que es el trabajo de tesis. Gracias también a la comisión de profesores por revisar este trabajo y aportar con sus comentarios y observaciones constructivas.

Finalmente, mencionar que tanto la consolidación de la base de datos, su preprocesamiento y el desarrollo de los algoritmos de esta tesis fueron en parte gracias a las herramientas analíticas y de visualización presente en el software R (R Core Team, 2017) con su IDE RStudio, disponibles en internet de forma gratuita. En particular se ocuparon los paquetes: geosphere, mapproj, maps, rgl, sp, rgdal, ggmap, ggplot2 y leaflet.

Tabla de Contenido

Índice de Tablas	vii
Índice de Ilustraciones	viii
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	3
1.2.1. Objetivos Generales	3
1.2.2. Objetivos Específicos	3
1.3. Resultados Esperados	3
1.4. Estructura de la tesis	4
2. Revisión Bibliográfica	5
2.1. Simulación microscópica de tráfico	5
2.2. Vehículos de Emergencia en el tráfico y como parte de los MTS	6
2.2.1. Vehículos de Emergencia como Autómatas Celulares	8
2.2.2. Simulación de Vehículos de Emergencia con estocasticidad	8
2.2.3. Propiedades físicas de los Vehículos de Emergencia	9
2.3. Heterogeneidad en la población de conductores	10
2.4. Software de microsimulación de tráfico	11
2.4.1. Modelos de Seguimiento Vehicular	11
2.4.2. El microsimulador de tráfico PARAMICS	13
2.4.3. Modelos de Seguimiento Vehicular: Estado del arte	15
2.5. Estudios de Comparación de microsimuladores de tráfico	17
2.6. Calibración y Validación de MTS	18
2.7. Síntesis	24
3. Interfaz de Programación de Aplicaciones para Microsimulación de Vehículos de Emergencia	25
3.1. Material Audiovisual del CBS	26
3.2. Características deseadas en el tráfico simulado	27
3.3. Modelo de cambio de pista para EV (ELCM)	28
3.4. Modelo de acera para EV (ESM)	30
3.5. Modelo de intersección para EV (EIM)	32
3.6. Parámetros y valores predeterminados	33
3.7. Algoritmos especiales para semáforos	34
3.7.1. Ejemplo numérico: simulación de intersección semaforizada	37

3.8. Síntesis de la API	39
4. Preparación de red de simulación, recopilación de datos y procesamiento de señales GPS	40
4.1. Datos para red de simulación	40
4.2. Codificación de Zona Centro de Santiago	44
4.3. Preparaciones para calibración y trabajo futuro	48
4.4. Metodología para procesamiento de datos GPS	49
4.5. Preprocesamiento	52
4.6. Algoritmos de <i>Clustering</i>	59
4.7. Cruce con actos y filtro según fechas de interés	65
4.8. Síntesis de procesamiento de datos GPS	66
5. Calibración en un caso ejemplo	68
5.1. Uso de videos para estimación de MRT y MTH	70
5.1.1. Resultados Mean Reaction Time y Mean Target Headway	72
5.2. Estimación de parámetros para API	74
5.3. Calibración de la red	76
6. Conclusión	82
6.1. Resumen y motivación del trabajo	82
6.2. Resultados y contribuciones	82
6.3. Limitaciones y trabajo futuro	84
Glosario	85
Bibliografía	88

Índice de Tablas

2.1.	Reacciones en distintas zonas del diagrama $h - \Delta V$	13
2.2.	Valores de parámetros calibrados en distintos estudios ocupando PARAMICS. Fuentes: Ratrouf et al. (2015); Reza et al. (2016)	22
3.1.	Videos CBS	26
3.2.	Programación de semáforo en ejemplo de figura 3.6	38
4.1.	Fuentes de información para cada elemento de la red de simulación	42
4.2.	Código de Vehículos CBS	50
4.3.	Cantidad de datos y rutas después de cada etapa de procesamiento	67
5.1.	Videos UOCT de Intersecciones	70
5.2.	Resultados MRT Pio Nono	72
5.3.	Resultados MTH Recoleta	73
5.4.	Incidentes identificados en video vehículo Q8 en ruta a emergencia	75
5.5.	Indicadores RMSP y correlación de escenarios simulados	79

Índice de Ilustraciones

1.1. Área de servicio del CBS y la posición de sus Compañías	2
2.1. Diagrama de Fase con umbrales de acción CFM PARAMICS (Duncan, 1997)	13
2.2. Histograma de variables <i>Aggressiveness</i> y <i>Awareness</i> en PARAMICS	14
3.1. Definiciones básicas en esquemas de arcos mostrados más adelante	28
3.2. Modelo cambio de pista en arcos: vehículos i y j están alerta	29
3.3. Modelo para aceras: vehículo i se encuentra alerta	31
3.4. Modelo para aceras: vehículo i vuelve a la normalidad, vehículos j y k aún no	31
3.5. Modelo para intersecciones: vehículo m cruza sin prioridad	32
3.6. Ejemplo algoritmo de semáforos	37
4.1. Fotos satelitales de Santiago, en rojo se muestra límites del área de interés .	41
4.2. Velocidad promedio por arco en centro de Santiago entre las 7:30 y 9:30 AM	43
4.3. Arcos según su tipo en centro de Santiago Centro	44
4.4. Interfaz Gráfica de Usuario de PARAMICS (Modeller) con capa de Bing Maps	46
4.5. Histograma del largo de arcos codificados	47
4.6. Red final del centro de Santiago codificada en PARAMICS	48
4.7. Datos por Vehículo	51
4.8. Datos por media-hora del día	51
4.9. Pulsos GPS por Vehículo	52
4.10. Histogramas de variable dT con valores entre 0 y 30	54
4.11. Valores de variable V_{media} en distintos meses	55
4.12. Gráfico de dispersión Velocidad media vs instantánea	56
4.13. Ruta recorre distancia (línea azul) en muy poco tiempo	56
4.14. Ruta pasa por túnel San Cristóbal, interrumpiendo registro GPS	57
4.15. Datos no se registran al estar en una carretera subterránea	58
4.16. Registro con máxima V_{media} a 175 km/h, pulso color rojo oscuro	58
4.17. Nube de Pulsos GPS a la mitad y al extremo de las ruta	59
4.18. Algoritmo para clusterizar extremos, en naranja el último dato agregado . .	61
4.19. Cubo con ruta de ejemplo anterior. En verde se destaca cluster encontrado .	61
4.20. Ejemplo Algoritmo <i>Snake Clustering</i> con parámetro $\sigma = 2$	64
4.21. Ruta de figura 4.17 antes y después de filtrar datos	65
4.22. Ejemplo de fin de rutas (marcadores naranjos) asignadas a acto (marcador rojo)	66
5.1. Ruta máquina Q8 - 15:18 lunes 19 de junio 2017	69
5.2. Ruta Q8 codificada en PARAMICS	69

5.3. Vista cámara UOCT: acceso Puente Pio Nono con Santa María hacia el sur .	70
5.4. Vista cámara UOCT: Santa María con Recoleta hacia el oriente	71
5.5. Vista cámara UOCT: Recoleta con Santa María hacia el sur	71
5.6. Vista cámara UOCT: Recoleta con Santa María hacia el norte	71
5.7. Diagrama de caja con datos de <i>headways</i> y tiempo de reacción	73
5.8. Dimensiones calle Santa María	74
5.9. Detención de vehículos (borde derecho) con luz verde en intersección semaforizada	76
5.10. Cruce de vehículo (borde derecho) con luz roja en intersección semaforizada .	76
5.11. Distancia acumulada v/s tiempo registrado por pulsos GPS de carro Q8 . . .	77
5.12. Perfil Distancia acumulada: GPS y simulaciones especiales	77
5.13. Perfil Distancia acumulada: GPS y escenario base (promedio y cada corrida)	78
5.14. Perfil Distancia acumulada: GPS y todos los escenarios	80
5.15. Perfil Distancia acumulada: GPS y escenario D_{125} (promedio y cada corrida)	81

Capítulo 1

Introducción

1.1. Motivación

El ruteo y modelación de la operación de vehículos de emergencia ha sido un tema de interés, aunque no muy presente en la literatura científica en el pasado. Los viajes involucrados se realizan con el propósito de atender urgencias y por ende es comprensible que se quieran realizar de la manera más rápida y eficiente posible. Una variable que siempre se buscará mejorar es el *tiempo de respuesta*, el cual se puede definir como el tiempo transcurrido entre la alerta y la llegada del vehículo a atender una emergencia. Entre las instituciones que hacen el trabajo de atender emergencias en Chile se encuentran Carabineros de Chile, Socorro Andino y Bomberos. En el caso de la ciudad de Santiago, el Cuerpo de Bomberos de Santiago (CBS) está encargado de prestar servicios de bomberos en 9 comunas de la ciudad¹ (ver Figura 1.1). El CBS es el cuerpo de bomberos más grande de Chile con respecto a número de Compañías al tener un total de 22 de ellas².

En trabajos previos entre la Universidad de Chile y el CBS se creó un software que entrega las rutas que minimizan el tiempo de respuesta llamado COMMANDER (Echeverría, 2013). Antes de este trabajo, las compañías del Cuerpo de Bomberos de Santiago se repartían las emergencias según la zona de ocurrencia. Esto es, un sistema de división territorial, el cual determinaba qué compañía se encargaba de una emergencia a partir de su ubicación, sin importar si la compañía asignada era la más cercana o si contaba con los vehículos necesarios disponibles. En el trabajo descrito en la tesis de Echeverría (2013) se usó una metodología a partir de los GPS de Buses Transantiago para estimar velocidades de vehículos de bomberos en las distintas calles de Santiago, logrando de esta forma alimentar el programa con un grafo de velocidades para la ciudad de Santiago.

A partir de diciembre del año 2015 aparece la oportunidad de realizar una mejora al software, dado que las distintas compañías del CBS instalaron aparatos que emiten pulsos GPS en algunos de sus vehículos. De esta forma, existe la posibilidad de alimentar COMMANDER

¹<http://www.cbs.cl/Institucion/Estructura> visitado el 9 de noviembre 2016

²https://es.wikipedia.org/wiki/Cuerpo_de_Bomberos_de_Santiago visitado el 9 de noviembre 2016

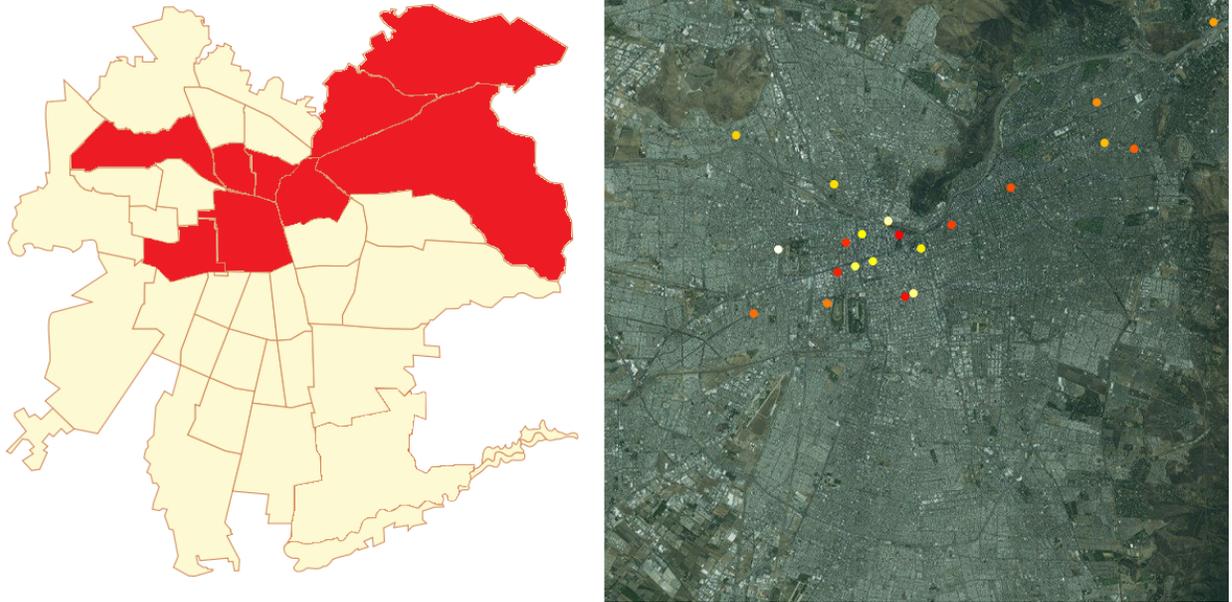


Figura 1.1: Área de servicio del CBS y la posición de sus Compañías

con un grafo de velocidades provenientes de datos GPS de vehículos de emergencia en operación, complementando la información proveniente de buses Transantiago utilizada en la actualidad.

Adicionalmente, surge la idea de utilizar videos de cámaras instaladas en los vehículos del CBS para lograr identificar el comportamiento extraordinario provocado por los vehículos de emergencia sobre el tráfico, de manera de modelarlo e incorporarlo en un Simulador Microscópico de Tráfico. Un Simulador Microscópico de Tráfico (MTS por *Microscopic Traffic Simulator*) es una herramienta computacional ocupada para simular situaciones de alta complejidad analítica, con el objeto de realizar alguna evaluación o predicción sobre el estado del transporte motorizado, tanto en carreteras como en vías urbanas. Las entidades simuladas en los distintos estudios que usan Microsimuladores de Tráfico son vehículos privados (POV por *Privately Owned Vehicle*) y buses de transporte público; incluso en algunos casos también se incorporan peatones con el propósito de simular intersecciones con alto flujo peatonal y de bicicletas.

En la literatura es posible encontrar estudios que incorporan vehículos de emergencia (e.g., carros bomba o ambulancias) en la simulación de tráfico. En general, esto se hace con tres propósitos: evaluar estrategias de evacuación ante catástrofes; analizar esquemas de control de semáforos para facilitar el paso de vehículos de emergencia; estimar el tiempo de respuesta de servicios de emergencia.

En esta tesis se pretende simular vehículos de emergencia (EV por *Emergency Vehicles*) con el fin de estimar sus tiempos de respuesta. Sin embargo, una vez que se incorporen al simulador, éste también se puede usar para otros propósitos. Además, gracias a los datos GPS mencionados anteriormente, será posible calibrar los parámetros del Microsimulador de Tráfico de manera que la simulación se asemeje lo más posible a lo que ocurre en la realidad ante la presencia de uno o más EVs. Con un simulador calibrado se hace posible la evaluación

de proyectos que afecten la operación de servicios de emergencia, por ejemplo, ante cambios estructurales en la red de calles.

1.2. Objetivos

1.2.1. Objetivos Generales

El objetivo principal de esta tesis es incorporar en un Simulador Microscópico de Tráfico el ruteo y operación de vehículos de emergencia sobre una zona urbana. En paralelo, se pretende detallar cómo codificar una red real compleja y de qué forma preparar los datos para alimentar una simulación de esta envergadura. Por último, se busca proponer un marco de calibración para estimar los distintos parámetros de la simulación y calibrar las variables de interés, de forma que las simulaciones sean lo más semejante posible a la realidad.

1.2.2. Objetivos Específicos

1. Programar y diseñar una API externa en C++ a usarse con el microsimulador PARAMICS para poder simular de forma realista las características especiales que tiene un EV al interactuar con otros vehículos en el espacio urbano de una ciudad.
2. Realizar minería de datos sobre la *raw data* de pulsos GPS de los vehículos del CBS de manera de preprocesarlos para su uso posterior tanto en COMMANDER como para la calibración del simulador.
3. Plantear una metodología para calibrar un simulador a partir de las bases de datos disponibles.
4. Codificar una red de una zona de Santiago en el software comercial PARAMICS, de forma que en el futuro sea posible simular el tráfico sobre esta red, incluyendo vehículos de emergencia.
5. Proponer y aplicar una metodología de calibración en la simulación de un expedición real hacia una emergencia.

1.3. Resultados Esperados

En esta tesis se busca contar con una red real codificada en PARAMICS junto a una metodología robusta de preprocesamiento de datos enfocada a la calibración de esta red. Además, se espera tener una API para PARAMICS que puede usarse en cualquier otra red donde se desee simular vehículos de emergencia de distinto tipo. Con lo anterior se espera crear una herramienta con el poder de evaluar proyectos y estimar escenarios futuros que podrían afectar los tiempos de respuesta de los servicios de emergencia de la ciudad.

1.4. Estructura de la tesis

A continuación, en el segundo capítulo de esta tesis se procede a realizar una revisión bibliográfica abarcando temas variados atinentes a este trabajo: microsimulación de transporte urbano en general; simulación de vehículos de emergencia; heterogeneidad en la población de conductores; literatura sobre los distintos microsimuladores de tráfico y sus diferencias y metodologías de calibración y validación para microsimuladores de tráfico.

Luego, en el capítulo tres se describe el desarrollo de una API en C++ para la microsimulación de Vehículos de Emergencia en el simulador comercial PARAMICS con el fin de incorporar comportamientos especiales en caso de presencia de EV en el tráfico. Para esto, se usará la librería de funciones que viene incorporada en el software PARAMICS, permitiendo manipular los modelos de Seguimiento Vehicular y Cambio de Pista.

En el cuarto capítulo se detalla la codificación de una red real del centro de Santiago en el microsimulador PARAMICS. Se muestran las distintas fuentes de información ocupadas para conseguir todos los datos necesarios junto con los métodos utilizados para traducir la información a la red codificada. Adicionalmente, en el mismo capítulo, se propone una metodología para el preprocesamiento de datos GPS de vehículos del CBS. Con la metodología se pretende preparar la data con dos propósitos: que pueda ser analizada y utilizada en conjunto con el grupo de analistas encargados de perfeccionar las estimaciones del software COMMANDER; y que sea posible ocupar los datos para la calibración de los modelos de simulación de vehículos de emergencia detallados en el capítulo anterior.

El quinto capítulo se enfoca en la calibración de un caso ejemplo de un viaje hacia una emergencia registrado por el CBS. De esta manera, se pretende proponer un marco de calibración para que la simulación se asemeje a la realidad. Para finalizar, en el último capítulo de esta tesis se presentan las conclusiones y se analizan líneas futuras de investigación.

Capítulo 2

Revisión Bibliográfica

2.1. Simulación microscópica de tráfico

Los microsimuladores de tráfico han sido utilizados en la ingeniería de transporte en las últimas dos décadas para realizar evaluación social de proyectos de transporte o de planificación urbana. Últimamente, estas herramientas también se han ocupado para estimar la emisión de gases de efectos invernadero debido al flujo de vehículos motorizados, e.g., Guo y Zhang (2014); Abou-Senna et al. (2013); Abou-Senna y Radwan (2013); Wang et al. (2016). En particular, para esta tesis son de interés aquellos estudios ocupando MTS que simulan vehículos de emergencia. En la literatura, gran parte de estos estudios consisten en evaluaciones de estrategias de evacuación, sea por catástrofes naturales o ataques terroristas, tanto en un contexto urbano como en carreteras interurbanas. Como ejemplo, Chen y Zhan (2008) comparan esquemas de evacuación en una zona poblada usando el MTS PARAMICS. Algo parecido se hace en Sisiopiku (2007) ocupando el MTS CORSIM. En esta misma línea están los trabajos de Song et al. (2010); VanLandegen y Chen (2012); Yu et al. (2014) donde en los últimos dos se ocupa el MTS VISSIM.

Un segundo caso de incorporación de EV al tráfico simulado son estudios de *Emergency Vehicle Preemption* (EVP), que consiste en sistemas de semáforos actuados o programación dinámica de semáforos. Estos sistemas asignan una fase de verde a vehículos de emergencia aproximándose a una intersección semaforizada, interrumpiendo de esta forma la operación regular del semáforo para dar paso al EV de manera preferencial. En el estudio de Yun et al. (2012) se simula un corredor urbano de cuatro intersecciones semaforizadas, ocupando VISSIM, para comparar distintos algoritmos de EVP y sus efectos sobre el tráfico, los autores concluyen que se pueden aumentar el tiempo de viaje hasta un 28 % para los vehículos transitando la avenida simulada, por lo que recomiendan a los ingenieros de tránsito decidir bien el método de transición para volver el semáforo a la normalidad. Kwon y Kim (2003) evalúan un sistema de EVP para el *Minnesota Department of Transportation*, el cual considera la integración con un sistema de ruteo de emergencia, donde se predefine una ruta para el EV y luego se coordinan los semáforos en el camino para dar la fase preferencial al EV de manera eficiente, como resultado se encuentra una disminución en los tiempos de respuesta de los servicios de emergencia entre 10 % y 16 %.

En los estudios mencionados anteriormente, es posible analizar el efecto en distintas variables gracias al uso de MTS. Entre estas variables se encuentran: las emisiones contaminantes, largos de cola, demoras de vehículos privados y tiempos de respuesta de EV. Sin embargo, a pesar de incorporar vehículos de emergencia dentro de sus simulaciones, los trabajos citados previamente no le asignan un trato especial a los EVs, ni tampoco incorporan reacciones o estímulos sobre los vehículos particulares alrededor del EV en el simulador. Estas conductas sí se observan en la realidad.

2.2. Vehículos de Emergencia en el tráfico y como parte de los MTS

Buchenscheit et al. (2009) observan 21 videos de EVs en ruta en la ciudad Ulm de Alemania, lo que equivale a 147 minutos de material, y concluyen que los EV cruzan las intersecciones semaforizadas aunque tengan luz roja. También afirman que en algunos casos, cuando se observa que el EV se acerca a una intersección con mucha congestión, los vehículos alrededor forman un “corredor” para dejarlo pasar entre las colas que se forman de automóviles. Adicionalmente, los autores calculan que, en promedio sobre las rutas observadas, 2.5 vehículos reaccionan de mala forma en cada ruta. Esto debido a que no cooperan o entran en pánico en el instante que se les acerca un EV, aumentando su tiempo de respuesta. Weinert y Düring (2015) analizan 90 minutos de videos de EV en ruta hacia una emergencia. En este material se observa que, de las 116 intersecciones semaforizadas, en el 48 % el EV cruza con luz roja. Además, se tiene que el 25 % de los POV reaccionan de alguna manera (i.e., acelerando o cambiando de pista) ante la presencia de un EV cuando se encuentran relativamente lejos de él (a más de 50 metros de distancia), mientras que el 50 % reaccionan entre los 50 y 20 metros y el restante 25 % reaccionaron cuando el EV ya estaba cerca, a menos de 20 metros de distancia.

En pocos casos se aprecia en la literatura un trato especial a los EV dentro de una simulación de tráfico. El caso más reciente es en Jordan et al. (2013) donde se deduce con una fórmula analítica el punto óptimo donde se debería detener un POV en una calle de dos pistas, de forma de ceder el paso a un EV en la pista contigua. Para esto, se usa la teoría de *Shockwaves* de Lighthill-Whitman-Richards (Lighthill y Whitham, 1955; Richards, 1956) y se microsimula usando VISSIM para comprobar la veracidad del cálculo. En el simulador se deja que los EV activen fases de emergencia en semáforos y provoquen el inicio de maniobras de cambio de pista en los POV alrededor.

Otro ejemplo de investigación con un trato especial a los EV en un MTS es Zhang et al. (2009). En este estudio se busca simular el paso de un EV por una carretera. Los autores realizaron estudios previos para carreteras usando CORSIM, pero en este caso se decide incorporar características especiales a los EV en la simulación y ver cuánto afectan en la estimación del tiempo de respuesta. En paralelo, también se compara los efectos en la disminución de tiempos de viaje estimados al dar trato especial a los EV vs un sistema de semáforos con EVP. Las nuevas acciones especiales permitidas a los EV simulados son: moverse por la berma, cruzar medianas, cruzar intersecciones (semaforizadas o no) sin tener prioridad, realizar

virajes no permitidos y moverse contra el sentido del tránsito. Además, se incorporan en la simulación interacciones EV-POV, lo que permite emular el efecto que se observa en la realidad, donde los POV cooperan con los EV al variar su velocidad o cambiarse de pista para facilitar su paso. Finalmente, con las reglas incorporadas a CORSIM mediante una API, se simula la autopista “Route 7” en torno a Washington DC. Al analizar los resultados, se encuentra una reducción en el tiempo de viaje de los EVs variando entre 19 % y 75 % al darles un trato especial. Por otro lado, el escenario con EVP implica una reducción entre 3 % y 34 % en los tiempos de viaje estimados por el simulador. Como conclusión, los autores sugieren que usar solamente reglas especiales para los EVs resulta en un tiempo de viaje estimado significativamente menor, en general, al que resulta de incluir un sistema de EVP sin trato especial a los EVs transitando.

En Yoo et al. (2010), usando el software de simulación IMAGES (Intelligent Multi-Agent System), se comparan distintas estrategias de cooperación POV-EV para una carretera con un sistema de transporte inteligente (ITS por *Intelligent Transportation System*) alimentado por una red de sensores ubicuos. Un ITS es un esquema de comunicación y coordinación automatizada entre vehículos en movimiento con el fin de optimizar la fluidez del tráfico al sincronizar las maniobras de sus agente, disminuyendo simultáneamente la tasa de accidentes. Los escenarios evaluados en la investigación son *No strategy vs Full Reservation vs Hop Reservation*. Full Reservation consiste en forzar sobre los POV un cambio de pista siempre, de manera de cederle el paso al EV, i.e., todos salen de la pista actual del EV. Hop Reservation, obliga al cambio de pista para formar un corredor dependiendo de la velocidad del vehículo aguas abajo del EV en cada pista. Se encuentra que ambas estrategias presentan un buen rendimiento, mejorando el tiempo de respuesta del EV en una magnitud similar al escenario sin estrategia, todo esto para distintos niveles de congestión. Sin embargo, se observa que la densidad vehicular aumenta considerablemente con el tiempo en la estrategia Full Reservation, no así en la estrategia Hop Reservation, la cual presenta el mismo comportamiento de la densidad a lo largo del tiempo que el caso base sin estrategia.

En Toy et al. (2002), se simula la operación del paso de EV por una carretera de 2 pistas con un sistema de “carretera automatizada” (AHS por *Automated Highway System*) similar al ITS de Yoo et al. (2010). Éste consiste en una carretera con vehículos autónomos conectados a un mismo controlador externo, que mediante algoritmos predefinidos, maneja el tráfico para maximizar el flujo de vehículos. Para facilitar el paso de un EV a través del AHS, los autores proponen distintos patrones de control vehicular, que implican maniobras grupales simultáneas del tráfico presente, coordinadas por el controlador externo de la autopista (AHS). Entre las maniobras están: *Vortex* consistiendo en un cambio de pista en pelotón; *Part-and-go*, donde en caso de haber tráfico detenido, los vehículos usan reversa para dar espacios al EV; *Zig-zag*, similar a Vortex, pero con pelotones pequeños y donde la pista libre se va alternando. El simulador ocupado “SmartAHS” es programado por los mismos autores en el lenguaje de programación SHIFT. Se decide modelar el tráfico vehicular como un fluido al no haber interacciones humanas, esto debido a que todos los vehículos se rigen por algoritmos propios y por el controlador externo del AHS.

Weinert y Düring (2015), citados anteriormente por sus aportes respecto del material audiovisual, escogen incorporar los fenómenos conductuales que identifican al simulador SUMO. El objetivo de su estudio es simular un ITS con tecnología de comunicación entre vehículos

y entre vehículo e infraestructura. Al igual que en otros artículos citados anteriormente, se deja que los EV crucen aunque su acceso esté con luz roja, fuerzan a los POV aguas abajo del tráfico a cambiarse de pista y además se les permite romper los límites de velocidad legales definidos dentro del simulador.

2.2.1. Vehículos de Emergencia como Autómatas Celulares

Como parte del área del conocimiento de la Física, se han realizado algunos aportes al estudio de la operación de vehículos de emergencia. Esto se debe a que los físicos se han interesado por los Autómatas Celulares (CA por *Cellular Automata*), quienes son entes virtuales que se rigen por reglas de evolución sencillas. El ejemplo más famoso es el Juego de la Vida de Conway (Gardner, 1970); este juego consiste en una grilla donde las casillas pueden estar vivas (de color negro) o muertas (de color blanco); las reglas del juego son condiciones sencillas para determinar la evolución de la vida en la grilla entre un instante discreto y el siguiente¹. Como parte de esta tesis, son relevantes los estudios con CA que emulan el tráfico vehicular incluyendo el paso de EVs, tal como en Moussa (2009), quien propone probar cuanto afecta la proporción de vehículos privados cooperando y la distancia en que éstos cooperan para ceder el paso a un EV. Para esto, genera un modelo discreto de tráfico con CA y simula una carretera de dos pistas. Con esto se encuentra que a mayor cantidad de vehículos cooperando y a mayor distancia de cooperación menor es el tiempo de respuesta del EV. Además, se observa que las maniobras especiales no afectan significativamente los tiempos de viaje de los POV. Por último, a diferencia del trabajo previo del mismo autor (Moussa, 2008), se permite que el EV cambie de pista en caso de mucha congestión en la pista actual, implicando una mejora considerable en el tiempo de respuesta.

Inspirado en el trabajo de Moussa, en Zhao et al. (2013) también se simula tráfico con EVs usando CA con reglas muy similares, pero se argumenta que los POV no perciben la presencia de un EV infinitamente hacia atrás como supuso Moussa, ni tampoco ocurre que el conductor del vehículo de emergencia tiene percepción infinita de la congestión en las pistas aguas abajo. En consecuencia, se propone un “largo de sirena”, distancia bajo la cual los vehículos interactúan. Los autores agregan una segunda probabilidad de cooperación, más alta que la primera, aplicada a los vehículos justo en frente del EV.

2.2.2. Simulación de Vehículos de Emergencia con estocasticidad

Aunque las investigaciones mencionadas incorporan comportamientos e interacciones especiales a los EVs, la gran mayoría utilizan un enfoque determinista al simular autos autónomos con algoritmos y controladores externos deterministas. Como excepción a lo anterior están Zhang et al. (2009) y los trabajos con autómatas celulares. No obstante, estos últimos no tienen modelos de seguimiento vehicular muy complejos; la carretera se modela como una grilla de $2 \times N$ y los vehículos en ésta avanzan a tasa constante en el espacio longitudinal discretizado. Más aún, no hay intersecciones ni semáforos o variaciones en el tipo de con-

¹https://es.wikipedia.org/wiki/Juego_de_la_vida

ductor presente en la simulación. En cambio, en Zhang et al. (2009), se ocupa el modelo de seguimiento vehicular integrado en CORSIM y con la API construida ad-hoc se incorpora la estocasticidad. Los conductores perciben la presencia del EV en la vía según una probabilidad en función de la distancia entre ambos vehículos. Una vez que el conductor está consciente de la presencia del EV aguas arriba, éste decide si cooperar dependiendo de un “factor de cooperación” el que a su vez depende de la distancia. En caso de decidir ayudar al paso expedito del EV, se asigna agresividad máxima al POV. Cabe destacar que CORSIM requiere que la ruta del EV sea declarada en forma previa a la simulación, por lo que el comportamiento de elección de ruta es de carácter determinista en este estudio.

2.2.3. Propiedades físicas de los Vehículos de Emergencia

Desde hace varios años se han estudiado los efectos psicológicos y el comportamiento humano frente a vehículos de emergencia. En el estudio descrito en Caelli y Porter (1980) se reportan las condiciones que influyen en la percepción de una sirena por parte de conductores de vehículos privados. En particular, se realiza un experimento que busca identificar desde qué posición los conductores creen que proviene el sonido de sirena y hacia dónde creen que se moverá el vehículo emitiendo el sonido. Como resultado se observa que todos los conductores afirmaban con seguridad de donde creían que venía el sonido de la sirena cuando en muchos casos, debido a un fenómeno psicofísico conocido, se apunta a zonas simétricamente opuestas con un ángulo de error de 180° . Más aún, en casos de tener las ventanas del vehículo arriba y la radio sonando, se tiende a subestimar la distancia a la que se encuentra el EV. En paralelo, se reporta que en entrevistas a conductores de ambulancias, éstos aseguran que en el tráfico habitual hay una distancia de alerta máxima de 100 metros por parte de los conductores y que alrededor de la mitad de aquellos conductores con los vidrios arriba y escuchando la radio no se percatan de la presencia de las sirenas del EV sonando a la distancia.

Con respecto a la tecnología de las sirenas presente en los vehículos de emergencia, De Lorenzo y Eilers (1991) buscan hacer una revisión bibliográfica sobre sistemas de alerta de presencia de EV. En consecuencia, se analizan aspectos técnicos tales como la frecuencia de las luces, su color y el rango de alcance para alertar a otros conductores. Sobre esto último, los autores afirman que: “una sirena debe competir con el ruido generado en las calles, radios de autos y ventiladores; además debe sobreponerse a las técnicas modernas de aislamiento de ruido”. Finalmente, se cita un estudio del Departamento de Transporte de EE.UU. (Potter et al., 1977) donde se reporta que la efectividad de sirenas para alertar en intersecciones urbanas es a lo más entre 8 y 12 metros, lo cual no aumenta significativamente en el caso de intersecciones en suburbios o en entradas a carreteras. Aparte de lo anterior, se recomienda una velocidad de entrada a la intersección de 15 km/hr con el fin de maximizar la seguridad. Además, los autores afirman que los resultados de estos reportes se corroboraron con otras investigaciones (Jones et al., 1974; Corliss y Jones, 1976). Por último, los autores mencionan que los reportes hechos por el Departamento de Transporte estadounidense (DOT) concluyen que la sirena no es un mecanismo efectivo para alertar a otros conductores sobre la presencia de un vehículo de emergencia.

No obstante, aunque los reportes del DOT hayan determinado que alertar mediante sirenas no es eficiente, Ho y Casey (1998) realizan un estudio que encuentra un ahorro significativo en

los tiempos de viaje de ambulancias si en el trayecto se ocupan sirenas para alertar al resto de los vehículos. Para llegar a esta conclusión, se comparan 64 viajes realizados en ambulancias con el propósito de atender una emergencia en una red urbana durante un periodo de nueve meses. Al analizar los datos, se calcula un ahorro en el tiempo de viaje promedio de 38.5 % ($p < 0,1$) al comparar rutas que ocuparon la sirena vs aquellas que no.

2.3. Heterogeneidad en la población de conductores

Respecto de la variación en los conductores, en Drucker et al. (2013) se realiza un estudio sobre las condiciones presentes en los accidentes POV-POV y EV-POV a partir de dos bases de datos de distintas instituciones conteniendo un historial de 8 años de accidentes en EE.UU. A partir de un análisis estadístico usando una regresión logística multivariada se observa que los datos no sugieren que la edad o el género sean variables relevantes para estimar riesgo de choque con un EV. En cambio, se identifican como factores relevantes la iluminación, distracciones internas en el vehículo y condiciones geométricas de la calle influyendo el campo visual. Panwai y Dia (2005) revisan literatura relacionada a la variación entre conductores y mencionan la presencia de atributos que influyen en el comportamiento de seguimiento vehicular. En este trabajo los autores los dividen en 2 categorías: “Diferencias individuales” (i.e., edad y género) y “Diferencias en el ambiente” que se relacionan al estado del individuo al momento de conducir, e.g., clima, estado del camino, distracciones repentinas, estrés, fatiga. Además se encuentra que, en promedio, si el conductor es mayor de edad o mujer, preferirá mantener una mayor distancia al vehículo que se encuentra al frente que si fuera menor de 65 años o hombre (Evans y Wasielewski, 1983; Ranney, 1999).

En Ossen y Hoogendoorn (2011) se encuentra que existen diferencias significativas en el comportamiento de seguimiento vehicular de personas manejando vehículos livianos. Se identifican variaciones en el espaciamiento deseado con respecto al auto del frente para distintas velocidades. Adicionalmente, se encuentra que los conductores de camiones conducen con una velocidad más constante que los conductores de vehículos livianos. También se observa que los conductores de vehículos livianos tienden a desear un menor espaciamiento temporal al momento de seguir un camión a otro vehículo liviano. Los autores además aseguran que hay dos tipos de heterogeneidad en la población de conductores, aquella proveniente de distintos estilos de conducción, la cual hace referencia a las diferentes reglas mentales que ocupan diferentes personas al momento de manejar un vehículo motorizado (e.g. fijarse en diferentes estímulos para reaccionar ante el vehículo del frente). La otra clase de heterogeneidad viene dada dentro de cada estilo de manejo, donde existe una variación en la población con respecto a la sensibilidad de las reglas mentales ocupadas por los conductores. Esta última variación de la población es la que se incorpora en la mayoría de los modelos de seguimiento vehicular a discutirse más adelante. Finalmente, los autores afirman que en Ossen et al. (2006) se confirma la existencia de ambos tipos de heterogeneidad en el tráfico real.

Haque y Washington (2014) investigan el efecto del uso de teléfonos celulares en el nivel de distracción al manejar. Entre la literatura que revisan encuentran que, en un meta-análisis de estudios, se encuentra que hablar por teléfono mientras se maneja un vehículo aumenta el tiempo de reacción en 0,25 segundos en promedio. Además, hay estudios que encuentran un

mayor aumento en el tiempo de reacción al estar hablando por teléfono que cuando se maneja bajo la influencia del alcohol. Por último, hay una variación en el efecto que tiene usar el celular al manejar dependiendo de la edad de la persona. En conductores de mayor edad, el aumento en el tiempo de reacción es mayor que en personas jóvenes; la diferencia entre estos dos grupos varía entre 0.11 a 0.40 segundos dependiendo del estudio. En esta investigación se encuentra que es igual de perjudicial para el tiempo de reacción el uso de manos libres que hablar acercando el teléfono a la oreja. En consecuencia, se argumenta que la raíz del problema se encuentra en la carga cognitiva que significa hablar por teléfono, al desviarse la atención del camino con el fin de poder seguir la conversación telefónica. Lo más importante a destacar para esta tesis es la variación en los tiempos de reacción que significa el hecho que algunos conductores se encuentran distraídos por distintos motivos en el tráfico normal.

2.4. Software de microsimulación de tráfico

En cuanto a los microsimuladores de tráfico disponibles para la realización de estudios en transporte urbano, los que están presentes con mayor frecuencia en la literatura son: VISSIM (PTV Group, 2017); PARAMICS (Quadstone Paramics, 2017); CORSIM (McTrans Center, 2017); AIMSUN (TSS-Transport Simulation Systems, 2017) y SUMO (DLR Institute of Transportation Systems, 2017). Confirmando la intuición, en Mubasher y ul Qounain (2015) se hace un meta-análisis de los estudios de simulación de tráfico encontrados en la literatura por un algoritmo de búsqueda. Posterior al análisis se concluye que estos 5 simuladores son los que se usan con mayor frecuencia en el orden que fueron mencionados anteriormente, siendo VISSIM el más popular.

Existen varias diferencias entre estos 5 simuladores. Por ejemplo, la interfaz de usuario o la calidad gráfica (un análisis comparativo de estos aspectos se puede encontrar en Kotusevski y Hawick, 2009). Sin embargo, las diferencias más relevantes son aquellas relacionadas a los modelos detrás de la simulación. Los tres modelos más importantes detrás de toda simulación de tráfico son: Seguimiento vehicular (CFM por *Car-following Model*), Cambio de pista (LCM por *Lane-changing Model*) y Aceptación de brecha (GAM por *Gap-acceptance Model*). Existen distintas variables y parámetros para cada modelo, pero hay casos en que ocupan mismos parámetros para distintos modelos, e.g., ocupar la aceleración máxima en el CFM y el GAM.

2.4.1. Modelos de Seguimiento Vehicular

En la revisión efectuada en Brackstone y McDonald (1999) sobre los distintos modelos de seguimiento vehicular, se pueden apreciar las historias detrás de las ideas que llevaron a la creación de cada modelo. Los autores proponen 4 categorías de CFM: *Gazis-Herman-Rothery* (GHR), en esta categoría entran los CFM de General Motors; *Collision Avoidance*, aquí están los CFM de AIMSUN y CORSIM; *Linear* (caso particular de GHR) y *Psychophysical* (PARAMICS y VISSIM). Los autores mencionan al final que existe una quinta categoría dado que a cualquier CFM se le podría incorporar Fuzzy-Logic.

Es necesario conocer los CFM *Collision Avoidance* y *Psychophysical* para comparar los simuladores mencionados anteriormente. En particular, es de suma importancia saber la cantidad de parámetros y variables que toma en cuenta cada CFM. Los dos artículos más usados como ejemplo de estos dos modelos son Gipps (1981) y Fritzsche y Ag (1994) respectivamente. En el primer tipo de modelo, lo que se busca es que cada entidad o vehículo simulado logre evitar chocar con el vehículo del frente inclusive en situaciones de desaceleración máxima debido a imprevistos en el camino o accidentes entre vehículos. Para esto, se formulan restricciones sobre la velocidad de cada individuo simulado. Dichas condiciones reflejan la existencia de una velocidad deseada en caso de flujo libre e incorporan la voluntad de los conductores de evitar colisiones con el vehículo del frente. Estas restricciones sobre la velocidad $v_n(t)$ del vehículo n -ésimo provocan que el conductor desacelere o acelere en cada instante discreto de simulación dependiendo del vehículo del frente, tomando en cuenta tanto su posición $x_{n+1}(t)$ como su velocidad $v_{n+1}(t)$. Como parámetros de este tipo de CFM se tienen (Brockfeld et al., 2004):

- a Máxima aceleración que se desea efectuar.
- b Máxima desaceleración se desea efectuar.
- V_{max} Velocidad máxima a la que el conductor desea viajar.
- τ Tiempo de reacción de cada vehículo.
- L Largo efectivo de vehículos, incluye distancia de seguridad en colas de detención.

Si se desea comprender en mayor detalle este tipo de modelos, es recomendable el libro de Barceló et al. (2010), donde también se puede encontrar una descripción de AIMSUN y SUMO. Este mismo autor es uno de los creadores de AIMSUN (Barceló y Casas, 2005). En el caso de CORSIM, se ocupa el modelo de seguimiento vehicular PITT descrito en Elefteriadou (2014, cap. 7). Ambos CFM, tanto PITT como el de AIMSUN son variaciones del modelo de Gipps, usando la misma base teórica y supuestos.

Para la segunda categoría de CFM de interés, los de naturaleza psicofísica, se postula que cada vehículo tenga un espaciamiento temporal o *headway* deseado h . Este *headway* corresponde a los “segundos de distancia” entre las posiciones del vehículo de interés y aquel situado en frente, tomando en cuenta las velocidades de ambos (Fritzsche y Ag, 1994). Otro aspecto clave de estos modelos es que los entes simulados no modifican su comportamiento en cada instante discreto de simulación, sino que esperan a que se llegue a ciertos umbrales preestablecidos, considerando tanto el *headway* entre los 2 vehículos como la diferencia entre sus velocidades. Es por esto que Brackstone y McDonald (1999) denominan a este tipo de modelos “*action point models*”. En la Figura 2.1 es posible apreciar los distintos umbrales en el CFM de PARAMICS (Duncan, 1997), donde el eje x corresponde al *headway* calculado como $h = \frac{\Delta x}{\Delta V}$, es decir, el cociente entre la separación espacial y la diferencia en velocidad. Además, en el diagrama se representa la aceleración para cada par $(h, \Delta V)$ con un vector. La Tabla 2.1 explica qué ocurre al entrar en cada zona y cómo reacciona el vehículo.

Con respecto al simulador VISSIM, una descripción completa del simulador se puede encontrar en Fellendorf y Vortisch (2010). Su CFM se basa en los trabajos de Wiedemann (1974); Wiedemann y Reiter (1992), también de carácter psico-físico como el de PARAMICS. Las diferencias entre el modelo de Wiedemann y el modelo de VISSIM no han sido reveladas públicamente hasta la fecha.

Tabla 2.1: Reacciones en distintas zonas del diagrama $h - \Delta V$

Zona	Acción
A	Estamos muy cerca, disminuir headway frenando proporcional a ΔV
B	Nos estamos separando, acelerar
C	Nos estamos acercando, frenar de a poco
D	Estamos muy separados, acelerar
E	Estamos muy separados y vehículo del frente acelera, acelerar lo máximo posible

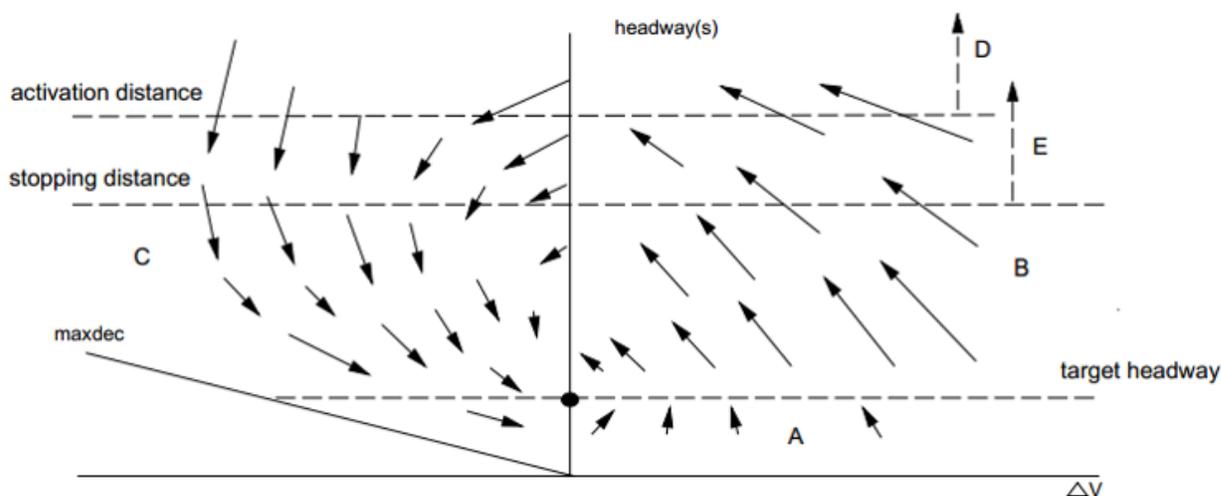


Figura 2.1: Diagrama de Fase con umbrales de acción CFM PARAMICS (Duncan, 1997)

2.4.2. El microsimulador de tráfico PARAMICS

En el caso particular del MTS PARAMICS, si bien por motivos comerciales no se entra en gran detalle sobre los modelos utilizados, en (Duncan, 1997) se describe de manera general el funcionamiento de algunos de ellos. En particular, en el reporte se habla sobre dos variables para modelar la variación entre conductores, fenómeno observado en terreno y en la literatura, tal como se explicó anteriormente. Para lograr esto, se incorporan factores que modifican h , siendo el valor promedio 1 segundo y pudiendo variar entre los 0,5 y 3 segundos. Las variables que modifican h para cada conductor se llaman *Awareness* y *Aggressiveness*; éstas son de carácter aleatorio y se generan al momento de crear cada entidad o vehículo. Para esta tesis se realizó un muestreo de 13.000 vehículos al generarlos en PARAMICS en una red de prueba para tres horas de simulación. Con esto, se encontró que los datos sugieren la presencia de una distribución Binomial de parámetros $p = 0,5$ y $n = 8$ en la generación de ambas variables (ver Figura 2.2). Tanto *Awareness* como *Aggressiveness* presentan valores entre 0 y 8, donde al aumentar el factor *Aggressiveness* disminuye el headway deseado. Por otro lado, a mayor *Awareness* el headway deseado aumenta, reflejando una mayor precaución por parte del conductor.

Además, de estas dos variables que modifican el valor de h , el headway deseado también puede ser modificado dependiendo del tipo de vehículo que se genere, e.g., liviano, de carga liviano, de carga pesado, etc. Incluso, dentro de cada simulación, se puede modificar h debido

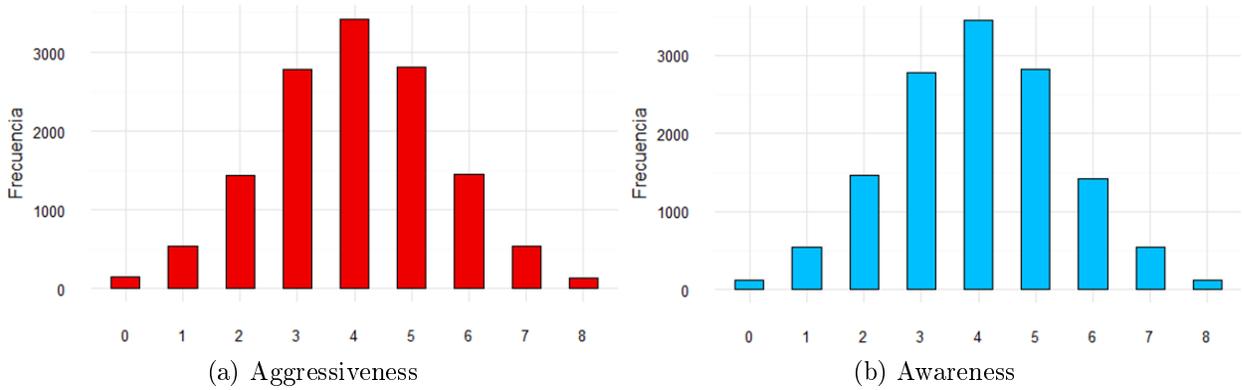


Figura 2.2: Histograma de variables *Aggressiveness* y *Awareness* en PARAMICS

a las condiciones de tránsito presentes. Por ejemplo, cuando el vehículo se va a enfrentar a una intersección y desea virar a la izquierda.

Adicionalmente, acorde a Duncan (1997), debido al deseo de emular las “shockwaves” que se observan en las colas vehiculares del tráfico común y corriente, se incorpora un retraso o *delay* de 1 segundo (en promedio) como tiempo de reacción de los POV simulados. Este retraso temporal de la reacción de cada conductor simulado varía en función de las velocidades y posiciones anteriores que “recuerda” el conductor. Duncan afirma que el tamaño de esta “memoria”, i.e., número de instantes en el pasado recordados por cada conductor, es inversamente proporcional al tiempo computacional requerido por la simulación, por lo que mediante una calibración de los creadores del modelo se fija en 3 el número de instantes recordados. El mecanismo que modifica el valor del tiempo de reacción no se ha detallado públicamente hasta la fecha en que se escribió este capítulo.

Si bien nunca se ha detallado el CFM de PARAMICS, es posible identificar algunos de los parámetros más importantes a partir de la experiencia con el simulador, la documentación oficial descargable del sitio de Quadstone Paramics (2017) y lo descrito en Duncan (1997):

- a Máxima aceleración.
- b_{max} Máxima desaceleración.
- b_{min} Mínima desaceleración.
- V_{max} Máxima velocidad a la que el conductor desea viajar.
- τ Tiempo de reacción promedio.
- h Headway o espaciamiento temporal promedio deseado.
- g_{min} Gap mínimo, distancia entre cada vehículo al estar detenidos en cola.
- d_t Distancia entre vehículos a partir de la cual vehículo detrás no percibe al otro.
- n_a Número de vehículos hacia adelante que cada vehículo percibe para reaccionar.

Existen muchos parámetros en los modelos psicofísicos en comparación a los otros tipos de modelos, lo cual es una desventaja ya que puede traer muchas complicaciones debido a que el problema de calibración requiere herramientas de optimización no-lineal. Sin embargo, en la práctica no se calibran todos los parámetros existentes. De acuerdo a Brockfeld et al. (2003), existen trece parámetros a calibrar en el modelo de Fritzsche (lo más parecido al CFM de PARAMICS), sin embargo, la mayoría de los trabajos en la literatura que ocupan

PARAMICS calibran los tres considerados más influyentes: τ , g_{min} y h . El motivo se debe a que existe un trade-off entre la cantidad de parámetros a calibrar y la facilidad del proceso (i.e, número de simulaciones necesarias, mínimo largo temporal de cada simulación requerido, resolución y cantidad de datos necesarios) (Truong et al., 2015; Saifuzzaman y Zheng, 2014; Olstam y Tapani, 2004).

2.4.3. Modelos de Seguimiento Vehicular: Estado del arte

Desde el artículo escrito por Brackstone y McDonald (1999) se han ideado nuevos Modelos de Seguimiento Vehicular, una revisión detallada de estos nuevos CFM y una nueva manera de categorizarlos se puede encontrar en Saifuzzaman y Zheng (2014). En este artículo, los autores proponen particionar los CFM en dos grupos: los de enfoque ingenieril y aquellos centrados en modelar el comportamiento humano. Dentro de cada grupo existen sub-categorías, entre los CFM de enfoque ingenieril se encuentran las sub-categorías: GHR, Desired Measure, Collision Avoidance, Optimal Velocity, Newell y Cellular Automata. Tanto el segundo como el cuarto forman parte de la categoría de modelos lineales de Brackstone y McDonald (1999). Entre los modelos enfocados en el comportamiento humano están: Perception Threshold (o psicofísicos), Driving Visual Angle (DVA), Driver Risk-taking (DRT) y Driver Distraction and Error (DDE).

Sobre los modelos ingenieriles, Saifuzzaman y Zheng (2014) afirman que carecen de la capacidad para simular situaciones de colisiones entre automóviles, dado que el 75 % de los accidentes son culpa de un error del conductor y en general las ecuaciones se conciben de forma que no haya conflictos espaciales entre las entidades simuladas. En particular, con respecto a los modelos GHR, los autores dicen que la mayor ventaja es su simplicidad, pero sus mayores defectos son no capturar la heterogeneidad en la población de conductores y sobrestimar la percepción de los conductores ante cambios pequeños en la velocidad relativa o el espaciamiento con respecto al vehículo en frente.

Por otro lado, se critica la posibilidad de agregar *Fuzzy-Logic* a los modelos GHR debido a la tarea complicada que sería definir las funciones de pertenencia y por dificultar aún más el proceso de calibración al aumentar el conjunto de parámetros. En cuanto a los modelos del tipo *Desired Measure*, se opina que sus variables no son observables, lo que se debe a que la mayoría de los modelos en esta categoría no se han calibrado empíricamente usando datos de tráfico real. De igual forma, tampoco se han realizado calibraciones de modelos de la categoría *Optimal Velocity*.

Sobre los modelos tipo Newell (2002) y sus extensiones los autores afirman que si bien estos modelos tienen dos parámetros, implicando un menor esfuerzo en su calibración, ninguno de éstos se relaciona a la variación en el comportamiento que presenta la población de conductores en la realidad. Finalmente, sobre los modelos que ocupan autómatas celulares, se tiene que al ser discretos no simulan el tráfico de manera realista. Si bien han habido esfuerzos para realizar simulaciones de autómatas continuas, éstas presentan un mejor desempeño a un nivel macroscópico.

Como crítica a los modelos enfocados en modelar el comportamiento humano, en aquellos de tipo psicofísico se observa que los valores de los umbrales de acción se obtienen de la literatura sobre estudios de comportamiento de conductores y no de datos para calibrar estos puntos de acción. Adicionalmente, las ecuaciones para los distintos umbrales no se publicaron. En cuanto a los otros modelos, los de tipo DVA no incorporan el tiempo de reacción en los conductores simulados ni tampoco heterogeneidad de la población. Los de la categoría DRT tampoco incorporan tiempo de reacción y al ser de carácter estocástico, su CFM presenta mayor ruido en los perfiles de aceleración de los vehículos. Por último, los DDE no han sido validados con datos reales.

Para concluir, Saifuzzaman y Zheng (2014) agregan observaciones adicionales sobre el estado del arte de la simulación de tráfico. En particular, con respecto a la recolección de datos, se destaca que la información normalmente proviene de registro de trayectorias (e.g. mediante GPS) y espiras que miden la tasa de ocupación de calles, lo cual es insuficiente para extraer las características de conducción de la población. Esta limitación provoca que los factores de comportamiento humano sean sobre-simplificados en los CFM que consideran factores humanos (como los psicofísicos). Debido a que parámetros relacionados a factores humanos no son observables en la naturaleza, se deben generar maneras innovadoras de recolectar otro tipo de datos para capturar las motivaciones psicológicas detrás del actuar de los conductores, la variación en sus percepciones y sus funciones cognitivas durante la conducción de un vehículo en el tráfico urbano. Es recomendable empezar a realizar experimentos sobre conducta humana usando simuladores de conducción para obtener parámetros críticos para los modelos de seguimiento vehicular. Los autores aseguran que estos experimentos son extremadamente escasos en la literatura, lo cual se condice completamente con lo observado en la realización de la presente revisión bibliográfica. En consecuencia, se observan dos maneras de sobrepasar esta limitante en los estudios usando MTS: datos de trayectoria son ocupados para estimar algunos factores humanos del CFM o los valores de estos factores se obtienen de la literatura o del sentido común.

Relacionado con la calibración, los autores mencionan que aún falta una metodología sistemática y rigurosa para calibrar. La mayoría de los modelos se prueban para ver si sus medidas de tráfico a nivel macroscópico se parecen a la realidad, lo cual solamente invalida un CFM en caso que éste se aleje mucho de las medidas reales, no lo valida como un buen modelo general en caso que arroje buenos resultados para un conjunto de datos en particular. Esta manera de enfrentar el problema de calibrar en los estudios que ocupan simulación dando tantas libertades provoca que se puedan generar malas prácticas de investigación acorde a los autores.

2.5. Estudios de Comparación de microsimuladores de tráfico

Brockfeld et al. (2003) comparan trece CFM ocupados en distintos simuladores, e.g. Gipps (1981), Fritzsche y Ag (1994) entre otros. Para esto se calibran con una misma base de datos de observaciones sobre una vía de una pista que termina en una intersección semaforizada. Posteriormente se validan al calcular el error en la estimación de los tiempos de viajes simulados vs reales. Se encuentra que los modelos con menos variables a calibrar tienen una mejor performance como el caso de Gipps con 6 parámetros (17% error), en cambio los más complejos no tienen tan buen rendimiento (Fritzsche con 12 parámetros, error mayor a 20%). No obstante, esto puede deberse en parte a que se usaron heurísticas para resolver el problema de optimización no lineal que busca los mejores parámetros de los modelos complejos.

Brockfeld et al. (2004) comparan modelos como Gipps, Fritzsche y Newell entre otros. En este estudio, se calibran los parámetros de los diez modelos usando una base de datos de 36 experimentos realizados en una pista de prueba registrando el movimiento de dos vehículos. La medida de error es la diferencia entre los *gaps* (espaciamiento espacial) de los vehículos simulados y los reales a través del tiempo. Como resultado, se observa que los modelos tienen un error promedio de 15% y ninguno destaca por sobre todos los otros dado que ninguno es superior al resto para todos los experimentos en cuanto a porcentaje de error. Como conclusión general de estas investigaciones, los modelos más complejos (i.e. con mayor cantidad de parámetros) no necesariamente proveen mejores resultados, por lo que los autores recomiendan ocupar modelos más sencillos que requieren menor esfuerzo de calibración.

Panwai y Dia (2005) hacen una comparación de AIMSUN, PARAMICS, VISSIM, versiones 4.15, 4.1 y 3.7 respectivamente. Se ocupan datos de perfiles de velocidad de 2 vehículos en una pista medidos mediante radares y se comparan las distancias entre los vehículos (sus *gaps*) en el simulador vs realidad para calcular el error. De esta manera, se busca saber cuál es el mejor modelo de car-following. En los resultados se encuentra un rendimiento promedio parecido entre PARAMICS y VISSIM. Por otro lado, AIMSUN presenta un menor error que los otros dos. Adicionalmente, se encuentra un comportamiento menos realista en PARAMICS al comparar los gráficos Δx vs ΔV con la realidad.

Zhi et al. (2005) comparan otros aspectos entre PARAMICS (versión 4.0) con AIMSUN (versión 4.0). Posterior a un análisis de sus distintos módulos, se concluye que AIMSUN presenta una curva de aprendizaje, tanto para codificar redes como para modelar sistemas de transporte público. En cambio, PARAMICS tiene gráficas más realistas en su interfaz junto a una mejor integración de sus distintos módulos. Por último, el universo de funciones que vienen con PARAMICS para desarrollar APIs es superior al de AIMSUN según los autores.

Olstam y Tapani (2004) buscan observar las diferencias entre los CFM de distintos simuladores: AIMSUN, PARAMICS, MITSIM y VISSIM. En primer lugar simulan usando MATLAB un escenario de un vehículo que realiza una serie de aceleraciones y desaceleraciones. Posteriormente, incorporan las ecuaciones de cada CFM en un vehículo que sigue a este vehículo líder simulado y observan las reacciones del vehículo siguiendo al simularlos simultáneamente. Como no se calibran los simuladores, se ocupan los parámetros que vienen de la

literatura. Con los resultados se concluye que PARAMICS, AIMSUN y VISSIM son los que presentan una similitud mayor entre la trayectoria del vehículo líder y el que sigue. MITSIM en cambio presenta reacciones más suaves del vehículo siguiendo, provocando muchos frenos de emergencia.

Geistefeldt et al. (2014) ocupan 5 simuladores, entre ellos PARAMICS (ver. 6.9.3); AIMSUN (ver. 7.0.3) y VISSIM (ver. 5.4), para simular una rampa de salida de dos pistas en una carretera con distintas combinaciones de flujo entrando y saliendo de ella. El propósito es observar si al calibrar los simuladores es posible reproducir las capacidades de diseño del manual de carreteras alemán (*Handbuch fuer die Bemessung von Strassenverkehrsanlagen*), el homólogo del manual de carreteras estadounidense (*Highway Capacity Manual*). Además, se espera poder entregar un conjunto de parámetros para que sean usados en el futuro en estudios involucrando carreteras alemanas ocupando los MTS. Como resultado se tiene que sí es posible reproducir las capacidades de diseño con los distintos MTS a excepción de SUMO. Cabe señalar que principalmente se calibraron τ y h en el caso de PARAMICS.

Una empresa estadounidense fundada por académicos e investigadores, CLR Analytics (2011), llevan a cabo un estudio para comparar la usabilidad de PARAMICS y VISSIM. En particular, observan que PARAMICS requiere de un menor tiempo para codificar una misma red. Por otro lado, VISSIM presenta un mejor módulo de programación de semáforos actuados dado que el de PARAMICS tiene unas pocas funciones básicas de programación actuada. Posterior a la codificación de la red se comparan los tiempos de simulación. Se aprecia un tiempo de ejecución similar en caso de no mostrar los vehículos simulados en la interfaz. Sin embargo, al elegir que sí aparezcan en la simulación VISSIM aumenta significativamente el tiempo de ejecución requerido para simular el escenario.

Se encontraron pocos trabajos en la literatura que comparen MTS populares usando versiones más recientes (en el caso de PARAMICS la última es la número 6.9.3, Quadstone Paramics, 2017). Esto es lamentable debido a que en muchas de las actualizaciones a través de los años ocurren cambios en la interfaz de usuario, algoritmos y modelos detrás de la simulación, por lo que las comparaciones realizadas por Panwai y Dia (2005), Olstam y Tapani (2004) y Geistefeldt et al. (2014) no son tan relevantes hoy en día.

2.6. Calibración y Validación de MTS

La calibración y validación de simuladores o modelos predictivos, son tareas fundamentales para las investigaciones que ocupan dichas herramientas. “La calibración es el ajuste de parámetros del modelo para mejorar la habilidad del modelo de reproducir el comportamiento de conductores locales y las características de desempeño del tráfico” (Dowling et al., 2004). El proceso de calibrar y validar permite conseguir un mejor poder predictivo ya que no se espera poder tener el mismo nivel de precisión en todos los escenarios de tráfico posibles con los parámetros que hay por defecto en los software de MTS (Olstam y Tapani, 2004; Dowling et al., 2004).

El objetivo concreto detrás de la calibración de un MTS consiste en maximizar la correlación entre las mediciones reales y las simuladas. En el proceso de calibración se han identificado tres decisiones importantes a tomar: qué conjunto de parámetros del simulador calibrar, cuál métrica o medida de bondad de ajuste usar y cuál heurística o método ocupar para optimizar el problema de calibración. Estas decisiones dependerán de lo que se desee del estudio (periodo de simulación, tamaño de la red) y la calidad de la información disponible. Sobre esto último, la calidad depende de: la cantidad de datos, la resolución de éstos (tiempo transcurrido o distancia transcurrida entre registros, según corresponda) y la cantidad de fuentes distintas de donde provienen. Según varios autores, es recomendable tener más de una fuente de datos, e.g. tener velocidades de vehículos de datos GPS y de espiras instaladas en las calles (Hollander y Liu, 2008; Saifuzzaman y Zheng, 2014; Zhong et al., 2016).

Con respecto al conjunto de parámetros, como se mencionó antes, es preferible calibrar solamente aquellos parámetros más relevantes, i.e., que causan un mayor impacto en la calidad del ajuste al cambiar de valor. Este conjunto de “parámetros relevantes” se puede armar a partir de un análisis numérico o también se puede extraer de la literatura en caso que alguien ya haya echo un análisis del MTS de interés. El interés detrás de intentar minimizar el número de variables surge debido a que al aumentar la cantidad de parámetros a calibrar, se incrementa el número de simulaciones mínima necesarias (MNR por *Minimum number of runs*), crece la cantidad de datos necesarios y también la resolución de datos requerida (Truong et al., 2015; Saifuzzaman y Zheng, 2014; Olstam y Tapani, 2004; Dowling et al., 2004).

Con el fin de cuantificar cuánto se acercan las simulaciones a la realidad es común ocupar algún indicador de la bondad de ajuste, por ejemplo, el RMSE: la raíz cuadrada del error cuadrático medio (2.1). Cada término de estos indicadores utiliza alguna medida del tráfico real y_i y lleva a cabo algún cálculo aritmético que busca cuantificar la distancia con respecto a la misma medición del tráfico simulado x_i . Estos indicadores luego se utilizan como la función objetivo a optimizar para perfeccionar los parámetros de la simulación. Un análisis comparativo y recomendaciones sobre las opciones usadas en la literatura para medir la bondad de ajuste (GOF por *Goodness of fit*) se puede encontrar en Hollander y Liu (2008). En este artículo, los autores destacan la opción de aplicar estos indicadores sobre distintas dimensiones; por ejemplo, el RMSE u otra métrica se puede calcular sobre medidas a través de la red (i.e. espacialmente) o temporalmente al comparar los datos como si fueran dos series de tiempo.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2} \quad (2.1)$$

Adicionalmente, Hollander y Liu (2008) dan una serie de recomendaciones sobre la calibración de MTS: calibrar todos los parámetros juntos, a menos que sea más apropiado calibrarlos con datos distintos (e.g. flujos para modelo de elección de ruta, velocidades para calibrar el CFM); no siempre es trivial emparejar una medida tomada en la realidad y_i con una medida simulada x_i para calcular los indicadores como el RMSE; utilizar indicadores de bondad de ajuste que penalicen errores grandes y no pequeños (esto ya que es común tener fluctuaciones pequeñas), en consecuencia se recomiendan aquellos que ocupan potencias como el RMSE;

definir con cuidado el rango factible de cada parámetro; calibrar de manera manual si son dos o tres parámetros los que se calibrarán, si no, entonces es recomendable automatizar el proceso; validar la calibración al ver el rendimiento con un nuevo conjunto de datos que no haya sido ocupado para calibrar; si es posible, una validación es más robusta ocupando datos de otra fuente (e.g. ocupar flujos de espiras para calibrar y datos GPS para validar) y por último nunca usar el simulador calibrado para estudiar fenómenos en otras zonas geográficas o periodos, ya que la calibración no es válida para estos casos.

Hourdakis et al. (2003) también dan sugerencias para calibrar un microsimulador de tráfico. Aseguran que para lograr una calibración robusta, no basta con tener un buen GOF, sino también indicadores de la calidad de la calibración que indiquen cuáles son los errores no-sistemáticos en los que se está incurriendo. Con esto en mente los autores presentan el RMSP (2.2) junto al Coeficiente de Correlación r (2.3). Por último, se habla de la inecuación de Theil que puede ser descompuesta en tres expresiones y permite interpretar los errores. Se aplica la teoría a la calibración de una carretera con varias entradas y salidas. Los datos provienen de *loop detectors* (espiras) instaladas en el suelo de la carretera a partir de las cuales es posible obtener congestión y flujo en distintas secciones de la carretera. Acorde a los autores, la calibración debería consistir primero en calibrar parámetros globales y después locales de cada vía. Además, se recomienda partir calibrando flujo y después velocidad, al ser la velocidad una variable muy sensible al cambio. Por último, en caso que se calibre la red de una carretera, es preferible partir la calibración de las variables desde aguas arriba hacia aguas abajo. Esto se debe a que en los primeros tramos es donde más se van a modificar los parámetros, pero éstos se estabilizarán después de un par de iteraciones a medida que se calibra más aguas abajo.

$$RMSP = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(\frac{x_i - y_i}{y_i} \right)^2} \quad (2.2)$$

$$r = \frac{1}{N-1} \sum_{i=1}^N \frac{(x_i - \bar{x})(y_i - \bar{y})}{\sigma_x \sigma_y} \quad (2.3)$$

Dowling et al. (2004) buscan describir un marco teórico para trabajar en la calibración de cualquier microsimulador. Se recomienda dividir los parámetros en dos categorías básicas: aquellos que el analista cree que tienen un valor razonable, por lo que no desea ajustarlos, y aquellos en donde el analista no está tan seguro de su valor actual y le gustaría ajustar. Además, se menciona que el analista no debe calibrar parámetros que no entiende o para los cuales no tiene datos. Adicionalmente, los autores aconsejan calibrar la menor cantidad de parámetros posibles para disminuir el esfuerzo requerido en la calibración, sin embargo, advierten que existe un trade-off ya que al tener más parámetros el analista tiene más grados de libertad para ajustar el modelo calibrado a la zona simulada. Se presenta el indicador GEH 2.4, con el que se puede medir la calidad de la calibración (GOF) con respecto al volumen del tráfico que debería observar. Este indicador es usado en general en la literatura para medir el error en cada arco y se acepta como bueno si en el 85 % de los casos presenta un valor menor al 5 % (Hollander y Liu, 2008); esto equivale a tener como criterio que la mayoría de los arcos tengan un error pequeño. Se menciona también que hay que definir una función objetivo; en

este caso se toma la función MSE (error medio cuadrático 2.5) de los volúmenes simulados vs reales, ya que como se ha dicho anteriormente, se recomienda penalizar más los errores grandes que los pequeños. Los valores de los volúmenes varían al ajustar los parámetros que se han decidido calibrar, que son las variables de decisión del problema de optimización.

$$GEH = \sqrt{\frac{2(x_i - y_i)^2}{(x_i + y_i)}} \quad (2.4)$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2 \quad (2.5)$$

En cuanto a los algoritmos disponibles para resolver la optimización del problema de calibración, en Ciuffo y Punzo (2014) se comparan 4 algoritmos (Genetic Algorithm o *GA*, SPSA, Simulated Annealing, OptQuest) para la calibración de AIMSUN con 4 parámetros. Los autores invocan el teorema “*No Free Lunch*” que establece que no existe algoritmo de optimización que rinda siempre mejor que otros para todos los posibles problemas de optimización. Basándose en este teorema, se argumenta que el hecho que un algoritmo sea el mejor para un estudio no implica que domine sobre todos los otros para todas las instancias factibles. En consecuencia, el algoritmo preferible para un tipo de problema depende de cada problema particular. En este trabajo se observa que el rendimiento de los algoritmos varían considerablemente en los 42 experimentos, con lo que los autores afirman que se prueba la validez del teorema “*No Free Lunch*” i.e. la dependencia del desempeño del algoritmo con respecto a los parámetros a calibrar, el indicador GOF escogido y la calidad de la data. El mejor algoritmo (aquel que más encontró el mínimo global) resultó ser el OptQuest; sin embargo, globalmente (promediando sobre todos los experimentos) GA tuvo el mejor rendimiento.

Ratrou et al. (2015) usaron un GA para calibrar PARAMICS. En particular se calibró el headway deseado (h) y el tiempo de reacción (τ). Se escogieron estos parámetros debido a que tienen una gran influencia sobre los modelos de tráfico presentes en PARAMICS, siendo en general los dos parámetros que se han calibrado en muchos otros estudios que ocupan PARAMICS. Para la calibración se emplea una metodología por etapas: se calibra un modelo predictivo de largo de colas con una red neuronal, donde las variables independientes son h y τ ; la variable independiente es el largo de cola promedio según las simulaciones de PARAMICS. Después, se optimizaron los parámetros usando un algoritmo genético para ver con cuáles la red neuronal arroja los largos de cola más similares a la realidad. Un trabajo parecido de los mismos autores se puede encontrar en Reza et al. (2016), donde se describe una metodología similar usando herramientas de *machine learning* para calibrar estas dos variables del simulador PARAMICS. En ambos trabajos se muestra una tabla con estudios que calibran PARAMICS ocupando las mismas dos variables de interés (h y τ) junto a los valores encontrados una vez terminada la calibración. Un resumen de la información de ambas tablas se encuentra en la Tabla 2.2 De la tabla se desprende que los parámetros calibrados varían entre 0,45 a 1,65 segundos para h y entre 0,41 a 1,00 segundos para τ .

Tabla 2.2: Valores de parámetros calibrados en distintos estudios ocupando PARAMICS. Fuentes: Ratrou et al. (2015); Reza et al. (2016)

Autores	Lugar de simulación	h [s]	τ [s]
Ozbay and Bartin (2003)	South Jersey, New Jersey, USA	0,70	0,50
Ma and Abdulhai (2002)	Toronto, Canada	0,86	0,71
Gardes et al. (2002)	San Francisco Bay, USA	1,65	0,42
Zhe et al. (2009)	Guangdong Province, China	0,45	0,43
Jobanputra and Vanderschuren (2012)	Cape Town, South Africa	0,50	1,00
Chu et al. (2004)	Irvine, Orange County, California	0,78	0,66
Lee et al. (2001)	NA	0,625	0,41
Li et al. (2010)	NA	0,45	0,43

Zhong et al. (2016) proponen una metodología de calibración que simultáneamente identifica los parámetros importantes a calibrar. En su revisión de la literatura mencionan problemas del proceso de calibración. Entre éstos: muchos modelos de tráfico a nivel microscópico no han sido validados mediante datos reales; falta una metodología de calibración rigurosa y sistemática; la resolución necesaria de los datos en general implican la presencia de mucho ruido y errores; la calibración de modelos estocásticos requiere aún más esfuerzo ya que se requiere una mayor cantidad de corridas para filtrar el ruido debido a la aleatoriedad; los métodos de optimización consistiendo en una búsqueda determinista, frecuentemente ocupados para el problema de calibración, encuentran óptimos locales debido a la gran complejidad de la estructura del problema y las distintas combinaciones posibles de los parámetros e interacciones desconocidas (posiblemente no-lineales) entre los parámetros. En consecuencia, se prefiere ocupar métodos de búsqueda aleatoria como GA, *Simulated Annealing*, *Ant Colony*, etc. Por último, los autores destacan que GA son los más popularmente usados para resolver el problema de calibración de MTS, los que coincidentemente, según el análisis comparativo de Ciuffo y Punzo (2014), tienen un mejor rendimiento global para resolver el problema de calibración que el resto de las heurísticas.

En el caso de la ciudad de Santiago, en un estudio realizado por Fernández et al. (2013) se calibra una sección de una calle importante de Santiago (Avenida Las Condes) con el microsimulador TSIS-CORSIM de manera de simular intersecciones semaforizadas con tráfico mixto. Como conclusiones del trabajo, los autores recomiendan calibrar los parámetros relacionados a los arcos de la red al ser aquellos con una mayor influencia sobre la bondad de ajuste, a su vez, también se concluye que existen parámetros que tienen poca relevancia sobre la red y por lo tanto no es necesario dedicar tiempo a calibrarlos. Además, se destaca que mediante un proceso simple de calibración por ensayo y error se pudo mejorar hasta un 33 % el indicador de bondad de ajuste utilizado en el trabajo.

Algunos MTS incorporan estocasticidad en sus simulaciones, ya sea mediante sus CFM, al incorporar reglas de *Fuzzy-Logic*, o en los modelos de elección de ruta de los entes simulados. Existen varias maneras posibles para incorporar aleatoriedad en las simulaciones. En consecuencia, cada vez que se corre un MTS para simular un escenario para un cierto periodo, los resultados varían en cada corrida. Con el fin de hacer un análisis consistente, se suele ocupar la media de las variables medidas en cada simulación. Surge entonces la pregunta: ¿cuál es el número de simulaciones necesarias para obtener una buena estimación de la media?

Para esto, en la literatura se habla del mínimo número de corridas necesarias (MNR por *Minimum Number of Runs*) para estimar estadísticos de las “medidas de performance” (MOP) de interés. Entiéndase MOP como variables ocupadas para comparar las simulaciones con la realidad, e.g., flujo, largo de cola, tiempo de viaje, etc. Para llevar a cabo el cálculo del MNR, se utilizan técnicas estadísticas obteniendo intervalos de confianza del promedio estimado de estas medidas. El problema radica en que a priori la persona llevando a cabo las simulaciones no conoce la desviación estándar (σ) del MOP de interés, la cual es necesaria para calcular el MNR requerido para obtener una estimación de la media con un cierto nivel de confianza.

Una técnica común en la literatura, de acuerdo con Hollander y Liu (2008), consiste en realizar un par de corridas al principio, con el fin de tener una estimación de la desviación estándar ($\hat{\sigma}$). Luego, se calcula el MNR ocupando el valor estimado $\hat{\sigma}$ y se procede a realizar este número mínimo de simulaciones. Otra técnica común según Truong et al. (2015) consiste en iterar, estimando nuevamente la desviación estándar de la muestra y el MNR en cada simulación realizada o, alternativamente, recalculando el MNR cada vez que se han realizado n corridas, donde $n \leq MNR$. Esto se realiza hasta que el número de simulaciones realizadas sea mayor o igual al MNR calculado en la iteración actual.

En el trabajo de Truong et al. (2015) se plantean tres metodologías nuevas y las aplican a la estimación del MNR para una red real simulada en VISSIM. La primera metodología consiste en estimar el MNR de cada MOP a partir de cinco corridas iniciales, posteriormente se realiza el máximo MNR entre los MOP. Este es el método más sencillo, sin embargo, asume que el MNR calculado en un principio es el correcto. El segundo método es el *Sequential Bonferroni*, en dicho método se va iterando y estimando nuevamente el MNR para el conjunto de MOP, por lo que se sabe inmediatamente si se han hecho suficientes corridas para tener una estimación de todos los MOP con un error aceptable. Esto permite ahorrar en iteraciones al tener un cálculo más preciso del MNR. Los autores concluyen que el segundo método propuesto cubre las falencias que los autores identifican en el método con corridas iniciales y además estima un menor MNR que métodos más complejos que sobre-estiman la cantidad de corridas necesarias. Otra observación es que ocupar técnicas consistiendo en realizar un número de simulaciones iniciales para estimar la desviación estándar es aceptable, pero que éste número de simulaciones iniciales debe ser al menos 5 y ojalá 10.

Adicionalmente, los autores ocupan un modelo de regresión lineal con el que concluyen que existe una correlación entre la complejidad de una red y el MNR de sus MOP. Por otro lado, se analiza el efecto del largo temporal de las simulaciones con el MNR, se ve que existe una fuerte relación entre éstos donde el MNR decae fuertemente hasta las 3 horas de simulación y después decrece de a poco a medida que se aumentan las horas del periodo simulado. Finalmente, se menciona la importancia de la frecuencia del transporte público en el largo necesario para obtener una buena confianza de sus MOP. Si la frecuencia es baja, es preferible tener simulaciones de mayor largo temporal.

2.7. Síntesis

En la literatura especializada, es posible encontrar trabajos utilizando MTS para distintos propósitos, ocupando a su vez diversos softwares de microsimulación de tráfico. Existen evaluaciones de proyectos de transporte reales, investigación sobre los factores humanos que afectan el tráfico, estudios que buscan encontrar la mejor forma de calibrar los simuladores o que intentan identificar el mejor software de microsimulación con el fin de poder recomendar alguno por sobre el resto.

Entre los trabajos ocupando MTS, existe un subconjunto que busca simular situaciones de emergencia. Por ejemplo, evacuaciones debido a terremotos o ataques terroristas, o el despacho de servicios de emergencia a zonas que necesitan de su ayuda. Para esto último, es necesario conocer en detalle los efectos que causan los vehículos de emergencia sobre el resto de los vehículos presentes en su ruta. En paralelo, se debe considerar la heterogeneidad en la población de conductores y factores físicos como, por ejemplo, la distancia a la que se escucha normalmente una sirena.

Al buscar incorporar vehículos de emergencia en algún simulador de tráfico, es importante tener en cuenta que existe una gran variedad de modelos de seguimiento vehicular detrás de los MTS. Además, existen diferencias prácticas, e.g., la interfaz gráfica, memoria computacional requerida, modelos vehiculares que es posible modificar mediante librerías (para el desarrollo de APIs). Simultáneamente, hay que tener en cuenta el esfuerzo de calibración que implica cada modelo de CFM dependiendo de los parámetros que se escoge calibrar y la métrica que se ocupará para evaluar la bondad de ajuste de las variables de interés. Notar que las variables a medir con dicha métrica dependerán de la calidad de los datos reales obtenidos para el proceso de calibración.

En este trabajo se escoge trabajar con el software PARAMICS debido al conocimiento adquirido previamente en otras investigaciones y al enorme potencial que existe para modificar los modelos detrás del simulador, gracias a la extensa librería de funciones disponible en el software. Además, se desarrolla una metodología de calibración acorde a la calidad de los datos y a las variables relevantes de forma de optimizar los parámetros, ocupando métricas de bondad de ajuste apropiadas de la forma que se describe en la literatura.

Capítulo 3

Interfaz de Programación de Aplicaciones para Microsimulación de Vehículos de Emergencia

Con el fin de incorporar en PARAMICS los efectos que los EV tienen sobre el tráfico observado en el material audiovisual del CBS, se programa una Interfaz de Programación de Aplicaciones (API por *Application Programming Interface*) en C++ que modifica la forma en que los POV reaccionan frente a la presencia de Vehículos de Emergencia. Para esto, se diseñan tres modelos a ser programados en la API: el *EV Lane Changing Model* (ELCM) o modelo de cambio de pista para EV, el *EV Sidewalk Model* (ESM) o modelo de acera para EV y el *EV Intersection Model* (EIM) o modelo de intersección para EV. Adicionalmente, en el caso especial de que exista más de un EV en un mismo arco, se programó una regla de seguimiento entre vehículos de emergencia (EV following). Además, debido a una limitación del software de simulación, que nunca permite que un vehículo cruce una intersección si su acceso tiene luz roja, se tuvo que implementar un algoritmo de semáforos actuados ad hoc.

Antes de explicar en detalle los modelos y algoritmos desarrollados en la API de PARAMICS, hay que destacar que para lograr comunicar una biblioteca de enlace dinámico (i.e., archivos de extensión .dll generados por el código) con el MTS PARAMICS, es necesario que el programa constructor (escrito en C++ con Visual Studio 2015) utilice en sus líneas de código un lenguaje común interpretable por el simulador. Aquí cumple un rol protagónico la librería de funciones diseñadas para programar APIs, que son parte del módulo *Programmer* del software PARAMICS. Según Zhi et al. (2005), una gran ventaja de PARAMICS por sobre otros MTS es la gran variedad de funciones disponibles en esta librería, permitiendo que un usuario avanzado sea capaz de re-configurar distintos aspectos esenciales de la simulación. Esta librería sería un ejemplo de lo que se conoce en las ciencias de la computación como API o interfaz de programación de aplicaciones. Simultáneamente, aquellos programas que hacen uso de estas librerías de funciones para comunicar dos softwares también reciben el nombre de API.

3.1. Material Audiovisual del CBS

Para identificar de qué manera se comportan los conductores de los vehículos rodeando a un EV que está en ruta hacia una emergencia se observaron videos grabados desde la perspectiva de los carros bomba del CBS en distintas horas del día, para diferentes tipos de calles. Este material audiovisual fue entregado por el CBS durante el transcurso del año 2016. En algunos casos, se desconoce la dirección de la emergencia y la patente o máquina desde la cual se efectúa la grabación. En general, la fecha de grabación de los videos varía entre abril y septiembre de los años 2013, 2015, 2016.

Algunos registros entregados inician en la estación de bomberos, otros parten a la mitad de un viaje hacia una emergencia. Sin embargo, también existen videos que corresponden a un registro del viaje de vuelta hacia la estación después de haber atendido una emergencia, estos videos fueron descartados ya que no son de utilidad para este estudio. La gran mayoría del material audiovisual se registró desde cámaras instaladas en el parabrisas frontal de cada máquina, el resto de los videos fueron capturados desde cámaras a un costado de las máquinas. En la Tabla 3.1 se presenta una lista de los videos con su duración y la dirección de la emergencia.

Tabla 3.1: Videos CBS

Dirección de Emergencia	Duración
Abate Molina - Grajales	0:15:02
Eduardo Matte - Victoria	0:05:00
Domey - Carrera	0:08:01
Ventura Lavalle - Portugal	0:12:01
Maule - San Francisco	0:11:01
Pedro Montt - Fábrica	0:06:01
San Isidro - Cóndor	0:09:01
Pedro Montt - Luis Cousiño	0:15:31
Glaziau - Pedro Lagos	0:08:00
Arturo Prat - Arauco	0:06:00
San Diego - Eyzaguirre	0:04:00
San Diego - Victoria	0:05:01
Sierra Bella - Matta	0:03:00
Santa Elena - Ñuble	0:06:00
Dieciocho - Alameda	0:12:10
15 Videos Sin Dirección	1:35:20
Total	3:40:09

3.2. Características deseadas en el tráfico simulado

Al estudiar el material audiovisual entregado por el CBS, se observa lo siguiente sobre el comportamiento de los vehículos privados (POV):

- Cooperan a distintas distancias del EV; esto varía según conductor, nivel de congestión y la maniobra a realizar.
- No cambian de pista si otra pista es mucho más conveniente para el EV.
- Se detienen en intersecciones y ceden el paso al notar la presencia de un EV intentando cruzar, aunque EV no tenga prioridad por el acceso desde donde se aproxima (cede al paso, disco pare o luz roja).
- Cruzan intersecciones aunque su acceso de acercamiento no tenga prioridad, en caso que haya un EV detrás con la intención de cruzar la intersección.
- Cambian de pista para dar paso al EV, despejando su camino, de esta manera minimizan las maniobras de cambio de pista que debe realizar el EV.
- En casos de extrema congestión, al notar que el tráfico no está avanzando, algunos vehículos livianos se suben a la acera para permitir el paso del EV.

De este comportamiento observado, no se incorporaron los primeros dos (variación en distancia de cooperación y cambio de pista dependiente del nivel de congestión en las otras pistas) debido a limitaciones del microsimulador. Los cambios sobre el tráfico vehicular inducidos en PARAMICS mediante la API para vehículos de emergencia, desarrollada en este trabajo, consisten en:

- Vehículos se hacen a un lado (cambio de pista) de manera anticipada si perciben la presencia de un EV aguas arriba del sentido del tráfico, a una cierta distancia.
- Vehículos ceden el paso a un EV en intersecciones, incluso si el acceso por donde se aproxima el EV no tiene prioridad, o tiene luz roja en el caso que la intersección esté semaforizada.
- Vehículos en la línea de detención cruzan intersecciones aunque no tengan prioridad, si uno de los 3 vehículos detrás en la fila corresponde a un EV.
- Se deja que los EVs circulen por pistas exclusivas (para bus y/o taxi).
- Vehículos livianos se suben a acera en casos de congestión, siempre y cuando tal maniobra sea factible.

Estas modificaciones se materializan en la simulación mediante la API siguiendo la lógica de los modelos descritos en las siguientes secciones. Antes de detallarlos, es necesario mencionar que la API funciona bajo la siguiente lógica: una vez que la simulación está corriendo, en cada instante discreto de simulación, más conocido como *time-step*, el software revisa cada arco en toda la red, por lo que los modelos están basados en las propiedades de cada arco, incluyendo como propiedad los vehículos que, en el respectivo *time-step*, están transitando por cada arco analizado. Para dichos vehículos, es posible consultar su velocidad y distancia a la línea de detención. Sobre el sentido de los arcos, si bien gráficamente (en la interfaz de usuario) se puede codificar una calle con tráfico en ambos sentidos, internamente PARAMICS interpreta calles de doble sentido como dos arcos de un sentido, por lo que un arco en la API tiene un único sentido siempre.

En la Figura 3.1 se muestra el esquema usado para explicar los modelos presentados en este capítulo. En éstos, se representa un arco de dos pistas con tres vehículos en donde el tráfico va en sentido izquierda-derecha. Una variable importante es d_i , la cual consiste en la distancia desde el parachoque frontal del vehículo i hasta la línea de detención. Por otro lado, la variable $l(h)$ corresponde al largo del vehículo h .

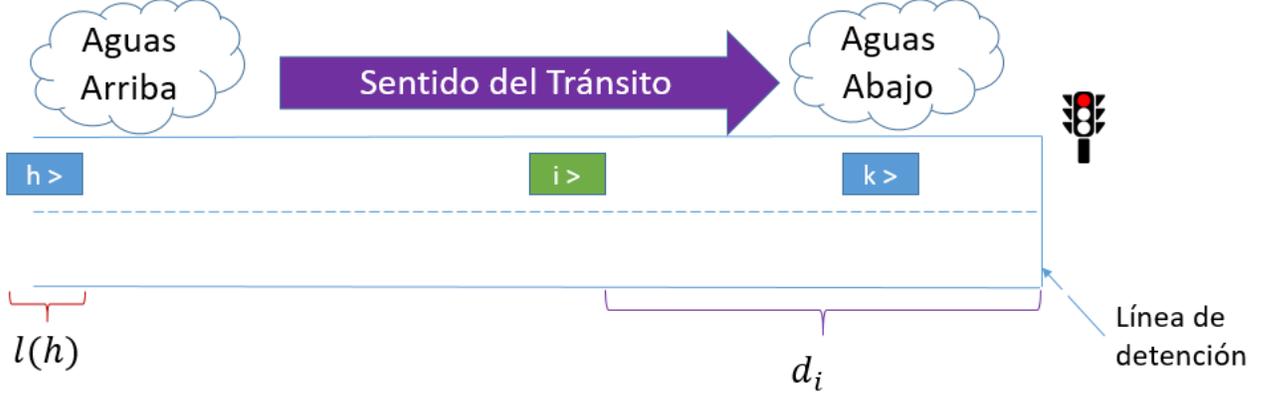


Figura 3.1: Definiciones básicas en esquemas de arcos mostrados más adelante

3.3. Modelo de cambio de pista para EV (ELCM)

Para comprender el modelo de cambio de pista en arcos, es necesario primero introducir cuatro nuevas variables:

- d_{min} : distancia mínima a línea de detención entre EVs presentes en el arco.
- d_{max} : distancia máxima a línea de detención entre EVs presentes en el arco.
- α_a : distancia aguas arriba de influencia de EV sobre otros POV.
- β_a : distancia aguas abajo de influencia de EV sobre otros POV.

Notar que en el caso que solamente exista un EV en un arco, se cumple que $d_{min} = d_{max}$. Adicionalmente, se define el EV líder como aquel vehículo de emergencia más cercano a la línea de detención.

En el modelo de cambio de pista en arcos, los vehículos que están *alerta* siguen una lógica para facilitarle el paso a los EV presentes. Un vehículo i cualquiera transitando por un arco con uno o más vehículos de emergencia debe cumplir con 3.1 y 3.2 simultáneamente para pasar a un estado alerta. Estas condiciones dictan que un vehículo privado está alerta si se encuentra entre dos o más EVs, o si está a lo más β_a metros aguas abajo de un EV, o si está α_a metros aguas arriba de un EV.

$$d_i \geq d_{min} - \beta_a \quad (3.1)$$

$$d_i \leq d_{max} + l(EV) + \alpha_a \quad (3.2)$$

En la Figura 3.2 se puede apreciar un esquema con un ejemplo. En éste se observa un arco con dos vehículos de emergencia en el cual los vehículos i y j cumplen las condiciones mencionadas anteriormente, por lo que se encuentran alerta. En contraste, los vehículos h y k no cumplen dichas condiciones geométricas y por ende no están alerta. Cabe destacar que el vehículo k no está alerta inclusive si parte de él se encuentra dentro de la “zona de alerta”. Esto debido a que, en la API, la distancia d_k usada para verificar las condiciones 3.1 y 3.2 se mide a partir del parachoque frontal del vehículo respectivo.

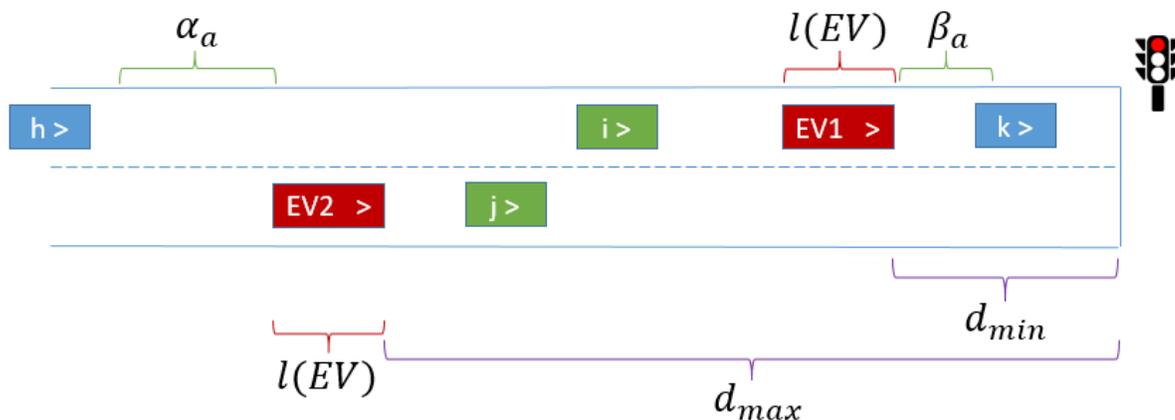


Figura 3.2: Modelo cambio de pista en arcos: vehículos i y j están alerta

Una vez que un vehículo se encuentra en estado de alerta, éste seguirá la siguiente lógica:

- En caso que no esté en pista de EV líder: no cambiarse a esta pista, es decir, no interferir el paso del EV.
- Si está en pista de EV líder: intentar cambiarse de pista, de esta manera el vehículo ayuda a facilitar el paso del EV.
- Si no se puede realizar este cambio de pista debido a congestión: avanzar y re-evaluar la situación, i.e., observar si EV líder sigue en misma pista o no y actuar acorde a estas reglas.

A modo de observación, en caso que un vehículo pudiera cambiarse tanto a la pista a su derecha como a la pista a su izquierda, si la API ordena a dicho vehículo una maniobra de cambio de pista debido a la presencia de un EV, ésta se percata de la posibilidad de cambiarse a cualquiera de las dos pistas, por lo que le ordenará en específico si cambiarse a la pista derecha o izquierda mediante la generación de una variable aleatoria binomial de parámetro $p = 0,5$. En palabras sencillas, si un vehículo alerta puede moverse hacia la izquierda o derecha, la API “lanzará una moneda” y le dirá a qué pista moverse. Esto ocurrirá solamente en caso de arcos con más de 2 pistas y en la cual el vehículo privado no se encuentra en una de las pistas al borde de la calle. En caso de no lograr cambiarse de pista después de intentarlo debido a que no se cumplen las condiciones del LCM que viene por defecto en PARAMICS, las cuales verifican que exista un espacio aceptable en la pista deseada, la API ordenará al vehículo que no siga intentando cambiarse de pista durante 3 segundos y luego, si persisten las condiciones, se decidirá nuevamente una pista de forma aleatoria para intentar cambiar a ésta. El ELCM no afecta a vehículos en arcos de una sola pista transitable y tiene reglas especiales en el caso de buses de transporte público, las cuales son:

Si un bus de transporte público está en estado de alerta, éste seguirá la siguiente lógica:

- Si está en la pista donde se encuentran los paraderos del recorrido y EV líder está en otra, se le ordena quedarse en pista con paradas.
- Si no está en pista donde se encuentran las paradas y EV líder está en su misma pista, cambiarse de pista de manera de acercarse a la pista con paradas.
- Aunque el EV líder esté en la misma pista, si el bus está en una pista con paraderos, no cambiarse de pista.

Por último, en caso de existir más de un EV en un arco, los EV que no son el líder, vale decir, aquellos que están más lejos de la línea de detención, son ordenados por la API a “seguir” al EV líder al recomendarles que busquen cambiarse a la pista del líder (*EV following*). Estas órdenes especiales solamente se generan entre vehículos de emergencia, de manera que aquellos que estén aguas arriba del líder busquen cambiarse a su misma pista.

3.4. Modelo de acera para EV (ESM)

En el desarrollo del modelo de aceras para EV se definieron cinco nuevos parámetros:

d_v^{min} : Distancia a la línea de detención hasta donde afecta el modelo de aceras.

β_v : Distancia hasta donde los vehículos aguas abajo del EV perciben su presencia.

v_h : Velocidad obtenida a partir de la distancia avanzada en los últimos t_h segundos.

t_h : Cantidad de segundos que recuerdan los vehículos para sacar su velocidad histórica.

s_h : Velocidad de congestión. Umbral usado para considerar un arco como congestionado.

En el modelo para aceras los vehículos *livianos* (i.e., no son camiones ni buses) cambiarán su comportamiento ante la presencia de un EV si están en la misma pista y no han podido realizar una maniobra de cambio de pista, lo que puede ser debido a congestión en las otras pistas o porque el arco tiene una única pista transitable. En el caso que dicho arco tenga una acera y haya tanta congestión que el vehículo percibe que no avanza (mediante condiciones que se explican más adelante), éste se subirá a la acera con el fin de dar paso al EV.

Para que un vehículo liviano llegue a subirse a la acera del arco que transita, antes tiene que estar alerta. Un vehículo i cualquiera pasa a estado alerta en el modelo de acera si se cumplen las condiciones 3.3 y 3.4. El vehículo buscará subirse a la vereda si además se cumplen las condiciones 3.5 y 3.6. Las primeras dos condiciones reflejan que detrás del vehículo, a una distancia cercana, se encuentra un EV. Por otra parte, las otras dos condiciones reflejan una situación de congestión extrema. Se consideró como congestión extrema que los tres vehículos aguas abajo en la cola se encuentren detenidos y que en los últimos instantes su velocidad haya sido de magnitud muy pequeña. Esto se definió de esta manera a partir de lo observado en los videos entregados por el CBS, aquellos conductores que ven pocos vehículos detenidos en frente esperan a que éstos crucen la intersección, a veces sin prioridad, o bien si ven que la cola avanza tampoco toman la decisión de subir a la acera. En caso que el vehículo haya cumplido todas las condiciones mencionadas anteriormente, pero la acera se encuentra ocupada por otro vehículo en el espacio donde éste intenta subirse, dejará de considerarlo y avanzará. En 5 segundos, si sigue alerta, el vehículo verificará si cumple las condiciones e intentará subir a la acera nuevamente.

$$d_i \geq d_v^{min} \quad (3.3)$$

$$d_{min} \geq d_i \geq d_{min} - \beta_a \quad (3.4)$$

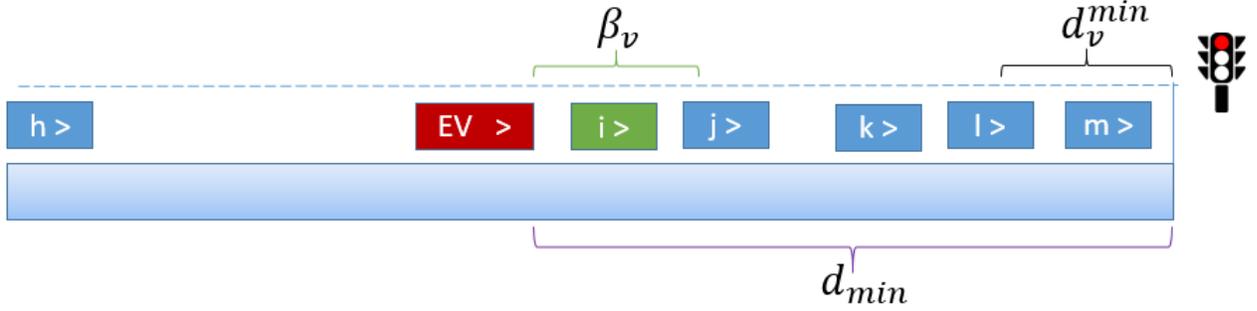


Figura 3.3: Modelo para aceras: vehículo i se encuentra alerta

$$v_i = v_{i+1} = v_{i+2} = v_{i+3} = 0 \quad (3.5)$$

$$v_h < s_h \quad (3.6)$$

En caso que el vehículo se haya subido a la acera, éste se mantendrá detenido sin avanzar horizontalmente hasta que el EV haya pasado. Una vez que pase, se intentará reincorporar al tráfico. Matemáticamente, esto significa que el vehículo se encontrará estático si cumple la condición 3.7 como se muestra en la Figura 3.4.

$$d_i \leq d_{max} + l(EV) \quad (3.7)$$

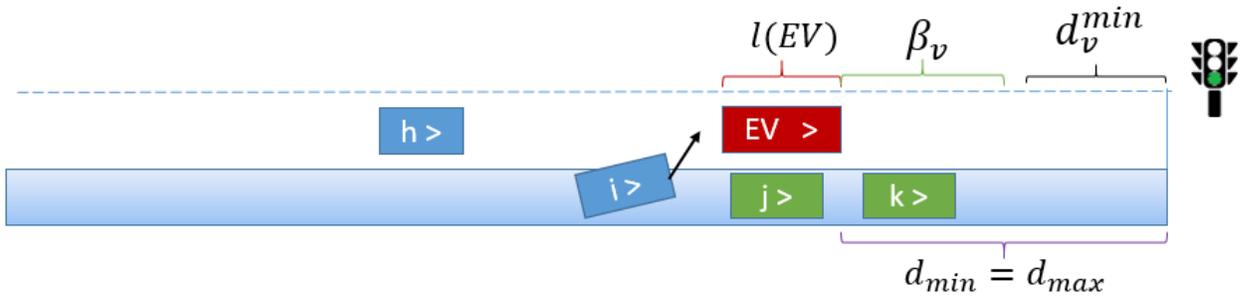


Figura 3.4: Modelo para aceras: vehículo i vuelve a la normalidad, vehículos j y k aún no

3.5. Modelo de intersección para EV (EIM)

Para el caso de EVs acercándose a la línea de detención en intersecciones donde su acceso no tiene prioridad, el modelo de veredas incorpora solamente un nuevo parámetro, δ_i , el cual es la distancia a partir de la cual los vehículos de emergencia acercándose a la intersección son percibidos en otros accesos; esta acción obliga a ceder el paso a los vehículos privados que se encuentren a menos de δ_i metros de la intersección, incluso si éstos tienen prioridad para pasar.

Como regla, un POV también cruzará sin prioridad si está en la misma pista aguas abajo del EV intentando cruzar la intersección (ver Figura 3.2). Los POV de otras pistas no cruzarán aunque estén en el mismo acceso. Además, en caso de existir más de un EV intentando cruzar una misma intersección en la simulación, los vehículos se registrarán por las reglas de tránsito comunes para decidir cuál EV cruzará primero.

Una vez que el EV cruce la intersección, ésta volverá a la normalidad. La API le ordenará a los vehículos que previamente estaban cediendo el paso forzosamente a moverse nuevamente acorde a la reglas del tránsito. Por último, para evitar conflictos internos entre este modelo con el de aceras, se recomienda que el parámetro del modelo de aceras d_v^{min} sea mayor o igual a δ_i ; por defecto, se ocupará $d_v^{min} = \delta_i$.

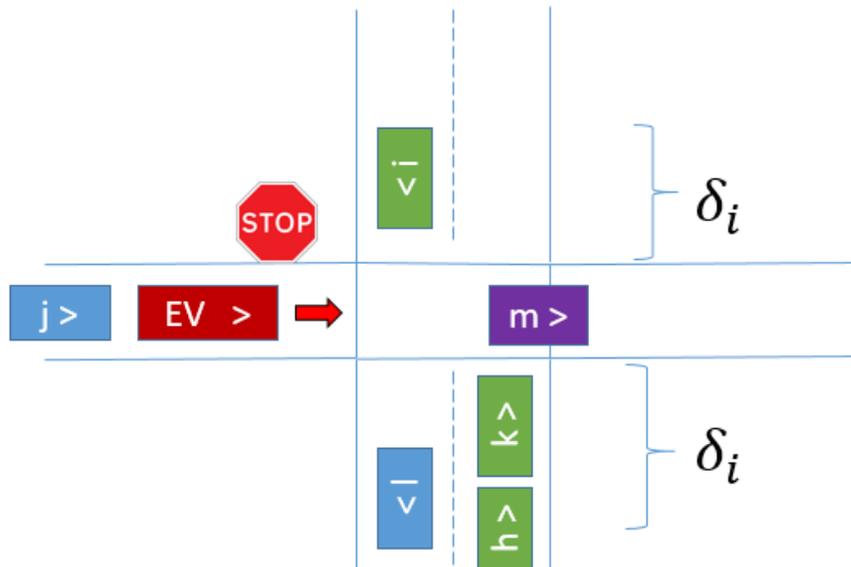


Figura 3.5: Modelo para intersecciones: vehículo m cruza sin prioridad

3.6. Parámetros y valores predeterminados

Cada modelo, programado e incorporado al simulador mediante la API, contiene parámetros que pueden ser ajustados, en principio, se usará los videos entregados por CBS para realizar una estimación gruesa. El modelo de arcos tiene dos: α_a y β_a . El modelo de veredas, en cambio, tiene tres: β_v , t_h y s_h . Por último, el modelo para intersecciones tiene δ_i que se asumirá igual a d_v^{min} ocupado en el modelo de aceras. Este último supuesto se debe a que d_v^{min} representa la distancia a la línea de detención en donde todos los vehículos ignoran el modelo de veredas, simultáneamente, δ_i es la distancia en donde se aplican las reglas del modelo de intersecciones si hay algún EV cerca. Es decir, se pasa de un modelo a otro en el instante que el vehículo está a menos de δ_i metros de distancia de la intersección.

En paralelo, se define un factor de cooperación (FC) entre 0 % y 100 %, parámetro general incorporado a la API de manera de reflejar el porcentaje de los vehículos particulares (POV) que cooperan con los vehículos de emergencia. En específico, FC es el porcentaje de los POV que escucharán órdenes de la API (i.e., cooperarán) al enfrentarse a EVs. De esta manera, se espera incorporar la variación en la población de conductores, ya que en los videos se puede observar que algunos vehículos no cooperan, lo que puede deberse a que los conductores prefieren no actuar ante la presencia del EV o puede ser porque no se percatan de la presencia del EV. Para simular esto, se genera una variable aleatoria binomial de parámetro $p = FC$ al crear cada POV en el simulador. Si la variable toma el valor 0 el vehículo cooperará en la eventualidad que se cruce con un EV. En caso contrario, el vehículo no coopera nunca. Con este mecanismo se tiene, en promedio, un porcentaje de vehículos cooperando igual a FC . Para estimar este porcentaje se debe contabilizar los vehículos que cooperan y aquellos que no, en condiciones ideales donde no se confunda la intención de no cooperar con la imposibilidad de cooperar.

En conjunto a los parámetros anteriores, que forman parte de la API, PARAMICS también tiene parámetros como se mencionó en el capítulo anterior. Entre éstos, los más importantes son el MTH (headway deseado promedio) y el MRT (tiempo de reacción promedio) (Ratrou et al., 2015). Por defecto estos parámetros tienen un valor de 1 segundo.

Con los videos mencionados anteriormente se realizó una estimación visual de los parámetros de la API explicados en la sección anterior. Se asignan valores para d_v^{min} y δ_i iguales a 15 metros, en el caso de los otros parámetros: $t_h = 10s$, $s_h = 2\frac{m}{s}$, $\alpha_a = 5m$, $\beta_a = 40m$, $\beta_v = 30m$ y $FC = 80\%$. Estos valores no resultaron de una estimación o calibración rigurosa, no obstante se consideran como los valores por defecto para usar en ejemplos. En el capítulo 5 se muestran valores estimados a partir de un proceso de estimación riguroso ocupando grabaciones de un viaje hacia una emergencia.

3.7. Algoritmos especiales para semáforos

Si bien el modelo para intersecciones afecta a todas las intersecciones, incluso aquellas con semáforos, en la realidad también se observa un comportamiento especial de los vehículos debido a la presencia de un EV al existir una señal semaforizada, lo que hasta ahora no se ha incorporado en los modelos ya descritos. Así, los vehículos comunes ignoran por completo la fase actual del semáforo cuando deben dar la pasada a un EV, aunque deban detenerse teniendo luz verde o bien cuando cruzan con luz roja en su acceso para no impedir el paso de un EV detrás de ellos. Por fortuna, sí es posible forzar a aquellos vehículos que enfrentan la luz verde a ceder el paso (i.e., no cruzar la intersección). Lamentablemente, por limitaciones del software PARAMICS, no es posible dejar que un vehículo simulado cruce si el semáforo está en luz roja, incluso si es un EV, por lo que se determinó simular este fenómeno de una manera indirecta.

Con el propósito de conseguir un comportamiento similar a la realidad, se implementó un algoritmo especial en la API que convierte todos los semáforos en semáforos actuados. La lógica detrás de un semáforo actuado consiste en utilizar información de operación y demanda que ocurre en tiempo real, obtenida de detectores dentro de la intersección, para alterar uno o más aspectos del ciclo actual del semáforo, i.e., acomodar la secuencia de las fases, los tiempos de verde de cada fase y el largo del ciclo. Un ejemplo de semáforo actuado típico es aquel con una fase que permite un movimiento con muy baja demanda; es decir, se observa una razón muy pequeña del flujo que desea realizar este movimiento. Si se detecta un vehículo en cola para este movimiento, se activa la fase pertinente. No obstante, mientras no se detecte la presencia de un EV, esta fase es ignorada.

En el caso de la API, los semáforos se activan cada vez que un EV está a menos de δ_i metros de la línea de detención con su acceso en luz roja, o bien si su acceso tiene la luz verde, pero falta menos de dos segundos para que termine esa fase del semáforo. En estos dos casos, el semáforo de la intersección pasa a ser controlado por los Algoritmos 1 y 2, diseñados para controlar estas situaciones.

Para controlar el estado de un semáforo, dentro de la librería de funciones de PARAMICS, existe la función $qps_SIG_action(intersección, fase, \dots, tiempoVerde)$, en donde *intersección* es el número asignado a la intersección de interés por PARAMICS; *fase* corresponde al número de fase a ser modificada en el semáforo de la intersección de interés; *tiempoVerde* es un número real positivo que representa el verde restante o verde programado. Para facilitar la notación, esta función se define como $T(fase)$ de aquí en adelante. Esta función puede realizar dos acciones: consultar el tiempo de verde que aún le queda a una cierta fase en una intersección, o bien asignar el tiempo de verde que le queda a la fase. En el último caso, si la fase modificada no es la actual al momento de llamar a la función, esto corresponde a cambiar solamente el tiempo de verde de dicha fase la próxima vez que se active. Posteriormente, la fase tendrá su tiempo de verde predefinido.

La lógica del Algoritmo 1 es la siguiente: se define fase actual f_A como la fase encontrada por el EV al momento de estar cerca de la línea de detención; se encuentra la fase más cercana a la actual donde el EV tenga permitido realizar el movimiento deseado; si dicha fase no es la fase actual, entonces se saltan todas las fases entre la actual y la fase de interés al definir

todos los tiempos de verde como cero; se le asignan dos segundos a la fase donde EV puede cruzar. Si estos dos segundos no son suficientes para la maniobra, se extiende la fase por dos segundos adicionales hasta que se materialice el cruce del EV.

Con el fin de volver a la normalidad una vez que el EV cruza la intersección, se ocupa un segundo algoritmo, el cual necesita determinar en qué instante del ciclo se debería dejar funcionando el semáforo. Para esto, se requiere capturar el tiempo de verde que hubiese transcurrido si el algoritmo no hubiese intervenido el semáforo. Con este fin se crea la variable V_h (“verde hipotético”). Esta variable captura el tiempo de verde que, hipotéticamente, transcurre de la fase actual f_A mientras el EV realiza la maniobra de cruzar sin prioridad.

Cabe destacar que, por limitaciones del software, no es posible modificar el tiempo de rojo-rojo entre las fases que deben saltarse para que se active la fase que permite el movimiento del EV. Es por esto que se descuenta este tiempo de V_h en la línea 12 del Algoritmo 1 cada vez que se salta una fase, ya que en la simulación hay un “tiempo muerto” entre fases igual al tiempo de rojo-rojo que obligatoriamente se debe esperar para cambiar de fase.

Algoritmo de Semáforos 1: Para intersección semaforizada con fases 1, 2, ..., N

```

1  Calcular:
2  número de fases  $N$ 
3  fase actual  $f_A$ 
4  fase más cercana que permita movimiento deseado por vehículo de emergencia  $f_{EV}$ 
5  tiempo rojo-rojo entre cada fase  $f_i$  y fase siguiente  $t_{RR}(f_i)$ 
6  tiempo que le falta a fase actual para terminar  $T(f_A)$ 
7  set  $V_h = T(f_A) - 2$ ;
8  set  $T(f_{EV}) = 2$ ;
9  if  $f_A \neq f_{EV}$  then
10 |   for fases  $f_i$  entre  $f_A$  y  $(f_{EV} - 1)$ 
11 |     set  $T(f_i) = 0$ ;
12 |     set  $V_h = V_h - t_{RR}(f_i)$ ;
13 |   if fase actual ==  $f_{EV}$  and  $T(f_{EV}) < 2$  then
14 |     set  $T(f_{EV}) = T(f_{EV}) + 2$ ;
15 |     set  $V_h = V_h - 2$ ;
16 end
```

Una vez que el EV ha cruzado la intersección semaforizada, se activa el Algoritmo 2. Éste se encarga de dejar el semáforo con la fase que estaba antes de la llegada del EV, con un verde V_h . Si ocurre que $V_h < 0$, implica que se agotó el verde hipotético de la fase actual mientras el EV cruzaba. Por lo tanto, se debería volver a la normalidad con la fase siguiente $f_A + 1$, con un tiempo de verde menor al que viene programado.

Con estos dos algoritmos se busca que la API reproduzca lo mejor posible la realidad en el software de simulación. En la práctica, la mayor diferencia entre la simulación y la realidad en intersecciones semaforizadas bajo la presencia de un EV es el “tiempo muerto” producto de los tiempos de rojo-rojo que se pierden al saltar a la fase f_{EV} .

Algoritmo de Semáforos 2: Para intersección semaforizada interrumpida

```
1 Requiere variables  $(N, f_A, f_{EV}, V_h)$ :
2 Calcular: tiempo rojo-rojo entre cada fase  $f_i$  y fase siguiente  $t_{RR}(f_i)$ 
3 if  $f_A \neq f_{EV}$  then
4   | for fases  $f_i$  entre  $f_{EV}$  y  $(f_A - 1)$ 
5   |   set  $T(f_i) = 0$ ;
6   |   set  $V_h = V_h - t_{RR}(f_i)$ ;
7 if  $V_h \geq 0$  then
8   | set  $T(f_A) = V_h$ ;
9 else
10  | set  $T(f_A) = V_{f_{A+1}} - |V_h|$ ;
11 end
```

Sin embargo, al observar la programación de los semáforos en Chile, se aprecian tiempos de rojo-rojo entre cero y dos segundos. Adicionalmente, los semáforos en general se programan con dos a cuatro fases, siendo la mayoría semáforos de dos fases. Debido a esto, en casi todos los casos, el “tiempo muerto” debido a saltar fases en los algoritmos no es significativamente grande, dado que gran parte de los semáforos de interés tienen dos fases y/o tiempo de rojo-rojo de un segundo.

Por último, notar que el “tiempo muerto” puede no estar tan alejado de la realidad, por el hecho de que en el material audiovisual provisto por el CBS, no se observa una cooperación inmediata una vez que un EV se aproxima a una intersección semaforizada, i.e., se observa un “tiempo muerto” también en la realidad donde ningún vehículo, incluido el EV, se atreve a cruzar la intersección, ya que aún no es claro que todos los vehículos presentes en la intersección estén cooperando cediendo el paso al EV.

3.7.1. Ejemplo numérico: simulación de intersección semaforizada

En la intersección de la figura 3.6 existe un vehículo de emergencia que desea cruzar la intersección; en particular, el EV desea hacer un movimiento Oeste-Este (OE). A su vez, la fase actual del semáforo permite solamente movimientos Norte-Sur (NS) y Sur-Norte (SN). Adicionalmente, el semáforo presenta dos fases adicionales detalladas en la Tabla 3.2.

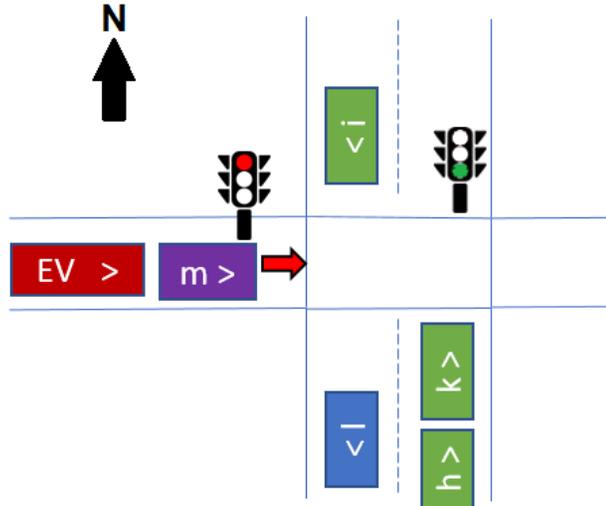


Figura 3.6: Ejemplo algoritmo de semáforos

Al simular esta situación, en el momento que el EV se encuentra a una distancia menor a δ_i de la intersección, la API lleva a cabo las órdenes que indica el modelo de intersecciones. Esto implica que los vehículos en calles sin presencia de un EV se detienen (vehículos i , k y h de la figura) y que vehículos delante del EV intentarán cruzar aunque no tengan la prioridad en la intersección (vehículo m).

Simultáneamente, la API activa el Algoritmo 1, de otra forma el vehículo m y el EV esperarían a tener verde para cruzar mientras el resto de los vehículos permanecen estáticos. El algoritmo obtiene la programación del semáforo que se muestra en la Tabla 3.2, donde V_i es el verde de la fase i -ésima, A_i su tiempo de amarillo y RR_{ij} representa el tiempo de rojo-rojo (i.e., periodo donde ningún movimientos se permite) programado antes de iniciar la siguiente fase. De la tabla se puede deducir que el tiempo de ciclo del semáforo es 64 segundos. Cabe mencionar que para la API, el tiempo de Amarillo programado es equivalente a una extensión del tiempo de verde. Es decir, en la API solo existe el tiempo de verde y el tiempo de rojo-rojo. Debido a esto, los tiempos de verde en cada fase para efectos del Algoritmo son 28, 10 y 23 segundos, respectivamente.

Una vez que se llama al Algoritmo 1, éste calcula el número de fases, identifica la fase actual (que corresponde a la fase 1 en este ejemplo) y busca la fase más cercana que permita el movimiento deseado por el EV, en este caso un movimiento OE. Para esto, se verifica si se permite el movimiento OE en la fase siguiente, correspondiente a la fase 2. Si esto no es así, se analiza la siguiente fase y así sucesivamente. En el ejemplo, la fase 3 permite el movimiento buscado, por lo que $f_{EV} = f_3$. Luego, suponiendo que al activarse el algoritmo quedan 10 segundos de verde en la fase 1, se crea la variable $V_h = T(f_A) - 2$, por lo que $V_h = 8$.

Tabla 3.2: Programación de semáforo en ejemplo de figura 3.6

Fase	Tiempo	Movimientos	Duración [s]
1	V_1	NS, SN	25
	A_1	NS, SN	3
	RR_{12}	-	1
2	V_2	SN, SE	7
	A_2	SN, SE	3
	RR_{23}	-	1
3	V_3	OE	20
	A_3	OE	3
	RR_{31}	-	1

Posterior a la inicialización, el algoritmo programa un tiempo de verde igual a cero para todas las fases entre la fase actual y f_{EV} , incluyendo la fase actual. Esto significa fijar en cero el tiempo de verde (y amarillo) de las fases 1 y 2. Además, por cada fase que se “salta” el algoritmo, se descuenta el tiempo de rojo-rojo a V_h . Como se tiene que fijar en cero el tiempo de verde de las fases f_1 y f_2 , se tiene que $V_h = 10 - RR_{12} - RR_{23} = 6$. Una vez que se hace esto, el algoritmo espera a que el semáforo pase a la fase de emergencia f_{EV} , cuyo tiempo de verde se programó en la inicialización como dos segundos. En caso de que estos dos segundos no sean suficiente para que el EV cruce, se le otorgan dos segundos adicionales, los que se descuentan de la variable V_h . Si suponemos que este fue el caso en el ejemplo, V_h queda en 4 segundos.

En el instante que el EV logra cruzar la intersección, se activa el Algoritmo 2, el cual busca volver el semáforo al estado en el que estaría si el Algoritmo 1 nunca lo hubiese alterado. Para conseguir esto, es necesario identificar la fase en la que estaría y el tiempo de verde restante que tendría dicha fase. Para lograrlo, se ocupa el valor de V_h y la fase que enfrentó el EV al llegar a la intersección f_A . Si $V_h > 0$, como en el ejemplo, se asigna verde cero a todas las fases entre f_{EV} y f_A (3 y 1 en el ejemplo). Por último, se asigna un tiempo de verde V_h a la fase f_A . Sin embargo, antes hay que contabilizar el rojo-rojo entre las fases 3 y 1 y descontarlo de V_h , ya que también consiste en un tiempo perdido. Para concluir, se redefine como $V_h = 4 - RR_{31} = 2$, por lo que finalmente el semáforo queda en la fase 1 de su ciclo, con dos segundos de verde restantes.

3.8. Síntesis de la API

La API diseñada en esta tesis consiste en un programa que ocupa la librería de funciones del módulo *Programmer* del software PARAMICS. Este programa contiene cuatro partes, cada una que extiende un control sobre la simulación bajo distintas situaciones. Dichos escenarios corresponde a cada vez que un EV:

- Transita una calle con más de una pista
- Se aproxima a una intersección cualquiera
- Transita por una calle congestionada con aceras
- Se aproxima a una intersección semaforizada

En cada uno de estos casos, las diferentes partes de la API se basan en los modelos descritos en este capítulo para controlar el simulador. La API se ejecuta simultáneamente con la simulación, en la cual pueden estar ocurriendo estos escenarios simultáneamente en distintos lugares de la red simulada, o bien puede que le estén sucediendo a un mismo EV, por ejemplo, cuando un EV se acerca a una intersección semaforizada entran en acción tanto el modelo de intersecciones como los algoritmos especiales para semáforos.

Capítulo 4

Preparación de red de simulación, recopilación de datos y procesamiento de señales GPS

En reuniones a principios del año 2016, el CBS expresó su interés por saber qué tan afectado se verá su operación debido al Plan Centro, el cual consiste en cambios a las calles del centro histórico de la ciudad de Santiago. Este plan fue ideado por la Municipalidad de Santiago en el año 2014 con el fin de efectuar un Plan Integral de Movilidad¹ que diera prioridad al flujo de peatones, ciclistas, transporte público y de carga por sobre al automóvil. Entre los cambios a considerar se encuentran medidas de aquietamiento de tráfico; cierre de calles; cambio en número y ancho de pistas; corredores para transporte público; ampliación de veredas y construcción de ciclovías². En consecuencia, se decidió codificar el casco histórico de Santiago y sus alrededores (ver figura 4.1).

4.1. Datos para red de simulación

Al decidir realizar una simulación microscópica de una red de tráfico real, surge la necesidad de codificar la topología de la red. Para cumplir este propósito, se requiere conseguir información sobre los distintos elementos que le dan sentido a una red conectada de calles en donde transitan vehículos motorizados. Estos elementos son: la geometría y topología de la red (ancho y largo de calles, número de pistas, ángulo entre calles en intersecciones); la regulación del tráfico (velocidad máxima, pistas especiales, prioridad en intersecciones); la programación de los semáforos que se encuentren en la red; el programa de operación del transporte público y la ubicación de las paradas de cada servicio; los flujos y tiempos de viaje del transporte privado (cantidad de vehículos transitando las distintas calles de la red en cada momento y sus velocidades); y la proporción de vehículos de cada tipo en el flujo de transporte privado (porcentaje de automóviles, camionetas, etc.).

¹<http://www.plataformaurbana.cl/archive/2015/12/07/plan-centro-restringira-acceso-vehicular-a-casco-historico-y-privilegiara-circulacion-de-buses/> visitado el 8 de marzo 2016

²<http://www.miratuentorno.cl/plancentro/wp-content/uploads/sites/0P-PIMS.pdf> visitado el 16 de mayo 2016

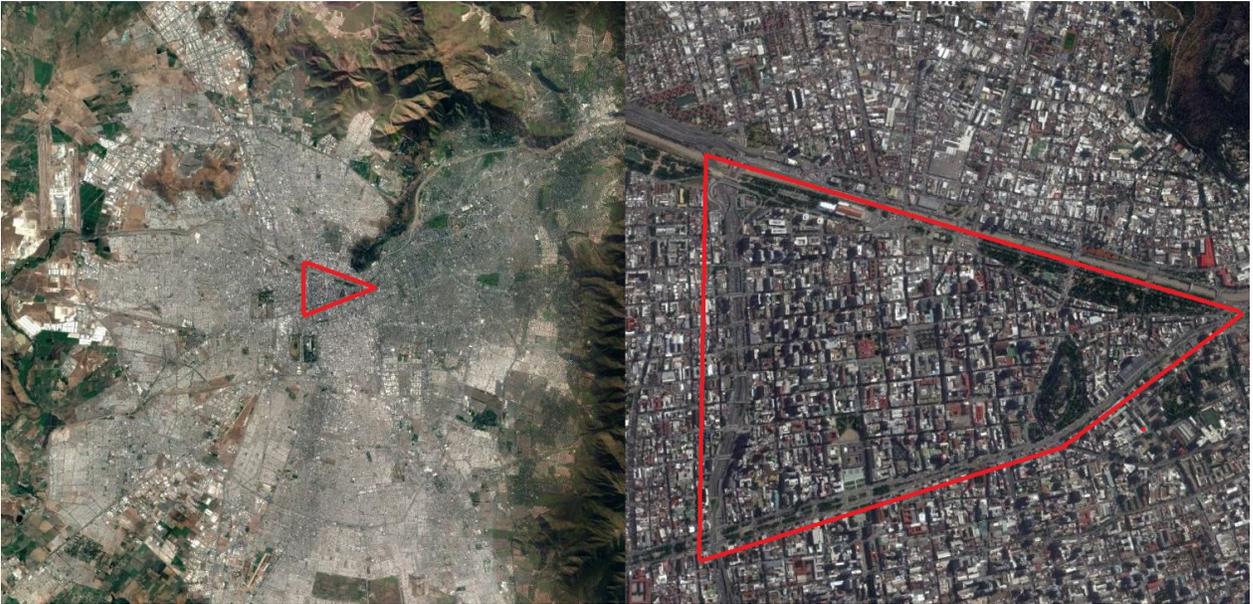


Figura 4.1: Fotos satelitales de Santiago, en rojo se muestra límites del área de interés

A priori, se requiere definir en detalle el lugar y periodo que se pretende simular. De esta manera, se evitan confusiones a futuro sobre qué codificar en la red. En el caso de este trabajo, se escogió recopilar datos para la zona de la figura 4.1 durante dos horas de la punta mañana de un día laboral promedio, específicamente, entre las 7:30am y las 9:30am.

La manera de obtener todos los datos y la fuente de éstos varía para cada caso. En la Tabla 4.1 se muestran los distintos elementos que requieren información y la fuente de donde se obtuvieron sus datos. En el caso de la información de transporte público, ésta se obtuvo del Directorio de Transporte Público Metropolitano (DTPM). El DTPM publica en su página web (www.dtpm.cl) el programa de operación y el trazado de recorridos Transantiago, sistema de transporte público de Santiago que hoy consiste en 5 líneas de metro y 378 servicios de buses³. Además, gracias a un convenio de investigación con la Universidad de Chile, fue posible ocupar los datos de transacciones (hechas con la tarjeta BIP, smartcard ocupada como forma de pago) y los datos GPS de buses Transantiago para obtener información de recorridos por paradero junto a la cantidad de subidas y bajadas promedio por parada, facilitando de esta manera el esfuerzo de codificación de recorridos y tiempos de detención por parada en la red de simulación.

En el caso de la información de programación de semáforos, se identificaron las intersecciones codificadas que presentaban semáforos. Luego, gracias a la ayuda de la Unidad Operativa de Control de Tránsito (UOCT)⁴ fue posible consultar la programación para el horario punta mañana de todos los semáforos de interés. En cuanto a la geometría, topología y regulación, esto se podía deducir de Google Street View, servicio de Google con el cual se pueden observar las calles de Santiago con fotos panorámicas tomadas desde un auto. Para obtener medidas oficiales de anchos de calzadas y la regulación de vías exclusivas (ejes que no permiten el paso de transporte privado en la mañana y/o tarde), se consultó a la Municipalidad de Santiago.

³https://www.dtpm.cl/archivos/INFORME_GESTION_DTPM_15_16_2.pdf visitado 1 de septiembre 2017

⁴<http://www.uoct.cl/>

Tabla 4.1: Fuentes de información para cada elemento de la red de simulación

Elemento	Fuente de Información
Geometría, Topología y Regulación	Google Street View Google Maps Bing Maps OpenMaps Municipalidad de Santiago Salida a terreno
Programación de semáforos	UOCT Salida a terreno
Flujo de buses y ubicación de sus paradas	Datos GPS Transantiago Planes Operacionales Información de Recorridos
Tiempos de viaje de Transporte Privado	Base de datos de TOMTOM

Entre los distintos datos necesarios para codificar una red, todos los datos de transporte privado son relevantes para la calibración del microsimulador. Con el fin de que la simulación se parezca lo más posible a la realidad, es necesario que los largos de cola, la congestión y la velocidad de los vehículos privados se asemejen a los datos reales. La obtención de dichos datos junto al proceso de calibración en la red de Santiago se escapan del alcance de esta tesis. No obstante, se obtuvieron datos de velocidad por arco en un grafo de Santiago creado por TomTom. La empresa TomTom⁵ recopila datos GPS de vehículos privados que tienen aparatos GPS de TomTom instalados o bien de vehículos cuyo conductor esté ocupando la aplicación Maps, la cual viene pre-instalada en los celulares iPhone de Apple⁶. Gracias a un convenio con el Departamento de Ingeniería Industrial de la Universidad de Chile, TomTom ha accedido a compartir parte de su base de datos para ser usados por la Universidad.

Los datos de TomTom obtenidos ya están procesados por investigadores del Departamento de Ingeniería Industrial. Estos datos toman en cuenta un periodo de un año (2015) para obtener la velocidad media por arco en intervalos de 30 minutos para un “día laboral promedio”. Con el fin de calcular la velocidad media sobre “días laborales promedio”, se decide descartar los fines de semana y los meses de diciembre, enero y febrero debido a que no presentan condiciones de congestión normales que sí se observan en el resto del año. Esto último ya que la gran mayoría de los estudiantes de educación básica, secundaria y superior ya no deben asistir a clases durante este periodo. En paralelo, es común también que los trabajadores pidan días de vacaciones durante estos meses, generando una menor cantidad de viajes en los días de semana. Adicionalmente, se descartan días laborales que ocurren justo antes de algún feriado, dado que presentan niveles de congestión fuera de lo común.

Con estos datos se calcula la velocidad promedio por arco para el periodo de dos horas establecido anteriormente. En la figura 4.2 se muestran los arcos con colores según su velocidad promedio para este intervalo de dos horas. Adicionalmente, se muestran los arcos según

⁵https://www.tomtom.com/es_es/

⁶<https://www.apple.com/cl/ios/maps/>

la categoría “Tipo de Calle” en la figura 4.3, variable que se genera con el propósito imputar velocidades promedio incógnitas, debido a un muestreo muy pequeño en algunos arcos. Es decir, si cierto arco no tiene datos para una cierta media hora, se le dan valores razonables para una calle de su tipo. El atributo “Tipo de Calle” viene dado por los mapas de TomTom e intenta agrupar las calles de manera de tener categorías de calles similares que se espera tengan patrones de comportamiento (velocidad a través de un día laboral) similares. Si bien se desconoce la metodología ocupada por TomTom para categorizar cada calle, lo que implica que no se sabe qué representa cada categoría de calle, sí se puede afirmar que cada calle perteneciente a un mismo grupo presenta curvas de tiempo de viaje similares a través del día.

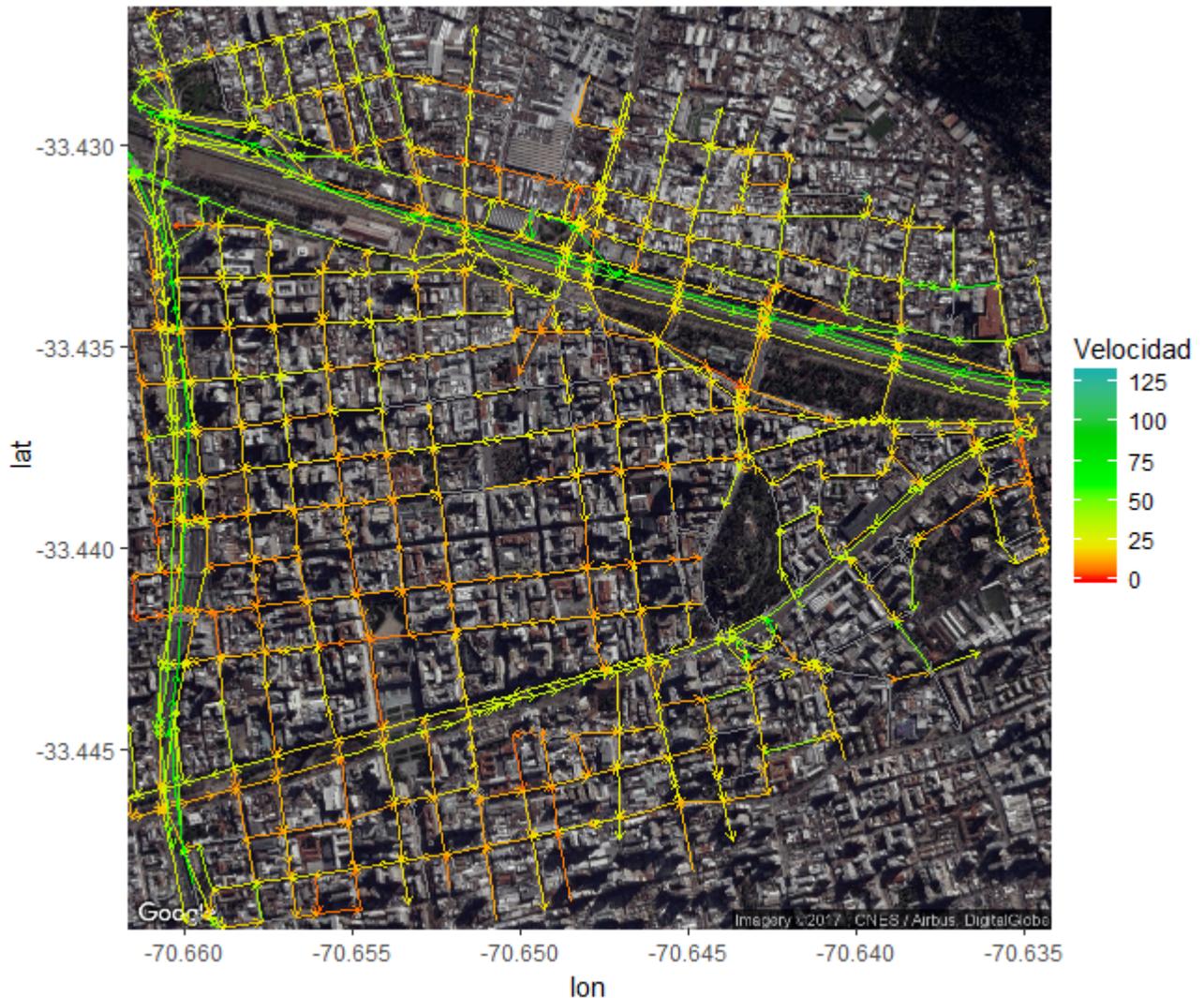


Figura 4.2: Velocidad promedio por arco en centro de Santiago entre las 7:30 y 9:30 AM

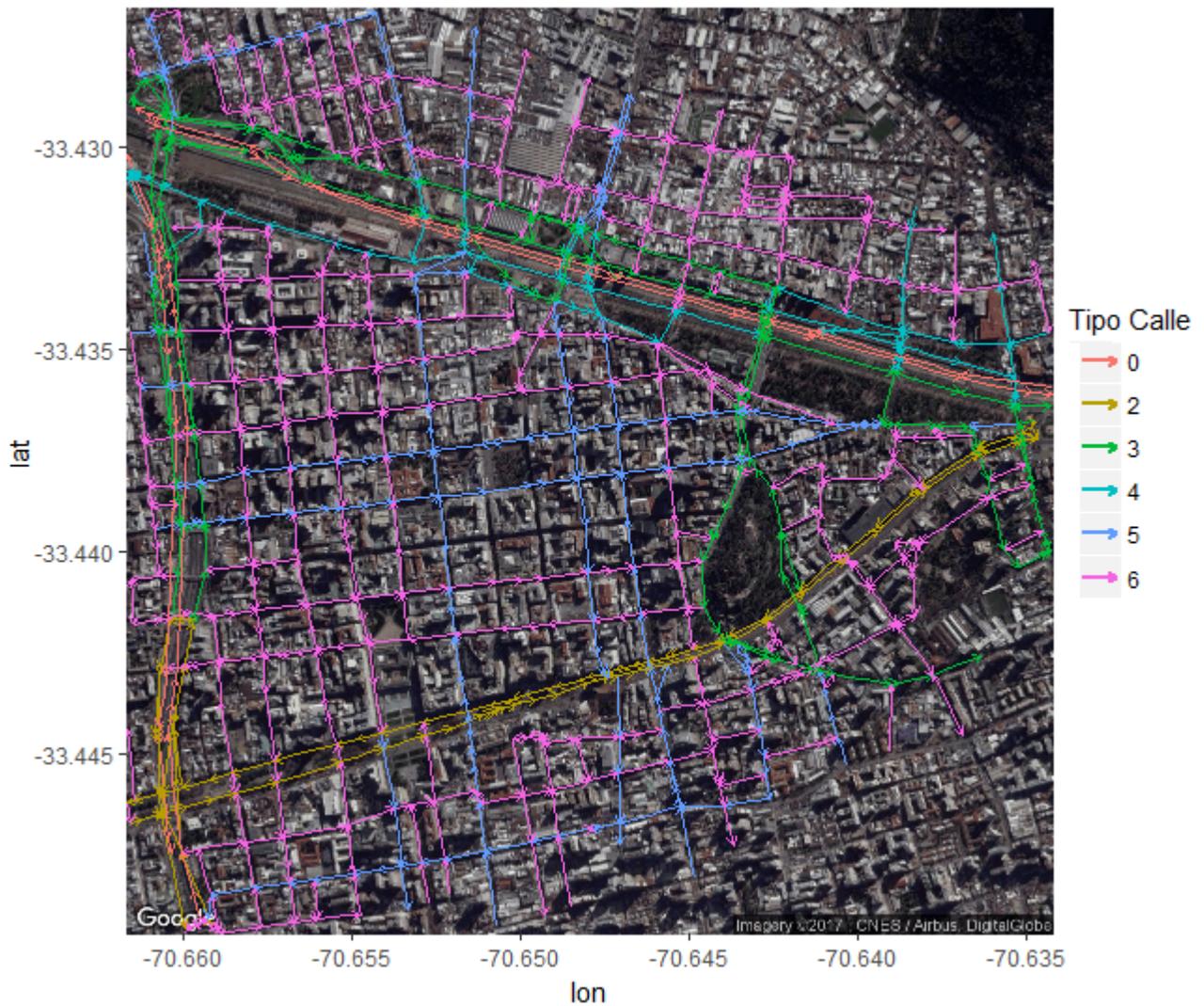


Figura 4.3: Arcos según su tipo en centro de Santiago Centro

4.2. Codificación de Zona Centro de Santiago

En paralelo, mientras se obtienen y procesan los datos necesarios indicados en la sección anterior, se codifica la red en el MTS PARAMICS. En primera instancia, se ocupa el programa Modeller, módulo de PARAMICS, el cual consiste en una interfaz gráfica de usuario (GUI) desde la cual se pueden “dibujar” las calles e intersecciones con el propósito de codificarlas en el simulador (figura 4.4). Además, la GUI tiene una opción para agregar una “capa” por debajo, con imágenes satelitales provenientes de Bing Maps. De esta manera, se puede codificar adecuadamente la orientación, sentido y geometría de las calles. Adicionalmente, para identificar detalles, por ejemplo, la distancia horizontal entre un paradero y la línea de detención en una calle, se ocupa la herramienta de distancia disponible en Google Maps. Si existen dudas sobre la cantidad de pistas o el sentido de alguna pista, se procede a ocupar la herramienta Google Street View para observar imágenes en terreno.

Cuando ya se han codificado las calles e intersecciones, es necesario agregar los elementos de control de tráfico. Por ejemplo, identificar aquellas calles con pistas exclusivas o solo bus, establecer la prioridad de cada intersección (i.e., si algún acceso tiene señal de pare o ceda al paso) y agregar semáforos en aquellas intersecciones semaforizadas. Para esto, se ocupa Google Street View junto a la base de datos de la UOCT para obtener la ubicación, planes y programación de los semáforos.

Respecto a las pistas con reglas especiales, se debe tener precaución de diferenciar aquellas pistas sólo bus de las pistas exclusivas. En particular, en la red de interés solamente hay pistas sólo bus en ciertos ejes especiales durante la punta mañana. Estas pistas sólo bus no permiten el paso de taxis. En cambio, las pistas exclusivas sí permiten el paso de taxis. No obstante, se consideró que ambos tipos de pistas especiales permiten el paso de EVs como se ha visto en los videos entregados por el CBS. También se codificaron pistas peatonales (ocupadas para simular paseos peatonales como el paseo Ahumada) que no permiten el paso de ningún vehículo simulado excepto por los EVs, con una velocidad baja, emulando situaciones donde los carros bomba transitan por los paseos peatonales con el propósito de alcanzar la emergencia que se encuentra en algún edificio accesible solamente desde algún paseo peatonal. Por último, se codificaron pistas para representar tanto calles con aceras anchas como con estacionamientos pagados; estos casos se agregan con el fin de incorporar los espacios al costado de las calles que permiten a vehículos particulares hacerse a un lado momentáneamente si así lo desean, acto que se incorpora en la simulación mediante el modelo de aceras explicado en el capítulo anterior.

En total, se codificaron 652 arcos, de los cuales 222 no contienen pistas especiales. Una cantidad de 213 tienen una pista de acera o estacionamiento (ocupadas por el modelo de aceras) y 237 presentan una pista exclusiva o sólo bus. Además, 42 arcos corresponden a calles peatonales que consisten en una pista especial (de tipo peatonal) tal como se describió anteriormente. Sobre el número de pistas por arco; sin contar las pistas para estacionamiento, aceras o paseos peatones; se encontraron 42 arcos sin pistas transitables (es decir, calles peatonales), 56 con solamente una pista, 245 con dos pistas, 181 con tres pistas, 95 con cuatro pistas, 29 con cinco pistas y 4 con seis pistas. En cuanto al largo de los arcos, en la figura 4.5 se muestra un histograma de esta variable. El arco más largo en la red es de 386 metros y el más corto tiene un largo de 6 metros. Cabe mencionar que tener 652 arcos en la red de simulación no implica que existan 652 calles codificadas, dado que una calle de doble sentido se representa por dos arcos en PARAMICS.

Sobre las intersecciones con semáforo, éstas se identifican mediante Google Street View anotándolas en una lista, ingresando las calles que se cruzan en la intersección semaforizada. De esta manera, con la lista de intersecciones se lleva a cabo una consulta en la base de datos facilitada por la UOCT. La base de datos entrega los horarios de los distintos planes para cada intersección. Un plan consiste en una forma de programar la duración y movimientos de las fases del semáforo. Dado que se pretende preparar la red para el periodo comprendido entre las 7:30 y 9:30 AM, se escoge el plan que ocurra durante este horario. Una vez identificado el plan de interés, se consulta la duración y los movimientos permitidos de cada fase, el tiempo de ciclo total del semáforo y el tiempo de rojo-rojo entre fases.

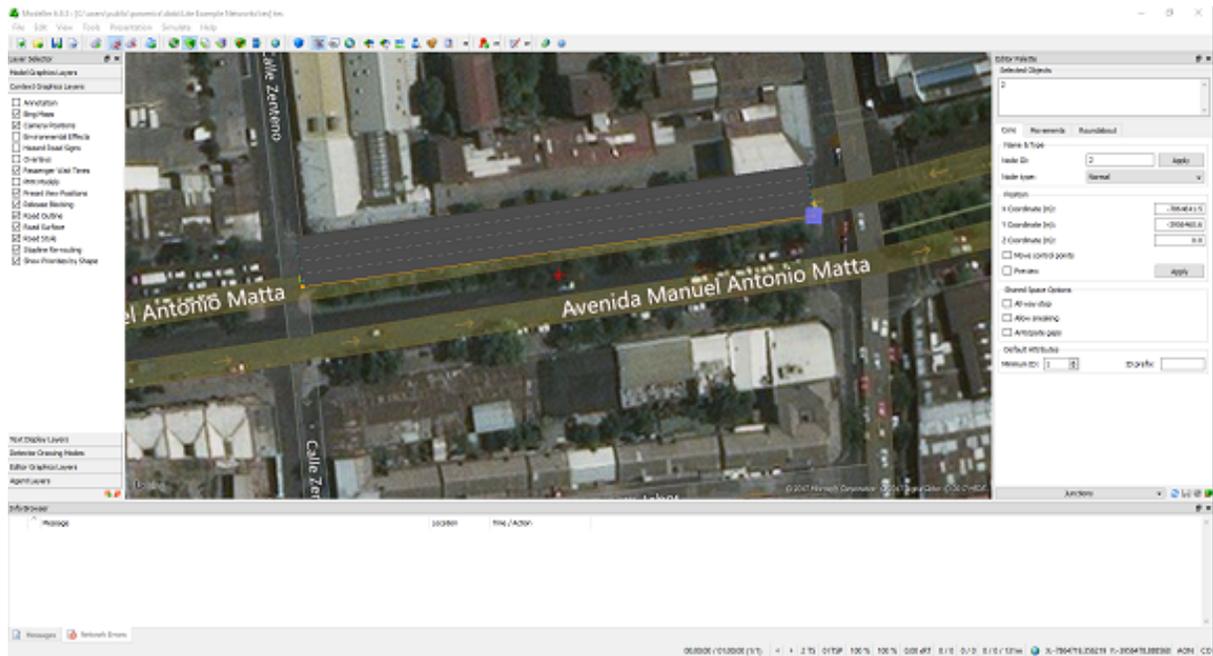


Figura 4.4: Interfaz Gráfica de Usuario de PARAMICS (Modeller) con capa de Bing Maps

En total, la red final tiene 453 nodos, de los cuales 228 representan una intersección. Los otros 225, si bien no representan una intersección, sí pueden contar con un semáforo; por ejemplo, calles con un paso peatonal al medio. Esto se codifica insertando un nodo con el fin de particionar el arco original en dos, de manera de poder asignar un semáforo al nuevo nodo. Este es el caso de 5 nodos que no son intersecciones, pero sí presentan un semáforo con dos fases programadas, donde una de éstas corresponde a una fase destinada al cruce de peatones. El resto de los nodos con un semáforo corresponden a 145 casos, que representan el caso común de un semáforo situado en una intersección de calles con flujo de vehículos motorizados. Entre los 150 semáforos codificados, 131 tienen dos fases, 15 tienen tres fases y los restantes 4 presentan cuatro fases. Como dato adicional, la gran mayoría de las fases de los semáforos terminan con un tiempo programado de rojo-rojo de un segundo (86.6 % del total).

Una vez codificada la prioridad de cada acceso en las 228 intersecciones junto a la programación de los 150 semáforos, las pistas especiales, la geometría (ángulo entre los accesos de cada intersección, ancho de pistas, posición de líneas de detención) se debe codificar el transporte público. Para esto, se obtienen primero los recorridos relevantes, que son los que en alguna parte del trazado de su recorrido cruzan la zona de interés codificada en el microsimulador. Con el fin de identificar dichos recorridos, se ocupa la herramienta de visualización de recorridos de Google Maps y la herramienta disponible en la página web de Transantiago⁷. Con la información de los recorridos y los paraderos en donde se detiene cada recorrido, se procede a codificar las paradas y los servicios de transporte público en la red de PARAMICS. Adicionalmente, se requiere ocupar la herramienta Google Street View, ya que es importante situar las paradas a la distancia de la línea de detención que se observa en la realidad.

⁷www.transantiago.cl/mapas-y-recorridos/conoce-los-recorridos visitado el 28 de julio 2017

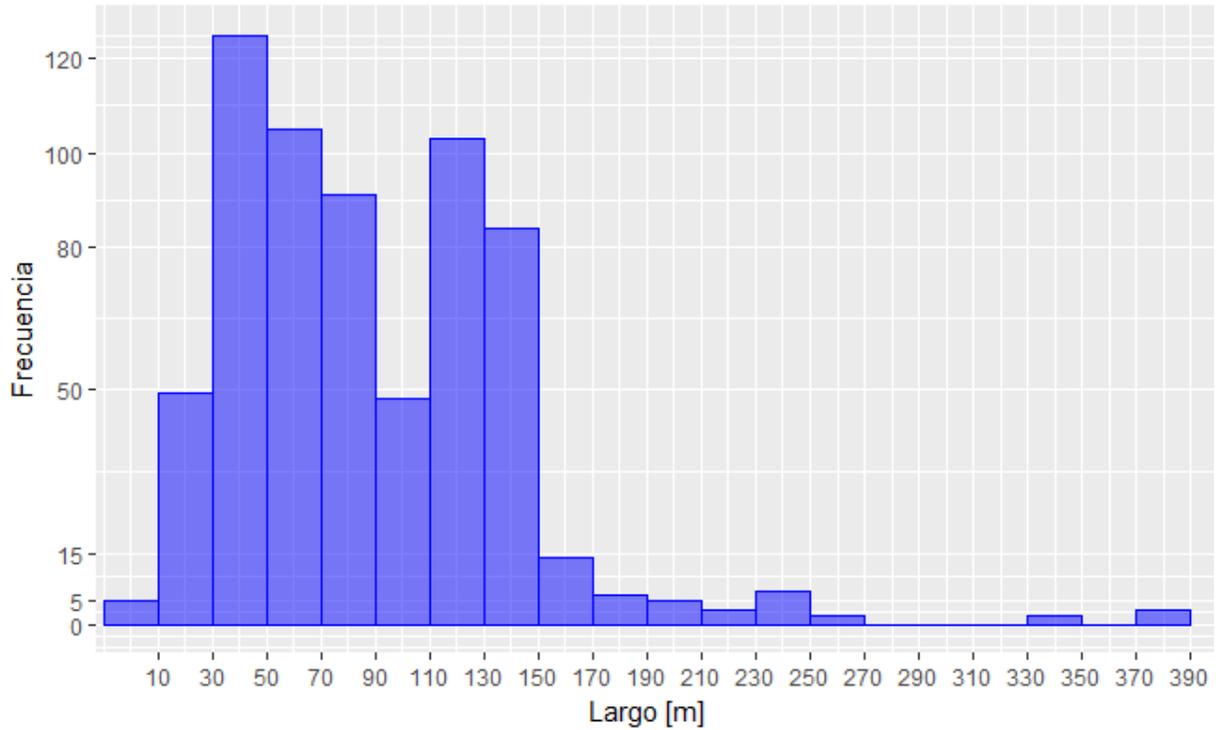


Figura 4.5: Histograma del largo de arcos codificados

Posterior a la codificación de los paraderos y recorridos al simulador, se deben programar los servicios (en particular su frecuencia) junto al tiempo de detención en cada parada. Con este propósito se ocupan los datos facilitados por el DTPM a la Universidad de Chile, procesados por el software ADATRAP. Mediante una consulta, es posible obtener la frecuencia de cada recorrido junto al número de subidas y bajadas estimadas para cada servicio por paradero, en el periodo de interés (7:30 a 9:30 AM), durante la última semana de Marzo 2016. La estimación del tiempo de detención por parada de los buses a partir del número de subidas y bajadas se lleva a cabo ocupando un modelo econométrico descrito en Tirachini et al. (2015). En este artículo, se proponen distintos modelos de regresión lineal para estimar el tiempo de detención de buses de transporte público en la ciudad de Santiago. Dichos modelos se calibran a partir de datos de conteo de subidas y bajadas obtenidos de grabaciones realizadas en distintos paraderos de la ciudad.

Para la red codificada, se ocupó el modelo que estima el tiempo de detención a partir del número de puertas para un paradero normal (i.e., una parada sin validadores externos). En particular, se utilizó solamente el modelo para buses de 3 o 4 puertas, dado que la mayoría de los servicios de transporte público de la red codificada son servidos por buses troncales de 3 o más puertas. El modelo, cuyos parámetros se muestran en la Tabla 9 del artículo, consiste en un “tiempo muerto” o constante de 6,71 segundos, un tiempo de bajada de 0,54 segundos por pasajero y un tiempo de subida igual a 2,35 segundos por pasajero. Con los datos de subidas y bajadas por parada para cada recorrido en el periodo de interés, se estima el tiempo de detención promedio para cada servicio en cada paradero mediante la ecuación 4.1, donde A_t y B_t son el número de bajadas y subidas, respectivamente.

$$T_{det} = 6,71 + 0,54A_t + 2,35B_t \quad (4.1)$$



Figura 4.6: Red final del centro de Santiago codificada en PARAMICS

Finalmente, con la topología, regulación (tanto en pistas como intersecciones), programación de semáforos y transporte público codificados para la zona de Santiago Centro, se agregan “zonas”, también conocidas como centroides. Las zonas se asignan a los arcos y corresponden a las ubicaciones donde se puede recibir o generar demanda. En otras palabras, corresponden a los orígenes o destinos de los viajes, tanto de transporte público como de transporte privado. Se agregaron zonas en cada calle que se situara al borde de la red y en cada ubicación de algún cabezal (o punto de regulación de frecuencia) de servicios de transporte público. En total, se codifican 79 zonas, donde 7 corresponden a calles de doble sentido. En la figura 4.6 se puede observar la red final con una capa por debajo, proveniente de una foto satelital de Bing Maps.

4.3. Preparaciones para calibración y trabajo futuro

Si bien la calibración de la red descrita en la sección anterior se escapa del alcance de este trabajo, se adelantó trabajo al completar dos tareas más. En primer lugar, se codificó una nueva red. Ésta se basa en la red original del Centro de Santiago descrita antes, pero con las modificaciones previstas por el “Plan Centro” o Plan Integral de Movilidad mencionado en la primera sección. Esto significa cerrar calles para el tráfico mixto, asignando pistas sólo bus en algunos ejes y cerrando calles debido a que se pretenden dejar como paseos peatonales. También en otros ejes se deben agregar pistas de acera y disminuir el ancho o número de pistas normales, debido a que se convertirán en calles semi-peatonales.

En segundo lugar, se creó un diccionario que permite automatizar la comparación entre las velocidades provenientes de TomTom y las encontradas por el MTS PARAMICS. Para esto, se generó manualmente una tabla donde cada registro contiene el nombre de un arco en PARAMICS junto al nombre del arco equivalente en TomTom. En algunos casos, una misma calle se codificó en PARAMICS como dos arcos y en TomTom como un solo arco o viceversa, lo cual se detalla en el diccionario. De esta manera, mediante un algoritmo, se pueden contrastar fácilmente las velocidades de las calles simuladas vs los valores registrados en la base de datos de TomTom.

Como trabajo futuro, dado que se desea ocupar la red como herramienta predictiva, será necesario incorporar el transporte privado a la red, con sus largos de cola y flujo o demanda por par origen-destino. Con este fin, se requerirá obtener datos confiables que indiquen el estado del tráfico en el periodo de interés. Fuentes posibles de esta información pueden ser: datos pasivos de movimiento registrados gracias a celulares (bluetooth, antenas de celular, GPS); datos registrados en aplicaciones (Waze, TomTom, GoogleMaps); videos de cámaras en intersecciones (UOCT) o conteos de flujo y largo de cola en terreno (SECTRA, universidades, consultoras). En paralelo, se tendrá que llevar a cabo un proceso de calibración que asegure que la simulación es una representación aceptable de lo que ocurre en terreno, tanto el nivel de congestión, como los tiempo de viaje. Adicionalmente, será vital llevar a cabo dicho proceso de calibración tomando en cuenta lo discutido al final de la revisión bibliográfica de esta tesis. De esta forma, al calibrar los parámetros de PARAMICS para esta red, se podrán simular escenarios con el propósito de hacer predicciones o evaluar proyectos de transporte o movilidad en la zona codificada.

4.4. Metodología para procesamiento de datos GPS

A fines del año 2015, la empresa Fleetup⁸ dedicada a prestar servicios de gestión de flota, instaló aparatos GPS en algunos vehículos de las distintas compañías del CBS. Por lo tanto, surge la oportunidad de aprovechar estos datos para alimentar un microsimulador de tráfico, dado que este tipo de datos contienen información muy valiosa y detallada sobre la operación de los vehículos registrados. En la literatura se han visto usos muy novedosos para datos GPS, como por ejemplo la deducción del trazado y dirección de rutas urbanas a partir de registros GPS (Biagioni y Eriksson, 2012); en otro estudio, Castro et al. (2013) ocupan los datos GPS provenientes de aparatos en taxis de la ciudad Hangzhou en China para analizar las actividades de la ciudad, la congestión y las operaciones de los taxis.

En el caso de los aparatos GPS instalados en vehículos del CBS, éstos tienen una precisión de 3 a 5 metros y se activan cada vez que el motor del vehículo asignado se prende, emitiendo un pulso con la latitud y longitud del vehículo cada 10 segundos. En total, se instalaron aparatos GPS en 61 vehículos, de los cuales dos correspondían a camiones que no se ocupan en emergencias (códigos L1 y L2) y dos tenían código sin sentido (REMOVE y BT18-HWGP56), por lo que finalmente se descargó el historial de pulsos GPS para 57 vehículos. Los códigos de los distintos vehículos y su significado se adjuntan en la Tabla 4.2. Además, se adoptó la siguiente notación: cada código siempre terminará con un número o una C, indicando

⁸<http://www.fleetup.cl/index.html>

la compañía a la que pertenece el vehículo o si es de la comandancia. Como excepción a esta regla, debido a que los vehículos pueden ser de una compañía, pero ser prestados a la comandancia, se escribe una “R” como segunda letra en el código. Por ejemplo, el código “BRC2” significa vehículo B2 en reemplazo usado en comandancia.

Tabla 4.2: Código de Vehículos CBS

Código	Significado
B	Carro Bomba
Q	Portaescalas
Z	Carro Aljibes para cisternas
K	Camioneta Comandancia
L	Camión de Logística
S	Ambulancia
X	Van de telecomunicaciones, transporte y apoyo
M	Camión con escaleras telescópicas
H	Carro de contención de materiales peligrosos
R	Carro de Rescate
BT	Carro Bomba Tanker
MX	Carro con escalera o brazo articulado
RX	Carro de Rescate técnico pesado
BX	Carro Bomba segundo socorro o intervención
BR	Carro Bomba de reemplazo
BRCX	Carro Bomba segundo socorro de reemplazo
RRCX	Carro de Rescate técnico pesado de reemplazo
QR	Portaescalas de reemplazo

Una vez que se obtuvieron los datos desde el servidor de Fleetup, se procedió a analizar los datos GPS. La primera observación es que tres de los vehículos no presentaron datos: BRC1, BRC4, RRCX7 (vehículo RX7 usado por comandancia). En consecuencia, se tienen los datos GPS de 54 vehículos del CBS en total. Por otro lado, debido a que la carga de combustible de los vehículos se hace en general el último día del mes, el sistema de Fleetup no registra los pulsos GPS emitidos durante estos días. Sin embargo, es posible que se registren GPS erradamente durante prácticas de conducción, viajes de vuelta a la estación y viajes por combustible en otros días del mes. La manera de solucionar este problema se explicará más adelante.

Al descargar todos los datos, éstos se consolidaron en una gran base de datos para ser analizados utilizando el lenguaje de programación R (R Core Team, 2017) mediante la plataforma open source RStudio⁹. En un principio, los datos consistían en una tabla de 2.637.191 registros de pulsos GPS de 54 máquinas del CBS emitidos entre el 1 de diciembre de 2015 y el 26 de mayo de 2016. Cada pulso consiste en información sobre 6 atributos: Vehículo, *Timestamp* (fecha y hora), Latitud, Longitud, Velocidad Instantánea y Revoluciones. El atributo Revoluciones no fue registrado por motivos que se desconocen, teniendo un valor constante e igual a 0 en todos los pulsos. Por esto, se decidió borrar el atributo de la base de datos.

⁹<https://www.rstudio.com/>

En paralelo, se creó un atributo “Compañía” que indica la compañía a la que pertenece el vehículo del registro.

En la figura 4.7 se muestra la cantidad de registros que se obtuvieron por cada vehículo durante el periodo de aproximadamente 4 meses especificado anteriormente. El vehículo con más datos fue RX17 con 114.182 pulsos GPS registrados. Por otra parte, el vehículo con menos pulsos emitidos fue el B11 con 418. Adicionalmente, en la figura 4.8 se muestra la cantidad de datos según la media-hora del día, siendo la primera media hora (marcada con un “1”) el intervalo entre las 0:00am y 0:30am.

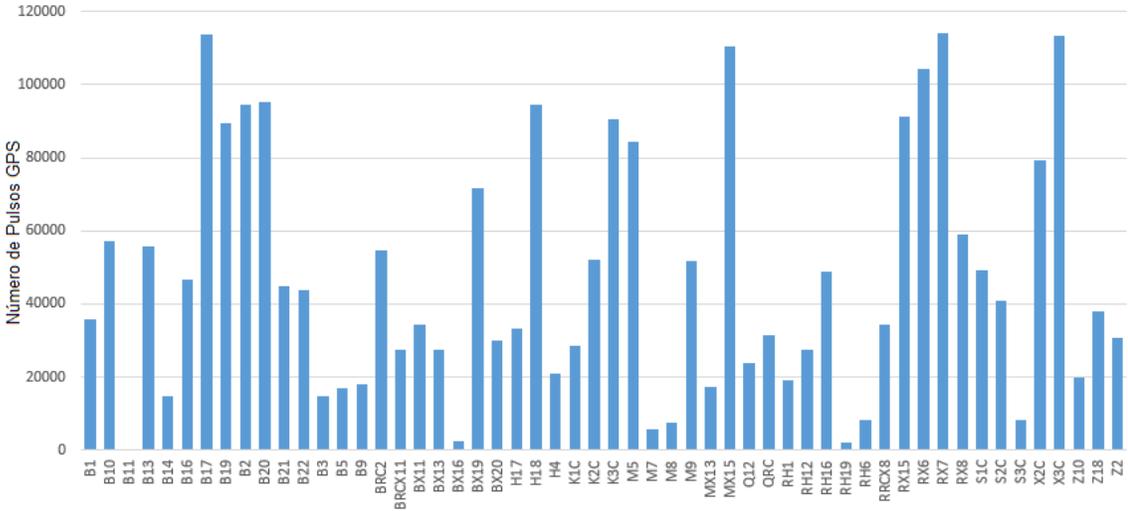


Figura 4.7: Datos por Vehículo

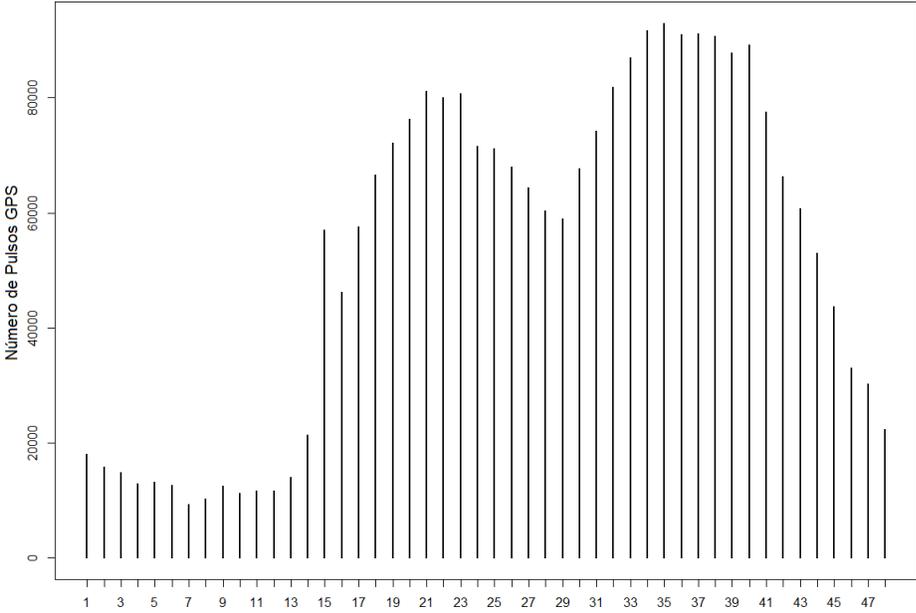


Figura 4.8: Datos por media-hora del día

Para tener una mejor idea de la zona que abarcan las distintas compañías y sus distintas máquinas, se muestran los pulsos GPS según vehículo para distintas compañías sobre un mapa del sector central de la ciudad de Santiago (figura 4.9).

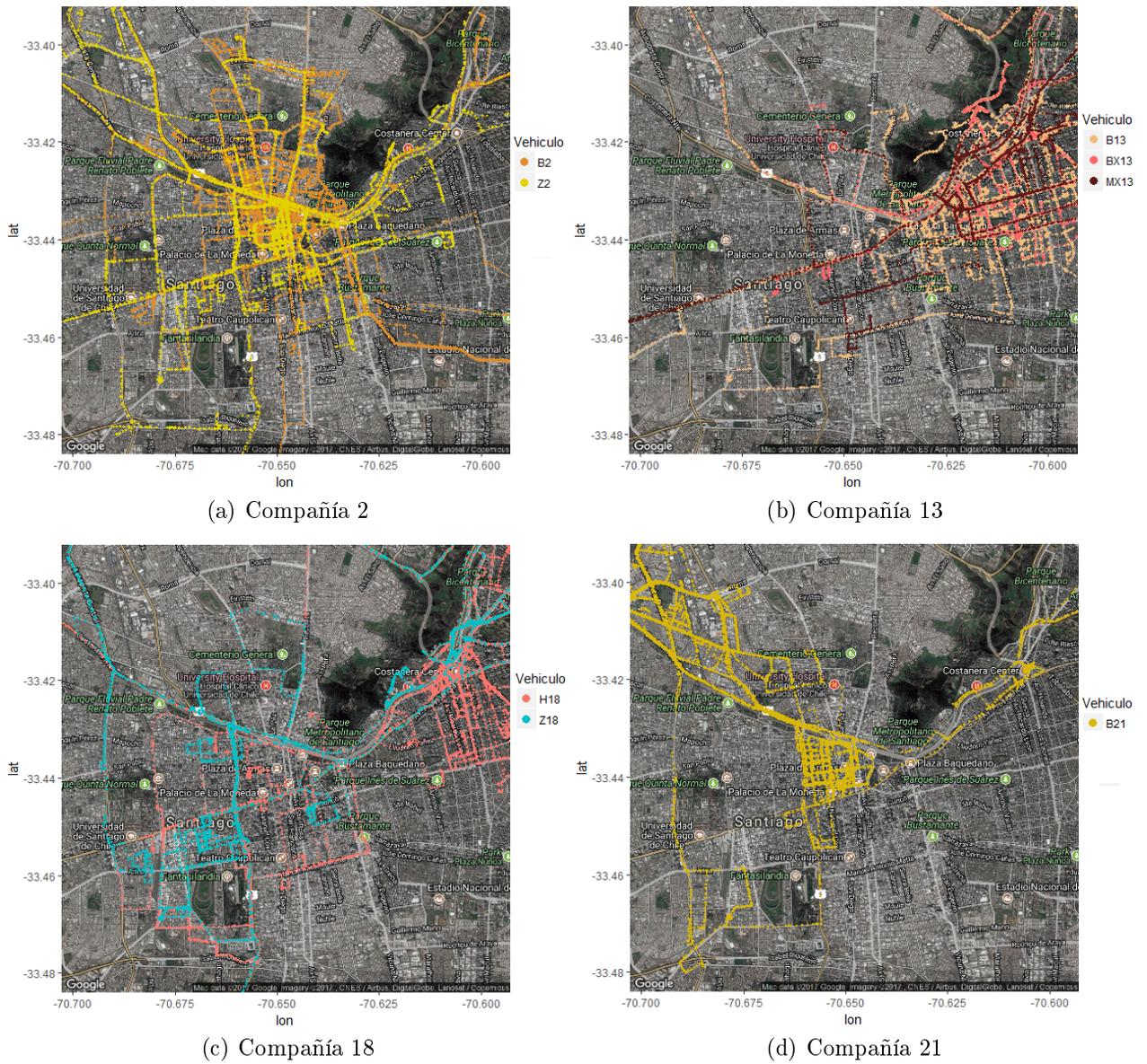


Figura 4.9: Pulsos GPS por Vehículo

4.5. Preprocesamiento

Posterior a la consolidación de la base de datos, se crearon nuevos atributos para ayudar a determinar rutas de interés, con un inicio y final bien identificado. Para este propósito, se define una “ruta” como una secuencia de pulsos GPS, donde el atributo Vehículo de un pulso es idéntico al Vehículo del pulso anterior, con un intervalo entre pulsos pequeño. Para determinar bien esto último, se crea la variable dT en la base de datos, definida como la diferencia (en segundos) entre el Timestamp del pulso respectivo y el del pulso anterior en caso que éstos pertenezcan a un mismo vehículo. Vale decir, el valor del atributo dT para el registro i -ésimo se calcula mediante la ecuación $dT_i = TS_i - TS_{i-1}$.

Creada la variable dT , se debe decidir cuál será el límite o umbral que define el término de una ruta en una secuencia de pulsos GPS. En la literatura, Alvares et al. (2007) proponen una metodología para identificar rutas en una secuencia de datos GPS mediante el conocimiento previo de posibles “paradas”. Esto no es posible debido que a priori no se sabe con certeza a qué emergencia fue cada vehículo en cada ruta. Por otra parte, Cao y Krumm (2009) tienen una base de datos de datos GPS con una resolución de 1 pulso cada segundo. Los autores definen en primera instancia una ruta como una secuencia de pulsos GPS donde cada GPS tiene a lo más 10 segundos o 100 metros de distancia con el pulso anterior. Además, se decide aplicar un segundo criterio, en el cual se determina que dos puntos pertenecen a una misma ruta si su distancia es menor a 30 metros. El criterio se baja a 10 metros en caso de que el ángulo entre los puntos consecutivos sea mayor a 10° . Fathi y Krumm (2010) desarrollan una metodología para encontrar intersecciones en una red urbana a partir de datos GPS que se almacenan con una resolución de 1 pulso cada 5 segundos. Los autores definen una trayectoria como una secuencia de datos GPS con velocidades razonables acorde a ellos (mínimo 5mph, máximo 90mph) o con un gap de a lo más 10 segundos entre pulsos. Por último, Giannotti et al. (2011) realizan minería de datos GPS de un sistema de arriendo de autos, en el cual los usuarios pagan según los kilómetros recorridos. Los datos GPS de cada auto se registran debido a que cada uno lleva un aparato GPS instalado para evitar robos. Los pulsos GPS se emiten cada 30 segundos y los autores definen una ruta como una secuencia de GPS usando un umbral de 30 minutos de separación máxima entre registro.

Al analizar los distintos criterios en la literatura, se observa que cada autor escoge arbitrariamente un umbral o criterio, dependiendo de la resolución de sus datos. Es decir, si los GPS tienen una resolución alta (1 por segundo) o baja (1 por minuto) se espera un umbral más pequeño (1-10 minutos) o bien un umbral menos restrictivo (10-60 minutos).

Aparte de lo anterior, hay que considerar que en el caso de la base de datos de este trabajo, las señales de los pulsos no siempre se enviaron cada 10 segundos. En algunos casos esto varía en unos segundos debido a errores desconocidos de los aparatos GPS. No obstante, esta es la minoría de los casos ya que aproximadamente el 80 % de los pulsos se enviaron exactamente a 10 segundos del pulso anterior. En la figura 4.10 aparecen dos histogramas con los valores de dT para los registros cuyo vehículo es idéntico al del pulso anterior; en particular, la figura 4.10b ayuda a apreciar la distribución de los valores menos frecuentes. De éstos gráficos se observa que los valores más frecuentes de dT son 10, 9, 11 y 0. Cabe mencionar que posterior a este análisis se eliminaron los datos con $dT \in [0, 3]$ dado que se cree que estos registros corresponden a datos duplicados (posición idéntica al registro anterior).

Con respecto a aquellos datos con $dT < 0$ que no se muestran en los histogramas, al analizarlos se observa que dT varía entre -86391 y -16235, implicando que estos pulsos siempre corresponden al registro del inicio de una ruta distinta efectuada en otro día u hora que la ruta del pulso previo. Finalmente, como conclusión del análisis se determina definir una ruta como una secuencia de GPS de un mismo vehículo con valores de dT menores a 240 segundos y mayores a 0 segundos.

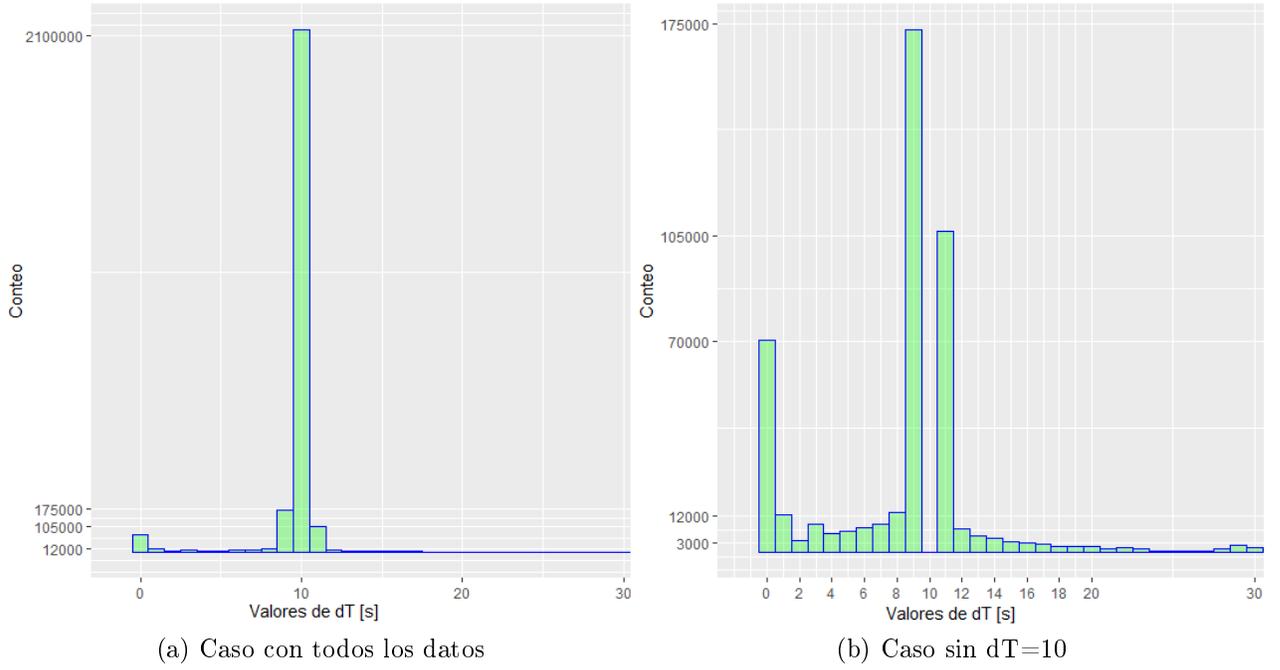


Figura 4.10: Histogramas de variable dT con valores entre 0 y 30

Después de crear la variable dT, se procedió a crear otra variable: la velocidad media entre dos pulsos V_{media} . Esta variable se calculó como el cuociente de la distancia y el tiempo transcurrido entre dos pulsos GPS. No obstante, calcular la distancia entre dos pulsos no es directo, debido a que la posición de los pulsos se encuentra especificada en Latitud-Longitud, i.e., ocupando el sistema de coordenadas WGS84. Para encontrar esta medida, se ocupó la función de distancia *VincentyEllipsoid*¹⁰ del paquete *geosphere* que se puede incorporar a la librería del software R. Esta función ocupa el modelo de la superficie terrestre del sistema WGS84 (esto es, una elipsoide con un radio horizontal mayor al vertical) para encontrar el valor del largo de la línea geodésica entre dos datos. Una línea geodésica consiste en una curva sobre una superficie que une dos puntos con el mínimo largo posible.

En la figura 4.11 se muestran todos los pulsos para distintos meses sobre un mapa satelital de Santiago. Los datos llevan un color según una paleta continua rojo-naranja-amarillo-verde proporcional al atributo V_{media} de cada registro. Cabe mencionar que si bien V_{media} describe la velocidad promedio entre dos pulsos; es decir, corresponde a una propiedad del tramo entre tales pulsos; por motivos prácticos, se asignaron los valores de V_{media} a cada pulso. En síntesis, el valor en el registro i -ésimo considera la distancia recorrida y el tiempo transcurrido con respecto al pulso anterior (ver ecuación 4.2).

$$V_{media_i} = \frac{\text{distancia}(Lat_i, Lat_{i-1}, Long_i, Long_{i-1})}{TS_i - TS_{i-1}} = \frac{dX_i}{dT_i} \quad (4.2)$$

¹⁰<https://www.rdocumentation.org/packages/geosphere/versions/1.5-5/topics/distVincentyEllipsoid> visitado 26 de septiembre 2017.

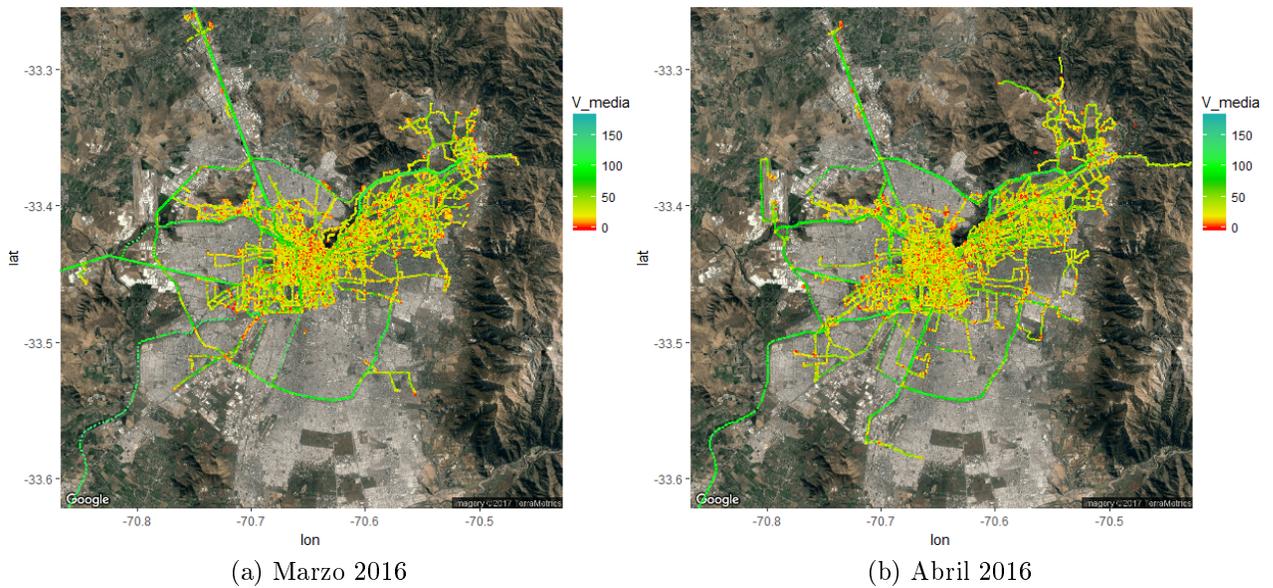


Figura 4.11: Valores de variable V_{media} en distintos meses

La variable V_{media} se establece con el propósito de tener información más detallada de las rutas de los vehículos. En particular, es preferible ocupar esta variable por sobre la variable “Velocidad Instantánea” (V^{inst}) que ya viene en la base de datos original. En primer lugar, dado que esta velocidad en realidad consiste en cuánto marca el velocímetro en el instante que se emite el pulso GPS, significa que no es representativa de lo que ha ocurrido en los últimos segundos. Como ejemplo, en el caso que un vehículo haya cambiado su posición en los últimos diez segundos, pero que en el instante de emisión del pulso GPS se encuentre detenido, implica que presentará una velocidad instantánea igual a 0. Sin embargo, la velocidad media entre pulsos será distinta de cero. En la figura 4.12 esto se refleja en el borde izquierdo del gráfico de dispersión, lugar donde se encuentran todos aquellos pulsos que cumplen las condiciones del ejemplo recién explicado. En segundo lugar, dado que los datos tienen una resolución alta, al ser la frecuencia de los pulsos igual a uno cada diez segundos, no significa una gran pérdida de información descartar la V^{inst} y ocupar la V_{media} .

Una vez que las rutas y la variable V_{media} están definidas, se procede a buscar rutas que contengan posibles errores a partir de velocidades muy altas en alguno de sus registros. De las 16.628 rutas identificadas, 9 contienen un dato con $V_{media} > 200km/h$. Al borrar el dato con velocidad alta y recalculer dT , dX y V_{media} aparece un nuevo registro dentro de estas rutas con una velocidad alta. Al dibujar una de estas rutas sobre un mapa (figura 4.13) se aprecia que esto se debe a que hay dos sub-rutas, donde una termina mientras simultáneamente comienza otra en otro lugar apartado. Esto puede deberse a errores al momento de registrar el *Timestamps* de los pulsos en una de las dos rutas. Dado que este error no se puede reparar borrando un par de datos, es preferible borrar por completo una de estas dos sub-rutas.

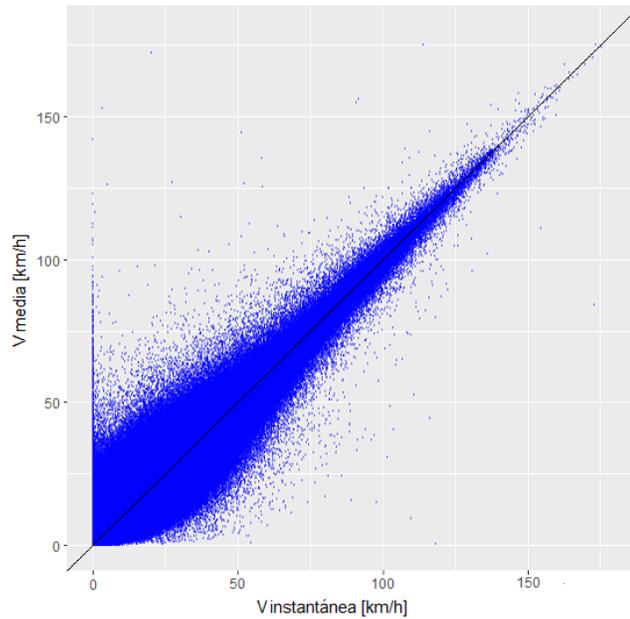


Figura 4.12: Gráfico de dispersión Velocidad media vs instantánea

En cada caso se borró la sub-ruta que tuviera un menor desplazamiento o una menor cantidad de registros. Hacer esto implicó eliminar menos de 900 pulsos de la base de datos, lo que corresponde a una pérdida poco significativa de información. Posteriormente, se buscaron datos con $V_{\text{media}} > 160 \text{ km/h}$ y se encontraron 22 pulsos, de los cuales 15 eran consecutivos en una carretera. De los otros siete, seis consisten en errores de registro puntuales, donde se aprecia una ruta bien definida con un pulso que salta a un lugar aleatorio. Al borrar estos errores puntuales en las seis rutas pertinentes y recalcular la variable V_{media} , se obtienen valores aceptables en las rutas.

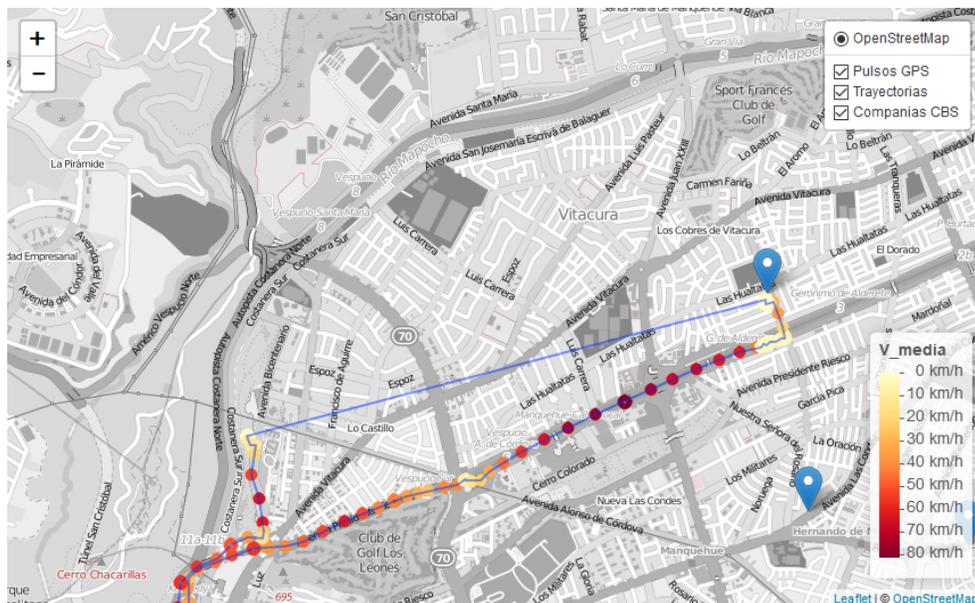


Figura 4.13: Ruta recorre distancia (línea azul) en muy poco tiempo

Finalmente, el último caso con una velocidad muy alta se consideró factible y al visualizar la ruta se aprecia que coincide con una falla común de los aparatos GPS que ocurren al estar adentro de un túnel o en un subterráneo. Este tipo de fallas observadas suceden mayoritariamente en el Túnel San Cristóbal (ver ejemplo en figura 4.14) y en algunos tramos de la autopista Costanera Norte (figura 4.15). Al perderse la señal se da un mayor espaciamiento temporal entre los registros. En este último caso, que coincide con una falla de este tipo, el vehículo avanzó 923 metros en 19 segundos (figura 4.16) resultando en una V_{media} de 175 km/h. Este valor máximo corresponde a una ruta del vehículo BX11 transitando por la autopista Costanera Norte, altura Lo Curro, un lunes a las 2:48pm.

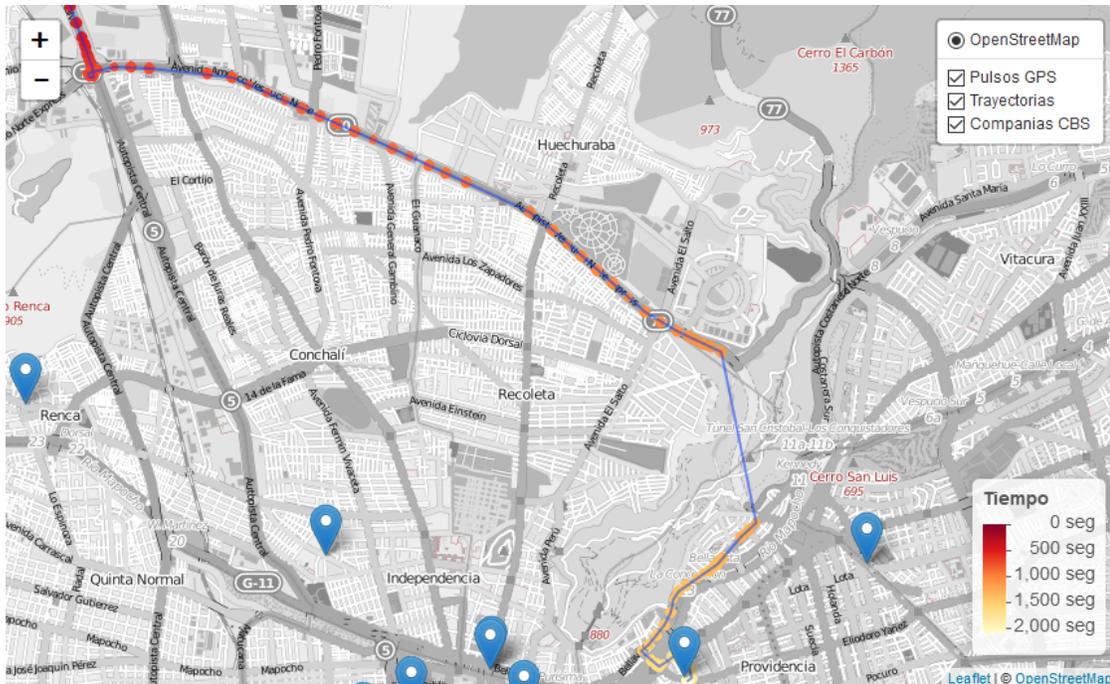


Figura 4.14: Ruta pasa por túnel San Cristóbal, interrumpiendo registro GPS

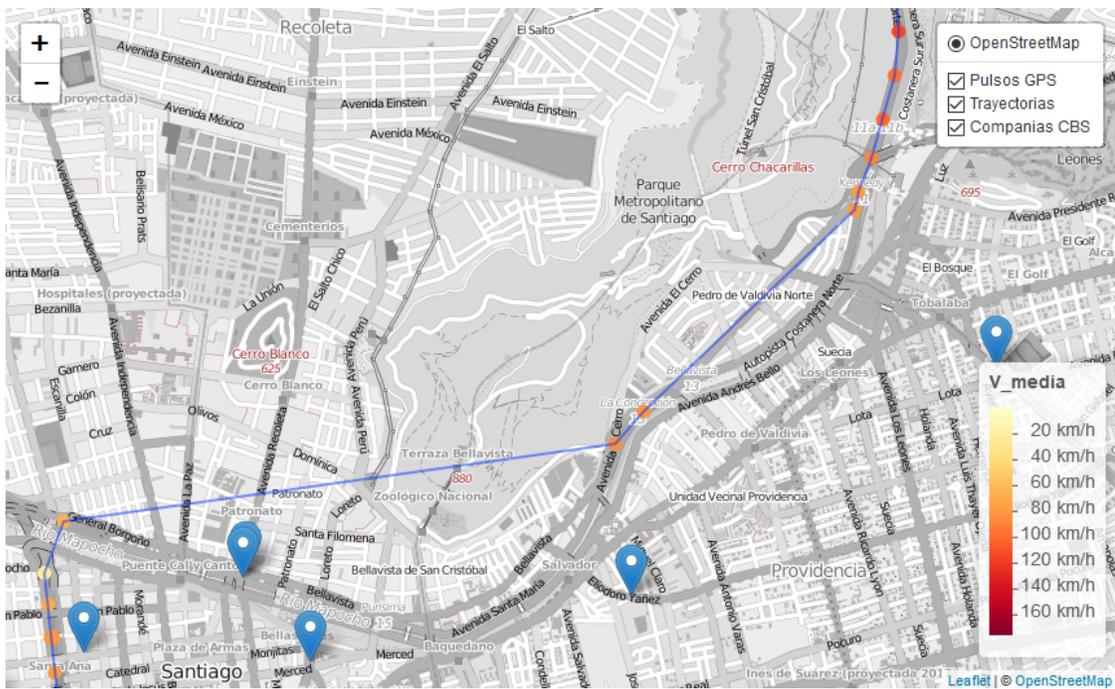


Figura 4.15: Datos no se registran al estar en una carretera subterránea



Figura 4.16: Registro con máxima Vmedia a 175 km/h, pulso color rojo oscuro

4.6. Algoritmos de *Clustering*

Continuando con la limpieza de datos, se decide borrar aquellas rutas de corta duración, con menos de 10 pulsos, equivalente aproximadamente a un minuto de duración. Finalmente, se tiene una base de datos de 2.469.889 registros (es decir, 130.000 pulsos menos que la base de datos original descargada de Fleetup). El siguiente desafío consiste en automatizar la detección de aglomeraciones de pulsos inservibles que se presentan frecuentemente al visualizar las rutas (ver figura 4.17). Estos datos no son de utilidad ya que no corresponden a un vehículo de bomberos viajando hacia una emergencia, por lo que es necesario filtrarlos.



Figura 4.17: Nube de Pulsos GPS a la mitad y al extremo de las ruta

Estas “nubes” de pulsos se presentan cuando un vehículo del CBS se encuentra detenido y con el motor encendido, provocando que los aparatos GPS estén activos enviando pulsaciones cada diez segundos con un error aleatorio. En consecuencia, los registros de estas aglomeraciones se caracterizan por estar confinadas en un radio limitado, con pulsos que además no presentan anomalías al analizarlos secuencialmente. Es por esto último que no fueron eliminados mediante el análisis previo ocupando las variables dT y V_{media} que solo capturan el comportamiento de un pulso con respecto al pulso anterior.

Al observar las rutas, es común encontrar nubes de puntos en su principio y final, sin embargo, también pueden presentarse al medio de las rutas. Esto puede deberse a que la máquina fue a una emergencia, no se apagó el motor, y posteriormente fue a otra emergencia o se devolvió a su estación para ser guardada. Esto último es lo que ocurre en la ruta de la figura 4.17, donde se ve que la máquina se detiene por un tiempo en un lugar y luego avanza hasta llegar a su compañía (cuya posición se visualiza con un marcador azul).

Para identificar las nubes o *clusters* se ocuparán algoritmos de agrupamiento o *clustering*, los cuales discriminan y agrupan datos con propiedades similares. Primero, se ocupará un algoritmo para identificar los extremos. Después se aplicará otro que identificará las nubes presentes a mitad de una ruta.

El algoritmo de *extremos* comienza tomando uno de los dos extremos de una ruta y verifica si los primeros diez datos están contenidos en un *cluster*. Para esto, calcula la “posición promedio” o *centroide* de estos diez datos. Esto último es un simple promedio aritmético de las coordenadas WGS84 proyectadas en sistema UTM (que ocupa un sistema cartersiano x-y). Una vez calculado el centroide, el algoritmo chequea si al menos ocho de los diez datos están a una distancia cercana (50 metros o menos) del centroide. Para esto, se ocupa la distancia euclideana, lo que es posible gracias a que las posiciones están en coordenadas UTM. Si los pulsos resultan estar cerca, el algoritmo marca estos diez datos como un cluster y comienza su segunda etapa. En esta etapa, se verifica si el *cluster* es más grande que los primeros diez datos. Para esto, se analizan secuencialmente los datos según su *Timestamp*, midiendo la distancia al centroide y al último dato agregado al *cluster*. Si una de estas dos distancias resulta ser menor a 20 metros, el registro se agrega al *cluster* y se recalcula la posición del centroide; de otra forma, se detiene la búsqueda y se procede a realizar el mismo método en el otro extremo de la ruta.

Algoritmo de Extremos (Ruta R): Encuentra *clusters* en extremo de rutas

```

1 Parámetros: int  $n = 10$ , int  $m = 8$ , int  $d_i = 50$ , int  $d_f = 20$ 
2 Calcular posición del centroide de primeros  $n$  datos en uno de los extremos  $\rightarrow P_C$ 
3 if Al menos  $m$  de los  $n$  pulsos están a menos de  $d_i$  metros de  $P_C$  :
4     Marcar los  $n$  datos como parte del cluster extremo  $X$ 
5 int  $k = n$ ;
6 while  $k \leq$  numero de pulsos en  $R$ 
7      $k = k + 1$ ;
8     if distancia(punto  $k$ -ésimo y  $P_C$ ) <  $d_f$ 
9         Marcar  $k$  como parte del cluster  $X$ ;
10    else if distancia(pulso  $k$  y el pulso  $k - 1$ ) <  $d_f$ 
11        Marcar  $k$  como parte del cluster  $X$ ;
12    else
13        break;
14    Recalcular posición del centroide del cluster  $X \rightarrow P_C$  ;
15 Repetir procedimiento en el otro extremo.
```

Como ejemplo se ocupa una ruta que termina en una compañía de bomberos visualizada en la figura 4.18

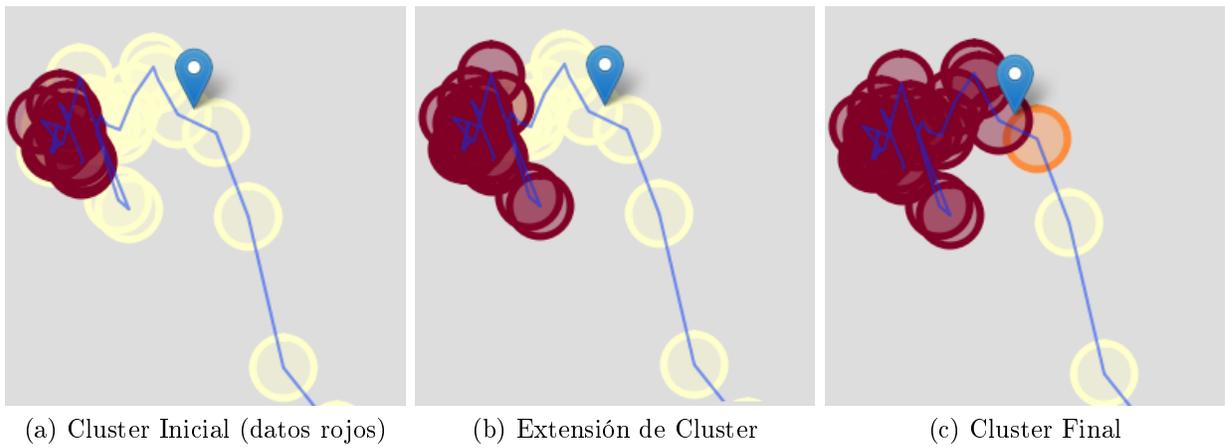


Figura 4.18: Algoritmo para clusterizar extremos, en naranja el último dato agregado

Adicionalmente, es posible visualizar el *cluster* del ejemplo de la figura 4.18 en un cubo tridimensional con las dimensiones espaciales x-y (en metros) junto al tiempo (en segundos) como el eje Z o altura de dicho cubo (figura 4.19).

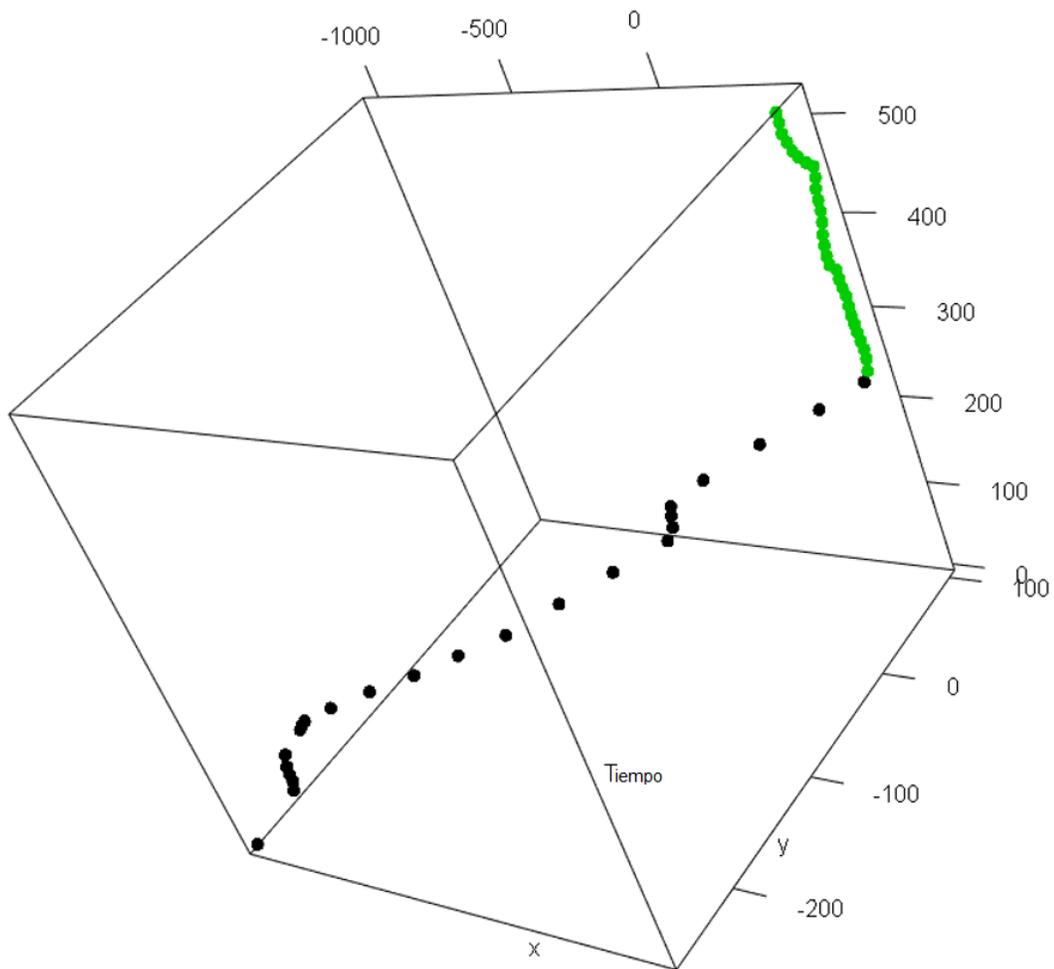


Figura 4.19: Cubo con ruta de ejemplo anterior. En verde se destaca cluster encontrado

Si bien con este procedimiento se han filtrado las aglomeraciones al principio y fin de cada ruta, se requiere otro algoritmo que se encargue de las nubes de puntos presentes a mitad de las rutas. Un problema adicional a considerar a la hora de hacer este algoritmo es no eliminar las detenciones en semáforos. Es decir, evitar falsos positivos donde el algoritmo identifique como *cluster* detenciones normales en intersecciones semaforizadas. Para lograr esto, se crea un algoritmo de agrupamiento llamado *snake clustering*.

El algoritmo *snake clustering* comienza tomando los datos de la ruta de interés y los enumera según su *Timestamp* o tiempo registrado. Dependiendo de dos parámetros, d_{max} y σ , el algoritmo analiza primero si el trío de datos número 1, $(1 + \sigma)$ y $(1 + 2\sigma)$ están a menos de d_{max} metros de distancia entre cada uno de ellos, e.g., si $d_{max} = 60$ y $\sigma = 6$, chequea si el dato número 1 está a menos de 60 metros del dato 7 y del dato 13, y si los datos 7 y 13 se encuentran a menos de 60 metros entre sí. Si esto no se cumple, chequea esta condición con los datos 7, 13 y 19 y así consecutivamente, “saltándose” cada seis datos para verificar la distancia entre éstos. En caso que para un trío de datos sí se cumpla que están a menos de 60 metros entre ellos, el algoritmo clasifica a los datos entre el menor y el mayor del trío como un *cluster*. Como ejemplo, si se encuentra que los datos 7, 13 y 19 están a menos de 60 metros entre ellos, el algoritmo determina que todos los datos enumerados entre el 7 y el 19 forman parte de un *cluster* (ver ejemplo en figura 4.20).

A partir de un *cluster* identificado, el algoritmo intenta agregar pulsos a éste, al continuar verificando si en el siguiente trío de datos (13, 19 y 25 en el ejemplo anterior) se cumple la misma condición de distancia anterior. Una vez que esto no ocurre, se concluye que ha terminado el cluster y se reinicia la búsqueda de *clusters* comenzando desde el dato con menor *Timestamp* que no forma parte del último grupo recién identificado. Por ejemplo, si se encuentra que los tríos 7, 13, 19 y 13, 19, 25 cumplen la condición de distancia, pero no así el trío 19, 25, 31; entonces se marcan los datos 7 al 25 como un *cluster* y se reinicia la búsqueda empezando en el dato 26 (tomando el trío 26, 32 y 38). En caso que se llegue al final de la ruta mientras se busca agregar datos a un *cluster*, se marcan los datos finales como parte de la última aglomeración encontrada.

Cabe mencionar que el parámetro σ se fijó como 6 asumiendo que un EV nunca estará más de dos minutos detenido una intersección debido a semáforos o condiciones de congestión en caso de ir hacia una emergencia. Esto debido a que, en general, entre cada pulso transcurren 10 segundos, por lo que al imponer $\sigma = 6$ el algoritmo analiza posibles *clusters* en tríos de datos donde el primer y último dato se encuentran aproximadamente a dos minutos de distancia temporal.

Una vez analizados todos los datos en una ruta e identificados sus *clusters*, se realiza un procedimiento similar al del algoritmo de extremos, en el que se busca expandir las aglomeraciones al verificar si los pulsos vecinos a cada *cluster* están lo suficientemente cerca como para considerarlos también. Para esto, se calcula el centroide de cada cluster identificado y se observa si los datos vecinos (anterior y siguiente al primer y último dato del *cluster*, respectivamente) están a menos de 60 metros del centroide. Es digno mencionar que no se recalcula el centroide a medida que se agregan datos a un *cluster*.

Snake Clustering (Ruta R): Encuentra clusters en rutas

```
1 Parámetros: int  $\sigma = 6$ , float  $d_{max} = 60$ 
2 Enumerar los datos según Timestamp desde 1 hasta  $N$ 
3 int  $a = 1$ ;
4 int  $b = a + \sigma$ ;
5 int  $c = b + \sigma$ ;
6 int  $i = 1$ ;
7 boolean agrandar cluster = False;
8 while  $c \leq N$  (numero de pulsos en  $R$ )
9     if distancia(datos número  $a$ ,  $b$  y  $c$ ) <  $d_{max}$ 
10         Marcar datos entre  $a$  y  $c$  como parte de cluster  $i$ ;
11         agrandar cluster = True;
12     else if agrandar cluster == True
13         agrandar cluster = False;
14          $i = i + 1$ ;
15          $b = b + 1$ ;
16          $c = c + 1$ ;
17      $a = b$ ;
18      $b = c$ ;
19      $c = c + \sigma$ ;
20 if agrandar cluster == True
21     Marcar datos entre  $a$  y  $N$  como parte de cluster  $i$ ;
22 for cada cluster  $k$  encontrado en ruta  $R$ 
23     Calcular el centroide  $P_C(k)$  para cluster  $k$ 
24     while se agregó un nuevo dato en la iteración anterior
25         Identificar primer y último dato de cluster:  $k_1$  y  $k_n$ 
26         if existe dato  $k_1 - 1$  y no pertenece a otro cluster
27             if distancia( $P_C(k)$ , pulso  $k_1 - 1$ ) <  $d_{max}$ 
28                 Agregar dato  $k_1 - 1$  a cluster como su primer dato;
29         if existe dato  $k_n + 1$  y no pertenece a otro cluster
30             if distancia( $P_C(k)$ , pulso  $k_n + 1$ ) <  $d_{max}$ 
31                 Agregar dato  $k_n + 1$  a cluster como su último dato;
```

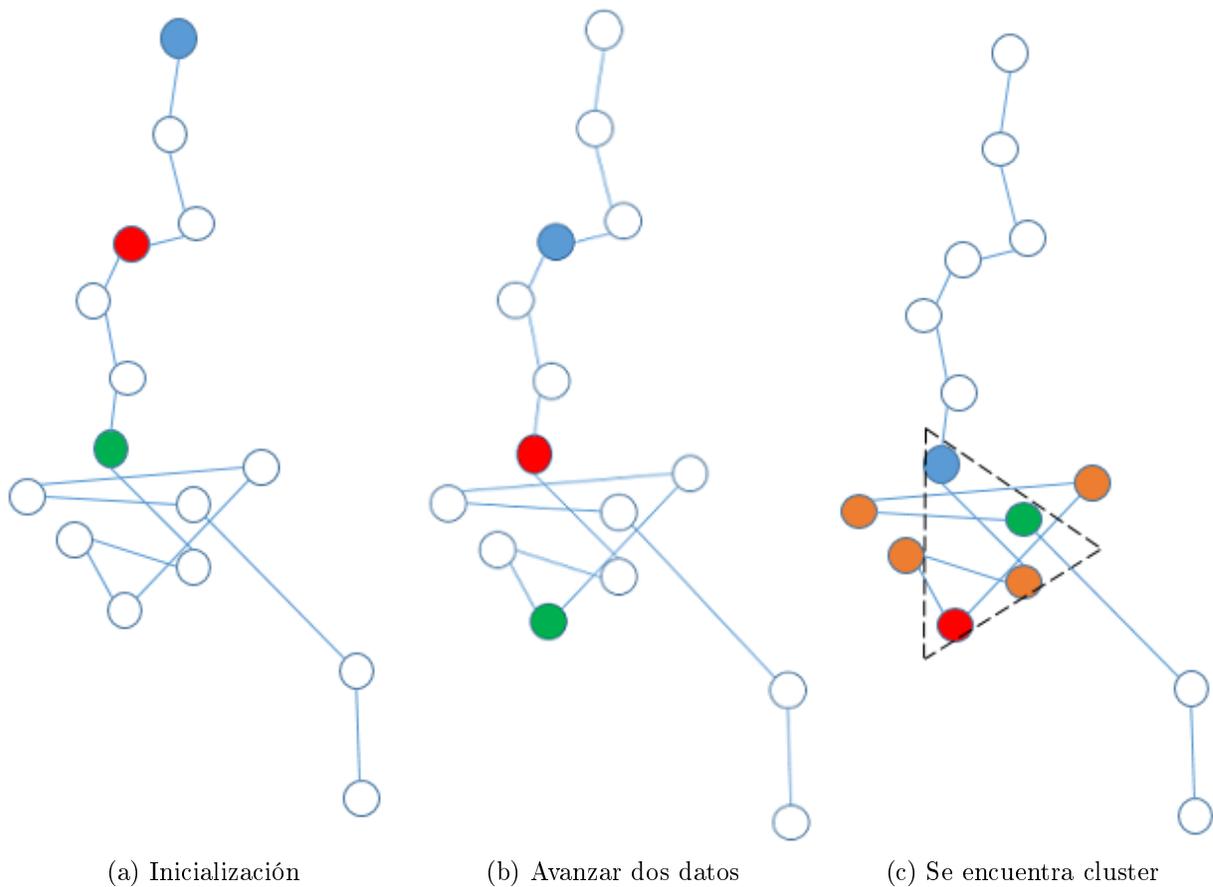
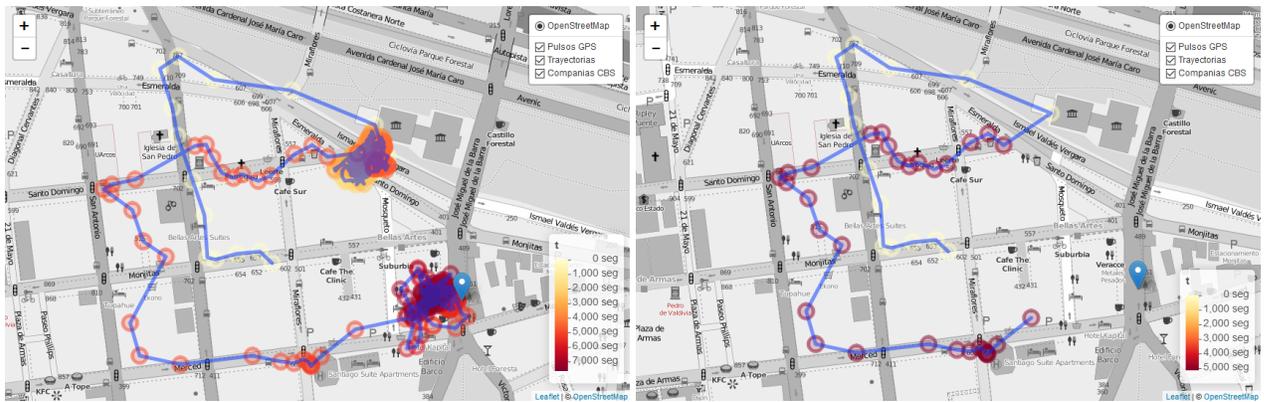


Figura 4.20: Ejemplo Algoritmo *Snake Clustering* con parámetro $\sigma = 2$

Luego de haber filtrado las nubes de puntos mediante el algoritmo de extremos, se pasó de tener 2.469.899 pulsos (11.422 rutas) a 1.572.343 (7.691 rutas). Al ocupar el algoritmo *snake clustering*, se terminó con una base de datos de 835.403 datos (7.498 rutas). Cabe destacar que estos algoritmos pueden haber identificado como un *cluster* datos que sí eran de utilidad ya que se ocuparon parámetros predefinidos manualmente, los que en la mayoría de los casos dieron los mejores resultados al visualizar el clustering de distintas rutas, teniendo precaución en el caso del *snake clustering*, de no borrar detenciones en intersecciones semaforizadas. Sin embargo, esto no está asegurado para todas las rutas de la base de datos. Como ejemplo, se pudo haber ocupado $d_{max} = 30$, lo que implica una mayor exigencia para identificar un grupo de pulsos como un *cluster*, pero quizá esto hubiera significado la permanencia de aglomeraciones de puntos en algunas rutas.

En síntesis, no se realizó una calibración de los parámetros de los algoritmos de aglomeración, lo que sería posible mediante un análisis con algunas de las métricas que cuantifican la eficacia de algoritmos de clasificación, por ejemplo, aquellas que se discuten en el trabajo de Majnik y Bosnić (2013) que analizan distintas maneras de analizar una tabla de contingencia, la cual ayuda a enumerar predicciones erróneas v/s predicciones acertadas. No obstante, un análisis de este tipo no está dentro de los objetivos de esta tesis, por lo que no se llevo a cabo.



(a) Ruta antes

(b) Ruta después

Figura 4.21: Ruta de figura 4.17 antes y después de filtrar datos

4.7. Cruce con actos y filtro según fechas de interés

A esta altura del procesamiento de los datos, se decide realizar un corte temporal a los datos, de forma que éstos contengan solamente registros durante días laborales (lunes a viernes, sin contabilizar días feriados) entre marzo y noviembre. Esto significa eliminar los datos de diciembre 2015, enero y febrero 2016. Esta decisión se realiza de forma que los tiempos de respuesta de los EVs en la base de datos representen los de un día laboral promedio. Se eliminan los meses diciembre, enero y febrero ya que no presentan un patrón de viajes normal, debido a que el resto del año los estudiantes de todas las edades asisten a clases. En total, posterior eliminar los datos de estos tres meses se tiene una base de datos de 485.794 (7.201 rutas), de los cuales 365.447 (5.293 rutas) corresponden a días laborales. Entre los meses de marzo y mayo existen 1.584 actos registrados.

Una vez realizada la limpieza de datos descrita anteriormente, se tiene una base de datos con las trayectorias de vehículos de emergencia en movimiento. Como último requisito para poder ocupar posteriormente estos datos, es necesario que estas rutas consistan en vehículos yendo hacia alguna emergencia. Para lograr esto, es posible cruzar la base de datos GPS final con el registro de Actos. Los Actos corresponden a emergencias registradas por el CBS con su dirección, posición (Latitud-Longitud), fecha y hora.

Mediante un algoritmo, se identificaron aquellas rutas que terminan cerca a alguna emergencia acorde al registro de Actos. Se determinó que aquellas rutas cuyo último pulso se encuentre a menos de 110 metros de un acto y que hayan ocurrido durante el mismo día consisten en viajes hacia dicha emergencia. Puede haber más de una ruta asociada a una misma emergencia (varios vehículo atendiendo una misma emergencia), sin embargo no puede ocurrir que hayan varias emergencias asociadas a una misma ruta. En el segundo caso, se asocia la ruta a aquel acto que esté más cerca de su último pulso. Con estos criterios es posible tener una base de datos de rutas hacia emergencias. Esta base de datos final contiene 16.362 pulsos con 360 rutas distintas. De estos datos, solamente 8.010 son pulsos que pasan por la zona de interés (i.e., la zona codificada en PARAMICS), formando 119 rutas en total.



Figura 4.22: Ejemplo de fin de rutas (marcadores naranjos) asignadas a acto (marcador rojo)

4.8. Síntesis de procesamiento de datos GPS

Para procesar y preparar los datos GPS, tanto para su calibración como para su uso por parte de los analistas a cargo del grafo de velocidades que ocupa COMMANDER, es necesario preocuparse de que la base de datos contenga GPS cada 10 segundos (o lo que establezca el proveedor). Borrar aquellos datos errados con pulsos duplicados o temporalmente muy cercanos (en este caso, 0 a 3 segundos). Ocupando el espaciado temporal como una variable, es posible identificar las rutas como secuencias de GPS sin saltos temporales muy grandes (se determinó el umbral de 240 segundos en este trabajo). Después, se crea la variable que cuantifica la velocidad entre pulsos, con ésta es factible identificar rutas con velocidades extremadamente altas. Utilizando herramientas de visualización, se pueden trazar estas rutas sobre un mapa de manera de determinar si dichas rutas tienen sentido, i.e., dada la calle o carretera, si la velocidad registrada es factible.

Posteriormente, se buscan filtrar las nubes de puntos provocadas por el ruido de los aparatos GPS en periodos donde el vehículo se encuentra estático. Para esto se propone ocupar dos algoritmos, uno pensado para filtrar aglomeraciones al inicio de las rutas y otro para identificar *clusters* a la mitad de éstas. Finalmente, dado que los datos se desean para analizar situaciones de emergencia, se requiere que la base de datos se cruce con el registro de actas para ubicar rutas que fueron efectivamente hacia alguna emergencia. Opcionalmente, si se desea se pueden llevar a cabo filtros por fecha o por zona si existen periodos del año o ubicaciones de interés particular para el estudio.

En la tabla 4.3 se muestran la cantidad de datos que permanecen luego de cada etapa del procesamiento de datos, junto al porcentaje de datos que quedan con respecto al tamaño de la base de datos “cruda”. En la última columna se dan a conocer la cantidad de rutas de cada conjunto de datos, identificadas acorde al criterio de los 240 segundos explicado antes.

Tabla 4.3: Cantidad de datos y rutas después de cada etapa de procesamiento

Corte	Datos	Porcentaje	Rutas
Base de datos “cruda”	2.637.191	100 %	16.628
Post procesamiento inicial	2.469.899	93.7 %	11.422
Post algoritmo de extremos	1.572.343	59.6 %	7.691
Post algoritmo <i>snake clustering</i>	835.403	31.7 %	7.498
Post filtro según mes (marzo a mayo)	485.794	18.4 %	7.201
Post filtro según día laboral	365.447	13.9 %	5.293
Post cruce con registro de actos	16.362	0.6 %	360
Post filtro según ubicación (zona centro)	8.010	0.3 %	119

Capítulo 5

Calibración en un caso ejemplo

Con el fin de simular una situación real de intervención de vehículos de emergencia, es necesario ocupar datos de distintas fuentes para realizar una calibración adecuada. Para este ejemplo, se buscó una expedición hacia una emergencia en el listado de Actos con algún vehículo en el registro de GPS. Finalmente, entre los casos encontrados, se pidió el registro audiovisual de la expedición hacia la emergencia, en caso de poder corroborar la existencia efectiva de una situación de emergencia.

En un principio, para realizar el cruce entre las bases de datos de actos y GPS, se revisaron los actos del mes de junio del año 2017 y se ocupó el paquete *geopy*¹ junto con una API de Google Maps para identificar la latitud y longitud de la dirección de las emergencia de cada acto. Este proceso de asignación de direcciones a coordenadas geográficas se conoce como *geocoding*. Con las coordenadas de cada acto, se ocupó un algoritmo que cruza las posiciones de los actos con los últimos pulsos GPS de cada ruta del mes de junio en la base de datos GPS. De esta forma, se filtran aquellos viajes que hayan terminado cerca de una emergencia. Para esto, se consideró como criterio de cercanía estar a menos de 500 metros y con una diferencia temporal menor a dos horas. Finalmente, se consideraron aquellos viajes que se encontraran mayoritariamente en la zona centro de Santiago, durante un día laboral y con una emergencia atendida en la misma zona.

De las rutas encontradas, cuatro cuentan con grabaciones de la expedición realizada. Al revisar los videos disponibles, se determinó cuales serían de utilidad dependiendo de sus propiedades. Principalmente, se requiere: que la grabación corresponda a la ruta hacia la emergencia y no de regreso a la compañía; que se observen interacciones con otros vehículos transitando; y que la ruta no sea de duración muy corta. Considerando estos criterios, se determinó que solamente el material capturado por una máquina Q8 (portaescalas de la octava compañía del CBS) sería útil. Este material consiste en cuatro videos, cada uno capturado desde el mismo vehículo con un ángulo diferente: frontal, trasero, piloto (lado izquierdo) y copiloto (lado derecho). La dirección del acto es Bellavista altura Capellán Abarzúa, comuna de Providencia, coordenadas WGS84 (-33.4319, -70.6291). En la figura 5.1 se dibujan los pulsos GPS registrados cada 10 segundos. En total, son 25 pulsos correspondientes a cuatro minutos y diez segundos, con una distancia recorrida por el vehículo de 2400 metros.

¹<https://github.com/geopy/geopy>

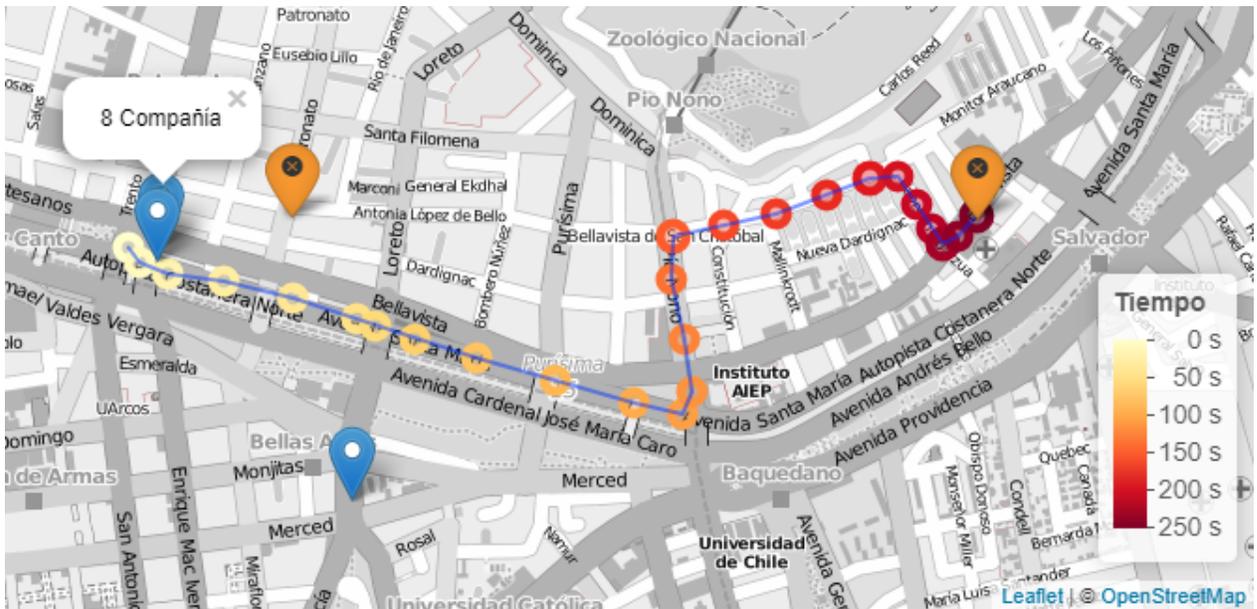


Figura 5.1: Ruta máquina Q8 - 15:18 lunes 19 de junio 2017

Luego de determinar qué ruta simular, se codificó en PARAMICS la red correspondiente a aquellas calles transitadas por el vehículo Q8 junto a las calles perpendiculares cruzadas en cada intersección. Adicionalmente, se codificó la red de semáforos con sus programaciones y las propiedades geométricas de las calles, entre otras cosas, siguiendo el procedimiento explicado en el capítulo anterior. En la figura 5.2 se observa la red codificada; las flechas azules indican el lugar desde donde parte la expedición el vehículo Q8 (flecha izquierda) y la ubicación de la emergencia o destino del viaje (flecha derecha). En rojo se muestran las intersecciones Santa María con Recoleta y con Pio Nono, a discutirse más adelante.

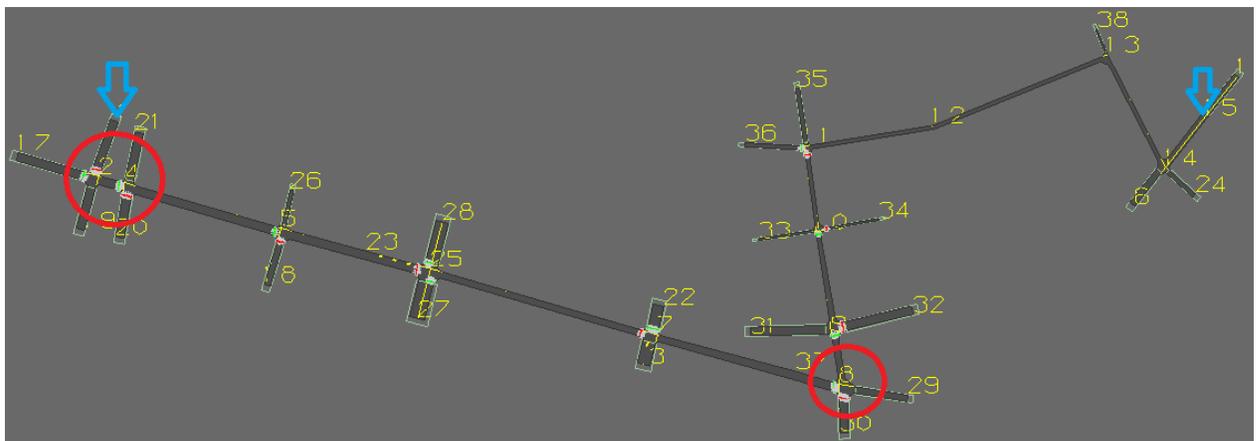


Figura 5.2: Ruta Q8 codificada en PARAMICS

5.1. Uso de videos para estimación de MRT y MTH

Con la red ya codificada, se buscó intersecciones en la ruta que estuvieran equipadas con cámaras de la UOCT y se solicitó videos en el mismo horario de la ruta durante un día laboral. Al final de la búsqueda, se logró obtener tres videos (ver Tabla 5.1). Con estos videos, se estimaron los dos parámetros más relevantes para calibrar en PARAMICS acorde a la literatura (Ratrout et al., 2015; Reza et al., 2016): MTH y MRT, i.e., *headway* deseado promedio y tiempo de reacción promedio.

Tabla 5.1: Videos UOCT de Intersecciones

Intersección	Fecha	Hora Inicio	Hora Fin	Duración
Pio Nono - Santa María	24 enero 2018	14:59:56	15:29:59	0:30:03
Santa María - Recoleta	27 diciembre 2017	14:59:58	15:29:59	0:30:01
Santa María - Recoleta	3 enero 2018	14:59:59	15:29:59	0:30:00

Para medir los tiempos de reacción, se observó la disipación de la cola formada debido a un cambio de luz roja a verde para el acceso que se observa desde la cámara (ver figura 5.3). La metodología se describe a continuación. Se mide el tiempo transcurrido entre el instante que cambia el estado del semáforo y comienza a moverse el primer vehículo. Luego, si es que existen vehículos detrás del primero y no se mueven antes que el primer vehículo, se mide el instante en que inicia el movimiento del segundo vehículo. Así hasta el tercer o cuarto vehículo. En general, la resolución de las cámaras no permiten medir con seguridad este dato para vehículos muy lejanos a la intersección en la cola.



Figura 5.3: Vista cámara UOCT: acceso Puente Pio Nono con Santa María hacia el sur

En el caso del primer video de Pio Nono con Santa María, la cámara muestra el mismo lugar durante los 30 minutos, sin embargo en el caso de los videos de Recoleta con Santa María, la cámara se mueve para mostrar distintos accesos. En las figuras 5.4, 5.5 y 5.6, se muestran imágenes de las cámaras con los distintos ángulos que se observan en las grabaciones.



Figura 5.4: Vista cámara UOCT: Santa María con Recoleta hacia el oriente



Figura 5.5: Vista cámara UOCT: Recoleta con Santa María hacia el sur



Figura 5.6: Vista cámara UOCT: Recoleta con Santa María hacia el norte

En el caso del video de Recoleta en el mes de diciembre, la cámara se emplaza primero hacia el oriente, luego de 14 minutos pasa a mostrar el sur durante 5 minutos para finalmente capturar el norte del acceso por los 11 minutos restantes. El otro video de Recoleta, grabado durante enero, inicia mostrando el sector norte durante 3 minutos, posteriormente se observa el oriente 7 minutos y termina hacia el sector sur durante los siguientes 20 minutos.

Con respecto a la estimación del *headway* deseado promedio, la metodología fue la siguiente: se busca pares de autos bajo condiciones ideales; estas son, ambos transitando la misma pista, con velocidad constante, en donde ninguno de los dos desea virar próximamente. Además, se requiere que no exista un espaciamiento muy grande, sino puede que el vehículo de atrás no esté afectado por el del frente. Por último, se necesita de un buen ángulo para corroborar estas cosas. Si se cumplen todas las condiciones para un par de vehículos, se procede a medir el espaciamiento temporal existente. Para medir esto, se pausa el video y se fija un punto de referencia en el pavimento, adelantando el video en instantes discretos, se anotan los tiempos en que el parachoque frontal de cada auto llega al punto de referencia. De esta manera, se obtiene una medición confiable del espaciamiento temporal entre los vehículos. Como supuesto, el espaciamiento medido bajo estas condiciones corresponde al *headway* deseado del conductor del auto de atrás.

5.1.1. Resultados Mean Reaction Time y Mean Target Headway

En la Tabla 5.2 se muestran los datos obtenidos para el MRT. Entre las razones para no contabilizar un auto reaccionando a un cambio en el semáforo están las siguientes: no se puede observar bien el momento en que éste empieza a avanzar, esto ocurre ya que a veces se tapa la vista desde el ángulo de la cámara, debido a la presencia de una moto, bicicleta o peatón entre las pistas de autos. Segundo, en el caso de que no corresponda al primer auto de una cola; si el auto avanza antes de que el auto que lo antecede avance, no se puede identificar su reacción al cambio en el semáforo. Por último, no se puede contabilizar en caso que no se pueda observar cuándo el semáforo cambia de luz roja a verde, ni cuando se observan los auto en cola desde atrás y no adelante. Por esto último, se tuvo que descartar los videos de Recoleta, ya que no se aprecia bien la disipación de la cola. En total, se identifica las reacciones de 104 vehículos, de las cuales 40 corresponden a vehículos que estaban primeros en la cola.

Tabla 5.2: Resultados MRT Pio Nono

Intersección	Vehículos	Número Datos	MRT [s]	Desv. Estándar [s]
Pio Nono - Santa María	Primeros	40	1.49	0.77
Pio Nono - Santa María	Otros	64	1.34	0.66
Total	-	104	1.4	0.71

Por otro lado, para estimar los espaciamientos temporales o *headways* deseados, más que colas en primer plano (grabadas de frente), se busca observar el perfil de autos transitando en situaciones de poca congestión. Por este motivo, al analizar los videos se ocupa solamente los videos de Recoleta. No es suficiente con tener poca congestión y un buen ángulo para observar el instante en que pasan los parachoques frontales de cada auto, también se necesita

observar una situación estable. Ésto es, se debe observar que se muevan a velocidad constante, i.e., que no se vea que uno está alejándose o acercándose al otro. Adicionalmente, no debe observarse que ambos están frenado debido a congestión visible más adelante. Más aún, se debe revisar que una vez cruzado en punto de referencia, ambos vehículos sigan todas las condiciones mencionadas. En el caso de Pio Nono, no es fácil apreciar pares de autos en estas condiciones, dado que hay pocos autos una vez lista la disipación de la cola del puente. En general, cuando se ha acabado la cola, se aprecian todos muy alejados del auto siguiente o en pistas distintas. En la Tabla 5.3 se muestran los resultados para cada uno de los videos.

Tabla 5.3: Resultados MTH Recoleta

Intersección	Fecha	Número Datos	MTH [s]	Desv. Estándar [s]
Santa María - Recoleta	27/12/17	20	1.43	0.24
Santa María - Recoleta	03/01/18	18	1.45	0.25
Total	-	38	1.44	0.24

En la figura 5.7 se grafica los diagramas de caja para los datos obtenidos. En cada uno se aprecian los percentiles 75 y 25 como los extremos superior e inferior de cada caja. La línea horizontal dentro de la caja marca la mediana o percentil 50. Los datos se incluyen como puntos ordenados en intervalos de 0.1 (segundos) de ancho. A partir de los diagramas se puede concluir la presencia de una mayor dispersión en las mediciones de los tiempos de reacción en comparación a los *headways*.

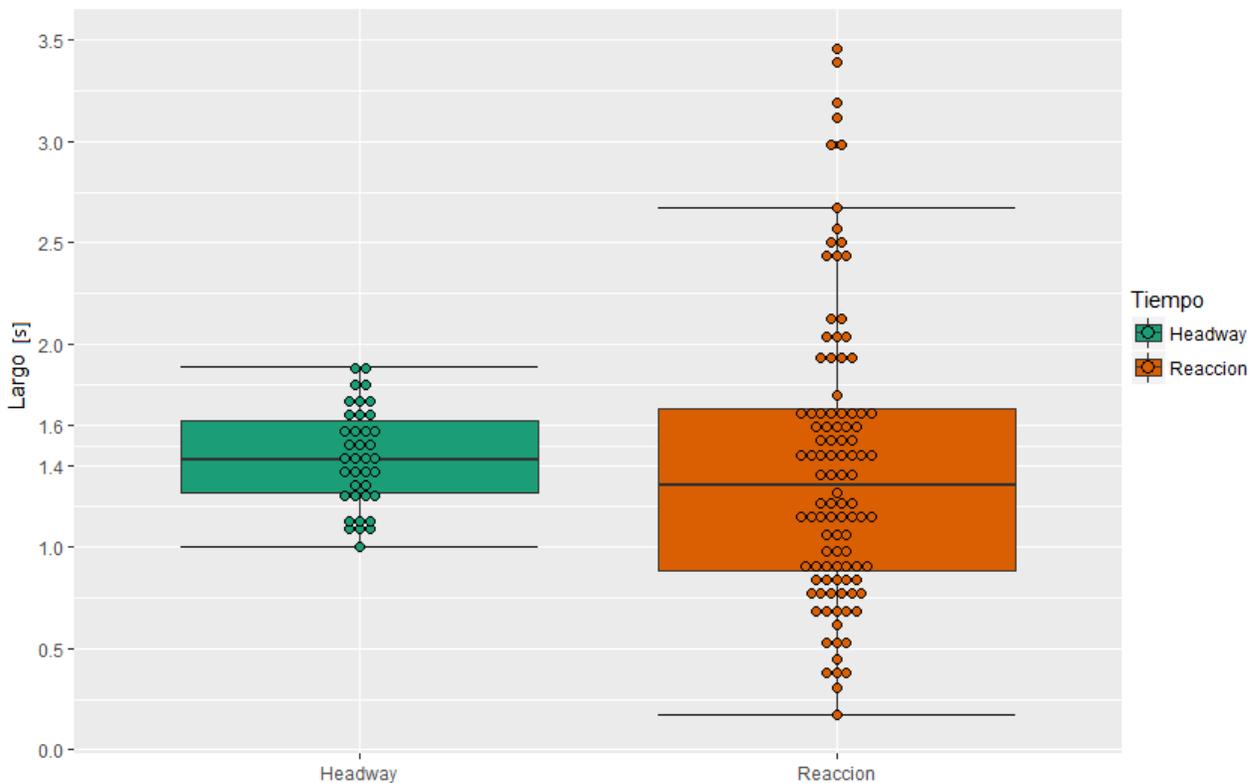


Figura 5.7: Diagrama de caja con datos de *headways* y tiempo de reacción

5.2. Estimación de parámetros para API

Una vez estimados los parámetros MRT y MTH (1.40 y 1.44 segundos, respectivamente) se busca estimar los parámetros útiles para los modelos de EV programados en la API. En particular, para este caso solamente se buscará estimar los parámetros α_a , β_a , δ_i (este último se supone igual a d_v^{min}) ya que no se observan autos moviéndose arriba de aceras y porque se observa que todos los vehículos cooperan eventualmente al verse enfrentados al vehículo de bomberos ($FC = 100\%$).

Para lograr estimar los parámetros, se observó las interacciones entre EV y POV presentes en los cuatro minutos que dura el viaje hacia la emergencia en cada uno de los videos grabados, desde los cuatro ángulos mencionados en la sección anterior. En la Tabla 5.4 se detallan los incidentes identificados junto al parámetro relacionado; adicionalmente, en las figuras 5.9 y 5.10 se muestran imágenes de dos eventos descritos en la tabla.

Para estimar las distancias de reacción, se miden los anchos de pista, el largo de las líneas de separación de pistas y el espaciamiento longitudinal entre las líneas. Esto se hizo en las calles recorridas por el vehículo Q8 donde se identificó reacciones de automóviles: la avenida Santa María y la calle Pio Nono. De esta forma, se encuentran las dimensiones como se muestran en la figura 5.8. Las tres medidas de interés resultaron iguales para ambas calles. Con estas medidas en consideración, se pudo realizar una mejor estimación de las distancias de reacción en los videos, al contabilizar el espaciamiento en función de las líneas que separan las pistas. Con las observaciones que se detallan en la Tabla 5.4, se promediaron respectivamente los eventos relacionados a un mismo parámetro. De esta manera se obtiene que $\alpha_a = 15m$, $\beta_a = 30m$ y $\delta_i = 10$.

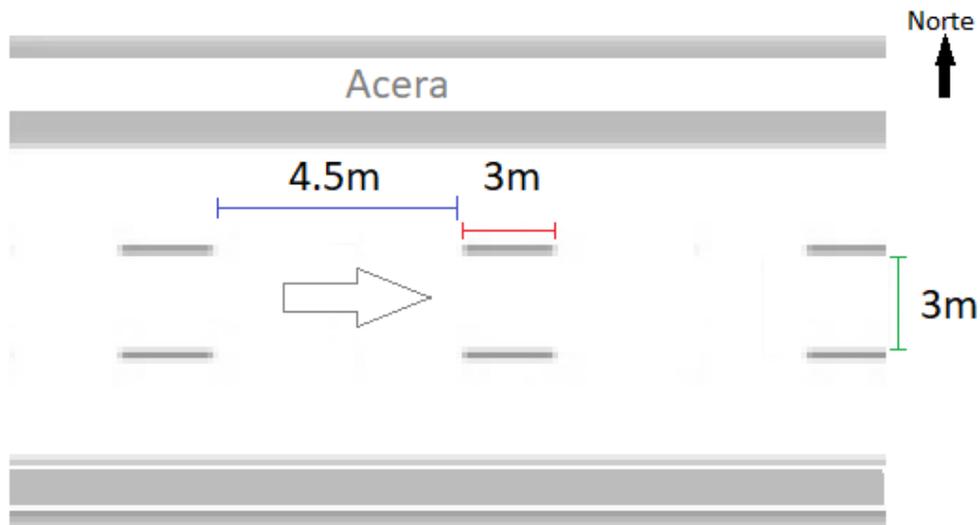


Figura 5.8: Dimensiones calle Santa María

Tabla 5.4: Incidentes identificados en video vehículo Q8 en ruta a emergencia

Ángulo	Hora	Reacción de POV	Distancia	Valor [m]	Parámetro relacionado
Frontal	15:16:54	Cambio de pista	con EV	10	β_a
Copiloto	15:17:11	Detención aunque tenga verde	a intersección (POV, EV)	(5,5)	δ_i
Trasero	15:17:15	Maneja normalmente	detrás de EV	15	α_a
Frontal	15:17:27	Cambio de pista	con EV	27	β_a
Frontal	15:17:44	Cambio de pista	con EV	40	β_a
Frontal	15:17:45	Cambio de pista	con EV	30	β_a
Cop./Con.	15:17:46	Detención aunque tenga verde	a intersección (POV, EV)	(10,10)	δ_i
Fron./Cop.	15:18:24	Cambio de pista	con EV	30	β_a
Frontal	15:18:40	Cambio de pista	con EV	30	β_a
Copiloto	15:18:42	Detención aunque tenga verde	a intersección (POV, EV)	(10,10)	δ_i
Frontal	15:18:44	Cruza con luz roja	con EV	10	δ_i
Copiloto	15:18:47	Cambio de pista	con EV	40	β_a
Trasero	15:18:46	Vuelve a pista original	detrás de EV	15	α_a
Frontal	15:19:23	Detención aunque tenga verde	a intersección (POV, EV)	(10,20)	δ_i
Frontal	15:20:34	Cruza sin prioridad	con EV	15	δ_i



Figura 5.9: Detención de vehículos (borde derecho) con luz verde en intersección semafORIZADA



Figura 5.10: Cruce de vehículo (borde derecho) con luz roja en intersección semafORIZADA

5.3. Calibración de la red

Con la red codificada y los parámetros estimados ocupando datos reales, se procede a calibrar el flujo de la red. El flujo o demanda en las calles afectan mediante la congestión al vehículo de emergencia, haciendo más lenta su expedición y afectando el perfil de distancia recorrida vs tiempo. Para medir la bondad de ajuste de las simulaciones, se ocupará dicho perfil obtenido mediante los GPS (figura 5.11) y se comparará con pulsos GPS simulados, i.e., mediciones de la distancia recorrida en el tiempo cada 10 segundos en cada corrida. En primer lugar, se simula tres escenarios especiales sin demanda: un caso sin la API, otro sin semáforos y un último con ambos API y semáforos. El EV se genera siempre en el minuto 5 de la simulación y se contabiliza el primer GPS cuando llega a la misma posición que marcó el primer pulso GPS de la ruta real. En la figura 5.12 se pueden ver los perfiles obtenidos. La curva superior corresponde al caso sin semáforos, lo que permite que el vehículo de emergencia llegue en menos tiempo que los otros casos; por otra parte, en el escenario con semáforos y sin API, el vehículo espera en caso de tener luz roja, siendo el caso que más demora en llegar a la emergencia. En este caso, al demorar más de 250 segundos no se puede ver en el gráfico el instante en que llega el EV.

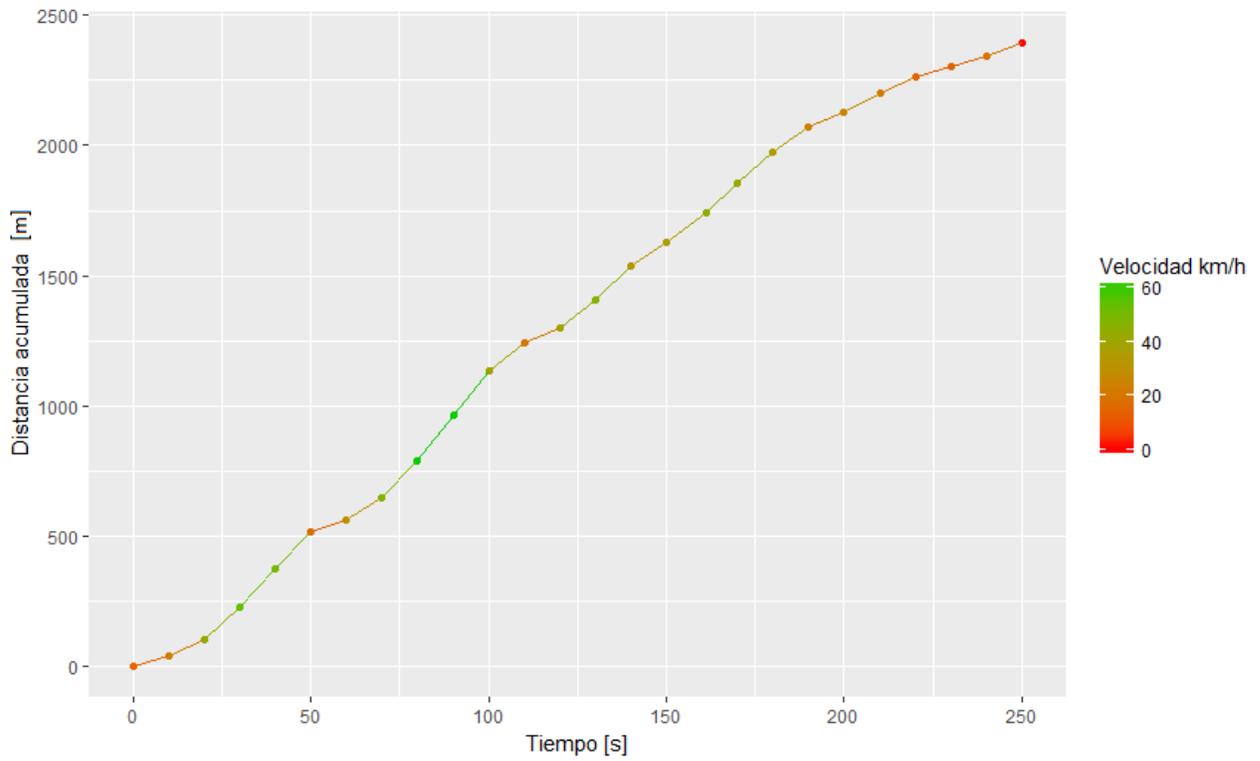


Figura 5.11: Distancia acumulada v/s tiempo registrado por pulsos GPS de carro Q8

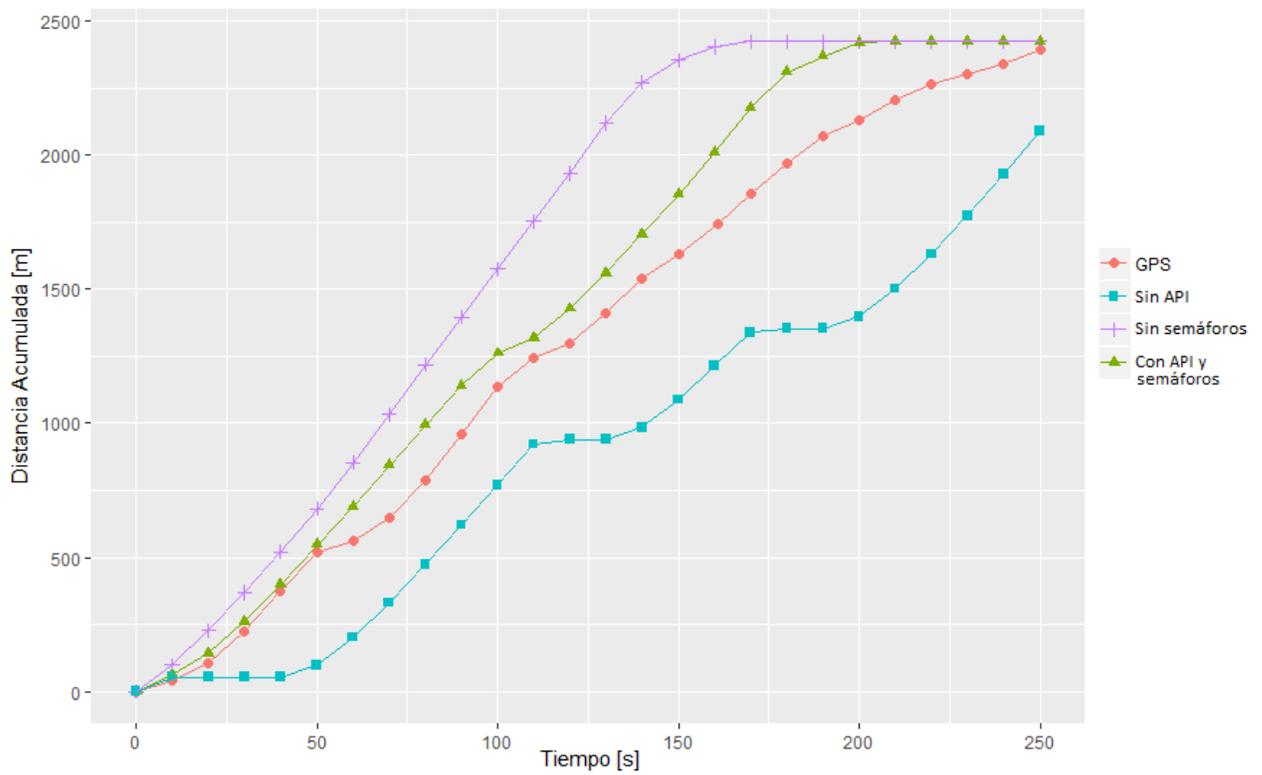


Figura 5.12: Perfil Distancia acumulada: GPS y simulaciones especiales

Luego de simular estos casos especiales, se procedió a cargar la red de manera que, visualmente, presente largos de cola similares a los que se observan del material audiovisual capturado por el vehículo Q8 en ruta. Con este escenario base, se procede a variar la carga total de la red. Es decir, se simulan escenarios con la misma matriz de demanda y estructura de viajes, dividida por un factor que se varió entre 0.5 y 2. Cada uno de estos escenarios se corrió 200 veces haciendo variar la semilla con la que se generan los números pseudo-aleatorios ocupados por PARAMICS. En la figura 5.13 se grafica el perfil para el escenario base, donde los perfiles de cada corrida corresponden a las curvas de color negro. Además, se muestra una distancia acumulada promedio de las 200 corridas, correspondiente a la curva de color azul.

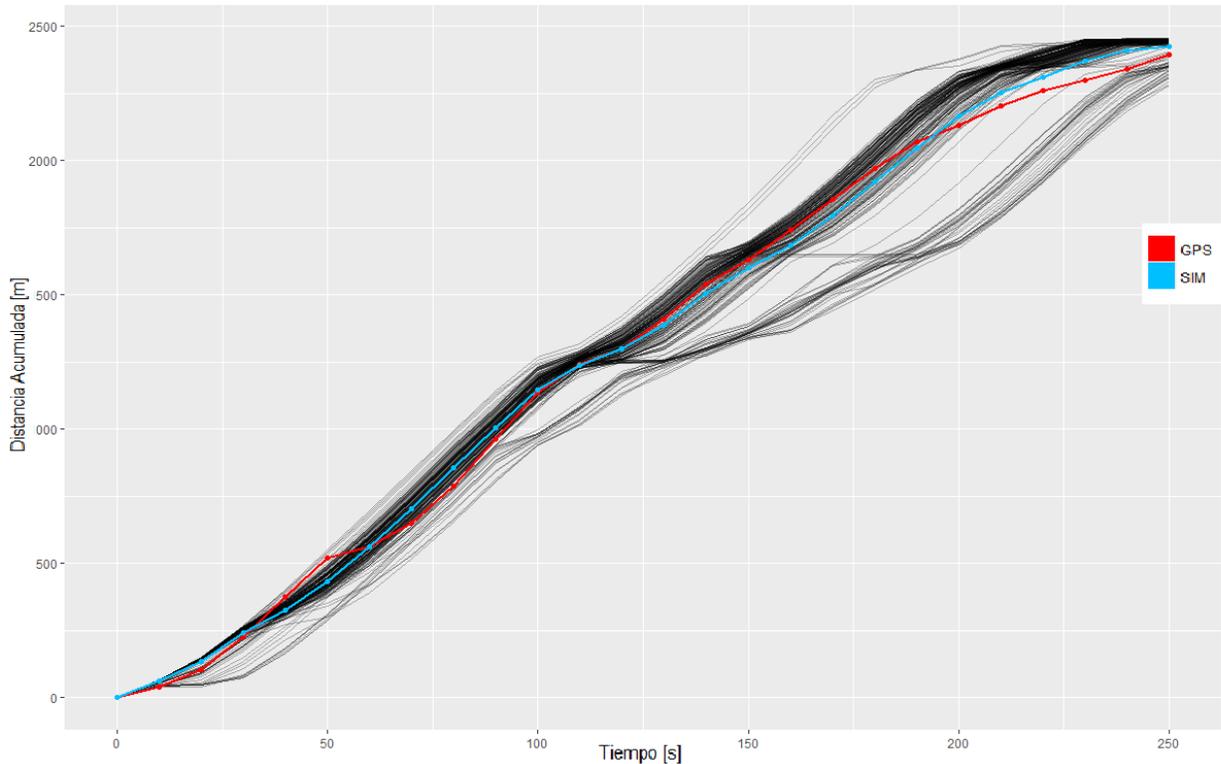


Figura 5.13: Perfil Distancia acumulada: GPS y escenario base (promedio y cada corrida)

Los factores que se ocuparon para variar la demanda fueron 0.50, 0.57, 0.66, 0.80, 1.25, 1.50, 1.75, 2.00, donde los primeros cuatro valores son el inverso multiplicativo de los últimos cuatro. Cada factor se ocupa para dividir la matriz de demanda del escenario base. Esto se definió así con el fin que fuese sencillo cambiar de escenario de manera manual en PARAMICS. Luego de generar las 200 corridas para cada escenario, se calculó el indicador RMSP (ecuación 5.1) entre cada corrida y los datos GPS. Posteriormente, se computó el promedio y desviación estándar (σ_{RMSP}) sobre los RMSP de las corridas de cada escenario para determinar cuál presenta el menor error. Adicionalmente, se calculó el coeficiente de correlación (ecuación 5.2) con el perfil promedio de cada escenario (Tabla 5.5).

Tanto el coeficiente de correlación r como el indicador RMSP se mencionaron anteriormente en la revisión bibliográfica y se explican en detalle en Hourdakis et al. (2003). El primero cuantifica la correlación existente entre dos serie de datos, tomando un valor igual a 1 si existe una correlación positiva. Estadísticamente significa existe una relación lineal y que si

una serie de datos aumenta la otra también lo hace. Por otra parte, el indicador RMSP busca cuantificar el error cuadrático medio porcentual, al ser cuadrático se consideran en menor proporción los errores pequeños y los errores grandes pasan a tener una mayor relevancia para el cálculo del error medio. El hecho que sea porcentual permite normalizar los errores para una misma variable en caso que ésta tome magnitudes diferentes en distintos instantes de la muestra.

$$RMSP = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(\frac{x_i - y_i}{y_i} \right)^2} \quad (5.1)$$

$$r = \frac{1}{N-1} \sum_{i=1}^N \frac{(x_i - \bar{x})(y_i - \bar{y})}{\sigma_x \sigma_y} \quad (5.2)$$

Tabla 5.5: Indicadores RMSP y correlación de escenarios simulados

Escenario	RMSP promedio	σ_{RMSP}	Coef. Correlación
Sin demanda - Con semáforo y API	0.1901	-	0.9953
Sin demanda - Sin semáforo	0.5324	-	0.9659
Sin demanda - Sin API	0.4320	-	0.9827
Escenario base con factor 0.50	0.1851	0.0462	0.9951
Escenario base con factor 0.57	0.1800	0.0552	0.9972
Escenario base con factor 0.66	0.1715	0.0392	0.9979
Escenario base con factor 0.80	0.1651	0.0373	0.9981
Escenario base sin factor	0.1635	0.0311	0.9983
Escenario base con factor 1.25	0.1630	0.0369	0.9988
Escenario base con factor 1.50	0.1676	0.0291	0.9989
Escenario base con factor 1.75	0.1729	0.0261	0.9988
Escenario base con factor 2.00	0.1711	0.0271	0.9988

Cabe mencionar que los indicadores se calcularon sin considerar el primer pulso simulado, ya que se fijó para que fuese idéntico al primer pulso GPS. Una vez realizados los cálculos, se concluye que todos los escenarios tienen una correlación lineal positiva fuerte con los datos reales al tener un coeficiente de correlación cercano a uno. Por otro lado, se encontró que el escenario con el factor de demanda 1.25 presentó el mejor ajuste al tener el mínimo RMSP. Esto quiere decir que el escenario que mejor se ajusta a la realidad tiene un 80 % de la demanda ingresada en el escenario base. Sin embargo, la desviación estándar de todos los RMSP promedio están en el orden del segundo decimal, al igual que las diferencias entre éstos al comparar escenarios. Más aún, el escenario sin demanda, con semáforos y API tiene un RMSP levemente superior al mejor escenario encontrado, lo que puede indicar que la congestión no es un factor importante al querer mejorar la bondad de ajuste de la simulación. Por otra parte, sí se puede apreciar una mejora considerable en el RMSP al incorporar la API y los semáforos a la simulación. Cabe destacar que si bien se observa un crecimiento en el RMSP al alejarse del escenario con el valor mínimo, se puede observar una baja nuevamente

en el RMSP del escenario menos congestionado D_{200} . La literatura alerta que esto puede suceder en problemas de calibración, al tener mínimos locales y comportamiento variable, tanto al escoger la función que mida la bondad de ajuste como los parámetros que se busca calibrar (Ciuffo y Punzo, 2014).

Para tener una mejor noción del ajuste promedio de cada escenario, en la figura 5.14 se muestran las curvas de perfil promedio (i.e. distancia acumulada promedio de las 200 corridas cada 10 segundos) en cada escenario. Adicionalmente, en la figura 5.15 se muestran las curvas con la distancia acumulada en el tiempo para todas las corridas del escenario con el menor indicador RMSP, D_{125} , junto al perfil de los pulsos GPS y el perfil promedio de todas las corridas para dicho escenario.

Finalmente, como resultado de este proceso de calibración, se tiene una simulación de una expedición de un vehículo de emergencia en la ciudad de Santiago con un error promedio de 16.3 %, lo que se puede interpretar como un resultado favorable para demostrar la utilidad de la API desarrollada para simular vehículos de emergencia. Para lograr esto, se estimaron los parámetros de los modelos para intersecciones y arcos de la API, en particular, la estimación se llevo a cabo para una zona y periodo en específico, por lo que a futuro se recomienda ocupar dichos parámetros obtenidos en caso que se desee hacer otra simulación de dicha zona y periodo incorporando vehículos de emergencia. Adicionalmente, en la práctica también es aceptable el uso de los parámetros MTH y MRT del microsimulador PARAMICS estimados. Esto en caso que se quiera simular el tráfico de días laborales en horario fuera de punta en la zona centro de Santiago y no se cuente con datos recientes de la zona y periodo que se desee simular.

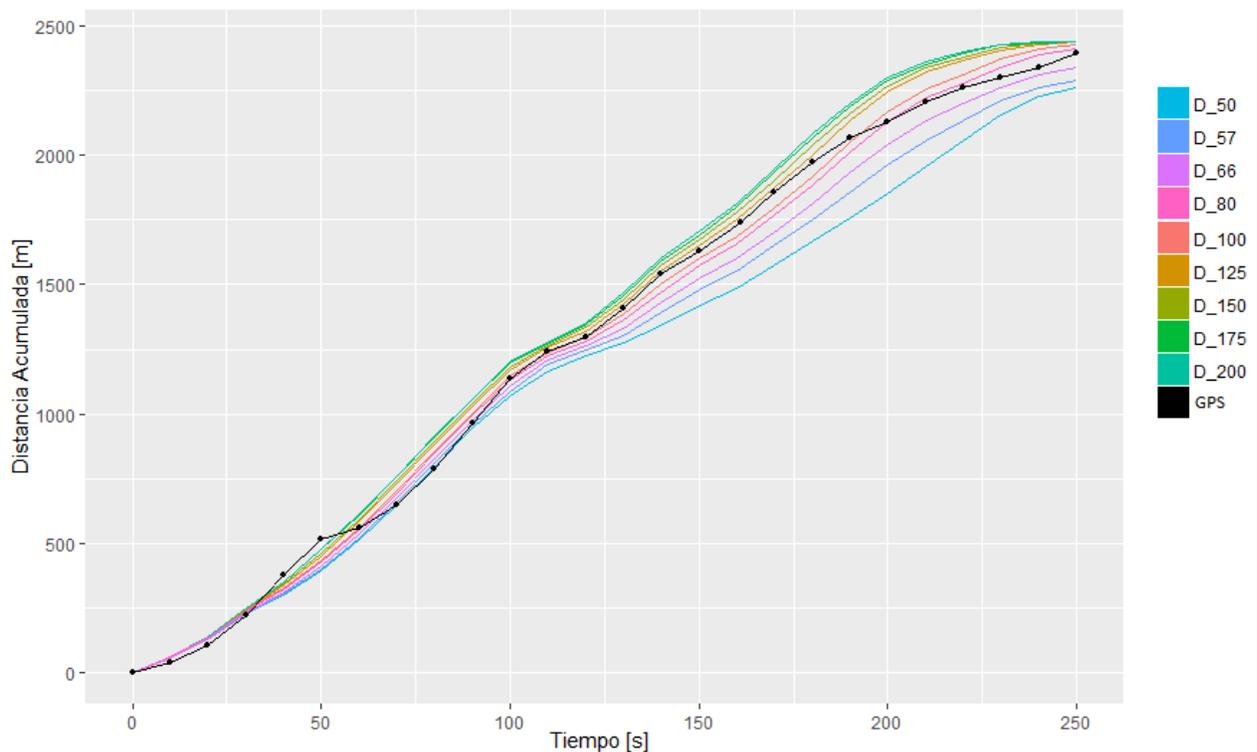


Figura 5.14: Perfil Distancia acumulada: GPS y todos los escenarios

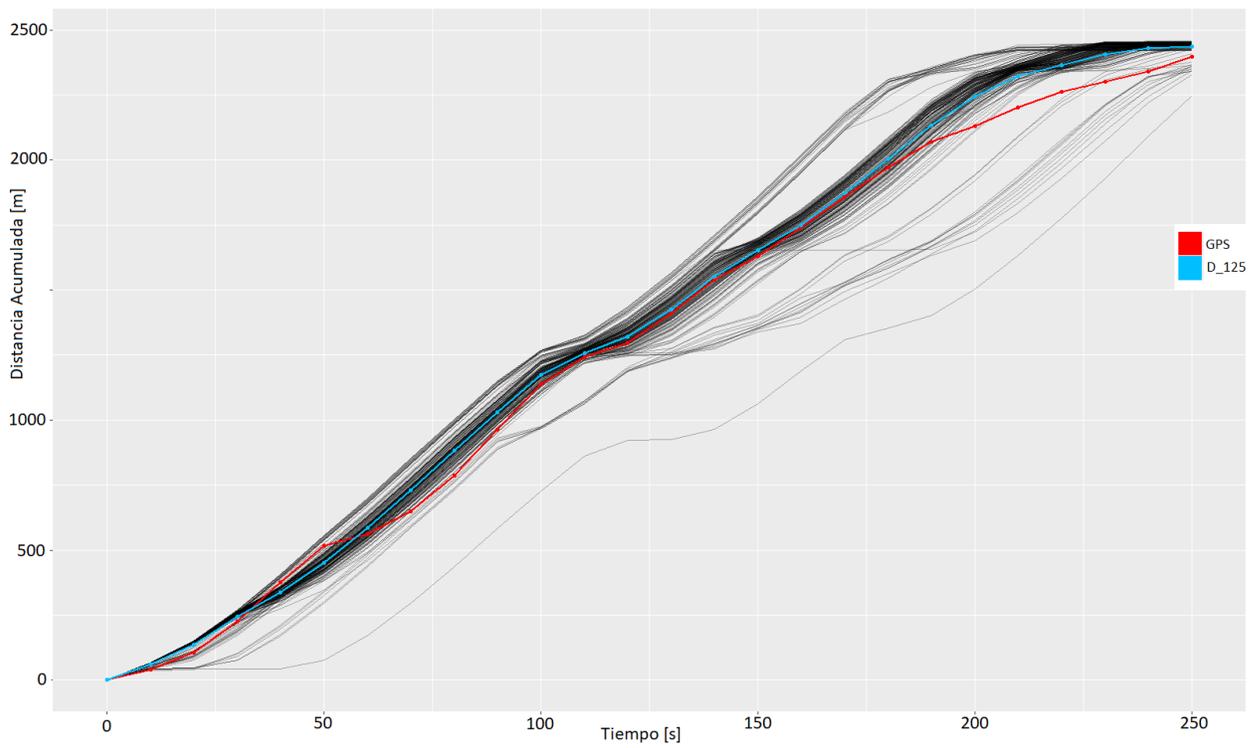


Figura 5.15: Perfil Distancia acumulada: GPS y escenario D_{125} (promedio y cada corrida)

Capítulo 6

Conclusión

6.1. Resumen y motivación del trabajo

En este trabajo de tesis se describe el desarrollo de una API que permite simular vehículos de emergencia en tráfico normal, con todos los efectos que se observan tanto en la literatura como en terreno. Además, se detalla la codificación de una red de gran tamaño: la red de calles del centro de Santiago. En tercer lugar, se explica una metodología para la preparación y procesamiento de los datos, de manera de poder ocuparlos tanto en la codificación como en la calibración de la red de Santiago. Por último, se describe un marco de estimación y calibración con el fin de simular una expedición real hacia una emergencia registrada por el CBS.

El propósito de esto consiste en tener una herramienta capaz de microsimular y evaluar la operación de vehículos de emergencia sobre una zona urbana. En particular, se pretende simular los tiempos de respuesta de vehículos de bomberos en la ciudad de Santiago. Adicionalmente, se busca discutir y proponer un marco de calibración para estimar los distintos parámetros necesarios para llevar a cabo este tipo de simulación, de forma que éstas sean lo más semejante posible a la realidad. Un objetivo a largo plazo es poder ocupar la API con PARAMICS en redes de ciudades de Chile calibradas con datos reales, de manera que se pueda usar esta herramienta para predecir o estimar de manera más certera los tiempos de viajes de servicios de emergencia.

6.2. Resultados y contribuciones

Como resultado de este trabajo, se logró codificar la zona centro de la ciudad de Santiago y se pudo ayudar a la estimación de tiempos de respuesta del CBS mediante el preprocesamiento de la base de datos de pulsos GPS. Además, se desarrolló una API con modelos inspirados en material audiovisual de EVs en ruta hacia una emergencia. Posteriormente, se calibró y ocupó la API junto al microsimulador PARAMICS para simular una expedición registrada por el CBS.

En el caso del procesamiento de datos, para lograrlo se desarrollaron dos algoritmos de *Clustering*, el algoritmo de extremos y el algoritmo *snake clustering*, los cuales resultaron ser de enorme utilidad para llevar a cabo una limpieza rápida de la base de datos. Aunque ambos algoritmos ocupaban parámetros que no se calibraron, se logró resolver de manera aceptable un problema de limpieza de base de datos. Gracias a ambos algoritmos se pudo filtrar un 68.3 % de los registros iniciales, correspondientes a aquellos pulsos GPS que representan 'ruido' o datos inútiles según los algoritmos de *Clustering*.

En paralelo, se codificó la red completa de la zona de interés de Santiago Centro, que se traduce en 453 nodos y 652 arcos, entre éstos se encuentran 228 intersecciones de las cuales 145 tienen un semáforo. De los 652 arcos, 42 corresponden a paseos peatonales presentes en el centro de la ciudad. Se incorporó tanto la programación de los semáforos, la geometría de las calles, las pistas especiales, los límites de velocidad y los recorridos de transporte público con sus frecuencias programadas y tiempos de detención por paradero.

Por otra parte, gracias a la extensa librería de funciones, fue posible desarrollar una API capaz de reproducir casi en su totalidad los comportamientos extraordinarios de los vehículos normales ante un EV, observados en los videos grabados por el CBS y que también se mencionan en la literatura. Esto se hizo mediante tres modelos, uno para la interacción entre vehículos en calles de más de una pista, otro en caso que los POV tengan la posibilidad de subirse a la acera y uno que describe el comportamiento de los vehículos en intersecciones. Además, aunque haya existido una limitante en cuanto a la simulación del cruce de intersecciones semaforizadas, ésta fue superada mediante un algoritmo ad-hoc en la API que ayuda a replicar lo que sucede en este tipo de intersecciones en la realidad.

En cuanto a la simulación de un caso real, registrado por cámaras instaladas en una máquina de bomberos, en primer lugar se estimaron los parámetros de PARAMICS que representan el tiempo de reacción (MRT) y el espaciamiento temporal deseado entre dos vehículos por los conductores en general (MTH). Esto se logró mediante cámaras de la UOCT situadas en intersecciones en la misma zona de interés y con videos durante el mismo periodo del día, como resultado se obtuvo un MRT estimado de 1.4 segundos y un MTH igual a 1.44 segundos. En cuanto a la estimación de los parámetros de la API, éstos se llevaron a cabo mediante la observación de los videos de las cámaras presenten en el vehículo de emergencia. De esta forma, se estima que la distancia promedio en la que un vehículo se percata de la presencia de un EV son 30 metros. Para que deje de estar alerta, debido a que ya pasó el EV, se observa una distancia de 15 metros en promedio. Respecto a la distancia a partir de la cual los vehículos se dan cuenta de la presencia de un EV acercándose a una intersección, ésta se estimó igual 10 metros en promedio. Estos valores estimados pueden usarse en otros estudios en caso que no se cuente con datos u otra estimación confiable.

Una vez estimados estos parámetros, se busco calibrar la congestión en la red simulada para mejorar la bondad de ajuste. Sin embargo, antes de cargar la red con flujos, se destaca que la simulación resulta ser mucho más semejante a la realidad usando la API desarrollada, al disminuir el error de 43 % a 19 % respecto de no usar la API en PARAMICS. Esto indica que la API cumple su objetivo al posibilitar una simulación de vehículos de emergencia mediante la incorporación al simulador del comportamiento observado en la realidad. Finalmente, al calibrar la red, el error se logro disminuir hasta 16.3 %

6.3. Limitaciones y trabajo futuro

Si bien se puede decir que los software de simulación microscópica de tráfico son una herramienta de una complejidad y potencial enorme, éstos a su vez exigen ser alimentados con una gran cantidad de datos, provenientes de distintas fuentes, para poder utilizarlos en forma razonable. Esto debido al proceso necesario de calibración y validación. Sin datos reales de los cuales apoyarse, no es posible explotar todo el potencial de los simuladores microscópicos de tráfico. Su poder predictivo será tan bueno como la cantidad y calidad de datos usados para calibrarlos. Simultáneamente, se reconoce también un desafío en el proceso mismo de calibración y validación, aunque se disponga de datos para dicho proceso. En efecto, hay variadas metodologías, indicadores y decisiones que quedan a criterio del modelador a la hora de calibrar. La complejidad del problema de calibración es alta y no necesariamente siempre se llegará a los parámetros óptimos, tanto por poseer de poco tiempo para llevar a cabo un análisis de sensibilidades detallado que permita hacer más eficiente el proceso de calibración, como por la escasez de recursos computacionales y pocas herramientas disponibles que permiten la automatización de dicho proceso en muchos casos.

Se debe recordar además que la API desarrollada solo se asemejará a la realidad si existe una calibración apropiada de los parámetros para cada red que se desee simular. Relacionado a esto último, se tiene como principal defecto de la API su gran cantidad de parámetros, siendo algunos difíciles de estimar o calibrar, implicando un mayor esfuerzo en el proceso de calibración.

Sobre la codificación de la red y la preparación de los datos, es un aspecto relevante el tiempo necesario para lograr todo el proceso descrito en este trabajo de tesis. La recopilación de las distintas fuentes de datos junto a la codificación en la interfaz de usuario de PARAMICS implicaron muchas horas, en varias ocasiones gastadas en trabajo repetitivo. Por último, dado que no se pudo obtener datos de flujo de la red de Santiago Centro, no fue posible simular la red al no tener ningún dato de congestión. Esto refleja lo que se mencionó anteriormente, para poder llevar a cabo una microsimulación es necesario obtener grandes cantidades de datos, de distintas fuentes, que no siempre son fáciles de obtener. En el caso de Santiago, el flujo vehicular no es algo disponible al público y la mayoría de las entidades que poseen estos datos no tienen intenciones de compartirlos. Adicionalmente, dado que se escapaba del alcance de este trabajo de tesis, no se calibraron los parámetros ocupados en los algoritmos de *Clustering*.

En cuanto a la calibración de la simulación del caso real, en particular, sobre la forma de estimación de parámetros mediante la observación de videos, se puede decir que tal método es poco eficiente al tener que observar horas de grabaciones, entre las cuales es poca la cantidad de momentos útiles, al existir muchos instantes en donde, por distintos motivos (e.g., ángulo de la cámara o nivel congestión), no es válido realizar mediciones para estimar los parámetros de interés. Además, en el caso de parámetros espaciales, las mediciones no son del todo fidedignas al ser estimaciones visuales de las distancias mediante referencias como las demarcaciones de las calles.

Finalmente, como trabajo futuro, se pretende ocupar la API con la red real del centro de Santiago, la cual actualmente se ve afectada por una serie de cambios estructurales debido a un proyecto de la Municipalidad de Santiago llamado “Plan Centro”. En paralelo, dado que el Cuerpo de Bomberos ha facilitado datos, es posible a futuro calibrar la API y el Simulador de manera de reflejar mejor la realidad en las simulaciones, pudiendo entonces efectuar una evaluación de impactos sobre tiempos de respuesta en la red del centro histórico o en otras redes en caso de existir los datos.

Glosario

AHS *Automated Highway Systems*: Sistemas de carreteras automatizadas. Es un esquema de funcionamiento para carreteras en donde todos los vehículos son autónomos y se encuentran comunicados a un sistema inteligente de la carretera a partir del cual pueden recibir órdenes de movimiento o pedir permiso para efectuar maniobras.

API *Application Programming Interface*: Interfaz de programación de aplicaciones. Es un conjunto de subrutinas o métodos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

CBS Cuerpo de Bomberos de Santiago. Institución que provee servicios de emergencia en 9 comunas de Santiago, responsable de ayudar en zonas de accidentes o catástrofes como incendios, aluviones, etc.

CFM *Car-following model*: Modelo de seguimiento vehicular. Estos modelos son responsables de guiar cómo reaccionan los vehículos al moverse en la red dentro de las simulaciones de los Microsimuladores de Tráfico, hay de distintos tipos y cada software tiene un modelo distinto.

EV *Emergency Vehicle*: Vehículo de Emergencia. Hace referencia a ambulancias, carros bomba, escalas entre otros. Se caracterizan por tener sirenas para alertar al resto de los vehículos sobre su presencia.

EVP *Emergency Vehicle Preemption*: Forma de programación para semáforos actuados, la cual consiste en activar una fase de emergencia si se detecta la presencia de un vehículo de emergencia, dando verde inmediato al acceso donde se encuentra.

GAM *Gap-acceptance model*: Modelo de aceptación de brecha. Se ocupan en los Microsimuladores de tráfico para simular el comportamiento de vehículos que desean cruzar una intersección sin tener prioridad (e.g. desde un acceso con cede al paso).

GOF *Goodness of fit*: Bondad de ajuste. Se refiere a una medida de lo parecido que es una simulación con la realidad. En general, se mide con alguna métrica que entrega un número para entender la precisión de la calibración de un simulador.

GUI *Graphical user interface*: Interfaz gráfica de usuario. Software que consiste en un entorno visual sencillo que facilita el uso del programa, mediante herramientas de ayuda gráfica que representan la información y acciones disponibles por el usuario.

ITS *Intelligent Transportation System*: Sistema de transporte inteligente. Consiste en un esquema de comunicación y coordinación automatizada entre vehículos autónomos en movimiento con el fin de optimizar la fluidez del tráfico y bajar la tasa de accidentes.

LCM *Lane-changing model*: Modelo de cambio de pista. Se ocupan en los Microsimuladores de tráfico para simular el comportamiento de vehículos que desean cambiar de pista.

MNR *Minimum number of runs*: Mínimo número de simulaciones necesarias. Es la cantidad objetivo calculada con el fin de tener una estimación confiable del promedio de las variables que se desean medir en la simulación, e.g., largos de cola, tiempos de viaje, etc.

MTS *Microscopic Traffic Simulator*: Microsimulador de tráfico. Software ocupado para emular el tráfico vehicular presente en las redes de transporte terrestre. Ejemplos serían PARAMICS, AIMSUN, VISSIM, CORSIM, etc.

POV *Privately Owned Vehicle(s)*: Vehículo(s) Privado(s). Son todos los vehículos motorizados que no son ni buses de transporte público ni vehículos de emergencia, e.g. camiones, autos, taxis, etc.

RMSP *Root Mean Squared Percentage Error*: Raíz del Error Cuadrático Medio Porcentual. Es uno de los indicadores que cuantifica el error en el proceso de calibración o validación de un simulador. Con esto es posible saber la bondad de ajuste de los parámetros de un simulador.

TSP *Traffic Signal Preemption* o simplemente *Preemption*. Consiste en un sistema de programación de semáforos donde se activa una fase especial en el semáforo de la intersección al detectar la presencia de ciertos vehículos.

Bibliografía

- Abou-Senna, H. y E. Radwan (2013). Vissim/moves integration to investigate the effect of major key parameters on co2 emissions. *Transportation Research Part D: Transport and Environment* 21, 39–46.
- Abou-Senna, H., E. Radwan, K. Westerlund y C. D. Cooper (2013). Using a traffic simulation model (vissim) with an emissions model (moves) to predict emissions from vehicles on a limited-access highway. *Journal of the Air & Waste Management Association* 63 (7), 819–831.
- Alvares, L. O., V. Bogorny, B. Kuijpers, J. A. F. de Macedo, B. Moelans y A. Vaisman (2007). A model for enriching trajectories with semantic geographical information. En: *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*. ACM, p. 22.
- Barceló, J. y J. Casas (2005). Dynamic network simulation with aimsun. En: *Simulation approaches in transportation analysis*. Springer, pp. 57–98.
- Barceló, J. et al. (2010). *Fundamentals of traffic simulation*. Vol. 145. Springer.
- Biagioni, J. y J. Eriksson (2012). Inferring road maps from global positioning system traces: survey and comparative evaluation. *Transportation Research Record: Journal of the Transportation Research Board* (2291), 61–71.
- Brackstone, M. y M. McDonald (1999). Car-following: a historical review. *Transportation Research Part F: Traffic Psychology and Behaviour* 2 (4), 181–196.
- Brockfeld, E., R. Kühne, A. Skabardonis y P. Wagner (2003). Toward benchmarking of microscopic traffic flow models. *Transportation Research Record: Journal of the Transportation Research Board* (1852), 124–129.
- Brockfeld, E., R. Kühne y P. Wagner (2004). Calibration and validation of microscopic traffic flow models. *Transportation Research Record: Journal of the Transportation Research Board* (1876), 62–70.
- Buchenscheit, A., F. Schaub, F. Kargl y M. Weber (2009). A vanet-based emergency vehicle warning system. En: *2009 IEEE Vehicular Networking Conference (VNC)*. IEEE, pp. 1–8.
- Caelli, T. y D. Porter (1980). On difficulties in localizing ambulance sirens. *Human Factors:*

- The Journal of the Human Factors and Ergonomics Society 22 (6), 719–724.
- Cao, L. y J. Krumm (2009). From gps traces to a routable road map. En: Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems. ACM, pp. 3–12.
- Castro, P. S., D. Zhang, C. Chen, S. Li y G. Pan (2013). From taxi gps traces to social and community dynamics: A survey. ACM Computing Surveys (CSUR) 46 (2), 17.
- Chen, X. y F. B. Zhan (2008). Agent-based modelling and simulation of urban evacuation: relative effectiveness of simultaneous and staged evacuation strategies. Journal of the Operational Research Society 59 (1), 25–33.
- Ciuffo, B. y V. Punzo (2014). No free lunch theorems applied to the calibration of traffic simulation models. IEEE Transactions on Intelligent Transportation Systems 15 (2), 553–562.
- CLR Analytics (2011). Paramics vs vissim on network coding and simulation speed. [Online; accessed 28-September-2016].
URL <http://www.clranalytics.com/resources/microsimulation-comparisons/paramics-vs-vissim>
- Corliss, E. L. y F. E. Jones (1976). Method for estimating the audibility and effective loudness of sirens and speech in automobiles. The Journal of the Acoustical Society of America 60 (5), 1126–1131.
- De Lorenzo, R. A. y M. A. Eilers (1991). Lights and siren: A review of emergency vehicle warning systems. Annals of emergency medicine 20 (12), 1331–1335.
- DLR Institute of Transportation Systems (2017). Sumo webpage. [Online; accessed 10-March-2017].
URL http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/
- Dowling, R., A. Skabardonis, J. Halkias, G. McHale y G. Zammit (2004). Guidelines for calibration of microsimulation models: framework and applications. Transportation Research Record: Journal of the Transportation Research Board (1876), 1–9.
- Drucker, C., S. G. Gerberich, M. P. Manser, B. H. Alexander, T. R. Church, A. D. Ryan y E. Becic (2013). Factors associated with civilian drivers involved in crashes with emergency vehicles. Accident Analysis & Prevention 55, 116–123.
- Duncan, G. (1997). Paramics technical report: Car-following, lane-changing and junction modelling. Edinburgh, Scotland: Quadstone, Ltd 5 (1), 5–2.
- Echeverría, A. (2013). Evaluación de tiempos de respuesta para el cuerpo de bomberos de santiago: nuevo modelo de ruteo, validación y métricas de desempeño del sistema actual. Tesis para optar al grado de Magíster en Gestión de Operaciones. Universidad de Chile.

- Elefteriadou, L. (2014). An introduction to traffic flow theory. Vol. 84. Springer.
- Evans, L. y P. Wasielewski (1983). Risky driving related to driver and vehicle characteristics. *Accident Analysis & Prevention* 15 (2), 121–136.
- Fathi, A. y J. Krumm (2010). Detecting road intersections from gps traces. En: *International Conference on Geographic Information Science*. Springer, pp. 56–69.
- Fellendorf, M. y P. Vortisch (2010). Microscopic traffic flow simulator vissim. En: *Fundamentals of traffic simulation*. Springer, pp. 63–93.
- Fernández, R., M. F. Lecaros y A. Valencia (2013). Calibración del microsimulador de tráfico tsis-corsim en chile. *Ingeniería de Transporte* 17 (1).
- Fritzsche, H.-T. y D.-b. Ag (1994). A model for traffic simulation. *Traffic Engineering and Control* 35, 317–321.
- Gardner, M. (1970). Mathematical games the fantastic combinations of john conways new solitaire game life. *Scientific American* 223 (4), 120–123.
- Geistefeldt, J., S. Giuliani, P. Vortisch, U. Leyn, R. Trapp, F. Busch, A. Rascher y N. Celik-kaya (2014). Assessment of level of service on freeways by microscopic traffic simulation. *Transportation Research Record: Journal of the Transportation Research Board* (2461), 41–49.
- Giannotti, F., M. Nanni, D. Pedreschi, F. Pinelli, C. Renso, S. Rinzivillo y R. Trasarti (2011). Unveiling the complexity of human mobility by querying and mining massive trajectory data. *The VLDB Journal and The International Journal on Very Large Data Bases* 20 (5), 695–719.
- Gipps, P. G. (1981). A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological* 15 (2), 105–111.
- Guo, R. y Y. Zhang (2014). Exploration of correlation between environmental factors and mobility at signalized intersections. *Transportation Research Part D: Transport and Environment* 32, 24–34.
- Haque, M. M. y S. Washington (2014). A parametric duration model of the reaction times of drivers distracted by mobile phone conversations. *Accident Analysis & Prevention* 62, 42–53.
- Ho, J. y B. Casey (1998). Time saved with use of emergency warning lights and sirens during response to requests for emergency medical aid in an urban environment. *Annals of emergency medicine* 32 (5), 585–588.
- Hollander, Y. y R. Liu (2008). The principles of calibrating traffic microsimulation models. *Transportation* 35 (3), 347–362.
- Hourdakis, J., P. Michalopoulos y J. Kottommannil (2003). Practical procedure for calibrating

- microscopic traffic simulation models. *Transportation Research Record: Journal of the Transportation Research Board* (1852), 130–139.
- Jones, F., T. Quindry y W. Rinkinen (1974). Summary report on emergency vehicle sirens. US Department of Justice, Law Enforcement Assistance Administration, National Institute of Law Enforcement and Criminal Justice.
- Jordan, C., M. Cetin y R. Robinson (2013). Path clearance for emergency vehicles through the use of vehicle-to-vehicle communication. *Transportation Research Record: Journal of the Transportation Research Board* (2381), 45–53.
- Kotusevski, G. y K. Hawick (2009). A review of traffic simulation software.
- Kwon, T. M. y S. Kim (2003). Development of dynamic route clearance strategies for emergency vehicle operations, phase i.
- Lighthill, M. J. y G. B. Whitham (1955). On kinematic waves. ii. a theory of traffic flow on long crowded roads. En: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. Vol. 229. The Royal Society, pp. 317–345.
- Majnik, M. y Z. Bosnić (2013). Roc analysis of classifiers in machine learning: A survey. *Intelligent data analysis* 17 (3), 531–558.
- McTrans Center (2017). Corsim webpage. [Online; accessed 10-March-2017]. URL <http://mctrans.ce.ufl.edu/mct/index.php/tsis-corsim/>
- Moussa, N. (2008). Modeling evacuation of emergency vehicles by cellular automata models. *International Journal of Modern Physics C* 19 (06), 947–956.
- Moussa, N. (2009). Evacuation model for emergency vehicles in highways. *International Journal of Modern Physics C* 20 (01), 59–69.
- Mubasher, M. M. y J. S. W. ul Qounain (2015). Systematic literature review of vehicular traffic flow simulators. En: *Open Source Software Computing (OSSCOM), 2015 International Conference on*. IEEE, pp. 1–6.
- Newell, G. F. (2002). A simplified car-following theory: a lower order model. *Transportation Research Part B: Methodological* 36 (3), 195–205.
- Olstam, J. J. y A. Tapani (2004). Comparison of carfollowing models. Swedish National Road and Transport Research Institute.
- Ossen, S., S. Hoogendoorn y B. Gorte (2006). Interdriver differences in car-following: a vehicle trajectory-based study. *Transportation Research Record: Journal of the Transportation Research Board* (1965), 121–129.
- Ossen, S. y S. P. Hoogendoorn (2011). Heterogeneity in car-following behavior: Theory and empirics. *Transportation research part C: emerging technologies* 19 (2), 182–195.

- Panwai, S. y H. Dia (2005). Comparative evaluation of microscopic car-following behavior. *IEEE Transactions on Intelligent Transportation Systems* 6 (3), 314–325.
- Potter, R., S. Fidell, M. Myles y D. Keast (1977). Effectiveness of audible warning devices on emergency vehicles. Tech. rep., US Department of Transportation, National Highway Traffic Safety Administration, publication No. DOT-TSC-OST-77-3g, 1977, Washington, DC.
- PTV Group (2017). Vissim webpage. [Online; accessed 10-March-2017].
URL <http://vision-traffic.ptvgroup.com/en-uk/products/ptv-vissim/>
- Quadstone Paramics (2017). Paramics version history. [Online; accessed 10-March-2017].
URL <http://www.paramics-online.com/>
- R Core Team (2017). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
URL <https://www.R-project.org/>
- Ranney, T. A. (1999). Psychological factors that influence car-following and car-following model development. *Transportation research part F: traffic psychology and behaviour* 2 (4), 213–219.
- Ratrout, N. T., S. M. Rahman y I. Reza (2015). Calibration of paramics model: Application of artificial intelligence-based approach. *Arabian Journal for Science and Engineering* 40 (12), 3459–3468.
- Reza, I., N. T. Ratrout y S. M. Rahman (2016). Calibration protocol for paramics microscopic traffic simulation model: application of neuro-fuzzy approach. *Canadian Journal of Civil Engineering* 43 (4), 361–368.
- Richards, P. I. (1956). Shock waves on the highway. *Operations research* 4 (1), 42–51.
- Saifuzzaman, M. y Z. Zheng (2014). Incorporating human-factors in car-following models: a review of recent developments and research needs. *Transportation research part C: emerging technologies* 48, 379–403.
- Sisiopiku, V. P. (2007). Application of traffic simulation modeling for improved emergency preparedness planning. *Journal of urban planning and development* 133 (1), 51–60.
- Song, F., X.-k. Yang y L. Du (2010). The development of paramics based metropolitan emergency evacuation transportation simulation system—meetsim. En: *ICCTP 2010: Integrated Transportation Systems: Green, Intelligent, Reliable*. ASCE, pp. 388–400.
- Tirachini, A., L. Camus y D. Cruz (2015). Modelación de tiempos de detención de buses en paraderos de santiago y valparaíso-viña del mar. *Ingeniería de Transporte* 19 (2).
- Toy, C., K. Leung, L. Alvarez y R. Horowitz (2002). Emergency vehicle maneuvers and control laws for automated highway systems. *IEEE Transactions on intelligent transportation systems* 3 (2), 109–119.

- Truong, L. T., M. Sarvi, G. Currie y T. M. Garoni (2015). Required traffic micro-simulation runs for reliable multivariate performance estimates. *Journal of Advanced Transportation*.
- TSS-Transport Simulation Systems (2017). Aimsun webpage. [Online; accessed 10-March-2017].
URL <https://www.aimsun.com/>
- VanLandegen, L. D. y X. Chen (2012). Microsimulation of large-scale evacuations utilizing metrorail transit. *Applied Geography* 32 (2), 787–797.
- Wang, F., A. R. Goparaj y D. Singh (2016). Reduce greenhouse gas emissions through ramp meter control.
- Weinert, F. y M. Düring (2015). Development and assessment of cooperative v2x applications for emergency vehicles in an urban environment enabled by behavioral models. En: *Modeling Mobility with Open Data*. Springer, pp. 125–153.
- Wiedemann, R. (1974). *Simulation des verkehrsfusses* Schriftenreihe des instituts für verkehrswesen. Universität Karlsruhe (8).
- Wiedemann, R. y U. Reiter (1992). Microscopic traffic simulation: the simulation system mission, background and actual state. Project ICARUS (V1052) Final Report 2, 1–53.
- Yoo, J. B., J. Kim y C. Y. Park (2010). Road reservation for fast and safe emergency vehicle response using ubiquitous sensor network. En: *Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*, 2010 IEEE International Conference on. IEEE, pp. 353–358.
- Yu, J., A. Pande, N. Nezamuddin, V. Dixit y F. Edwards (2014). Routing strategies for emergency management decision support systems during evacuation. *Journal of Transportation Safety & Security* 6 (3), 257–273.
- Yun, I., B. B. Park, C. K. Lee y Y. T. Oh (2012). Comparison of emergency vehicle preemption methods using a hardware-in-the-loop simulation. *KSCE Journal of Civil Engineering* 16 (6), 1057–1063.
- Zhang, L., J. Gou, K. Liu, G. McHale, R. Ghaman y L. Li (2009). Simulation modeling and application with emergency vehicle presence in corsim. En: *Vehicular Technology Conference Fall (VTC 2009-Fall)*, 2009 IEEE 70th. IEEE, pp. 1–7.
- Zhao, H.-T., H.-Y. Mao y R.-J. Huang (2013). Cellular automaton models for mixed traffic flow considering passage way of emergency vehicle. *Modern Physics Letters B* 27 (08), 1350052.
- Zhi, T. Y., C. R. Long y L. Der Horng (2005). Comparison of paramics and getram/aimsun for its simulations.
- Zhong, R., K. Fu, A. Sumalee, D. Ngoduy y W. Lam (2016). A cross-entropy method and probabilistic sensitivity analysis framework for calibrating microscopic traffic models. *Transportation Research Part C: Emerging Technologies* 63, 147–169.