

## APLICACIÓN DE VITERBI SOBRE MODELOS OCULTOS DE MARKOV PARA LA ESTIMACIÓN DE TRÁFICO VEHICULAR

Alejandro J. Perez<sup>a,c</sup>, Leonardo D. Dominguez<sup>a,b</sup>, Pablo A. Lotito<sup>a,b</sup>, Juan P. D'Amato<sup>a,b</sup> y  
Aldo J. Rubiales<sup>a,c</sup>

<sup>a</sup>*Instituto de Investigación PLADEMA- Facultad de Ciencias Exactas - U.N.C.P.B.A., Paraje Arroyo  
Seco, (B7001BBO) Tandil, Buenos Aires, Argentina, <http://www.pladema.net/>*

<sup>b</sup>*CONICET (Argentina)*

<sup>c</sup>*CICPBA (Pcia. de Buenos Aires, Argentina)*

**Palabras Clave:** Tráfico, Modelos Ocultos de Markov, Viterbi, Simulación

**Resumen.** En los últimos años se ha observado un incremento en la incorporación de dispositivos de tecnología inalámbrica en los vehículos particulares, tales como sistemas de sonido Bluetooth o mecanismos de comunicación de componentes internos. Interceptar estas señales ofrece una nueva forma de hacer un seguimiento de cada vehículo y así obtener datos para estimar la movilidad urbana de manera pasiva. La naturaleza estocástica de las transmisiones recolectadas tanto en ocurrencia como en intensidad de señal, hacen que su interpretación sea un desafío significativo. En este trabajo se desarrolla un método de estimación del estado de una red de tráfico a partir de las trayectorias de los vehículos, basado en modelos ocultos de Markov y resuelto aplicando el algoritmo de Viterbi. A su vez, el método propuesto es validado por simulación microscópica.

## 1. INTRODUCCIÓN

La organización del tráfico urbano es uno de los principales retos a los que se enfrentan las ciudades. El mundo se urbaniza rápidamente y la densidad de población aumenta. Este crecimiento conlleva una expansión de la demanda y ha llevado a priorizar la aplicación de estrategias que permitan disminuir la congestión global. Estas técnicas se ubican en tres áreas principales: gobierno, optimización de la red de transporte y servicios de transporte integrados (en este último caso se ubican los sistemas de información al viajero que proveen información en tiempo real).

Sin embargo, dentro de las áreas mencionadas, no es posible llevar a cabo estrategias si no se conocen las variables que describen el estado real del sistema: densidad de ocupación, flujo vehicular y velocidades medias. Para obtener estos datos existen productos comerciales, como *Mistic* de Swarco y *CitySolver* de Bitcarrier, pero requieren elevados costos de instalación y mantenimiento. Para las autoridades, es indispensable contar con información del estado de la red ya que permite tomar decisiones fundamentadas que mejoren las condiciones de tráfico y en efecto la calidad de vida de los habitantes.

En lo que refiere a los sistemas de transporte, la mayoría de los líderes gubernamentales coinciden en que es necesario invertir en infraestructuras. Sin embargo, las restricciones impuestas por la limitación presupuestaria obligan a disminuir la congestión gestionando más eficazmente la demanda y el suministro mediante la utilización de sistemas de transporte inteligente (ITS) [Jamie Houghton \(2009\)](#). En cuanto a los costos que implica la congestión vehicular, la operación de los vehículos que circulan en las calles de ciudades de más de 100.000 habitantes consume alrededor del 3.5 % del PBI de América Latina y el Caribe [Bull \(2003\)](#). Dicho esto, es notable la necesidad de contar con herramientas que permitan a las autoridades aplicar políticas que mejoren las condiciones del transporte.

Por otro lado, la utilización masiva de teléfonos móviles con sensor GPS integrado permite a las autoridades desarrollar aplicaciones para los usuarios que deseen enviar y recibir información relativa al tráfico de la red. Además, la incorporación de dispositivos de tecnología inalámbrica Wi-Fi y Bluetooth en automóviles particulares, lleva a nuevas formas de obtener datos de la movilidad urbana realizando capturas de los vehículos en tránsito, permitiendo inferir su ubicación. Este mecanismo de recolección de datos se considera no invasivo ya que no es necesario alterar la vía para su instalación, como en el caso de los detectores por inducción magnética. Esto lo convierte en un método de bajo costo y fácil instalación, con posibilidad de ser extendido a múltiples aplicaciones, como por ejemplo: posicionamiento en interiores basado en señales Wi-Fi [Jekabsons et al. \(2011\)](#) y análisis de uso de líneas de colectivos en base al seguimiento de usuarios vía Wi-Fi y Bluetooth [Dunlap et al. \(2016\)](#).

## 2. OBJETIVO

El objetivo es producir una estimación de tiempos de viaje entre ubicaciones de una red espacial conocida. El método propuesto utiliza información generada por distintos dispositivos utilizados en cada uno de los vehículos que circulan dentro de la red, los cuales pueden tener una aplicación especificada para informar su posición vía GPS (modo activo), o a través de la detección de señales Bluetooth (modo pasivo). Para estimar tiempos de viaje a partir de muestras posicionales se requieren dos elementos: la diferencia de tiempo observada y la trayectoria realizada. La diferencia de tiempo entre dos puntos puede ser calculada directamente como la diferencia entre el *timestamp* de ambos. La trayectoria recorrida debe ser determinada con respecto a la red espacial correspondiente. Identificar una trayectoria puede ser trivial si los

dos puntos se encuentran cercanos (en tiempo y espacio). De otro modo, se recurre a métodos más complejos para buscar la trayectoria más probable. A continuación se detallan los métodos desarrollados y los resultados obtenidos para la estimación de tiempos de viaje a partir de muestras GPS y Bluetooth.

## 2.1. Presentación del problema

A fin de definir el problema es necesario introducir los siguientes componentes:

**Una red espacial digital** es un grafo  $\mathcal{G} < \mathcal{V}, \mathcal{E} >$  donde  $\mathcal{V}$  es un conjunto de vértices y  $\mathcal{E}$  un conjunto de arcos. Cada arco tiene las propiedades: vértices, longitud, dirección. Adicionalmente, una red tiene propiedades relacionadas al tránsito, como restricciones de giro, límites de velocidad, información de semáforos, etc.

**Un conjunto de datos de localización** consiste de: (a) latitud, (b) longitud, (c) tiempo del muestreo, (d) identificación del dispositivo. La ubicación reportada por un dispositivo es también llamada sonda. Una secuencia de sondas provenientes de un mismo vehículo se denota por  $\mathcal{P} = p_1, p_2, \dots, p_{t-1}$ .

**Una trayectoria real** es una secuencia de arcos que representan el camino real atravesado por un vehículo a medida que se generan las observaciones  $\mathcal{P}$ . Es denotada por  $\mathcal{T}$  y no es considerada en el contexto de map-matching (excepto como datos ground truth).

**Una trayectoria observada** es una secuencia de segmentos de línea recta, tal que cada línea conecta dos posiciones observadas consecutivas.

El problema es entonces definido como: buscar la secuencia de arcos más probable ( $\tau_{opt}$ ) para una secuencia de observaciones  $\mathcal{P}$  y una red espacial digital  $\mathcal{G}$ :

$$\tau_{opt} = arg \max_{\tau} \mathbb{P}(\tau = \mathcal{T} | \mathcal{P}, \mathcal{G})$$

donde  $\mathbb{P}(\tau = \mathcal{T} | \mathcal{P}, \mathcal{G})$  es la probabilidad ser  $\tau$  la verdadera trayectoria dado  $\mathcal{P}$  y  $\mathcal{G}$ .

## 3. ESTIMACIÓN DE LA TRAYECTORIA

Asumiendo que un dispositivo viaja a través de una red espacial de calles o caminos, el objetivo es producir una estimación segundo a segundo de su ubicación en base a una serie de transmisiones de cada dispositivo. Esta tarea presenta los siguientes desafíos:

**Consumo de energía:** En este caso se considerará la dispersión espacial de detecciones, es decir, es deseable partir de observaciones que se realizan con baja periodicidad, ya que una alta periodicidad en las transmisiones implica mayor consumo energético del dispositivo. Esto hace que métodos estándar basados en triangulación no sean aptos.

**Muestras de baja precisión:** Incluso cuando el consumo de energía no es una preocupación, los sensores GPS no siempre están disponibles. Algunos celulares usan WiFi y la señal de celular para el posicionamiento, siendo este de menor calidad. Incluso las experiencias con GPS pueden no ser las esperadas, por ejemplo, cuando el usuario se encuentra con el celular en el bolsillo o está atravesando zonas de edificios de gran altura o túneles donde la precisión se reduce a decenas o cientos de metros.

### 3.1. Solución trivial

Considere la siguiente solución:

**Al realizar una observación, tomar la posición de menor distancia al segmento de calle más cercano al origen de la transmisión. Interpolar las ubicaciones entre observaciones.**

En primer lugar, utilizar el segmento más cercano a cada posición frecuentemente no es el segmento real por donde circula un vehículo. En segundo lugar, las variaciones en el muestreo por error de los instrumentos de posicionamiento aumentan la complejidad en situaciones de dispositivos estacionarios, donde pueden ocurrir observaciones intermitentes. En el algoritmo planteado, esto se refleja como saltos entre las ubicaciones, ubicando las detecciones en posiciones hacia delante y hacia atrás.

Por último, interpolar ubicaciones distantes por línea recta producen trayectorias de baja calidad. El uso de una red espacial, como el mapa de una ciudad puede mejorar significativamente el resultado, pero introduce ambigüedades en las rutas que deben ser resueltas. Utilizar el camino más corto entre las muestras no es aplicable por la alta incertidumbre de la ubicación. Como se sugiere en este ejemplo, soluciones por algoritmos determinísticos no se ajustan al problema, debido a la naturaleza estocástica de las detecciones.

### 3.2. Algoritmo básico de estimación de trayectoria

A continuación se presenta un método probabilístico para la estimación de la trayectoria. En lugar de asociar el segmento más cercano a una observación en cada instante de tiempo, se modela la distribución de posibles ubicaciones a lo largo del tiempo, y se toma la trayectoria que maximiza la probabilidad en el modelo.

Para reducir el conjunto de ubicaciones posibles y restricciones de movimiento, se asume que el dispositivo viaja a través de una red espacial conocida. El modelo, tomado como entrada del algoritmo, consiste en vértices localizados espacialmente, conectados por arcos dirigidos.

Dada la red espacial, el problema de estimación de trayectoria tiene cierta similitud con map-matching, que busca coincidir una secuencia de observaciones escasas y ruidosas con una serie de estimaciones de ubicación. El método expuesto a continuación está basado en map-matching de Viterbi.

Formulamos el problema usando Modelos Ocultos de Markov (HMM, por sus siglas en inglés: Hidden Markov Models) para estados, probabilidades de transición y probabilidades de emisión, utilizando decodificación por el algoritmo de Viterbi para determinar la trayectoria más probable, representada por una secuencia de estados ocultos en el Modelo de Markov. Utilizado en conjunto con el algoritmo de Viterbi, un HMM espera una entrada periódica y produce una transición de estados  $i \rightarrow j$  por período. Las transiciones son regidas por probabilidades de transición  $p(i \rightarrow j)$ , y probabilidades de emisión  $p(obs|i)$ , donde *obs* es la observación actual. Las auto-transiciones son permitidas. A continuación se describe en detalle la formulación de los Modelos Ocultos de Markov.

### 3.3. Modelos ocultos: Segmentos de calles

Un mapa de calles simple consiste de vértices representando las intersecciones, arcos representando los segmentos de una calle, y (mayormente para vehículos) restricciones de giro que determinan las transiciones permitidas entre arcos.

Nuestro modelo debe ser capaz de reconocer el trayecto más probable segundo a segundo siendo este un trayecto válido. Para esto, es necesario contar con estados de extensión espacial significativamente menor al segmento de una calle o cuadra, las cuales pueden superar los cientos de metros (Figura 1).

Cada estado representa un área rectangular en la cual puede estar ubicado un emisor GPS. Una calle puede ser atravesada pasando por todos los segmentos, sin saltos entre segmentos no adyacentes. Se mantienen las restricciones de giro para el último segmento de cada cuadra.

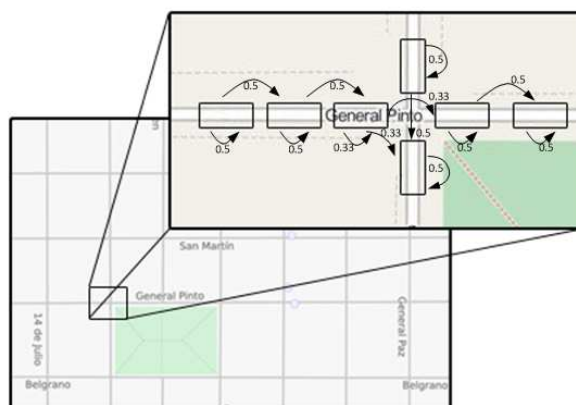


Figura 1: Modelo oculto de Markov de dos calles interseccionadas. Los segmentos son sub-divididos para aumentar la granularidad.

De esta forma definimos  $\mathcal{G}' = \langle \mathcal{V}', \mathcal{E}' \rangle$  como un nuevo grafo compuesto por los segmentos, con estados como vértices y transiciones basadas en las restricciones de giro como arcos.

### 3.4. Probabilidades de transición

Las probabilidades de transición reflejan los siguientes conceptos: 1) Para un segmento dado, existe una probabilidad de que en el siguiente instante, el vehículo se encuentre en ese mismo segmento. 2) Un vehículo puede viajar solo del final de un segmento al comienzo del siguiente si usa la misma intersección; esto asegura que el camino de segmentos saliente sea continuo. 3) Un vehículo no puede viajar irrazonablemente rápido en cualquier segmento. En otras palabras, las transiciones modelan el comportamiento en las intersecciones: continuando o girando. Se usan probabilidades uniformes entre segmentos adyacentes. Por lo tanto, para cada segmento  $i$ , con segmentos adyacentes  $n \in N$ ,  $p(i \rightarrow i) = p(i \rightarrow n) = \frac{1}{|N|+1}$ .

### 3.5. Probabilidades de emisión

Describe la probabilidad  $p(obs|s)$  de hacer una observación  $obs$  en el estado actual  $s$ .

Una observación  $obs$  consiste en un nuevo punto  $p_i$  (lat,lng) de la trayectoria  $\mathcal{P}$ . Para cada segmento candidato  $n \in N$  se modela la probabilidad de emisión como una función de distribución normal que depende de la distancia del segmento a la observación  $p_i$ . Concretamente, la densidad de probabilidad de emisión para el segmento  $i$  con la observación  $p$  es  $N(dist(i, p))$ , donde  $N$  es una función Gaussiana con media  $\mu = 0$ , y  $dist(i, p)$  es la distancia Euclídeana entre  $i$  y  $p$ . La varianza de  $N$  depende del tipo de sensor (A-GPS/Wi-Fi/Red celular/Bluetooth). Se toma como referencia un error de 10 metros (medido en estudios previos sobre precisión de sistemas de posicionamiento en dispositivos móviles Jones et al. (2015)). Este modelo representa el concepto de que es probable que un punto particular haya sido observado desde un segmento próximo, pero no necesariamente el más cercano.

Adicionalmente, se introducen dos métodos que permiten calcular  $p(obs|s)$  considerando a cada segmento como un punto (método 1), o como una superficie (método 2). Se utiliza la ecuación (1) que describe valores de probabilidad bajos al aumentar las distancias.

$$p(obs|s) = \frac{e^{-d((x,y);X_j)^2}}{\sum_{k=1} e^{-d((x,y);X_k)^2}} \quad (1)$$

### 3.5.1. Método 1

Para cada observación  $p_i$  se calcula la distancia entre cada segmento  $n \in N$  y la observación. La ubicación del segmento se corresponde con el centroide de la superficie que lo define. Se aplica un factor de corrección sobre la distancia para escalar los valores pequeños provenientes de las distancias geodésicas expresadas en Km. El valor que se devuelve es  $p(obs|s)$

---

#### Algorithm 1 Método 1

---

```

1: procedure PROBABILITY( $p_i$ )
2:    $pow \leftarrow e^{-distancia(p_i,centroide)^2}$ 
3:   for each  $s \in \mathcal{S}$  do ▷ Normalización
4:      $norm \leftarrow norm + e^{-distancia(p_i,s)^2}$ 
5:   end for
6:   return  $pow/norm$ 
7: end procedure

```

---

### 3.5.2. Método 2

A diferencia del método 1, se considera la superficie del segmento en el cálculo de  $pow$ . Se calcula la suma de  $e^{-d^2}$  sobre el área de cada segmento. Se aproxima sobre la superficie dividida en intervalos pequeños con una separación  $(\delta x, \delta y)$ , más precisamente, la décima parte del tamaño de cada segmento en x e y

---

#### Algorithm 2 Método 2

---

```

1: procedure POW( $p_i,s$ )
2:   for  $x \leftarrow xMin; x < xMax; x + \delta x$  do
3:     for  $y \leftarrow yMin; y < yMax; y + \delta y$  do
4:        $centroide \leftarrow Point(x,y)$ 
5:        $pow \leftarrow pow + e^{-distancia(p_i,centroide)^2}$ 
6:     end for
7:   end for
8:   return  $pow$ 
9: end procedure

```

---

## 3.6. Algoritmo de Viterbi

Dado un HMM y un conjunto de observaciones  $obs$ , se extrae la trayectoria que maximiza la probabilidad de las observaciones en el modelo. El modelo oculto estará definido por: los estados, el vector de probabilidades iniciales  $\pi_i$  para cada estado, la matriz de probabilidades de transición  $a$ , y un vector de funciones de distribución de probabilidades de observaciones  $opdfs$ .

$\pi$  Valores de probabilidad inicial,  $\pi_i$  es la probabilidad inicial del estado  $i$ .

**a** Un arreglo de probabilidades de transición de estados.  $a_{ij}$  es la probabilidad de ir desde el estado  $i$  al estado  $j$ .

**opdfs** Son las distribuciones de observación,  $opdfs[i]$  es la distribución de observación asociada al estado  $i$ .

Serán tantos estados como segmentos existan en la red espacial definida  $\mathcal{G}'$ . Para construir el modelo, se itera sobre los segmentos definidos en  $\mathcal{G}'$  y se crea un estado para cada segmento, con probabilidad  $p_i = \frac{1}{cant(segmentos)}$ .

Una distribución  $opdf$  está dada por una función de distribución normal bivariada con media en el centroide del segmento  $s$ ,  $\mu = (avg(longitud), avg(latitud))$  y la matriz identidad como matriz de covarianza.  $A_{ij}$  estará dado por  $\frac{1}{(d_{max}+1)}$ , donde  $d_{max}$  es el grado máximo saliente para  $s$ , incluyendo la auto-transición de cada estado. En la Figura 2 se ejemplifica la función de distribución para un segmento  $s'$  con centroide  $\mu' = (a, b)$ . Se observa la variación de la probabilidad  $p(obs|s')$  en función de la distancia respecto a  $\mu'$ . Los métodos 1 y 2 definidos en la sección 3.5 se presentan como alternativas a la función de distribución normal bivariada.

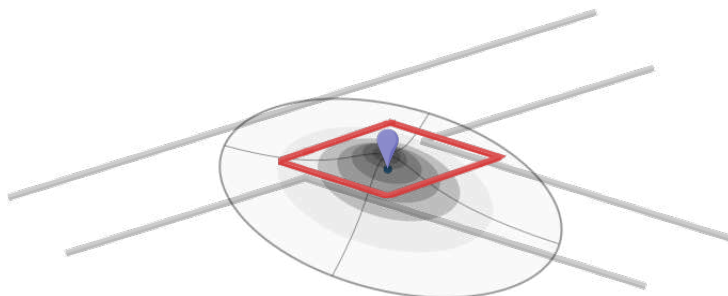


Figura 2: Función de densidad de probabilidades bidimensional.

La decodificación por Viterbi es una técnica de programación dinámica para buscar la secuencia de estados ocultos más probable dado un conjunto de observables y la distribución de probabilidades de emisión y probabilidades de transición. En nuestro caso, los estados ocultos se corresponden con la trayectoria recorrida, y los observables son las muestras de posición.

### 3.6.1. Ejemplo HMM

En la Figura 3 se muestra un ejemplo donde S1, S2, y S3 son segmentos de calle y p1, p2, p3, y p4 son observaciones realizadas. Hay igual probabilidad de hacer una transición desde S1 a S1, S2, o S3. Como la función de densidad de probabilidad de emisión es una función decreciente de la distancia, la secuencia de segmentos más probable para la secuencia de observaciones dada es S1, S3, S3, S3, y S3. Si bien p2 es más cercano a S2, el estado oculto más probable de ese punto es S3 dadas las restricciones de transitividad. Si se toma el camino atravesando S2, no es posible girar luego a S3.



Figura 3: Ejemplo de HMM

Es claro en este ejemplo la utilidad de HMM, es robusto frente a muestras posicionales cercanas a un segmento que no sea el correspondiente a la observación, y representa correctamente la idea de una ruta continua en lugar de una secuencia de segmentos.

### 3.6.2. Flujo del algoritmo

- Para cada punto de una trayectoria observada, se computa la probabilidad de emisión para cada segmento candidato (estados ocultos), representados por los círculos vacíos en la Figura 4. Mientras que la probabilidad de transición es asignada a cada estado incidente en el estado oculto.

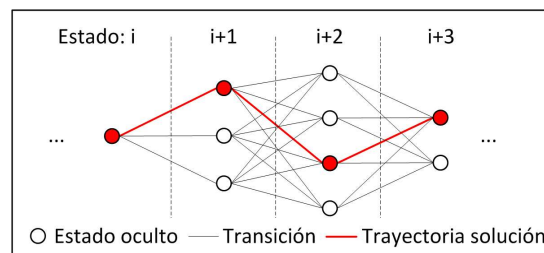


Figura 4: Diagrama de funcionamiento del algoritmo de Viterbi

- El algoritmo realiza backtracking sobre la cadena de Markov actualizada y arroja una solución parcial.
- El proceso anterior es repetido para el próximo punto de la trayectoria. El algoritmo termina cuando el último punto es alcanzado.

## 4. ESTIMACIÓN A PARTIR DE DISPOSITIVOS INALÁMBRICOS

Aquí,  $p(obs|s)$  describe la probabilidad de hacer una observación  $obs$  si el estado actual es  $s$ , con la siguiente variación: una observación  $obs$  consiste en un conjunto de detecciones del dispositivo en cuestión, por todos los monitores (detectores), en el tiempo actual. El modelo de probabilidades de emisión describe la distribución de probabilidades de la ubicación de un dispositivo en toda la red, para cada segundo.

En el caso de la localización basada en lecturas de señales provenientes de dispositivos inalámbricos, se busca que el modelo capture la relación estocástica entre la intensidad de señal recibida (RSS, en inglés Received Signal Strength) y la distancia.

Las probabilidades de emisión se ven influenciadas por diversos factores, como por ejemplo la distancia, siendo este el de mayor interés. Intuitivamente, la probabilidad de hacer una



detección decrece a medida que aumenta la distancia entre el dispositivo emisor y el monitor, alcanzando cero en el rango máximo de recepción. Por otro lado, la variabilidad del canal (*fading*) también es un factor significativo, influenciado principalmente por la ubicación (el entorno) y los obstáculos presentes. Esto implica que una transmisión puede no ser detectada, incluso a corta distancia.

Una observación  $obs$  consiste de un conjunto de eventos  $e_m \in obs$ , uno por cada monitor  $m$ , tal que:

$$p(obs|s) = p_{tx} \prod_{m \in obs} p(e_m|s, tx) \quad (2)$$

donde  $tx$  indica que una transmisión ocurrió, y  $p_{tx}$  es la probabilidad de que una transmisión esté ocurriendo. Se fijó  $p_{tx} = 0,97$ , medido experimentalmente en reportes previos sobre identificación efectiva de dispositivos Bluetooth en distancias hasta 100 metros Huang et al. (2014). Cabe aclarar que el método no es sensible a este parámetro.

Cada evento  $e_m$  puede tomar los valores  $deteccion_m$  o  $nodeteccion_m$ , donde  $nodeteccion_m$  indica que el dispositivo no fue detectado por ese monitor. Las *no-detecciones* pueden no ser reportadas al servidor: la ausencia de una detección indica una no-detección. Cada  $deteccion_m$  es acompañada de una intensidad de señal.

Para todos los estados  $s$ , vale que  $p(nodeteccion_m|s) = 1 - p(deteccion_m|s)$ . Para estados fuera del alcance máximo de detección, es  $p(deteccion_m|s) = 0$ .

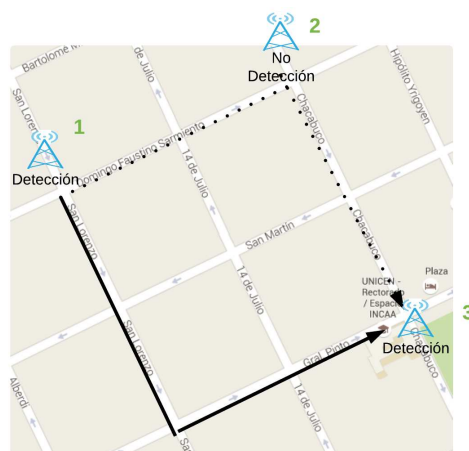


Figura 5: Ejemplo para diseño del modelo de emisión de probabilidades

La Figura 5 ilustra un ejemplo a partir del cual surge el modelo de emisión de probabilidades. Un dispositivo atraviesa el área de detección del monitor 1 y a continuación viaja a través de un camino desconocido sin ser detectado hasta el monitor 3. Aquí, el hecho de que ocurra una no-detección entre los monitores 1 y 3 aporta cierta información. Siendo que nos encontramos en igualdad de condiciones, es más probable que el dispositivo haya viajado sobre la ruta sin detectores intermedios.

Para determinar  $p(deteccion_m|s, tx)$ , nos basamos en medidas experimentales de la señal de radiofrecuencia, utilizando un vehículo y detectores Bluetooth diseñados para tal fin (monitores). Para recolectar medidas se condujo a lo largo del alcance de detección de un monitor con el

dispositivo Bluetooth encendido, registrando los paquetes transmitidos. Posteriormente se utilizó un estimador de funciones de densidad no paramétrico (método Kernel) sobre la distancia de los paquetes recibidos, o bien:

$$p(dist|RSS) = \sum_{dist \in \text{paquetes\_recibidos}} N(dist, \sigma^2). \quad (3)$$

La Figura 6 ilustra, para un paquete recibido con una intensidad RSS dada, la densidad de probabilidad de la distancia en la que se encontraría el transmisor  $p(dist|RSS)$ . A valores de intensidad altos, hay mayor probabilidad de que un vehículo se encuentre a corta distancia (ej. entre 0 y 50 metros para un  $RSS > -60$ ). En el caso opuesto, a menor intensidad, es mayor la incertidumbre sobre la distancia entre dispositivos (ej. valores por debajo de  $-70$ dB pueden provenir de cualquier ubicación en el área de cobertura del detector).

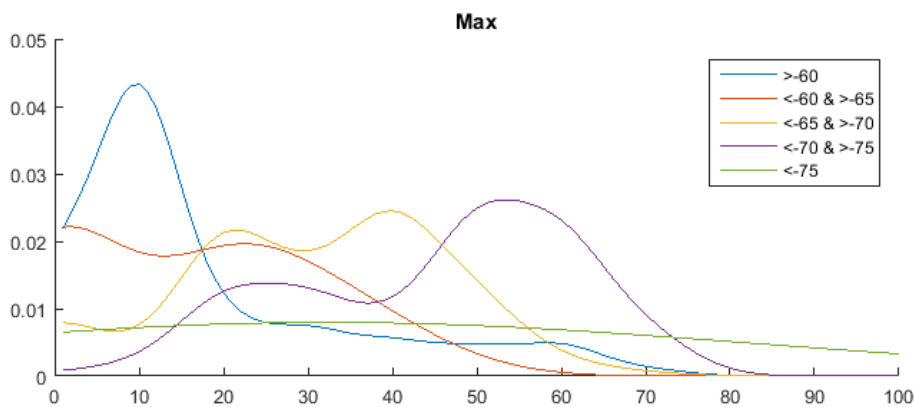


Figura 6: Funciones de densidad de probabilidad de observación Bluetooth.

Definida la función de densidad de probabilidades, es posible calcular la probabilidad de hacer una detección  $p(deteccion_m|s, tx)$  para un paquete de una intensidad de señal dada sobre la superficie que define al estado  $s$ :

$$p(deteccion_m|s, tx) = \int_x \int_y p(dist(x, y, m)|RSS) dx dy \quad (4)$$

donde  $dist(x,y,m)$  es la distancia euclídea entre la coordenada  $x, y$  y el monitor  $m$ .

## 5. RESULTADOS

Los métodos propuestos para la estimación de tiempo de viaje, aplicados a observaciones GPS y Bluetooth, fueron evaluados a través de diferentes enfoques, incluyendo un análisis cuantitativo y una discusión general de la calidad de los métodos presentados, considerando además, la caracterización de la señal Bluetooth.

El objetivo de utilizar estos métodos es medir el tiempo de viaje y compararlo con el tiempo real de la simulación. Las evaluaciones se realizan sobre la red de ejemplo  $\mathcal{G}$ , representada por la figura 7. La red cuenta con uniones internas que separan la calle en tramos (o cuadras) de aproximadamente 200 metros cada uno y, en los extremos de la red, se encuentran los distritos que se utilizan para modelar la demanda. Por otro lado, los identificadores corresponden a la

nomenclatura que llevan los arcos originales de la red en el simulador *SUMO* (Simulation of Urban MObility). Esta es una herramienta *open source* que permite modelar y simular de manera microscópica las interacciones entre vehículos individuales. A través de la definición de una red de tráfico y de las características de los vehículos involucrados, *SUMO* obtiene la velocidad y la posición de los mismos en cada intervalo de tiempo. La herramienta fue extendida en su funcionalidad básica para soportar la simulación de los sensores GPS y Bluetooth, permitiendo la colocación de detectores de señales inalámbricas, incluyendo la posibilidad de ajustar el error de los dispositivos. Si bien trabajos previos como el de [Musa y Eriksson \(2012\)](#) presentan un enfoque de estimación similar utilizando señales Wi-Fi, no validan el método por simulación. En la red se despliegan monitores Bluetooth de igual modo a como fueron utilizados para la simulación de detecciones. En la simulación, todos los monitores fueron configurados para realizar detecciones hasta 100 metros de distancia entre un vehículo y el detector, respetando el comportamiento observado en las pruebas de campo.

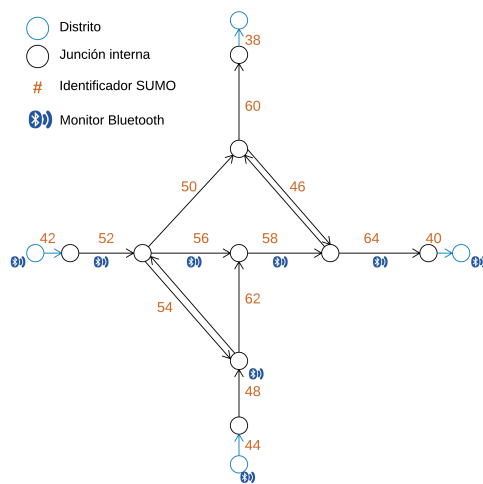


Figura 7: Red de calles para las pruebas de la plataforma.

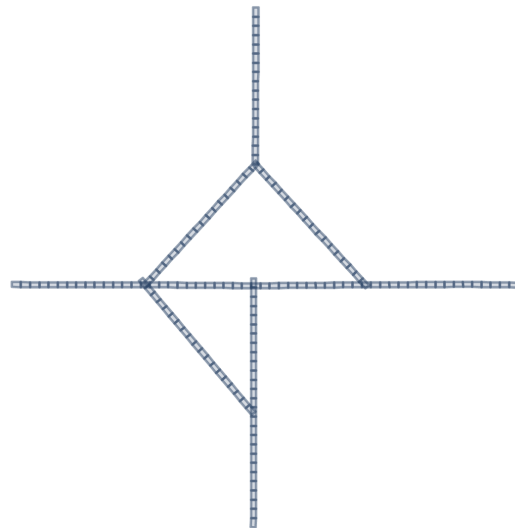


Figura 8: División de la red en segmentos (*SegmentoViterbi*).

Para la ejecución del algoritmo de estimación, los segmentos son divididos aplicando la metodología propuesta en el apartado 3.3, formando  $\mathcal{G}' = \langle \mathcal{V}', \mathcal{E}' \rangle$ , con un total de 154 estados  $\mathcal{V}'$  (Figura 8).

Se establecen dos recorridos  $R1$  y  $R2$  de longitud  $l = 1,22$  km. Se realizan simulaciones para ambos recorridos, variando los parámetros índice de penetración y probabilidad de observación. Finalmente se realizan pruebas sobre los recorridos  $R1$  y  $R2$  conjuntamente.

### 5.1. Estimación GPS

En el caso de observaciones GPS, *SUMO* arroja posiciones en cada ciclo de simulación (Figura 10 - A) y sobre estas se aplican dos procesos. En primer lugar, se descartan observaciones con cierta probabilidad  $p_{obs}$  (B) y, en segundo lugar, se modela el error de una observación como una variable aleatoria distribuida normalmente con media cero y desvío dado (C, D y E).

Luego, el algoritmo estima una secuencia de segmentos que describe la trayectoria más probable  $\tau_{opt}$  dada la secuencia de observaciones  $\mathcal{P}$ . Se considera como ground truth a los segmentos recorridos por la secuencia original  $\mathcal{T}$  en el intervalo de tiempo definido por el *timestamp*

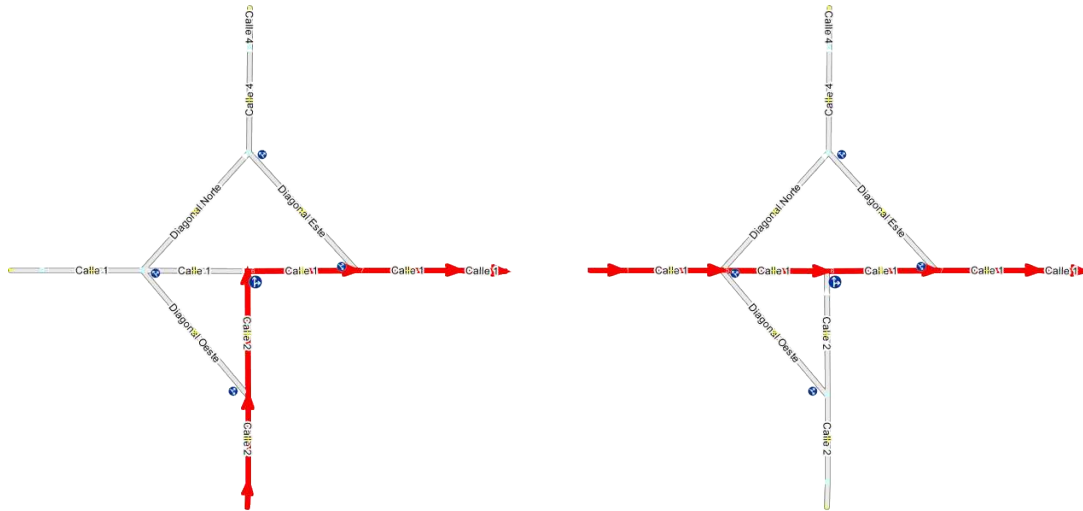


Figura 9: Recorridos definidos R1 y R2, respectivamente.

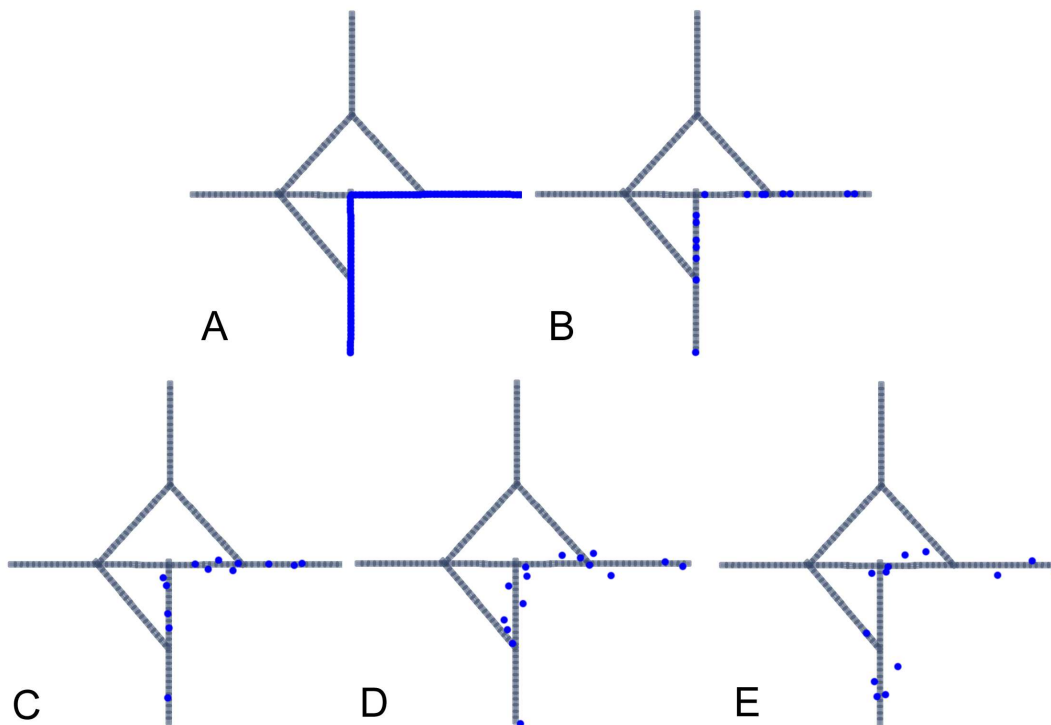


Figura 10: Recorrido simulado para 1 vehículo en el recorrido R1, con probabilidades de observación:  $p_{obs} = 1$  (A) y  $p_{obs} = 0,125$  (B); a partir de simulaciones con  $p_{obs} = 0,125$ , en C, D y E se visualizan perturbaciones (PF) de 1, 5 y 10 metros, respectivamente.

de la primera y última detección, resultado del descarte de observaciones.

Para las comparaciones se tendrán en cuenta los siguientes aspectos:

- Velocidad media general (por simulación).
- Tiempo de viaje medio por tramo.
- Relación tiempo - segmento.

A partir de la secuencia de segmentos  $\tau_{opt}$ , se determina el tiempo de viaje medio por segmento. Esto es, por ejemplo, si un vehículo se traslada por los segmentos 1, 2 y 3 con la frecuencia:

$$\tau_{opt} = \{1, 1, 2, 3, 3\},$$

como resultado se tiene que el vehículo permaneció dos segundos en el segmento 1, un segundo en el segmento 2 y dos segundos en el segmento 3. Este cálculo se corresponde con la frecuencia absoluta de las detecciones en cada segmento  $F_i$  y se realiza para todos los vehículos, tomando el valor medio de cada segmento. Luego, se agrupan las demoras obtenidas según el tramo al que pertenece cada segmento. El tiempo de viaje estimado en cada tramo  $e$  ( $TTEst_e$ , por sus siglas en inglés: Estimated Travel Time) se calcula a partir de la suma de las demoras de cada segmento  $s \in S$  tal que  $s$  surge de la división del tramo  $e$ .

$$TTEst_e = \sum_{i=1}^S SegmentTT_i \quad (5)$$

El tiempo de viaje en cada segmento es, entonces, la relación entre la suma de la frecuencia absoluta de cada estado  $F_i$  y el total de vehículos  $V$ , o bien:

$$SegmentTT_i = \frac{\sum F_i}{V} \quad (6)$$

A continuación se detallan las series de simulación para los recorridos  $R1$  y  $R2$ . En primera instancia se muestran las distancias y tiempos totales para cada recorrido y luego, se analizan los resultados desde otra perspectiva analizando los tiempos por cada tramo.

- $P_{obs}$ : Probabilidad de observación (o probabilidad de lectura), es la probabilidad de que un vehículo realice una transmisión dada su ubicación.
- PR: Índice de penetración, se corresponde con el porcentaje de vehículos que portan un dispositivo de posicionamiento.
- DistSim: Distancia recorrida por simulación (m).
- DistEst: Distancia Estimada (m).
- TTEst: Tiempo de viaje total estimado por arco (s).
- TTSim: Tiempo de viaje total Simulado (s)
- VelSim: Velocidad media Simulado (m/s)
- VelEst: Velocidad media Estimada (m/s)

Recorrido	Pobs	PF	TTSim	TTEst	DistSim	DistEst	VelSim	VelEst
<i>R1-1v (1)</i>	1	0	93	93	1208.40	1200.0	12.99	12.90
<i>R1-1v (2)</i>	1	10	93	93	1208.40	1200.0	12.99	12.90
<i>R1-1v (3)</i>	0.125	0	88	88	1120.45	1134.34	12.73	12.89
<i>R1-1v (4)</i>	0.125	5	85	85	1101.09	1088.89	12.95	12.81
<i>R1-1v (5)</i>	0.125	10	92	92	1194.72	1200.00	12.99	13.04

Tabla 1: Resultados de tiempos y distancias en la simulación de un vehículo, sobre el recorrido R1, aplicando el Método 1

- PF: Factor de perturbación.
- Veh: Cantidad de vehículos observados.

La Tabla 1 detalla los resultados de la aplicación del método 1 sobre la simulación de un vehículo en el recorrido R1, variando los parámetros  $p_{obs}$  y el factor de perturbación  $PF$ . Aquí, los tiempos coinciden entre los casos simulados y estimados porque se obtienen a partir de la diferencia entre las marcas de tiempo de la primera y última observación. Las distancias **estimadas** se corresponden con la distancia que abarcan los estados (segmentos) resultantes; en otras palabras, los valores de distancia serán múltiplos de la longitud del segmento  $l = 22,22$  m. Finalmente, podemos observar una comparación de las velocidades medias obtenidas en la Figura 11. La diferencia máxima registrada es de 0.16 m/s.

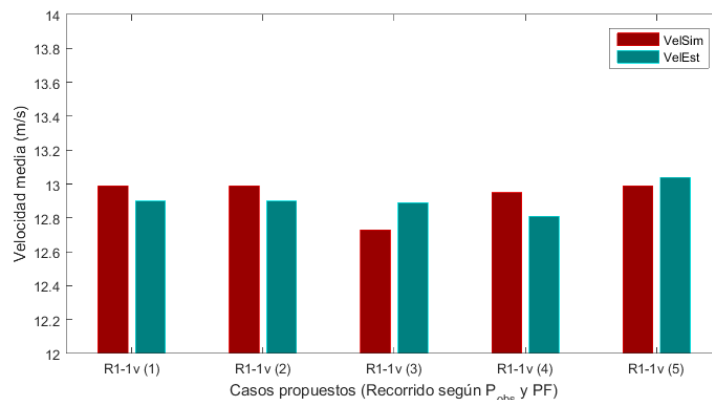


Figura 11: Estimación de la velocidad media general para las variaciones de parámetros  $p_{obs}$  y  $PF$  sobre el recorrido R1

Es importante considerar otros aspectos en el análisis de las estimaciones. Para el estudio de los tiempos de viaje sobre los tramos (cuadras) de la red original, se utilizan valores agregados (promedios) de las demoras estimadas. En la Figura 12 se muestran los resultados para el tiempo de viaje estimado por tramos (cuadras o *edges*) para 10, 100 y 200 vehículos, respectivamente.

Recordando la estructura original de la red  $\mathcal{G}$ , puede observarse que los arcos que presentan mayor error (arco 40 y arco 44) son los que corresponden a los distritos para modelar la demanda. En estas zonas de pequeña longitud es donde comienzan a ingresar y retirarse los vehículos simulados. Por lo mencionado anteriormente, el tiempo de viaje asignado a estos tramos es más corto respecto a los demás. En ensayos generales, se obtuvo un error máximo de 2.72 segundos en el tramo número 56 en la simulación de 200 vehículos.

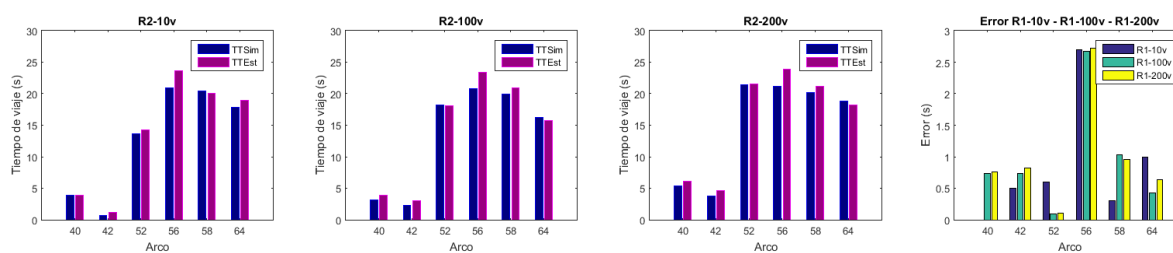


Figura 12: Estimación del Tiempo de Viaje para 10, 100 y 200 vehículos en el recorrido R2, con  $p_{obs} = 0,125$  y  $PF = 10$

La Tabla 2 incluye una comparación del error del tiempo de viaje obtenido en simulaciones para 10 vehículos entre los métodos propuestos. En términos generales, se observó un leve aumento del error medio en el método #2 respecto al método #1. Debido a que el método #2 es de mayor complejidad en cuanto a la cantidad de operaciones que se realizan y no significa un aporte en la calidad de la estimación, en la evaluación Bluetooth se consideró utilizar únicamente el método #1.

Método	Recorrido	$P_{obs}$	PF	Error Medio (seg)
1	1	0.5	5	0.7667
1	1	0.125	5	0.4500
1	2	0.5	5	1.0833
1	2	0.125	5	1.0500
2	1	0.5	5	0.6333
2	1	0.125	5	0.6333
2	2	0.5	5	1.0833
2	2	0.125	5	1.7333

Tabla 2: Cuadro comparativo del error medio en la estimación del Tiempo de Viaje (TT) obtenido en simulaciones para 10 vehículos, en los recorridos R1 y R2, estimadas a partir de los métodos 1 y 2.

## 5.2. Estimación Bluetooth

En el caso de las estimaciones Bluetooth, se valida el funcionamiento del modelo planteado y se compara con la estimación obtenida a partir de la caracterización de la señal BT.

En un escenario ideal, la potencia de señal recibida en función de la distancia puede modelarse a través de la ecuación de transmisión de Friis [Rappaport et al. \(1996\)](#). Para simular señales Bluetooth se aplica una metodología sobre los datos experimentales utilizados en la estimación de las funciones de densidad *opdf*.

SUMO registra la distancia de los vehículos a los monitores desplegados en cada instante de simulación. Esto ocurre sólo si el vehículo se encuentra en el rango de detección del monitor. Aquí, es necesario traducir el valor de distancia a una intensidad de señal. La traducción se realiza a través de dos métodos propuestos.

El primer método genera datos utilizando la ecuación (7), obtenida por regresión exponencial sobre el conjunto de medidas experimentales descrito previamente (6). Se obtiene entonces una

intensidad de señal a partir de la distancia del vehículo al detector, registrada por el simulador.

$$signal = -62,85 e^{d 0,001936} \tag{7}$$

Respecto a los datos simulados, es deseable contar con el error inherente a los dispositivos Bluetooth. Si bien es sabido que el error mencionado no sigue una distribución normal, estas son ampliamente utilizadas en estudios de investigación por los buenos resultados que ofrecen Cinefra (2014). Se introduce entonces el error a través de una aleatorización basada en distribuciones normales, con medias y desvíos que dependen de la distancia de la detección: Dada una distancia  $d$ , se asocia una distribución normal con media  $\mu(d)$  y desvío  $\sigma(d)$ , calculadas por interpolación sobre las medias y desvíos registrados (Figura 13). Para cada distancia registrada se obtienen valores aleatorios según la distribución.

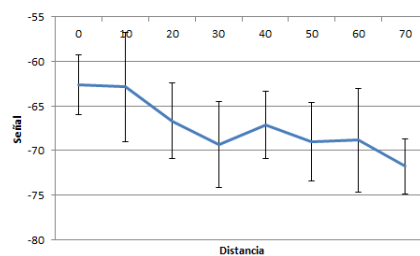


Figura 13: Señal media y desvío estándar de las muestras experimentales según la distancia al monitor

En la Figura 14 se observan tiempos de viaje aplicando el método a detecciones Bluetooth. Los valores de intensidad de señal recibida se obtuvieron evaluando la ecuación de Friis con las distancias entre el vehículo simulado y cada monitor.

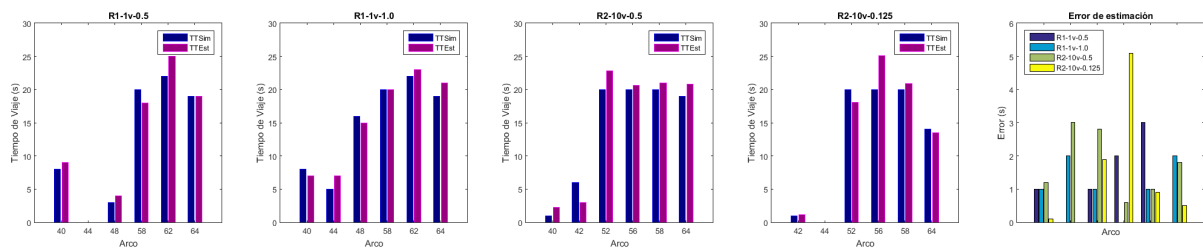


Figura 14: Estimación del Tiempo de Viaje vía detección Bluetooth para 1 y 10 vehículos en el recorrido R1 y R2, con diferentes combinaciones de  $p_{obs}$

En la Figura 15 se representa el avance de un vehículo sobre los segmentos (eje vertical) a lo largo del tiempo (eje horizontal). En este caso, se trata de un único vehículo en el recorrido 1, y la diferencia media entre tiempo simulado y tiempo estimado entre segmentos es de apenas 1 segundo, validando de esta forma la metodología utilizada.

Analizando el comportamiento de la estimación para un vehículo, en la Tabla 3 se realiza una comparación del tiempo de viaje estimado a partir la caracterización experimental de la señal de Bluetooth. Se observan valores para distintas probabilidades de lectura  $p_{obs}$ , sobre el trayecto de un vehículo en el recorrido 2, utilizando el método de simulación Bluetooth por perturbación



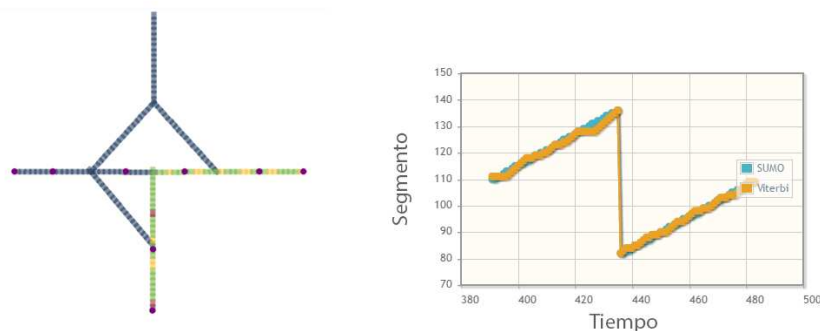


Figura 15: Representación visual en la plataforma de la estimación de tiempo de viaje con monitores Bluetooth

Arco	TTSim	TTEst-1	TTEst-0.8	TTEst-0.5
52	20.00	22.00	23.00	25.00
56	21.00	24.00	24.00	21.00
58	21.00	22.00	21.00	22.00
64	19.00	20.00	20.00	19.00

Tabla 3: Tiempos de viaje estimados para 1 vehículo en el Recorrido R2, a distintos valores de  $p_{obs}$ .

(segundo método de simulación). Aquí se utilizan las medidas experimentales para generar observaciones aleatorias con el error medio observado en el estudio de campo realizado.

Sobre los valores estimados, se percibe un error máximo de 5 segundos y un error medio máximo de 1.75 segundos, lo cual es relativamente bajo respecto al tiempo de viaje real de cada tramo en el recorrido (menor al 8%). Cabe destacar que el filtro  $p_{obs} = 0,5$  arroja alrededor de 30 detecciones entre los 6 monitores de la red y, en las pruebas de velocidad, se observaron como mínimo 5 detecciones por monitor (a 80 Km/h), por lo cual, la probabilidad de lectura indicada permite simular un tamaño de muestra comparable a la realidad.

Finalmente, en la Tabla 4 se comparan las estimaciones aplicando distintos índices de penetración (PR) y probabilidades de lectura (Obs). En los dos primeros casos el índice de penetración utilizado es del 100% (PR1), lo cual representa que todos los vehículos tienen un dispositivo Bluetooth activo, y para el tercer y cuarto caso (PR0.1) un índice del 10%. Por otra parte, la probabilidad de lectura utilizada es del 100% (Obs1) en el primer y tercer caso y del 50% (Obs0.5) en el segundo y cuarto caso. Como se puede observar, el error no aumenta en demacía cuando el índice de penetración disminuye. Considerando que en pruebas de campo hemos registrado un factor de penetración de equipos con Bluetooth del 15%, se puede concluir que el método propuesto debería funcionar con un error aceptable en un caso real. Por otro lado, conforme se reduce la probabilidad de lectura, aumenta el error debido a la alta incertidumbre de la ubicación.

## 6. CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo se introdujo un método de estimación de tiempos de viaje en redes de tráfico a partir de señales GPS y Bluetooth. Las metodologías propuestas en ambos casos fueron validadas mediante simulación microscópica.

Arco	TTSim	PR1/Obs1	PR1/Obs0.5	PR0.1/Obs1	PR0.1/Obs0.5
40	8	1.71	0.76	1.5	3.67
44	5	1.48	2.72	1.33	2.89
52	20	1.42	0.42	1.17	2.56
62	22	3.63	4.64	3.83	4.56
64	19	5.76	7.87	5.67	8.56

Tabla 4: Cuadro comparativo del error (seg) en la estimación del tiempo de viaje (TT) respecto al tiempo real TTSim, aquí se simularon 200 vehículos en el recorrido #2, con tasas de penetración  $PR = \{1, 0.1\}$  y probabilidades de observación  $P_{obs} = \{1, 0.5\}$ .

La utilización de detectores inalámbricos implica una evolución en los sistemas de transportes inteligentes ya que estos aportan características de identificación y seguimiento vehicular simplificando la infraestructura y costos de los sistemas actuales de medición del tráfico.

En cuanto a los resultados, se obtienen tiempos de viajes más precisos utilizando la técnica activa, aunque esta implica que el usuario instale una aplicación específica en su celular a través de la cual envíe su posición a intervalos de tiempos regulares.

Desde un punto de vista práctico este hecho es difícil de lograr y se considera con mayor posibilidad de implementación un enfoque basado en registrar de manera pasiva la interacción entre equipos Bluetooth (donde no sea necesario la instalación de ninguna aplicación por parte de los usuarios). Además, se observa que con este enfoque y con un índice de penetración del 10 % (similar al obtenido a través de pruebas de campo) se obtienen resultados más que aceptables. Cabe destacar que el mencionado índice se irá incrementando con el paso del tiempo.

Como trabajo futuro, se plantea la necesidad de validar los algoritmos presentados en un escenario real para lo cual se precisa instalar varios monitores Bluetooth en un ambiente urbano.

## REFERENCIAS

- Bull A. *Congestión de tránsito: el problema y cómo enfrentarlo*. 87. United Nations Publications, 2003.
- Cinefra N. An adaptive indoor positioning system based on bluetooth low energy rssi. 2014.
- Dunlap M., Li Z., Henrickson K., y Wang Y. Estimation of origin and destination information from bluetooth and wi-fi sensing for transit. En *Transportation Research Board 95th Annual Meeting*, 16-6837. 2016.
- Huang J., Albazraqoe W., y Xing G. Blueid: a practical system for bluetooth device identification. En *INFOCOM, 2014 Proceedings IEEE*, páginas 2849–2857. IEEE, 2014.
- Jamie Houghton J.R.y.C.L. Transporte inteligente: Cómo mejorar la movilidad en las ciudades. IBM Institute for Business Value, Sector Público, 2009.
- Jekabsons G., Kairish V., y Zuravlyov V. An analysis of wi-fi based indoor positioning accuracy. *Scientific Journal of Riga Technical University. Computer Sciences*, 44(1):131–137, 2011.
- Jones T.W., Marzen L., y Chappelka A. Horizontal accuracy assessment of global positioning system data from common smartphones. *Papers in Applied Geography*, 1(1):59–64, 2015.
- Musa A. y Eriksson J. Tracking unmodified smartphones using wi-fi monitors. En *Proceedings of the 10th ACM conference on embedded network sensor systems*, páginas 281–294. ACM, 2012.
- Rappaport T.S. et al. *Wireless communications: principles and practice*, volumen 2. Prentice Hall PTR New Jersey, 1996.