Mathematics for trajectory extrapolation using vehicle and human traffic data towards zero traffic fatalities

Final Presentation, G-RIPS 2023

August 7, 2023

Melissa Adrian, The University of Chicago
Namito Asano, Musashino University
Ann Mwangi, African Institute for Mathematical Sciences
Tomoki Okamoto, Musashino University
Aaron Scheiner, Northwestern University

Mentors:
Natsuo Miyatake, Tohoku University
Masao Ono, IHI Corporation

Date of presentation August 7, 2023.
Revised August 8, 2023.
Finalized by mentors August 10, 2023.
Available online August 25, 2023.
IHI Doc. No. HAPD116432.

Outline

- Introduction
- Data Exploration & Preprocessing
- Methods
  - Gaussian State-Space Model
  - Ensemble Kalman Filter (EnKF) and Smoother (EnKS)
  - Car Following Model
  - 2D to 1D Coordinate Conversion
- Conclusion
- Future Directions

Introduction

Motivation [1]

Why do we analyze traffic data?

- Drastically improve both safety and mobility in traffic systems
- Feed data to autonomous vehicles both for in-house development and on-road transmission

What is our group doing?

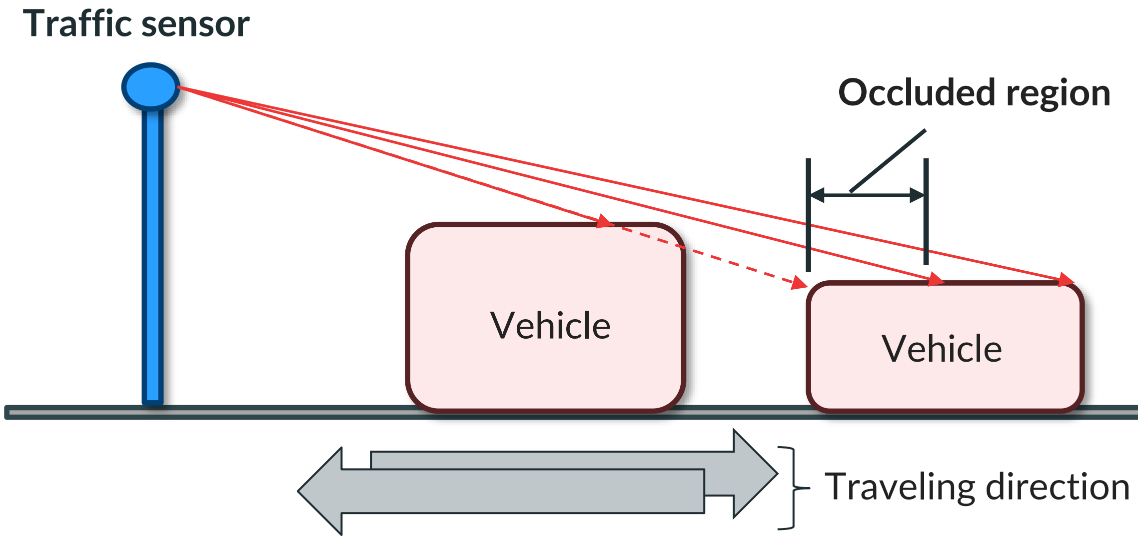
- Filtering out noise
- Extrapolating trajectories from partial data

What are our main end goals?

- To reduce the number of traffic sensors needed at an intersection
- To develop an advanced technology for traffic flow analysis based on actual driver's behavior

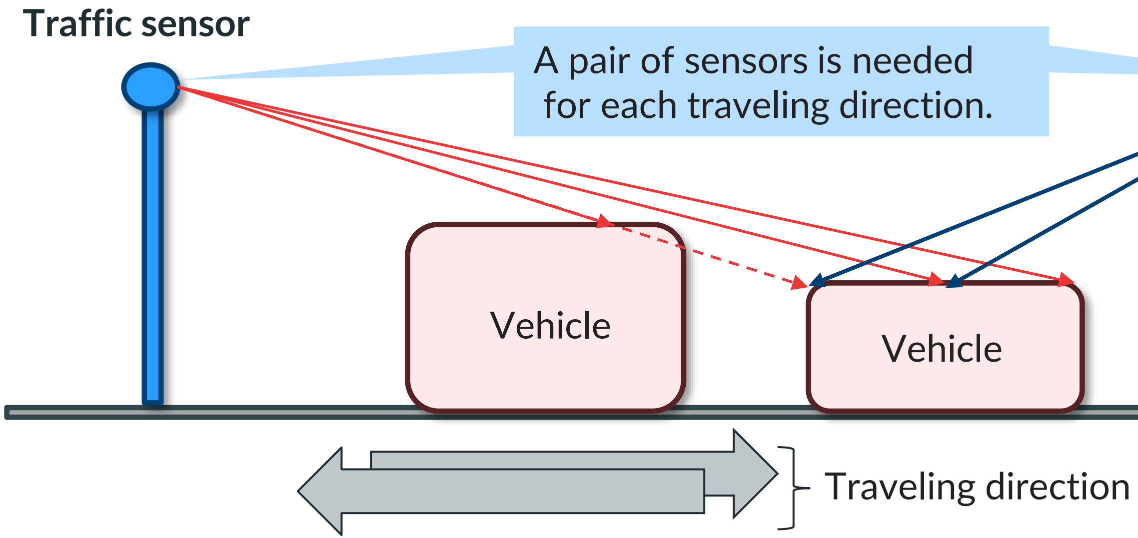
Problem Statement

- A traffic sensor inevitably yields occluded regions due to overlapping objects.
- This leads to incomplete trajectories, hindering the analysis of incidents and congestion.
- At a typical crossroad, 4 sensors give full data, and anything less results in incomplete data.



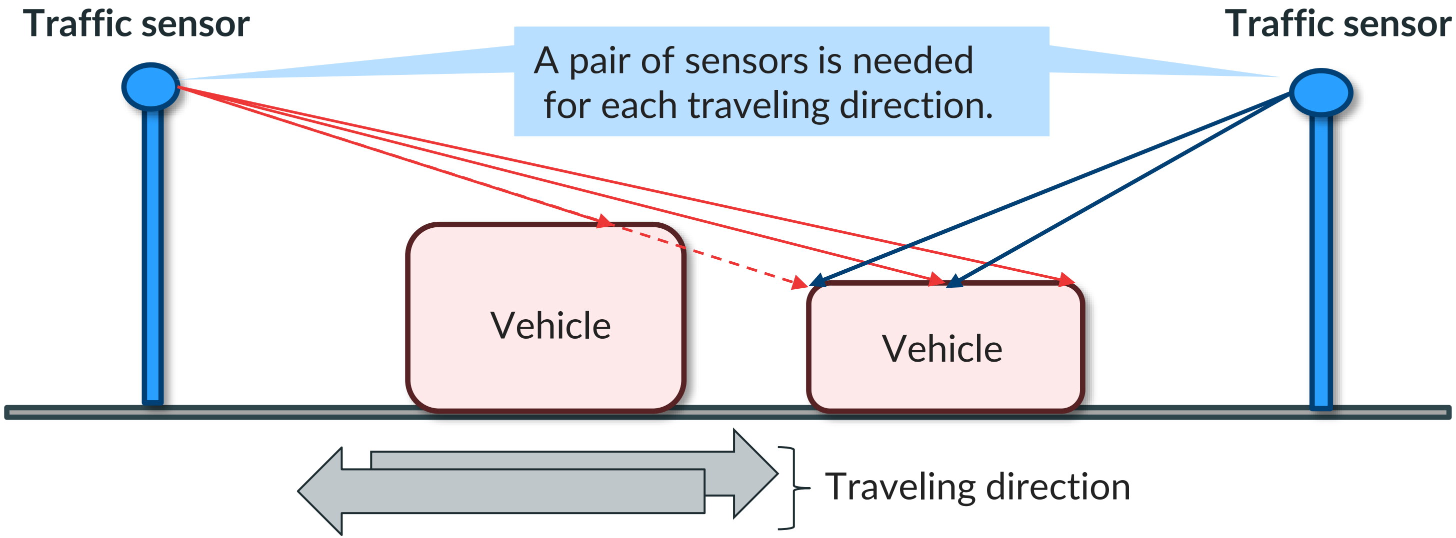
Problem Statement

- A traffic sensor inevitably yields occluded regions due to overlapping objects.
- This leads to incomplete trajectories, hindering the analysis of incidents and congestion.
- At a typical crossroad, 4 sensors give full data, and anything less results in incomplete data.



Problem Statement

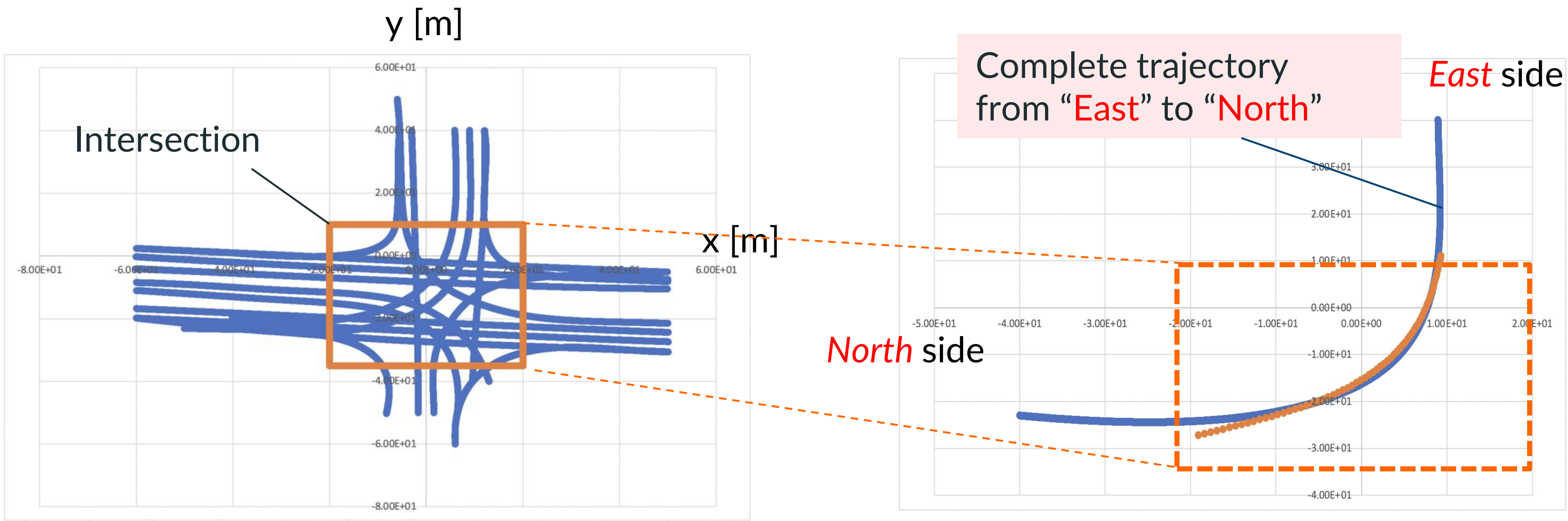
- To address the issue, we want to generate continuous data of driving behaviors by extrapolating physically probable values from incomplete traffic flow data.



Data Description

- Each vehicle has 2D time series position (x, y) on the local coordinate system.
- Each vehicle is given labels of an inflow direction and an outflow direction at an intersection.
- “Unknown” is given to the vehicles whose trajectory is incomplete in the intersection.

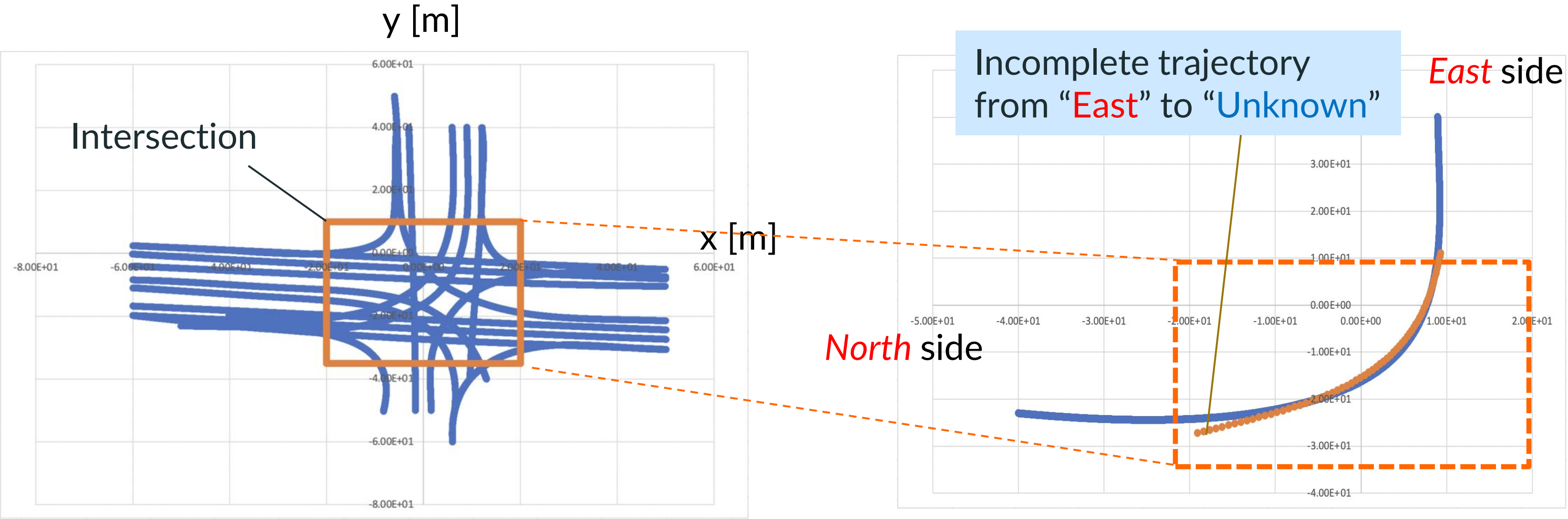
Examples of trajectories at an intersection



Data Description

- Each vehicle has 2D time series position (x, y) on the local coordinate system.
- Each vehicle is given labels of an inflow direction and an outflow direction at an intersection.
- "Unknown" is given to the vehicles whose trajectory is incomplete in the intersection.

Examples of trajectories at an intersection

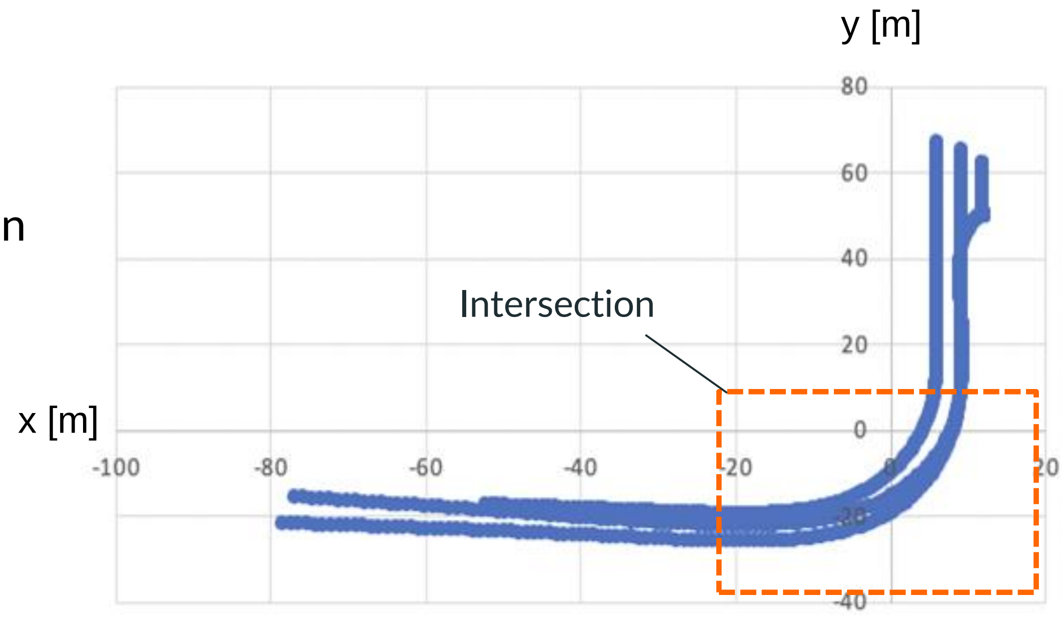


Data Exploration & Preprocessing

Lanes and Lane Changes

Examples of intersection behavior around lanes include:

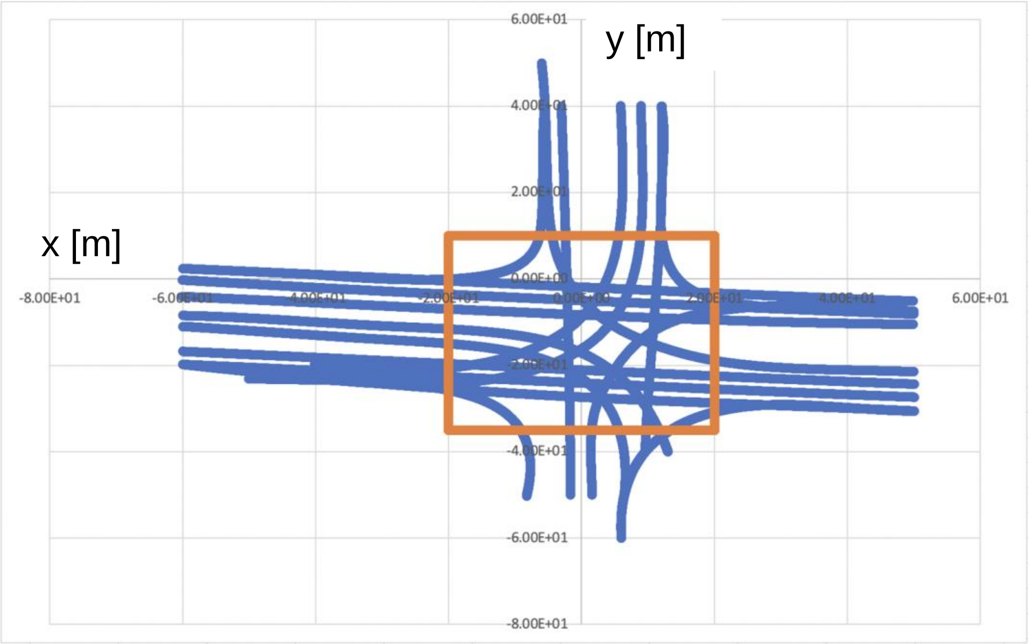
- Inner, outer, and center lanes
- Lane changes before the intersection
- Lane changes in the intersection



Our Spatial Region of Interest

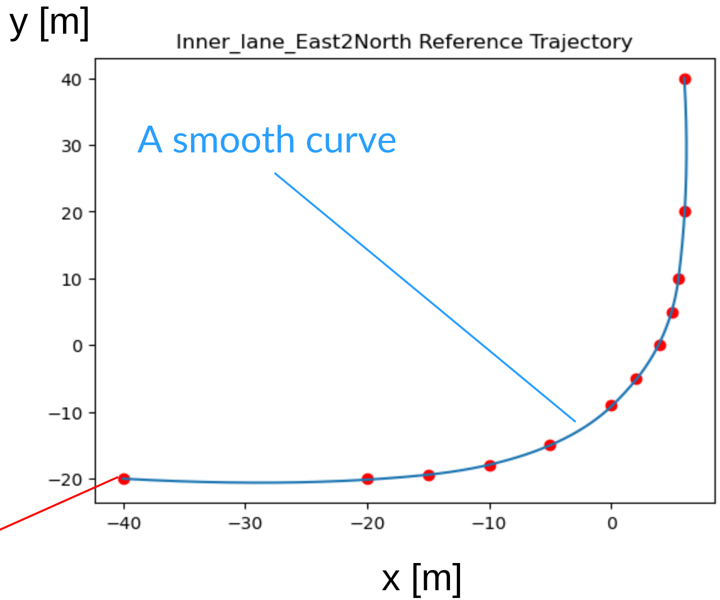
We decided to only estimate trajectories within the box.

This box represents the intersection, where each edge is a crosswalk in the intersection.



Preprocessing the Reference Trajectory

We experimented with linear interpolation, cubic spline and quadratic spline techniques to come up with a suitable smooth curve representing the reference trajectories.



A reference trajectory (shown by some way-points)
which represents an average path of vehicles

Methods

Defining terms in our traffic context

Latent states X_t

$$\begin{bmatrix} x_1 \\ v_1 \\ x_2 \\ v_2 \end{bmatrix}$$

- x_1, v_1: car 1's position and velocity at time t
- x_2, v_2: car 2's position and velocity at time t

Observations Y_t

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- x_1: car 1's position at time t
- x_2: car 2's position at time t

Observation operator H_t

Example:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

both cars visible

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

car 1 hidden

Gaussian State-Space Model [2]

car-following model

$$
\begin{aligned}
&\text{(transition)} && X_t = F_\alpha(X_{t-1}) + \xi_t, && \xi_t \sim N(0, Q_\beta), && 1 \le t \le T, \\
&\text{(observation)} && Y_t = H_t X_t + \eta_t, && \eta_t \sim N(0, R_t), && 1 \le t \le T, \\
&\text{(initialization)} && X_0 \sim p_0(X_0),
\end{aligned}
$$

X_t : latent states at time *t*

F_α : operator mapping the latent space from time *t-1* to *t* parameterized by α

Q_β : measurement error variance-covariance matrix parameterized by β

Gaussian State-Space Model [2]

$$\begin{aligned}
&\text{(transition)} \quad X_t = F_\alpha(X_{t-1}) + \xi_t, \qquad \xi_t \sim N(0, Q_\beta), \quad 1 \le t \le T, \\
&\text{(observation)} \quad Y_t = H_t X_t + \eta_t, \qquad \eta_t \sim N(0, R_t), \quad 1 \le t \le T, \\
&\text{(initialization)} \quad X_0 \sim p_0(X_0),
\end{aligned}$$

X_t : latent states at time *t*

F_α : operator mapping the latent space from time *t-1* to *t* parameterized by α

Q_β : measurement error variance-covariance matrix parameterized by β

Y_t : observations at time *t*

H_t : operator mapping the latent space to the observation space at time *t* (**known**)

R_t : observation error matrix at time *t* (**known**)

Gaussian State-Space Model [2]

$$
\begin{aligned}
&\text{(transition)} && X_t = F_\alpha(X_{t-1}) + \xi_t, && \xi_t \sim N(0, Q_\beta), && 1 \le t \le T,\\
&\text{(observation)} && Y_t = H_t X_t + \eta_t, && \eta_t \sim N(0, R_t), && 1 \le t \le T,\\
&\text{(initialization)} && X_0 \sim p_0(X_0), && &&
\end{aligned}
$$

p_0 assumed known

X_t : latent states at time *t*

Y_t : observations at time *t*

F_α : operator mapping the latent space from time *t-1* to *t* parameterized by α

H_t : operator mapping the latent space to the observation space at time *t* (**known**)

Q_β : measurement error variance-covariance matrix parameterized by β

R_t : observation error matrix at time *t* (**known**)

Gaussian State-Space Model [2]

$$
\begin{aligned}
&(\text{transition}) && X_t = F_\alpha(X_{t-1}) + \xi_t, && \xi_t \sim N(0, Q_\beta), && 1 \le t \le T,\\
&(\text{observation}) && Y_t = H_t X_t + \eta_t, && \eta_t \sim N(0, R_t), && 1 \le t \le T,\\
&(\text{initialization}) && X_0 \sim p_0(X_0), && &&
\end{aligned}
$$

X_t : latent states at time *t*

Y_t : observations at time *t*

F_α : operator mapping the latent space from time *t-1* to *t* parameterized by α

H_t : operator mapping the latent space to the observation space at time *t* (**known**)

Q_β : measurement error variance-covariance matrix parameterized by β

R_t : observation error matrix at time *t* (**known**)

Goal: Estimate X_t for $1 \le t \le T$

Ensemble Kalman Filter [2]

Two step procedure:

1. Forecast into the future using the car following model

(forecast) $\hat{X}_t^n = F_\alpha(X_{t-1}^n) + \xi_t^n$

ensemble index

car-following model

Ensemble Kalman Filter [2]

Two step procedure:

1. Forecast into the future using the car following model

(forecast) $\hat{X}_t^n = F_\alpha(X_{t-1}^n) + \xi_t^n$

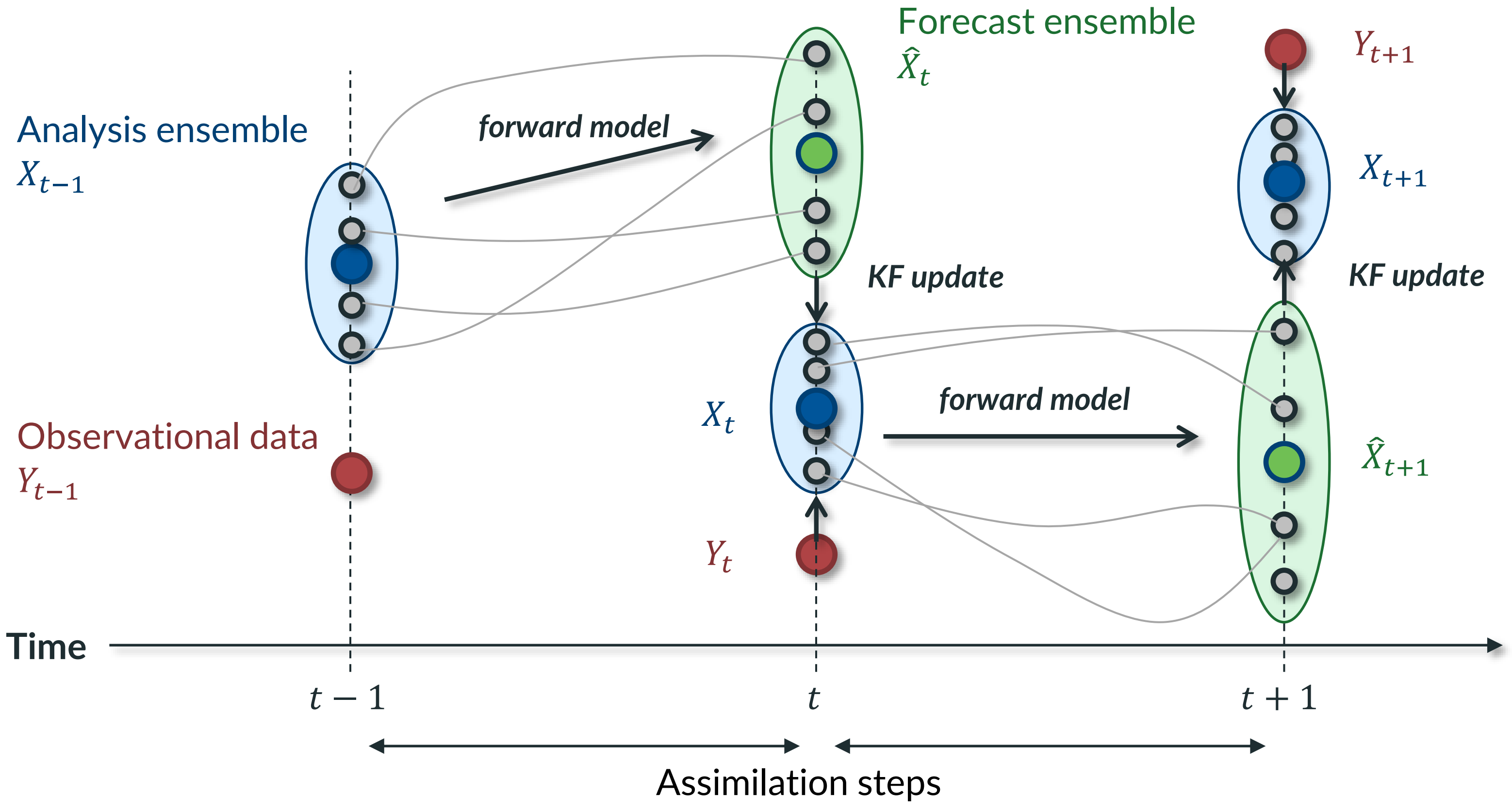
2. Analysis based on observational data up to time *t*

(analysis) $X_t^{n,a} = \hat{X}_t^n + \underbrace{\hat{C}_t H^T (H \hat{C}_t H^T + R)^{-1}}_{K_t} (Y_t + \eta_t^n - H \hat{X}_t^n)$

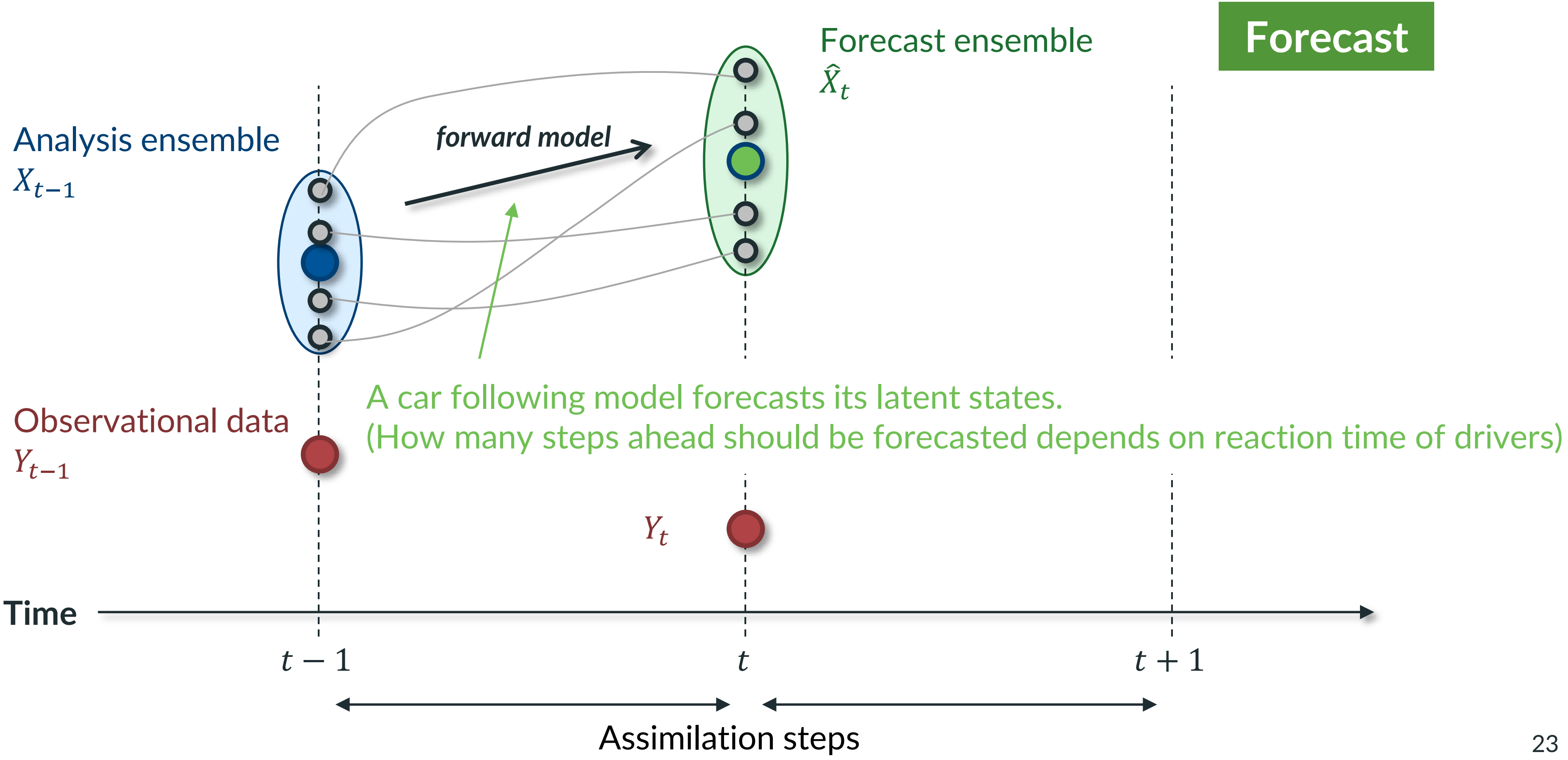
Kalman gain K_t

- specifies the weight given to the observations vs. the predictions

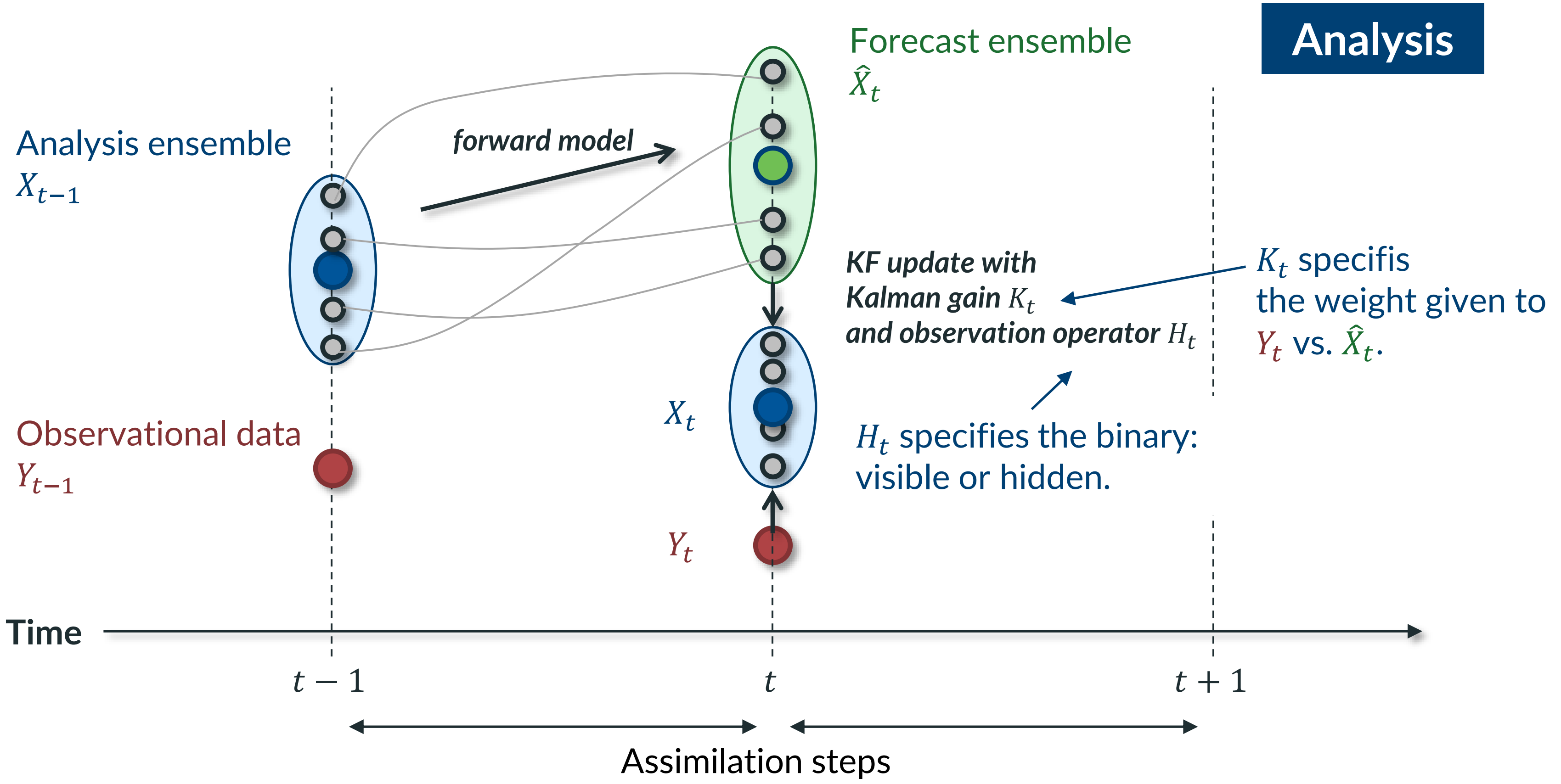
Ensemble Kalman Filter [2]



Ensemble Kalman Filter [2]



Ensemble Kalman Filter [2]



Ensemble Kalman Smoother [3]

Two step procedure:

1. Perform EnKF to obtain estimates $X_t^{n,a}$ for $1 \leq t \leq T$.

2. Analysis based on observational data up to time *t+L*

(analysis) $$X_t^n = X_t^{n,a} + \sum_{l=1}^{L} K_{t,t+l}(Y_{t+l} + \eta_{t+l}^n - H_{t+l}(\hat{X}_{t+l}^n))$$

EnKF analysis
(result with no look-ahead)

Ensemble Kalman Smoother [3]

Two step procedure:

1. Perform EnKF to obtain estimates $X_t^{n,a}$ for $1 \leq t \leq T$.

2. Analysis based on observational data up to time *t+L*

# of look-ahead time steps

(analysis) $X_t^n = X_t^{n,a} + \sum_{l=1}^{L} K_{t,t+l}(Y_{t+l} + \eta_{t+l}^n - H_{t+l}(\hat{X}_{t+l}^n))$

Ensemble Kalman Smoother [3]

Two step procedure:

1. Perform EnKF to obtain estimates $X_t^{n,a}$ for $1 \leq t \leq T$.

2. Analysis based on observational data up to time *t+L*

(analysis) $X_t^n = X_t^{n,a} + \sum_{l=1}^{L} K_{t,t+l}(Y_{t+l} + \eta_{t+l}^n - H_{t+l}(\hat{X}_{t+l}^n))$

- Kalman gain between times t and t+l
- Scales the residuals based on:
  - covariance between times t and t+l

Ensemble Kalman Smoother [3]

Two step procedure:

1. Perform EnKF to obtain estimates $X_t^{n,a}$ for $1 \leq t \leq T$.

2. Analysis based on observational data up to time *t+L*

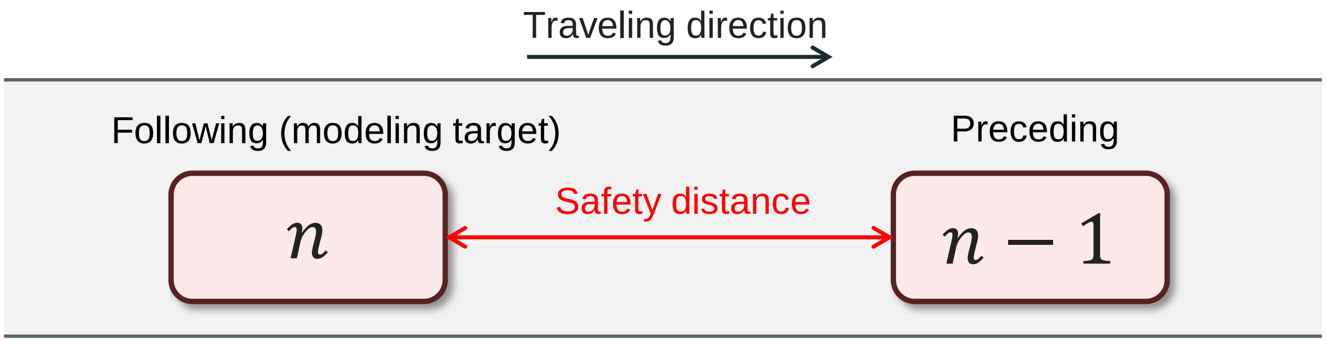
(analysis) $$X_t^n = X_t^{n,a} + \sum_{l=1}^{L} K_{t,t+l}(Y_{t+l} + \eta_{t+l}^n - H_{t+l}(\hat{X}_{t+l}^n))$$

Residual of observation vs. prediction

Car Following Model [4]

Gipps model is a car following model featured by a safety distance between vehicles.

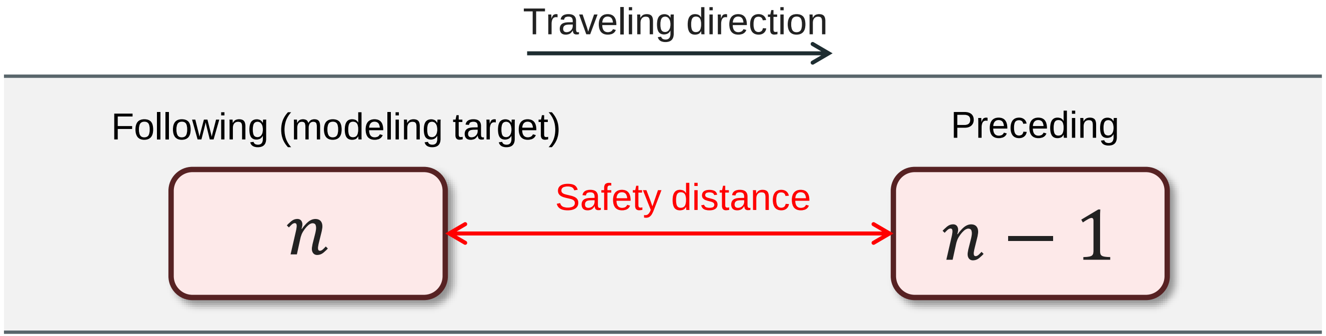
- The purpose of this model is to have a following vehicle maintain the distance.
- The following vehicle can safely stop even if the preceding vehicle brakes suddenly.



Car Following Model [4]

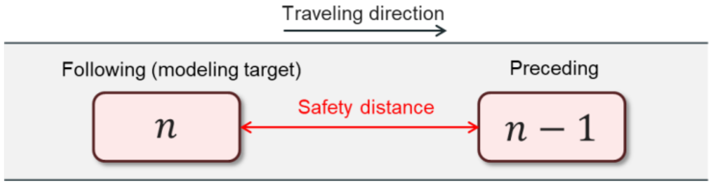
Gipps model is a car following model featured by a safety distance between vehicles.

- This model is relatively simple compared to other models.
- However, it seems to have sufficient expressive power for modeling of the driver's behavior in typical intersections.



Car Following Model [4]

Velocity of the following vehicle n



$$v_n(t+T) = \min\{v_n^a(t+T), v_n^b(t+T)\} \qquad (4.1)$$

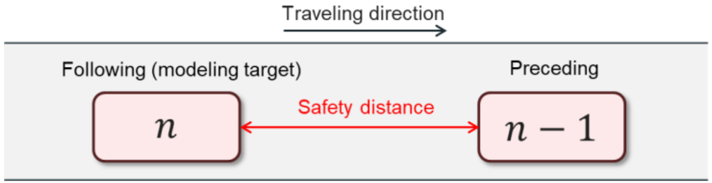
$$\begin{cases} v_n^a(t+T) = v_n(t) + 2.5 \cdot a_n^{max} \cdot T \cdot \left(1 - \dfrac{v_n(t)}{v_n^{desired}} \cdot \sqrt{0.025 + \dfrac{v_n(t)}{v_n^{desired}}}\right) \\ v_n^b(t+T) = d_n^{max} \cdot T + \sqrt{(d_n^{max} \cdot T)^2 - d_n^{max} \cdot \left[2\{x_{n-1}(t) - s_{n-1} - x_n(t)\} - v_n(t) \cdot T - \dfrac{v_{n-1}(t)^2}{\hat{d}_{n-1}}\right]} \end{cases}$$

- T is the reaction time.
- n is the follower index.
- $n-1$ is the leader index.
- $v_n(t), v_{n-1}(t)$ are the vehicle speeds of the follower and leader respectively at the time t.
- $v_n^{desired}$ is the follower's maximum desired speed.
- a_n^{max} is the follower's maximum acceleration.
- d_n^{max} is the follower's maximum deceleration
- $\hat{d}_{n-1}$ is estimation of maximum deceleration desired by *n-1*
- $x_n(t), x_{n-1}(t)$ are the location of the follower and leader respectively at the time t.
- S_{n-1} is the inter-vehicle spacing at a stop. This includes the length of the leader's vehicle added to the follower's desired inter-vehicle spacing at the stop.

Car Following Model [4]

Velocity of the following vehicle n

$$v_n(t+T) = \min\{v_n^a(t+T), v_n^b(t+T)\} \qquad (4.1)$$



$$\begin{cases} v_n^a(t+T) = v_n(t) + 2.5 \cdot a_n^{max} \cdot T \cdot \left(1 - \frac{v_n(t)}{v_n^{desired}} \cdot \sqrt{0.025 + \frac{v_n(t)}{v_n^{desired}}}\right) \\ v_n^b(t+T) = d_n^{max} \cdot T + \sqrt{(d_n^{max} \cdot T)^2 - d_n^{max} \cdot \left[2\{x_{n-1}(t) - s_{n-1} - x_n(t)\} - v_n(t) \cdot T - \frac{v_{n-1}(t)^2}{\hat{d}_{n-1}}\right]} \end{cases}$$

Issue: the above model can describe the behavior of a following vehicle only in the one-dimensional space such as a straight lane.

Curvilinear Coordinate Conversion [5]

We decided to use curvilinear coordinate conversion in order to change **a 2-D problem to a 1-D problem**.

In Cartesian coordinates, the position is represented by straight vertical and horizontal axes.
In Curvilinear coordinates, the position is represented by a straight axis and a curved axis:
lateral (n) and longitudinal (s) along the lane.

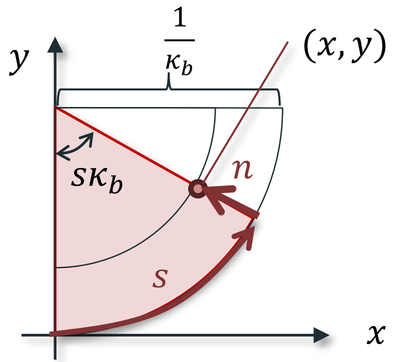
Cartesian coordinates

$$\begin{cases} x = \left(\frac{1}{\kappa_b} - n\right) \cdot \sin(s\kappa_b) \\ y = \frac{1}{\kappa_b} - \left(\frac{1}{\kappa_b} - n\right) \cdot \cos(s\kappa_b) \end{cases}$$

κ_b : curvature at $(x,\ y)$

Curvilinear coordinates

$$\begin{cases} s = \frac{1}{\kappa_b} \arctan\left(\frac{\kappa_b x}{1 - \kappa_b y}\right) \\ n = \frac{1}{\kappa_b} - \frac{x}{\sin\left\{\arctan\left(\frac{\kappa_b x}{1-\kappa_b y}\right)\right\}} \end{cases}$$



Curvilinear Coordinate Conversion [5]

We decided to use curvilinear coordinate conversion in order to change **a 2-D problem to a 1-D problem**.

In Cartesian coordinates, the position is represented by straight vertical and horizontal axes.
In Curvilinear coordinates, the position is represented by a straight axis and a curved axis:
lateral (n) and longitudinal (s) along the lane.

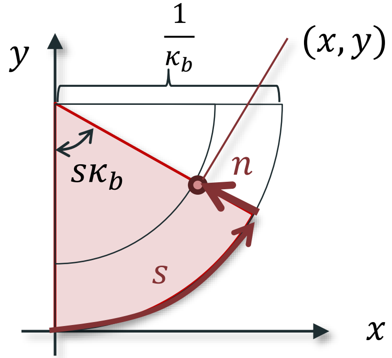
Cartesian coordinates

$$\begin{cases} x = \left(\frac{1}{\kappa_b} - n\right) \cdot \sin(s\kappa_b) \\ y = \frac{1}{\kappa_b} - \left(\frac{1}{\kappa_b} - n\right) \cdot \cos(s\kappa_b) \end{cases}$$

κ_b : curvature at $(x,\ y)$

Curvilinear coordinates

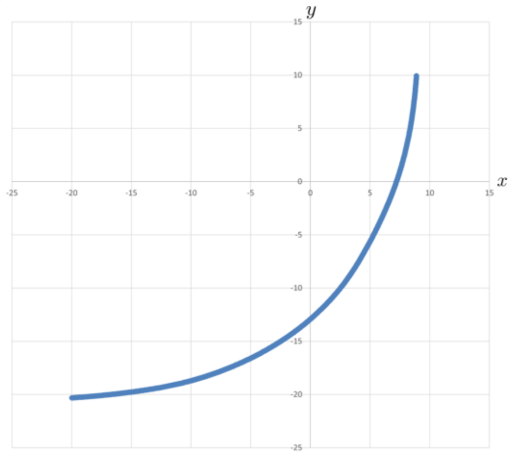
$$s = \frac{1}{\kappa_b} \arctan\left(\frac{\kappa_b x}{1 - \kappa_b y}\right)$$



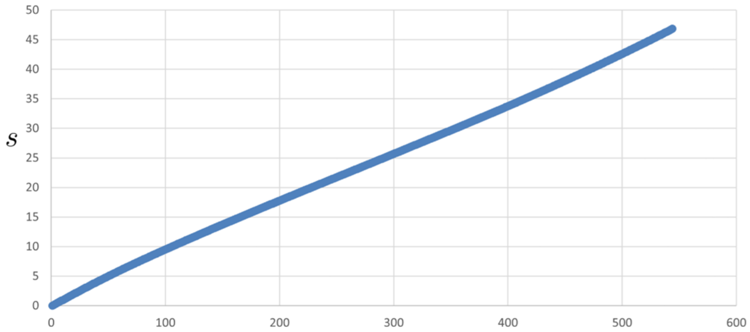
Assume that the lateral motion is sufficiently small so that the n-coordinates can be ignored.

Computationally Tractable Method

Then, we attempted the conversion using the curvature κ_b , but faced computational issues such as difficulty in applying to straight lines and complicated curves.



(x, y) reference trajectory

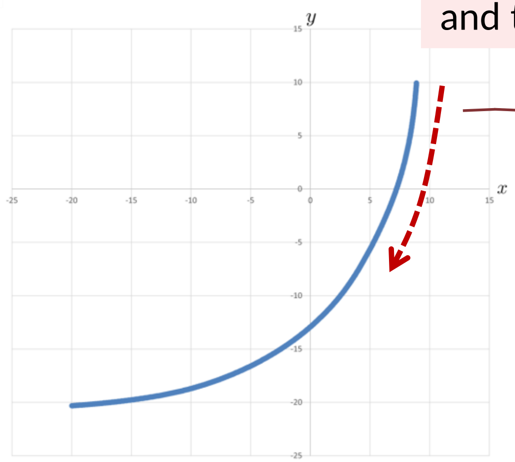


converted trajectory
only using s coordinate

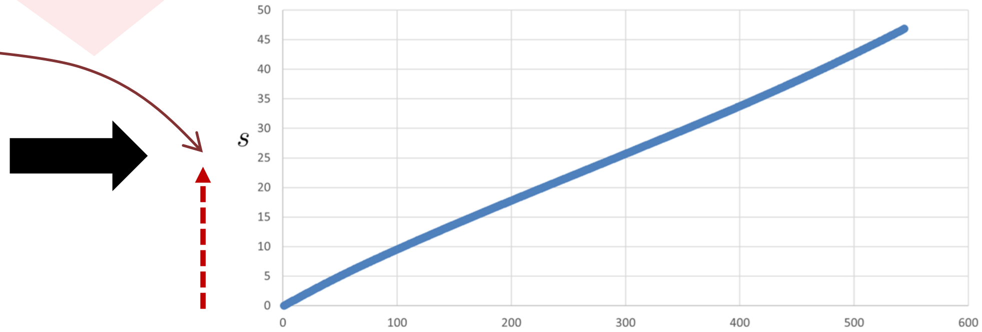
Computationally Tractable Method

By considering that “s” stands for the distance along reference trajectories, we had conclusion that the line segment distance method is suitable.

We add these up to the closest reference trajectory point to an x and y point, and that sum is the s coordinate.



(x, y) reference trajectory



converted trajectory
only using s coordinate

Results

Results

- *We are describing the following results :*
    - Assigning Labels to Unknown Trajectory Directions.
    - Denoising.
    - Extrapolating Trajectories.

Results

Objective : Assigning Labels to Unknown Trajectory Directions

- ***We*** **:**
  1. Pulled out vehicles' data having unknown trajectory directions from the given dataset.

Results

Objective : Assigning Labels to Unknown Trajectory Directions

- ***We*** **:**
  2. Quantified the similarities between the observational trajectory and each reference trajectory by calculating mean squared error (MSE).

$$MSE_j = \frac{1}{N}\sum_{i}^{N}(y_i - f_j(x_i))^2$$

- N is the number of trajectory points.
- x_i, y_i are observation trajectory coordinates of "i".
- $f_j(x)$ is the reference trajectory function of "j".

Results

Objective : Assigning Labels to Unknown Trajectory Directions

- ***We*** **:**
  3. Assigned directions of the reference trajectory having minimum MSE (MSEmin) to the observational trajectory.

$$MSE_{\min} = \min\{MSE_j\} \ \ (j = 1, 2, \ldots, M)$$

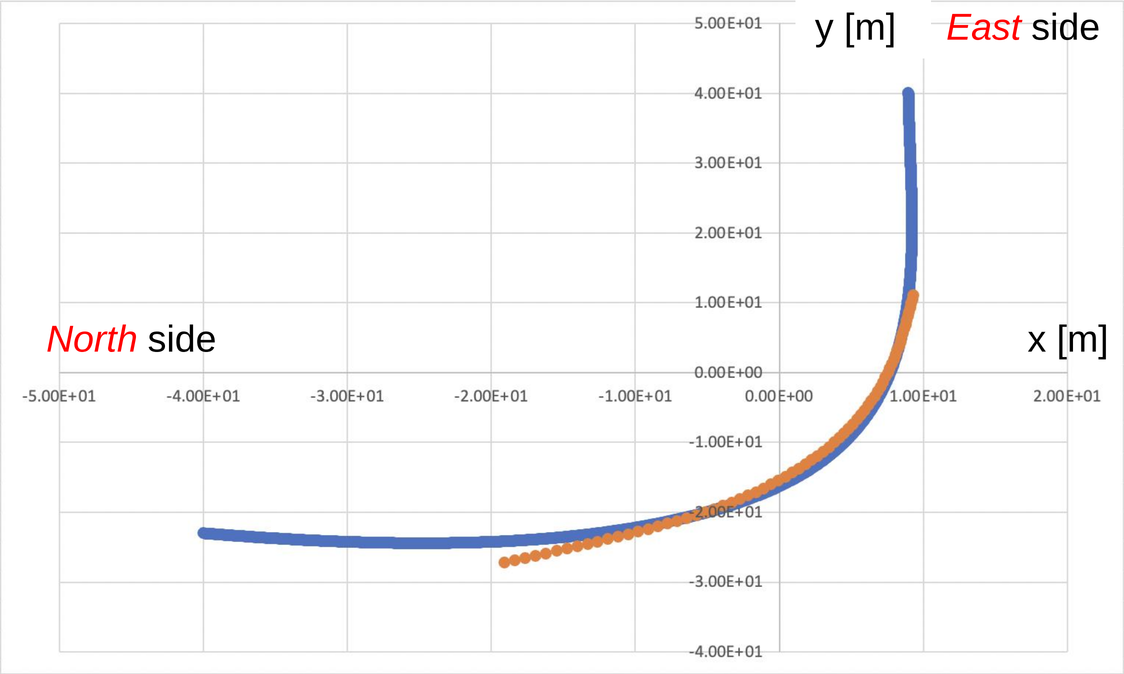
- $MSE_{\min}$ is the minimum MSE.
- "M" is the number of reference trajectories.

Results

Objective : Assigning Labels to Unknown Trajectory Directions

This trajectory was marked "East" to "Unknown."
(orange makers)

We matched it to the "from East to North Outer Lane" reference trajectory.
(blue markers)



Results

Objective : Denoising

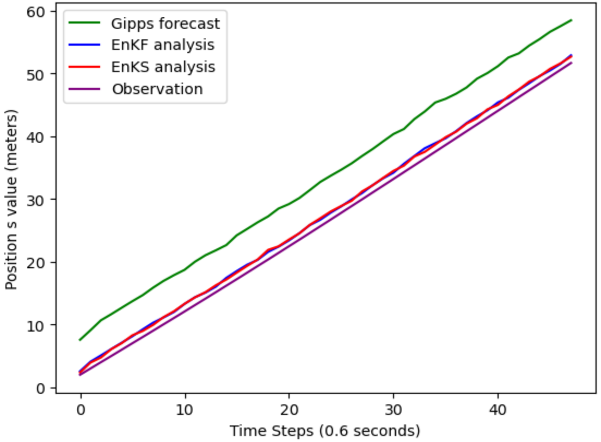
Parameters (not optimized yet)

- Forward model: Gipps
- EnKS lookahead: 7
- Gipps noise level: $Q = qI = 0.5I$
- Obs. noise level: $R = rI = 0.05I$

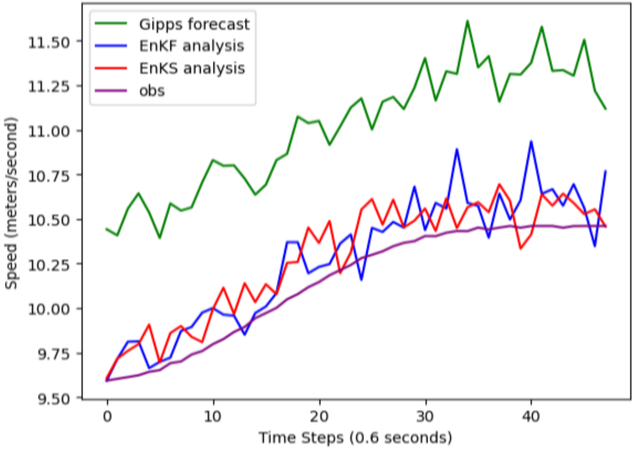
Results for tracking **one vehicle** through the intersection

Gipps forecast and EnKF/EnKS analysis can be assimilated into the trend of velocity.

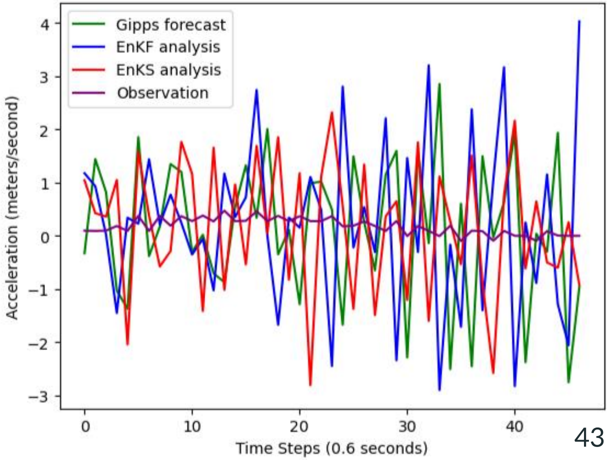
Position



Velocity



Acceleration



Results

Objective : Denoising

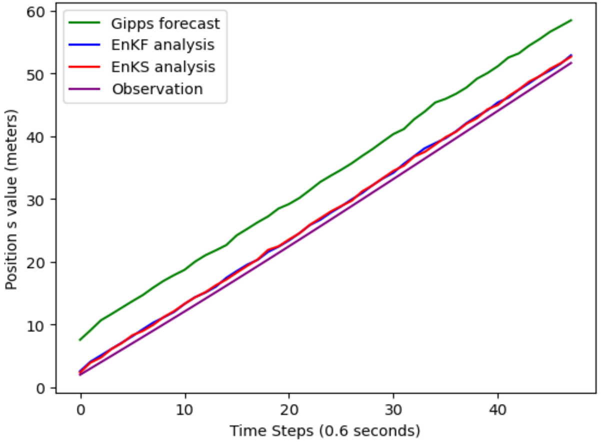
Parameters (not optimized yet)

- Forward model: Gipps
- EnKS lookahead: 7
- Gipps noise level: $Q = qI = 0.5I$
- Obs. noise level: $R = rI = 0.05I$

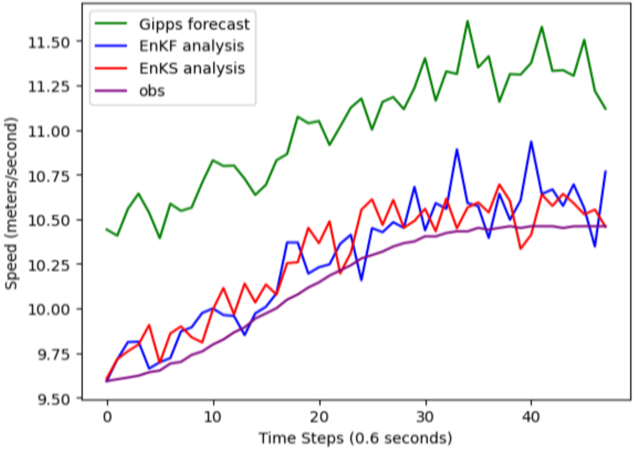
Results for tracking **one vehicle** through the intersection

The forward model have no information on 2nd or higher derivative of "s."
Thus, the acceleration cannot be estimated, and the noise is amplified compared to the complete observation.

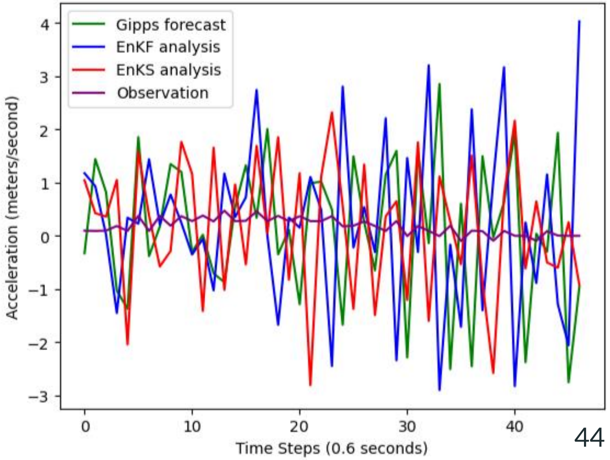
Position



Velocity



Acceleration



Results

Objective : Denoising

Parameters (not optimized yet)

- Forward model: Gipps
- EnKS lookahead: 7
- Gipps noise level: $Q = qI = 0.5I$
- Obs. noise level: $R = rI = 0.05I$

Results for tracking **one vehicle** through the intersection

The forward model have no information on 2nd or higher derivative of "s."
Thus, the acceleration cannot be estimated, and the noise is amplified compared to the complete observation.

↓

Important takeaways:

If both vehicles are visible, we should pay attention to the forward model as follows:

- Suppress its forecast (*in practice, specifying* r *to be very small and* q *to be very large*)
- Adopt a forward model whose errors are on a smaller scale compared to the observations (*whose errors, however, are very small as far as vehicles are visible*) .

Results

Objective : Extrapolating Trajectory

Algorithm 3 Extrapolating Trajectories

1: Initialize $X_0^{1:N}$
2: **for** $t = 1, \ldots, T$ **do**
3: Apply the Gipps forward model from equation 4.1 to $X_{t-1}^{1:N}$.
4: Check if the latent dimension is correct (remove vehicles no longer in our region of interest, unless their observed position is still in the region, and add vehicles that enter the region)
5: Construct a new H_t based on which latent states are observed versus unobserved (including hidden vehicles)
6: Construct R_t and Q_t based on d_{x_t} and d_{y_t}.
Apply the EnKS correction from Alg. 2. (See the next side)
7: **end for**
8: **Output:** $X_{1:T}^{1:N}$**, which includes estimated speeds, positions, and accelerations for both hidden and observed vehicles.**

Results

Objective : Extrapolating Trajectory

Algorithm 1 Ensemble Kalman Filter (EnKF)

Input: $Y_{1:T}, X_0^{1:N}$. (If $X_0^{1:N}$ is not specified, draw $X_0^n \sim^{\text{i.i.d.}} p_0(X_0)$)
for $t = 1, \ldots, T$ **do**
  Set $\hat{X}_t^n = F(X_{t-1}^n) + \xi_t^n$, where $\xi_t^n \sim^{\text{i.i.d.}} N(0, Q_t)$ ▷ Forecast
  Compute $\hat{m}_t$ and $\hat{C}_t$ using equations 4.7 and 4.8, and set $\hat{K}_t = \hat{C}_t H_t^T (H_t \hat{C}_t H_t^T + R_t)^{-1}$.
  Set $X_t^n = \hat{X}_t^n + \hat{K}_t (Y_t + \gamma_t^n - H_t \hat{X}_t^n)$, where $\gamma_t^n \sim^{\text{i.i.d.}} N(0, R_t)$ ▷ Analysis
end for
Output: $X_{1:T}^{1:N}$

Algorithm 2 Ensemble Kalman Smoother (EnKS) [3]

1: **Input:** $Y_{1:T}, X_0^{1:N}$. (If $X_0^{1:N}$ is not specified, draw $X_0^n \sim^{\text{i.i.d.}} p_0(X_0)$)
2: Compute $\hat{X}_{1:T}^{1:N}$ using Algorithm 1.
3: **for** $t = 1, \ldots, T$ **do**
4:   Compute $\hat{m}_t$ and $\hat{C}_t$ using equations 4.7 and 4.8
5:   Compute $\hat{K}_{t,t+l} = \hat{C}_{t,t+l} H_{t+l}^T (H_{t+l} \hat{C}_{t,t+l} H_{t+l}^T + R_{t+l})^{-1}$.
6:   Set $X_t^n = \hat{X}_t^n + \sum_{l=1}^{L} \hat{K}_{t,t+l} (Y_{t+l} + \gamma_{t+l}^n - H_{t+l} \hat{X}_{t+l}^n)$, where $\gamma_{t+l}^n \sim^{\text{i.i.d.}} N(0, R_{t+l})$
7: **end for**
8: **Output:** $X_{1:T}^{1:N}$

$$\hat{m}_t = \frac{1}{N} \sum_{n=1}^{N} \hat{X}_t^n \quad (4.7)$$

$$\hat{C}_t = \frac{1}{N-1} \sum_{n=1} (\hat{X}_t^n - \hat{m}_t)(\hat{X}_t^n - \hat{m}_t)^T \quad (4.8)$$

Conclusion

Conclusion

- ***We worked on :***
  - **Filtering out observation data noise.**
  - **Extrapolating trajectories from given partial data.**

In order to

- Reduce the number of traffic sensors needed at an intersection.
- Develop an advanced technology for traffic flow analysis using continuous data of actual driver's behavior

Conclusion

- ***We used :***
  - Ensemble Kalman smoother (EnKS).
  - Gipps model.

↓ ***When***

- ***We :***
  - Filtered out observation data noise.
  - Completed partial vehicle trajectories.

Conclusion

- ***We*** **:**
  - Implemented a tractable method for assigning probable labeles to trajectories with "unknown" directions.
  - Validated estimation performance of a proposed smoothing method based on Gipps model and EnKS.
  - Devised an algorism for extrapolating incomplete trajectory by integrating all proposed methods in this project.

Future Directions

Future Directions

- Improvements to the forward model
  - Fine-tune the Gipps model parameters based on our dataset
  - Consider more expressive models, such as a 2D prediction framework and a model with higher order derivative (acceleration and jerk)

- Improvements to the data assimilation framework
  - Account for spatial correlations between vehicles to increase effective ensemble size
  - Assimilate temporally intermediate observations between the forward model passes

Acknowledgements

We appreciate everyone involved in the program, such as

- g-RIPS staff
- our academic mentor, Natsuo-san
- our industrial mentor, Ono-san
- Institute for Pure and Applied Mathematics (IPAM) at UCLA
- Tohoku University Advanced Institute for Materials Research (AIMR)

Thank you for listening!

References

[1] Realizing a sustainable mobility society with traffic flow data, IHI Engineering Review, 56(1), 2023.
Available online: https://www.ihi.co.jp/en/technology/techinfo/contents_no/1199408_13586.html

[2] Geir Evensen.
The Ensemble Kalman Filter: Theoretical formulation and practical implementation,
Ocean Dynamics, 53(4), 343–367, 2003.

[3] Geir Evensen, Peter Jan, and Van Leeuwen.
An ensemble kalman smoother for nonlinear dynamics,
Monthly Weather Review, 128, 1852–1864, 2000.

[4] Johan Janson Olstam and Andreas Tapani.
Comparison of Car-following models,
Swedish National Road and Transport Research Institute: Linkoping, Sweden, 2004.

[5] Junsoo Kim, K. Jo, Wontaek Lim, Minchul Lee, and M. Sunwoo.
Curvilinear-Coordinate-Based Object and Situation Assessment for Highly Automated Vehicles,
IEEE Transactions on Intelligent Transportation Systems, 16, 1559–1575, 2015.