New environmental dependent modelling with Gaussian particle filtering based implementation for ground vehicle tracking

Miao Yu*, Yali Xue*, Runxiao Ding*, Hyondong Oh*, Wen-Hua Chen* and Jonathon Chambers+

*Department of Aeronautical and Automotive Engineering
Loughborough University

+School of Electrical and Electronic Engineering
Newcastle University

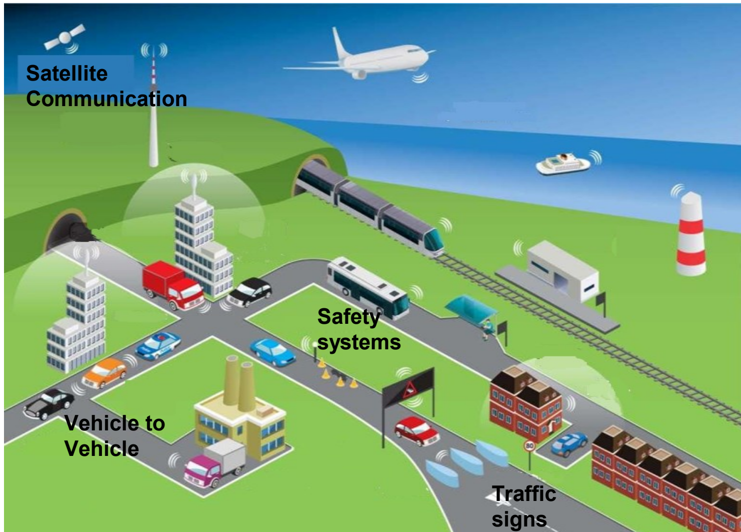
Sensor Signal Processing for Defence 2016, Edinburgh

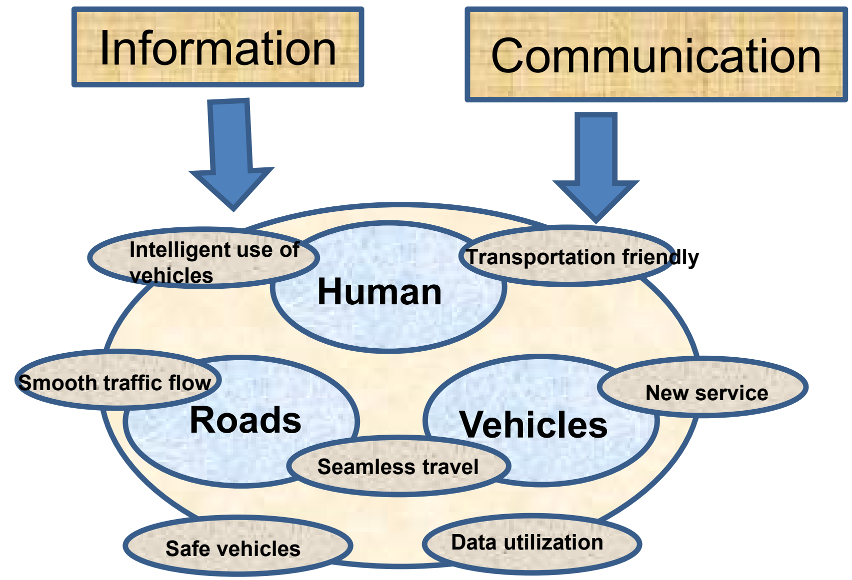
Outline of the presentation

- Overview of the intelligent transportation system
- Traditional methods for the ground vehicle state estimation
- The developed method for tracking the ground vehicle:
  - Model
  - Algorithm
- Simulation studies
- Conclusions and Future plans

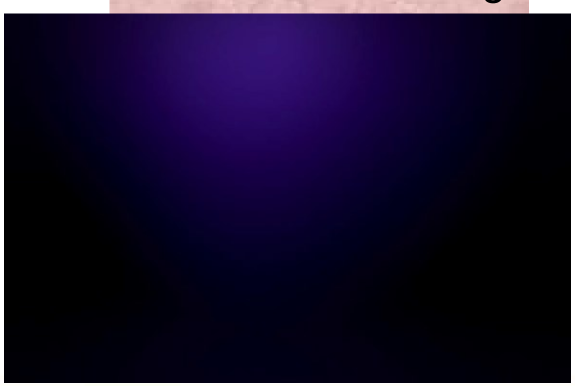
Overview of the intelligent transport system

Intelligent transportation system:

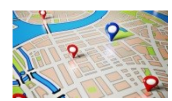




Autonomous driving



Sensor



Database

Environmental perception

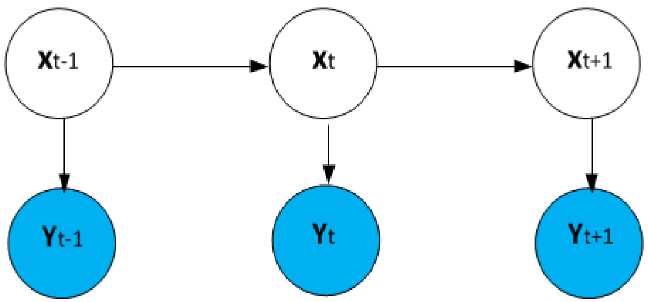
Traffic participants state estimation

Traffic situation prediction

Taking Manoeuvring

Reviews of the traffic participants state estimation methods

Filtering based algorithm:



- Kalman filter and its variances, extended, unscented ones
- Particle filter and its variances, auxiliary, unscented ones
- Multiple model based filtering, IMM,IMMPF
- Filtering with data association, MHT, PHD

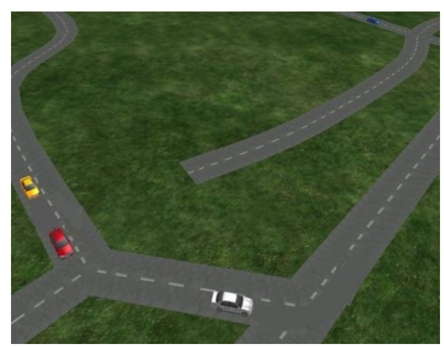
Domain knowledge aided:

Physical constraints (road network)

M. Ulmke and W. Koch, “Road-map assisted ground moving target tracking,” *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 42, no. 4, pp. 1264–1274, 2006.

H. Oh, H. Shin, S. Kim, A. Tsourdos, and B. White, “Airborne behaviour monitoring using Gaussian processes with map information,” *IET Radar, Sonar and Navigation*, vol. 7, no. 4, pp. 393–400, 2013.

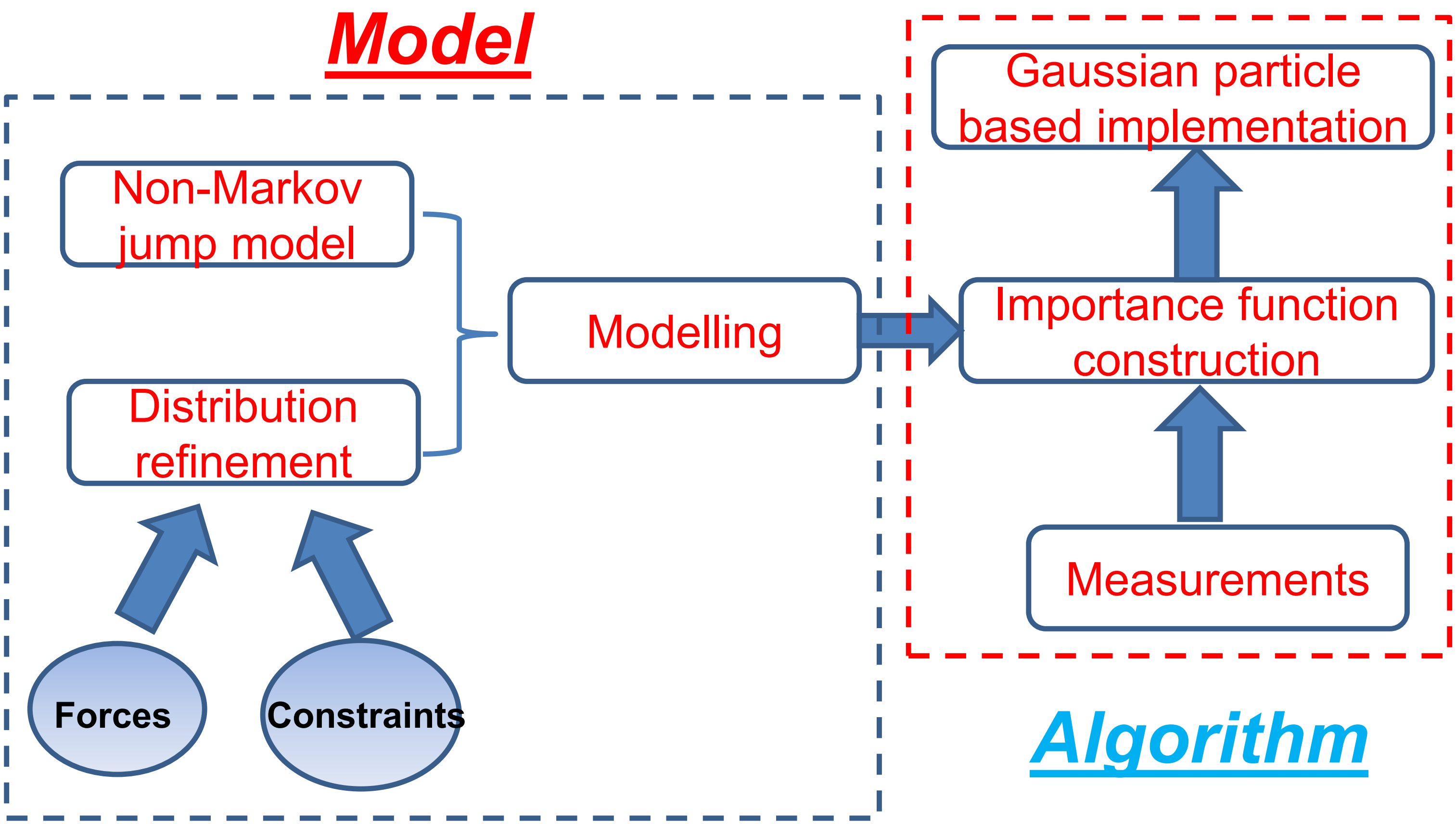
Context information



T. Gindele, S. Brechtel, and R. Dillmann, “A probabilistic model for estimation driver behaviors and vehicle trajectory in traffic environment,” in *13th International IEEE Annual Conference on Intelligent Transportation Systems, Madeira Island, Portugal*, 2010.

Liu, C. and Chen, B. L. W., “Road network based vehicle navigation using an improved IMM particle filter,” *2013 IFAC Intelligent Autonomous Vehicles Symposium, Gold Coast, Australia* (2013).

Outline of the proposed algorithm



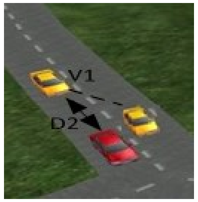
Non-Markov jump modelling for incorporating context information

Multiple models representing vehicle manoeuvres are involved

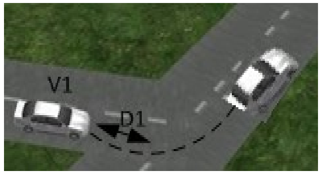
Transition between different models is context-dependent

$$p(m_t|m_{t-1}, context)$$

Context: relationship between the vehicle state and the environment

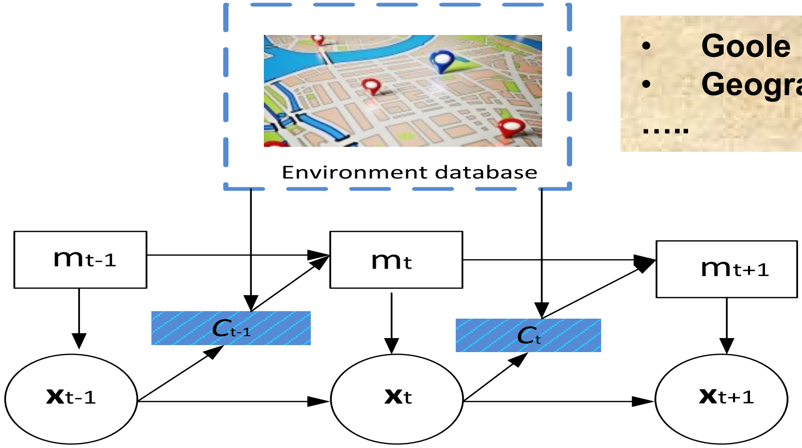


Scenario1: A vehicle moves close to another vehicle, overtaking manoeuvring occurs



Scenario2: A vehicle moves close to a junction, turning manoeuvring occurs

General Bayesian network representation



- **Goole map**
- **Geographical information system**

…..

$$\boldsymbol{x}_t: state\ m_t: movement\ mode\ C_t: context$$

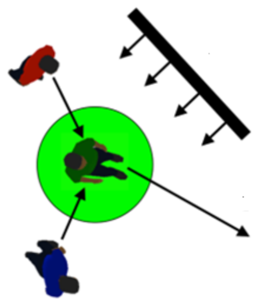
State model refinement

A general state model in the non-Markov jump model system: $\boldsymbol{x}_{t+1} = F_{m_t}(\boldsymbol{x}_t) + G\boldsymbol{w}_t \rightarrow p(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, m_t, \boldsymbol{Y}_{t-1})$

X. Li and P. Jilkov, "Survey of maneuvering target tracking. Part I: Dynamic models", *IEEE Trans. on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.

Two-step refinement:

I. Social force:

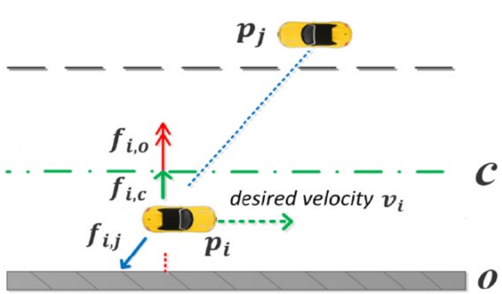


Originally developed for modelling the human movements

Helbing, D., Farkas, I., Viscek, T.: 'Simulating dynamic features of escape panic', Nature, 2000, 407, pp. 487–490.

Extended for modelling the traffic participants in a shared space (including vehicles):

B. Anvaria, M. Bella, A. Sivakumara, W. Ochienga, " Modelling Shared Space Users Via Rule-based Social Force Model ", Transportation Research Part C: Emerging Technologies, Volume 51, 2015, Pages 83–103



Forces:

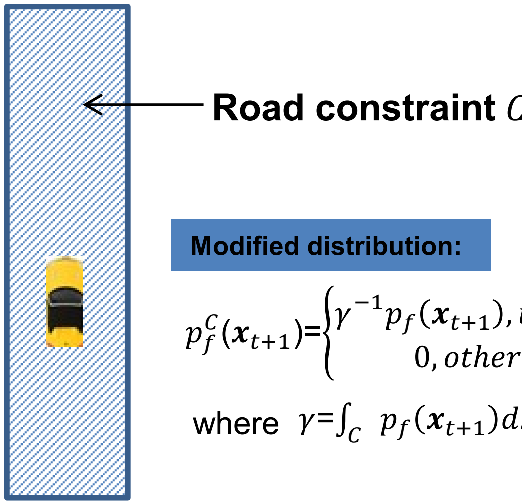
$$\boldsymbol{f}_{i,o}^{repulsive} = A \cdot \exp(\frac{-d_{i,o}}{B})\boldsymbol{n}_{io}$$

$$\boldsymbol{f}_{i,c}^{attractive} = A' \cdot (1 - \exp(\frac{-d_{i,c}}{B'}))\boldsymbol{n}_{ic}$$

Refined model:

$$\boldsymbol{x}_{t+1} = F_{m_t}(\boldsymbol{x}_t) + I(\boldsymbol{f}_t) + G\boldsymbol{w}_t \rightarrow \boldsymbol{p}_f(\boldsymbol{x}_{t+1}|\boldsymbol{x}_t, m_t, \boldsymbol{Y}_{t-1})$$

II. Constraints:

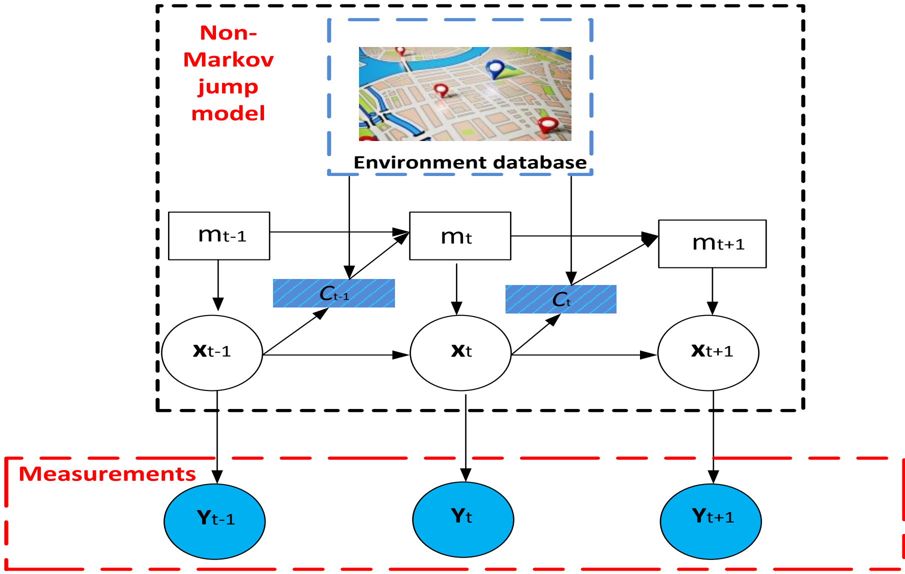


Modified distribution:

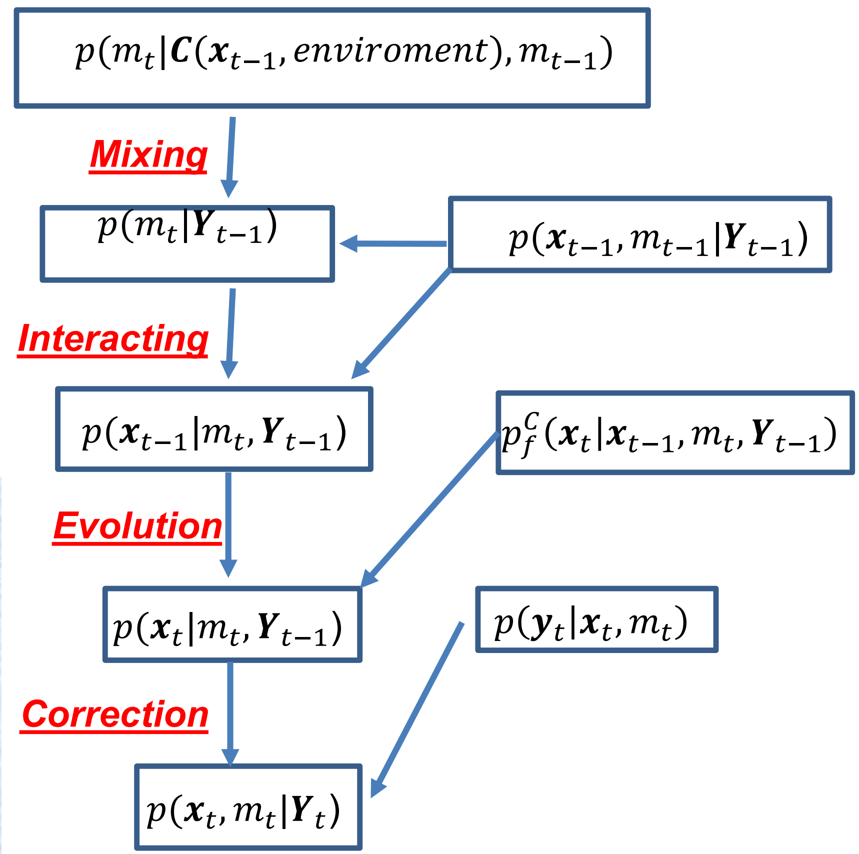
$$p_f^C(\boldsymbol{x}_{t+1}) = \begin{cases} \gamma^{-1} p_f(\boldsymbol{x}_{t+1}), if\, \boldsymbol{x}_{t+1} \in C \\ 0, otherwise \end{cases}$$

where $\gamma = \int_C p_f(\boldsymbol{x}_{t+1}) d\boldsymbol{x}_{t+1}$

Algorithms development—Bayesian inference



Probability inference flowchart:



Detailed inferences:

- $p(m_t|\boldsymbol{Y}_{t-1}) = \int \sum_{m_{t-1}\in M} p(m_t, \boldsymbol{x}_{t-1}, m_{t-1}|\boldsymbol{Y}_{t-1}) d\,\boldsymbol{x}_{t-1} = \int \sum_{m_{t-1}\in M} p(m_t|\boldsymbol{C}(\boldsymbol{x}_{t-1}, enviroment), m_{t-1}) p(m_{t-1}\boldsymbol{x}_{t-1}|Y_{t-1}) d\,\boldsymbol{x}_{t-1}$

$\boldsymbol{C}(\boldsymbol{x}_{t-1}, enviroment)$ →context information extraction function

- $p(\boldsymbol{x}_{t-1}|m_t, \boldsymbol{Y}_{t-1}) = \frac{\sum_{m_{t-1}\in M} p(m_t|x_{t-1}, m_{t-1}) p(x_{t-1}, m_{t-1}|\boldsymbol{Y}_{t-1})}{p(m_t|\boldsymbol{Y}_{t-1})}$

- $p(\boldsymbol{x}_t|m_t, \boldsymbol{Y}_{t-1}) = \int p\,(\boldsymbol{x}_{t-1}|m_t, \boldsymbol{Y}_{t-1}) p_f^C(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, m_t, \boldsymbol{Y}_{t-1}) d\boldsymbol{x}_{t-1}$

- $p(\boldsymbol{x}_t, m_t|\boldsymbol{Y}_t) \propto p(m_t|\boldsymbol{Y}_{t-1}) p(\boldsymbol{y}_t|\boldsymbol{x}_t, m_t) p(\boldsymbol{x}_t|m_t, \boldsymbol{Y}_{t-1})$

Algorithms development—Generic particle filtering based implementation

Initially, we have a set of particles $\{\mathbf{x}_{t-1}^{r,i}, u_{t}^{r,i}\}_{i-1,\dots,N}$ to approximate $p(m_{t-1} = r, \boldsymbol{x}_{t-1}|Y_{t-1})$ for every mode r:

- ***Mode mixing:***

$$p(m_t = m|\boldsymbol{Y}_{t-1}) \approx \sum_{r\in M}\sum_{i=1}^{N} p\left(m_t = m\middle|\boldsymbol{C}(\boldsymbol{x}_{t-1}^{r,i}, enviroment), m_{t-1} = r\right)u_t^{r,i} \equiv \Lambda_{t-1}^{u}$$

- ***Interacting:***

$$p(\boldsymbol{x}_{t-1}|m_t = m, \boldsymbol{Y}_{t-1}) \approx \frac{\sum_{r\in M}\sum_{i=1}^{N} p\left(m_t = m\middle|m_{t-1} = r, \boldsymbol{x}_{t-1}^{r,i}\right)u_{t-1}^{r,i}\delta(x_{t-1} - x_{t-1}^{r,i})}{\Lambda_{t-1}^{u}}$$

N particles $\{\overline{\boldsymbol{x}}_{t-1}^{m,i}, \bar{u}_{t-1}^{m,i}\}_{\boldsymbol{i=1,\dots,N}}$ are resampled to solve the exponential increasing of particles number

- ***Evolution:***

Sampling particles $\{\boldsymbol{x}_t^{m,i}\}_{i=1,\dots,N}$ from $p_f^C(\boldsymbol{x}_t|m_t = m, \overline{\boldsymbol{x}}_{t-1}^{m,i}, \boldsymbol{Y}_{t-1})$

$$p(\boldsymbol{x}_t|m_t = m, \boldsymbol{Y}_{t-1}) \approx \bar{u}_{t-1}^{m,i}\delta(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^{m,i})$$

- ***Correction***:

$$p(\boldsymbol{x}_t, m_t = m|\boldsymbol{Y}_{t-1}) \propto \sum_{i=1}^{N} p(m_t = m|\boldsymbol{Y}_{t-1})\mathrm{p}\left(\boldsymbol{y}_t\middle|\boldsymbol{x}_t^{m,i}, m_t = m\right)\bar{u}_{t-1}^{m,i}\delta(\boldsymbol{x}_t - \boldsymbol{x}_t^{m,i})$$

H. Blom and E. Bloem, “Exact Bayesian and particle filtering of stochastic hybrid systems”, IEEE Transaction on Aerospace and Electronic Systems, Vol.43, No.1, Pages 55-70, 2007.

Algorithms development—Gaussian particle filtering based implementation

Initially, we have a set of particles $\{\mathbf{x}_{t-1}^{r,i}, u_t^{r,i}\}_{i-1,\dots,N}$ to approximate $p(m_{t-1} = r, \boldsymbol{x}_{t-1}|Y_{t-1})$ for every mode r

- ***Mode mixing:***

$$p(m_t = m|\boldsymbol{Y}_{t-1}) \approx \sum_{r\in M}\sum_{i=1}^{N} p\left(m_t = m\middle|\boldsymbol{x}_{t-1}^{r,i}, m_{t-1} = r\right)u_t^{r,i} \equiv \Lambda_{t-1}^{u}$$

- ***Interacting:***

$$p(\boldsymbol{x}_{t-1}|m_t = m, \boldsymbol{Y}_{t-1}) \approx \frac{\sum_{r\in M}\sum_{i=1}^{N} p\left(m_t = m\middle|m_{t-1} = r, \boldsymbol{x}_{t-1}^{r,i}\right)u_{t-1}^{r,i}\delta(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-1}^{r,i})}{\Lambda_{t-1}^{u}}$$

N particles $\{\bar{\boldsymbol{x}}_{t-1}^{m,i}, \bar{u}_{t-1}^{m,i}\}_{\boldsymbol{i=1,\dots,N}}$ are resampled to solve the exponential increasing of particles number

- ***Evolution and correction:***

Initialization of mean and covariance : $\widehat{\boldsymbol{x}}_{t-1|t-1} = \sum_{i=1}^{N} \bar{u}_{t-1}^{m,i} \cdot \bar{\boldsymbol{x}}_{t-1}^{m,i}$ $\quad \hat{P}_{t-1|t-1} = \sum_{i=1}^{N} \bar{u}_{t-1}^{m,i} \cdot (\bar{\boldsymbol{x}}_{t-1}^{m,i} - \widehat{\boldsymbol{x}}_{t-1|t-1}) \cdot (\bar{\boldsymbol{x}}_{t-1}^{m,i} - \widehat{\boldsymbol{x}}_{t-1|t-1})^T$

- A Gaussian distribution $N_c(\boldsymbol{x}_t|\widehat{\boldsymbol{x}}_{t|t}, \hat{P}_{t|t})$ is obtained using ***truncated unscented Kalman filter***
- New particles $\{\boldsymbol{x}_t^{m,i}\}_{i=1,\dots,N}$ are sampled from $N_c(\boldsymbol{x}_t|\widehat{\boldsymbol{x}}_{t|t}, \hat{P}_{t|t})$ for approximating $p(\boldsymbol{x}_t|m_t = m, \boldsymbol{Y}_t)$ as:

$$p(\boldsymbol{x}_t, m_t = m|\boldsymbol{Y}_t) \propto \sum_{i=1}^{N} u_t^{m,i}\delta\left(\boldsymbol{x}_t - \boldsymbol{x}_t^{m,i}\right), \quad with \quad u_t^{m,i} = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t, m_t = m)N(\boldsymbol{x}_t^{m,i}|\widehat{\boldsymbol{x}}_{t|t-1}, \hat{P}_{t|t-1})}{N(\boldsymbol{x}_t^{m,i}|\widehat{\boldsymbol{x}}_{t|t}, P_t)}$$

Truncated unscented Kalman filter

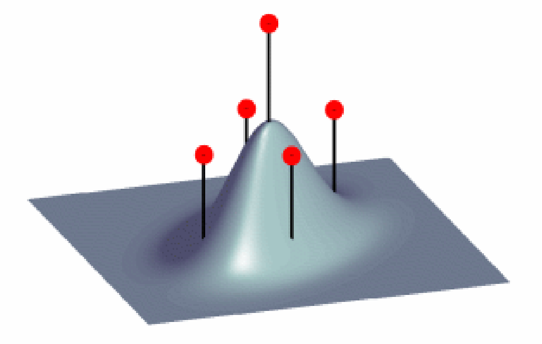
Based on the initial mean $\overline{\boldsymbol{x}}$ and covariance P of $\boldsymbol{x}_{t-1}$ related to $p(\boldsymbol{x}_{t-1}|m_t, \boldsymbol{Y}_{t-1})$

Obtaining σ-points:

$$\boldsymbol{x}_0 = \overline{\boldsymbol{x}} \quad u_0 = \frac{\kappa}{n_\chi+\kappa} \quad \boldsymbol{y}_0 = h(\overline{\boldsymbol{x}}, \boldsymbol{e}_0)$$

$$\boldsymbol{x}_i = \overline{\boldsymbol{x}} + (\sqrt{(n_\chi+\kappa)P})_i \quad u_i = \frac{1}{2(n_\chi+\kappa)} \quad \boldsymbol{y}_i = h(\boldsymbol{x}_i, \boldsymbol{e}_i)$$

$$\boldsymbol{x}_{n_\chi+i} = \overline{\boldsymbol{x}} - (\sqrt{(n_\chi+\kappa)P})_i \quad u_{n_\chi+i} = u_i \quad \boldsymbol{y}_{n_\chi+i} = h(\boldsymbol{x}_{n_\chi+i}, \boldsymbol{e}_{n_\chi+i})$$



Mean and covariance updating:

$$P_{y_t|y_t} = \sum_{j=0}^{2n_\chi} u_j(\boldsymbol{y}_j - \widehat{\boldsymbol{y}})(\boldsymbol{y}_j - \widehat{\boldsymbol{y}})^T \quad \text{where } \widehat{\boldsymbol{y}} = \sum_{j=0}^{2n_\chi} u_j \boldsymbol{y}_j$$

$$P_{x_t|y_t} = \sum_{j=0}^{2n_\chi} u_j(\boldsymbol{x}_j - \overline{\boldsymbol{w}})(\boldsymbol{y}_j - \widehat{\boldsymbol{y}})^T$$

$$K = P_{x_t|y_t} P_{y_t|y_t}^{-1}$$

$$\overline{\boldsymbol{x}}' = \overline{\boldsymbol{x}} + K(\boldsymbol{y}_t - \widehat{\boldsymbol{y}})$$

$$P' = P - K P_{y_t|y_t} K^T$$

Constraint information incorporation:

M samples $\{\boldsymbol{x}_i\}_{i=1,\dots,M}$ are obtained from $N(\mathbf{x}_t \,|\overline{\boldsymbol{x}}', P')$, new mean and covariance are obtained after considering the constraint:

$$\overline{\boldsymbol{x}}'_C = \frac{1}{N}\sum_{j=1}^{M'} \boldsymbol{x}_j$$

$$P'_C = \frac{1}{N}\sum_{j=1}^{M'}(\boldsymbol{x}_j - \overline{\boldsymbol{x}}'_C)(\boldsymbol{x}_j - \overline{\boldsymbol{x}}'_C)^T$$

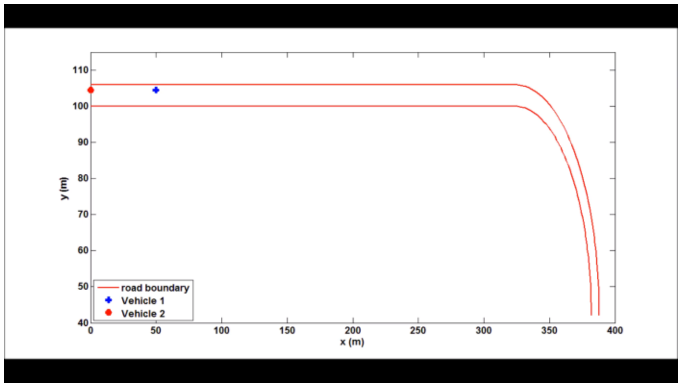
where $\{\boldsymbol{x}_j\}_{i=1,\dots,M} \in C$

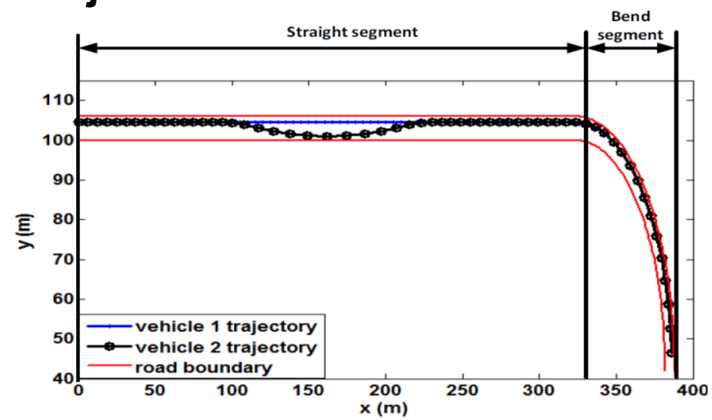
Samples are obtained from $N(\mathbf{x}_t \,|\overline{x}'_C, \boldsymbol{P}'_C)$ incorporating the measurement and constraint information

O. Straka, J. Dunik and M. Simandl, "Truncated nonlinear filters for state estimation with nonlieanr inequality constraints", Automatica, Vol.48, No.2, Pages 273-286, 2012.

Simulation studies--scenario

Simulated vehicles moving scenario and trajectories:





Vehicles manoeuvring behaviours:

The first vehicle:

moving with a constant velocity of 10 (m/s) for 27.5 (s)
turning with an angular velocity of 0.2 (rad/s) for 2.5 (s).

The second vehicle:

moving with a constant velocity of 12.5 (m/s) for 8s
overtaking vehicle 1
moving again with 2.5 (m/s) for 7s along the straight segment.
moving along the bend road segment with an angular velocity of 0.2 (rad/s).

Simulation studies--models

State models:

Constant velocity model:

$$\begin{pmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{pmatrix} = \begin{pmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{k-1} \\ y_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{pmatrix} + \begin{pmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{pmatrix} \begin{pmatrix} n_x \\ n_y \end{pmatrix}$$

$$\begin{pmatrix} n_x \\ n_y \end{pmatrix} \rightarrow N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0.1^2 (m/s^2)^2, 0 \\ 0, 0.1^2 (m/s^2)^2 \end{pmatrix}\right)$$

$$\begin{pmatrix} n_x \\ n_y \end{pmatrix} \rightarrow N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 2^2 (m/s^2)^2, 0 \\ 0, 1^2 (m/s^2)^2 \end{pmatrix}\right)$$

Constant turning (CT) model:

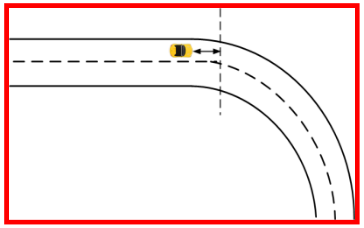
$$\begin{pmatrix} x_k \\ \dot{x}_k \\ y_k \\ \dot{y}_k \\ w_k \end{pmatrix} = \begin{pmatrix} 1 & \frac{\sin wT}{w} & 0 & -\frac{1-\cos wT}{w} & 0 \\ 0 & \cos wT & 0 & -\sin wT & 0 \\ 0 & \frac{1-\cos wT}{w} & 1 & \frac{\sin wT}{w} & 0 \\ 0 & \sin wT & 0 & \cos wT & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \\ w_k \end{pmatrix} + \begin{pmatrix} T^2/2 & 0 & 0 \\ T & 0 & 0 \\ 0 & T^2/2 & 0 \\ 0 & T & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} n_x \\ n_y \\ n_w \end{pmatrix}$$

Each model is refined by both force (from the road boundaries and another vehicle) and constraint (road)

Model transitions:

Model transition probabilities are defined in a context dependent way, e.g.

$p(m_t = HICV | m_t = LICV) = exp(-a \cdot d_{vehicle})$, $d_{vehicle}$: distance to the front vehicle
$p(m_t = CT | m_t = LICV) = exp(-a' \cdot d_{turn})$, d_{turn}:distance to the turn
$a = 0.1,\ a' = 0.15$: empirical set parameters

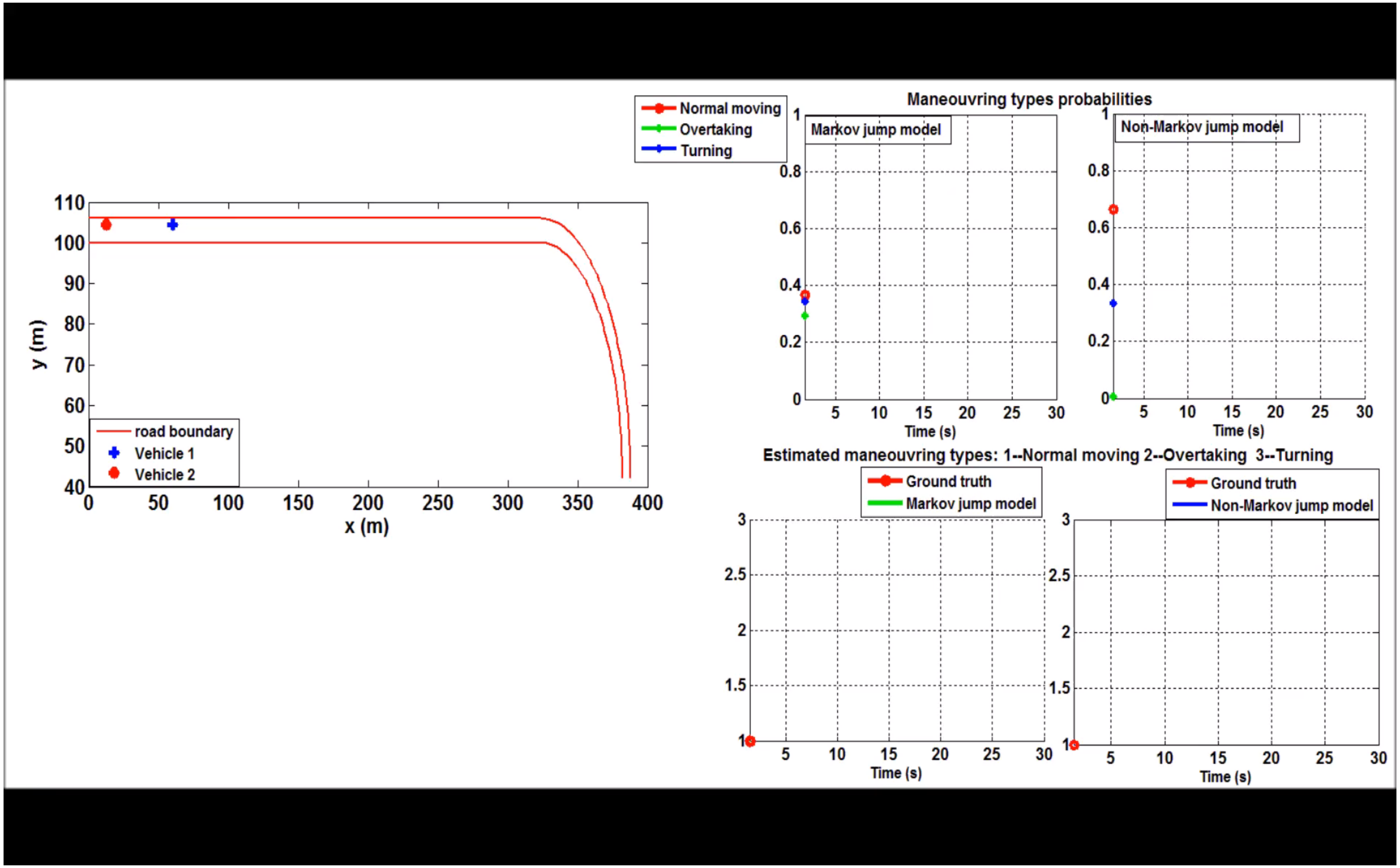


Measurements:

A sensor positioned at [200,30] (m) is applied to measure the range r_t and bearing angle θ_t of a particular vehicle with:

$$\begin{pmatrix} r_t \\ \theta_t \end{pmatrix} = \begin{pmatrix} \sqrt{(x_s - x_t)^2 + (y_s - y_t)^2} \\ \tan^{-1} \frac{y_s - y_t}{x_s - x_t} \end{pmatrix} + \begin{pmatrix} n_r \\ n_\theta \end{pmatrix}, \begin{pmatrix} n_r \\ n_\theta \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 5^2 (m^2), 0 \\ 0, 0.02^2 (rad^2) \end{pmatrix}\right)$$

Simulation studies—manoeuvring type determination



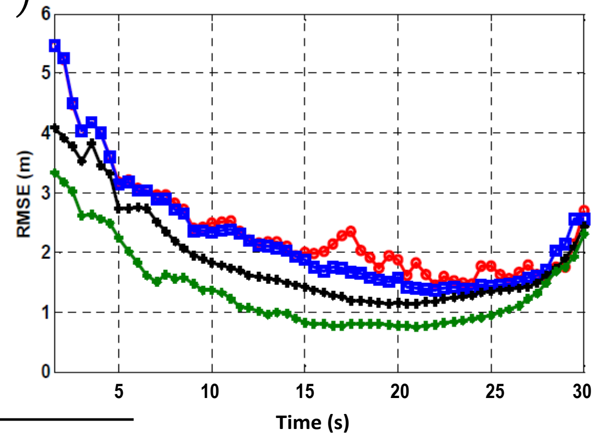
Simulation studies—modelling comparisons

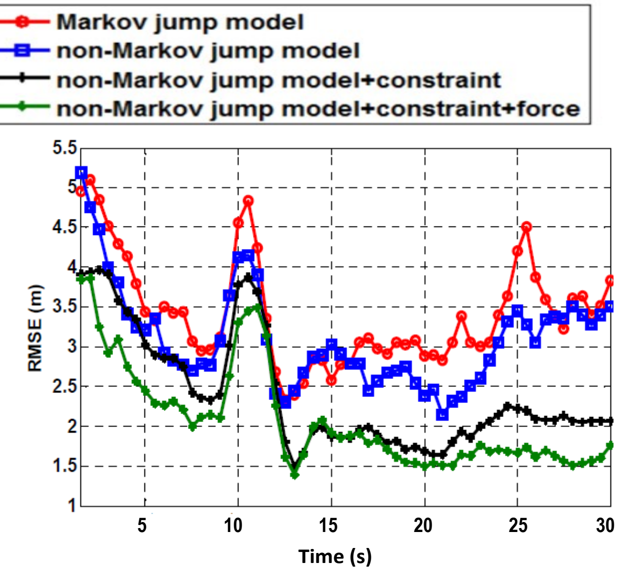
- **Comparisons between different modelling approaches are made from 100 Monte-Carlo simulations**
- **Gaussian particle filtering based approach (with the same particle number) is applied for every model**

$$\boldsymbol{RMSE_t} = \sqrt{\frac{\mathbf{1}}{\boldsymbol{M}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{M}}\left((\boldsymbol{x_t^g} - \boldsymbol{x_t^{i,e}})^2 + (\boldsymbol{y_t^g} - \boldsymbol{y_t^{i,e}})^2\right)}$$

$(\boldsymbol{x_t^g}, \boldsymbol{y_t^g})$: ground truth position at time t
$(\boldsymbol{x_t^{i,e}}, \boldsymbol{y_t^{i,e}})$: estimated position for the $i-th$ MC at t
M: number of MCs L: length of the trajectory

$$Ave\_RMSE = \frac{1}{M}\sum_{i=1}^{M}\sqrt{\frac{1}{L}\sum_{t=1}^{L}\left((x_t^g - x_t^{i,e})^2 + (y_t^g - y_t^{i,e})^2\right)}$$



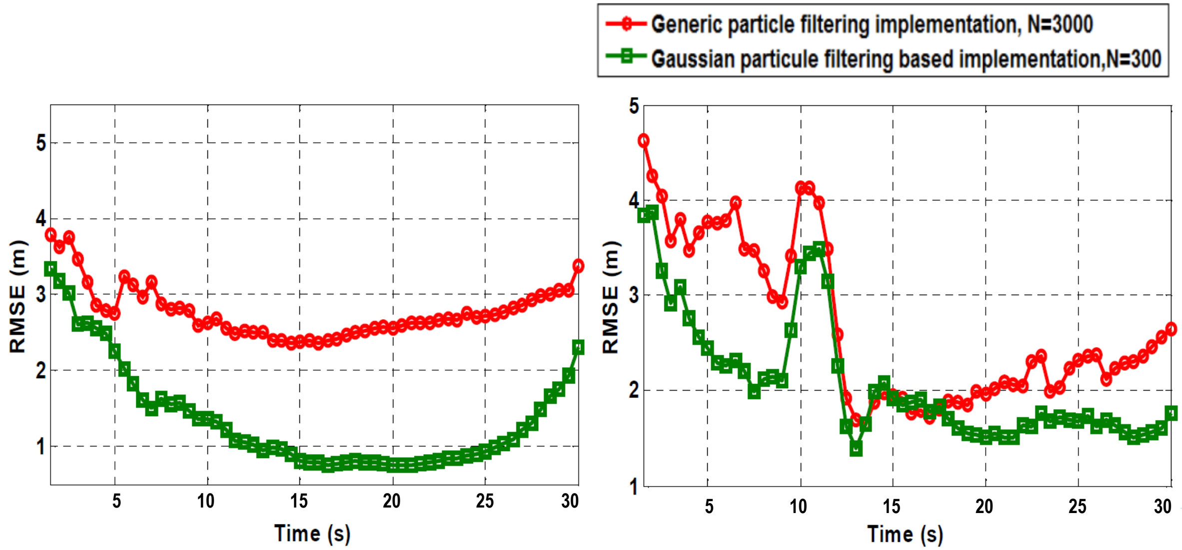


Ave_RMSEs (m) for different modeling approaches

	Markov jump model (T.Kirubarajan, 2003)	Non-Markov jump model (context information)	Non-Markov jump model+constraint	Non-Markov jump model+constraint+ force
Vehicle 1	2.52	2.43	2.17	1.51
Vehicle 2	3.45	3.25	2.42	2.10

Simulation studies—algorithms comparison

- Under the same model, comparisons have been made with the generic particle filtering based and Gaussian particle based approaches



AVE_RMSEs (s) for different implementation algorithms

	Generic particle filtering	Gaussian particle filtering
Vehicle 1	2.77	1.51
Vehicle 2	2.65	2.10

Computational costs for trajectories tracking (per sample):

- Generic particle filtering: 0.12 (s)
- Gaussian particle filtering: 0.03 (s)

Conclusions and future works

- In this work, we have developed a new traffic participants state estimation method:

Model aspect:

- Non-Markov jump model is applied to model the transition probabilities based on the context information
- For a particular model, both the force and constraint are applied for refining the model

Algorithm aspect:

- Bayesian inference algorithm is developed for the model
- Gaussian particle filtering based algorithm is applied for implementation

- Future work:

- Data associations (miss detection/false alarms)
- System parameters calibration (machine learning)

Questions?