# How can be improve bus ETAs?
# Using real-time position data to estimate road state

**Tom Elliott and Thomas Lumley**

Department of Statistics, University of Auckland, New Zealand

tom.elliott@auckland.ac.nz

## 1. Introduction

- real time information (RTI) has become an important component of public transit systems, providing commuters with up to date information on the location and estimated arrival time (ETA) of buses
- transit vehicle tracking has been used for several decades to obtain ETAs, with much research into methods for improving the vehicle models used [1–4]
- the main source of uncertainty is congestion, particularly in networks with poor public tranport infrastructure (e.g., dedicated bus lanes)
- research shows that ETA accuracy can be improved by using travel time information from previous buses along a road [5], however this only applied to a specific, manually selected road
- we propose a generalised approach to modeling transit vehicles and network congestion
  - a vehicle model estimates vehicle speed/travel time along a road
  - the state (vehicle speed) of the road is updated to reflect real-time congestion
  - ETAs are updated using congestion information along all intermediate roads

## 2. GTFS network construction

- **GTFS** is an API specification for transit data [6], and includes route shape information
- available in over 500 locations worldwide
- **transit network** consisting of intersections and connecting road segments constructed from the raw GTFS data

1. raw GTFS data provides **one shape per route**
2. **identify points of intersection** between one or more routes using algorithm adapted from [7]
3. **split shapes at intersections** to obtain shapes for each individual road segment
4. express each route as a sequence of road segments

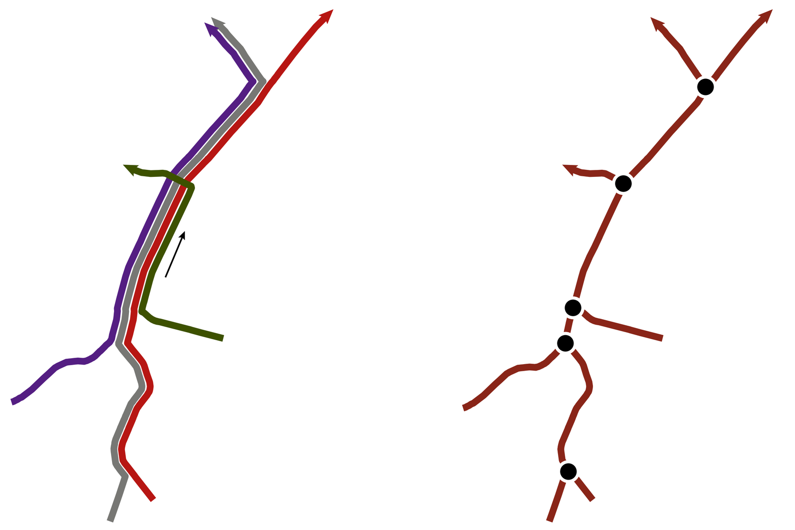

**Figure 1:** *An example transit netork produced from five routes. Left: the raw GTFS shapes; Right: the generated transit network with intersections shown as dots.*

- Implementation in progress: the `gtfsnetwork` R package, github.com/tmelliott/gtfsnetwork

## References

[1] Z. Wall and D.J. Dailey. An algorithm for predicting the arrival time of mass transit vehicles using automatic vehicle location data. In *Proceedings of the Transportation Research Board Annual Meeting*, 1999.

[2] D. Dailey, S. Maclean, F. Cathey, and Z. Wall. Transit vehicle arrival prediction: Algorithm and large-scale implementation. *Transportation Research Record: Journal of the Transportation Research Board*, 1771:46–51, jan 2001.

[3] F. W. Cathey and D. J. Dailey. A prescription for transit arrival/departure prediction using automatic vehicle location data. *Transportation Research Part C: Emerging Technologies*, 11(3-4):241–264, jun 2003.

[4] Etienne Hans, Nicolas Chiabaut, Ludovic Leclercq, and Robert L. Bertini. Real-time bus route state forecasting using particle filter and mesoscopic modeling. *Transportation Research Part C: Emerging Technologies*, 61:121–140, dec 2015.

[5] Bin Yu, William H. K. Lam, and Mei Lam Tam. Bus arrival time prediction at bus stop with multiple routes. *Transportation Research Part C: Emerging Technologies*, 19(6):1157–1170, dec 2011.

[6] Google Developers. What is GTFS? https://developers.google.com/transit/gtfs/, 2006.

[7] Yongchuan Zhang, Jiping Liu, Xinlin Qian, Agen Qiu, and Fuhao Zhang. An automatic road network construction method using massive gps trajectory data. *ISPRS International Journal of Geo-Information*, 6(12), 2017.

## 3. Vehicle state model

- sequential Bayesian methods well suited to **real-time vehicle tracking**
- in Auckland, GTFS realtime positions obtained approximately every 30 seconds, but high variability
- **particle filter**: general, flexible estimation method
  - vehicle state approximated by a sample of particles $\tilde{\boldsymbol{X}}_k = (\boldsymbol{X}_k^{(i)})_{i=1}^N$
  - particles transitioned independently, so multimodal distributions are no issue (e.g., at bus stops and intersections)
  - likelihood is intuitive: the distance between the estimated and observed location
- measurement function $h : \mathbb{R} \mapsto \mathbb{R}^2$ calculates particle's map position from distance traveled along shape
- our particle filter estimation procedure is as follows:

1. predict trajectory of each particle using transition function $f$ and system noise parameter $Q_k$ as shown in figure 2

$$\boldsymbol{X}_k^{(i)} = f(\boldsymbol{X}_k^{(i)}, w_k), \quad w_k \sim N(0, Q_{k-1})$$

2. assume $\boldsymbol{Y}_k$ is a noisy measurement of true position with GPS error $\sigma_y^2$ (fig 3), and define $g : \mathbb{R}^2 \mapsto \mathbb{R}^2$ such that $dist(g(\boldsymbol{Y}_1), g(\boldsymbol{Y}_2))$ is the ground distance between the points, then the measurement model is

$$g(\boldsymbol{Y}_k) \sim N\left(g(h(\boldsymbol{X}_k)), \begin{bmatrix}\sigma_y^2 & 0\\ 0 & \sigma_y^2\end{bmatrix}\right)$$

and $(\delta_k^{(i)})^2 = dist(g(h(\boldsymbol{X}_k^{(i)})), g(Y_k))^2$ is the sum of two independent normal r.v.'s with mean 0 and variance $\sigma_y^2$

$$\left((\delta_k^{(i)})^2/\sigma_y^2\right) \sim \chi^2(2) \sim \mathrm{Exp}(0.5)$$

3. evaluated the likelihood for each particle

$$p(\boldsymbol{Y}_k | \boldsymbol{X}_k^{(i)}) = 0.5 e^{-(\delta_k^{(i)})^2/2\sigma_y^2}$$

4. update state by resampling particles with replacement, using likelihood weights (fig 4)

$$w^{(i)} = p(\boldsymbol{Y}_k | \boldsymbol{X}_k^{(i)}) / \sum_{j=1}^{N} p(\boldsymbol{Y}_k | \boldsymbol{X}_k^{(j)})$$

5. use resulting trajectories to estimate vehicle speed along road segments to update network in section 4

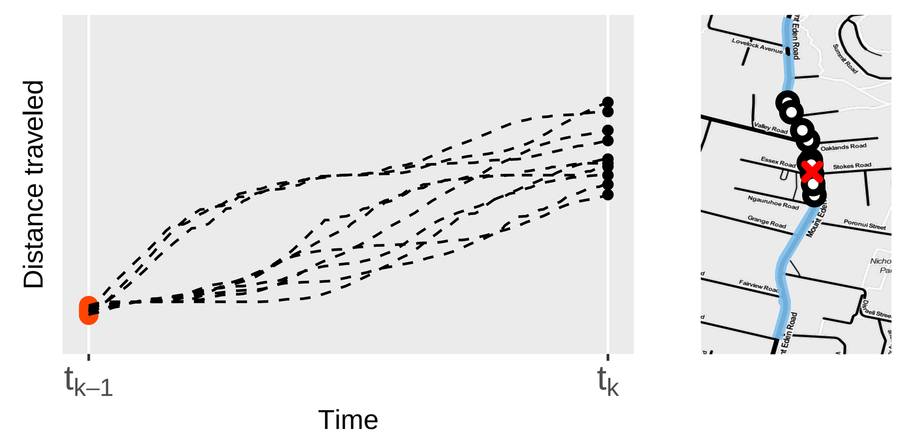


**Figure 2:** *Left: simulated particle trajectories. Right: particle positions $h(\boldsymbol{X}_k^{(i)})$; observation $\boldsymbol{Y}_k$ in red.*

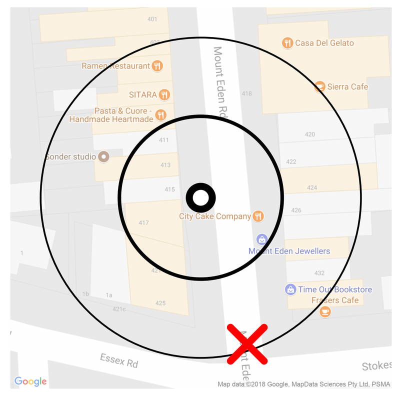

**Figure 3:** *$Y_k$ (red cross) is a bivariate normal r.v. with mean and variance represented by the black dot and concentric rings, respectively.*

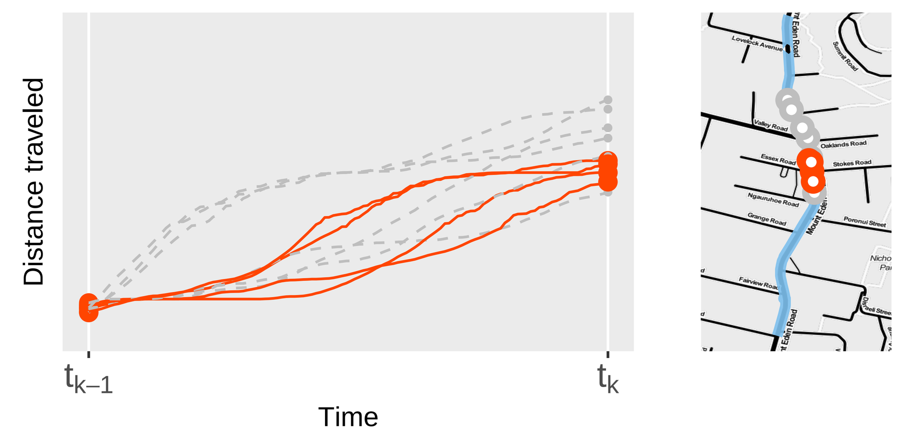


**Figure 4:** *After resampling, a posterior sample of trajectories is obtained (orange).*

## 4. Network state model

- network model to estimate and predict network state
- segment $j$ has state $\beta_r^j$ (vehicle speed) at time $t_r$ with variance $P_r^j$
- estimate state using an adapted extended Kalman filter (EKF) algorithm
- Step 1: predict future state
  - define **transition function** $a$ such that the state converges to the prior (fig. 5)
  - prior mean speed $\mu_j(t)$ and variance $\psi_j(t)$ at time $t$ from **historical data** (blue lines in fig. 5)
  - use EKF equations to recursively predict state in 1 second intervals
  - define **system noise** so $P_r^j$ converges to $\psi_j(t_r)$
- Step 2: update state when observations recieved from step 5 in section 3 (red point fig. 6)
  - **measurement error:** variance of particle speeds
  - use EKF update equations to update state at time of observation

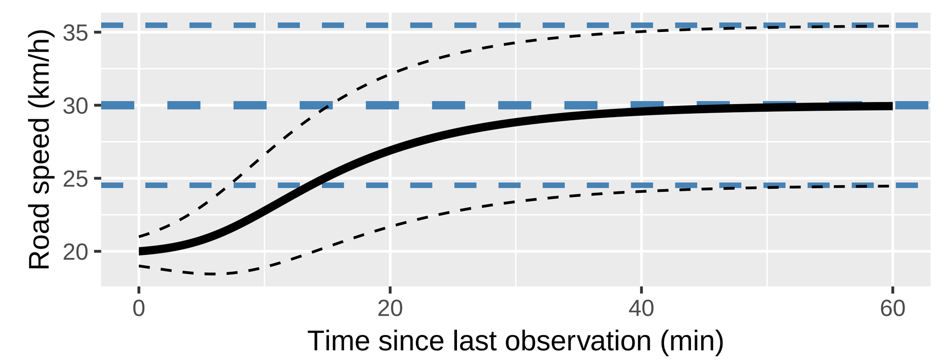


**Figure 5:** *Road state $\hat{\beta}^j$ (with variance $P_r^j$, dashed black lines) converge to the prior (blue lines).*

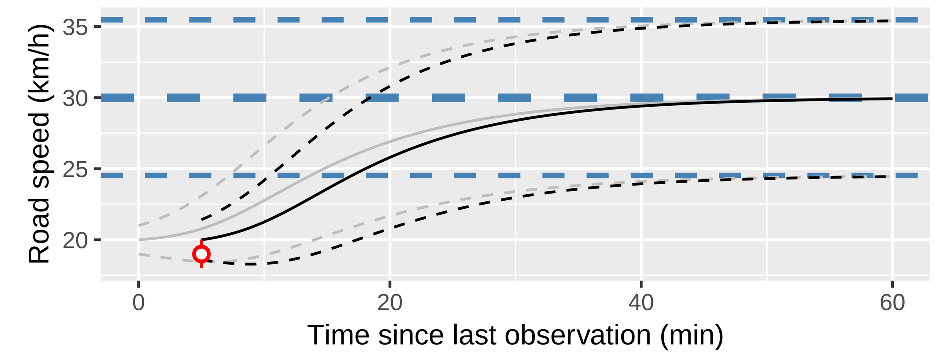


**Figure 6:** *After recieving a vehicle speed estimate (red dot), the state is updated.*

## 5. Predicting arrival time

- for each particle, simulate journey along remainder of route
- **simulate speed** $v_t^j \sim N(\hat{\beta}_t^j, P_t^j)$ for each upcoming segment $j$
- simulate intersection and bus stop wait times and compute arrival time at each upcoming stop
- resulting ETA distribution can be conveyed to passengers
  - a point estimate
  - and/or a prediction interval (for commuters, this would be understood as a min and "max" wait time)
- ETAs are typically reported in discrete minutes. For the example in figure 7, the distribution might be summarised with
  - a **point estimate** of 5 minutes
  - a **prediction interval** of 4–8 minutes
- summary statistics need to be chosen such that, as the bus approaches, the estimates decrease

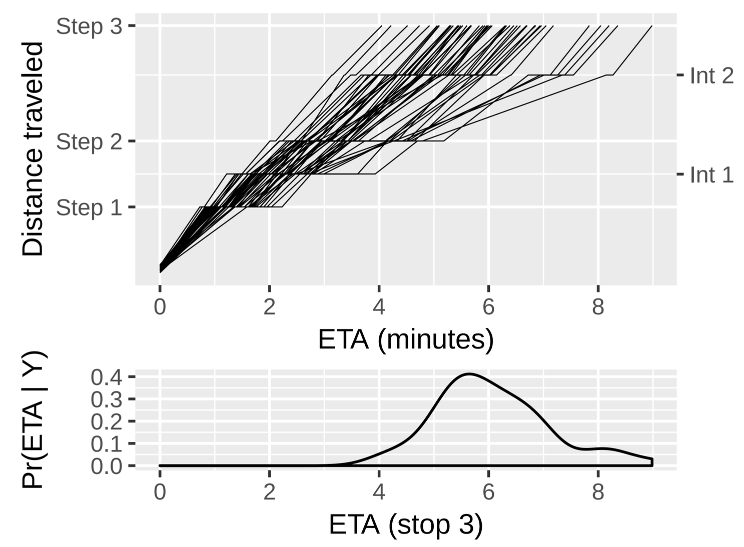


**Figure 7:** *Top: travel time predictions for a bus, showing locations of stops (left axis) and intersections (right axis). Bottom: posterior density of ETAs for stop 3.*

## 6. Conclusion and future work

- segmenting routes into route-independent segments allows vehicle observations to update the road network
- real-time network state used to predict arrival time

**Next steps:**

- improve the network state model: variable speeds along a roads (i.e., $\mu_j(t, d)$ depends on time and distance along segment), include covariates in state transition (adjacent segments, yesterday's traffic, weather, etc.)
- develop a stop-time and intersection-wait time model to more accurately predict wait times
- investigate ideal summary statistics for ETAs (both point and interval prediction)

2018 ISBA World Meeting, Edinburgh, UK, June 24–29 2018