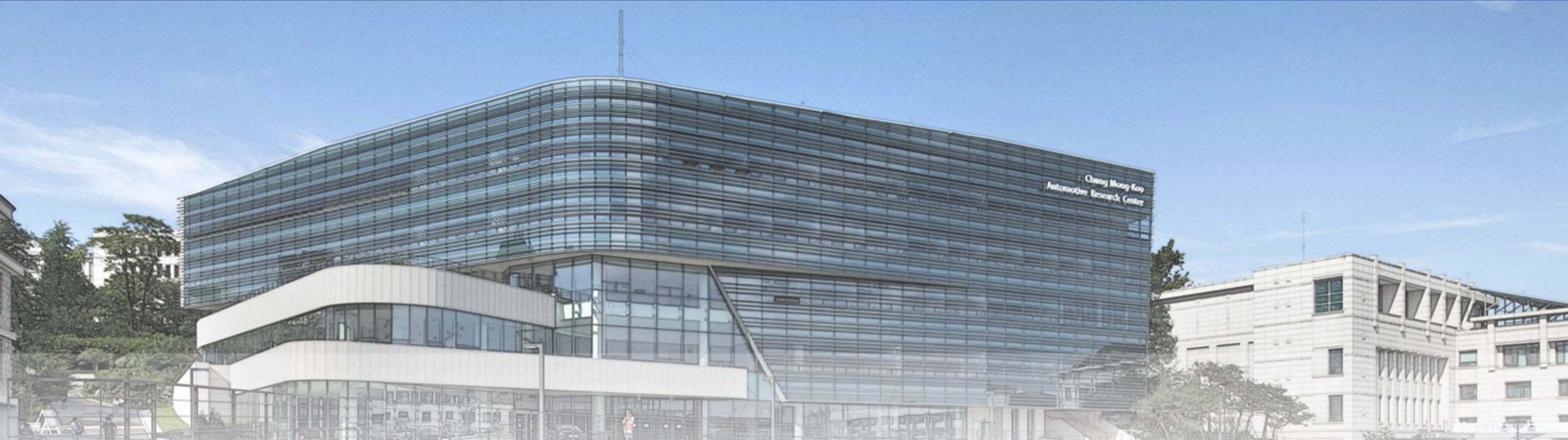


Ego-vehicle speed prediction using long short-term memory based recurrent neural networks

Kyuhwan Yeon

Advising professor: Myoungho Sunwoo

ACE Lab control hub

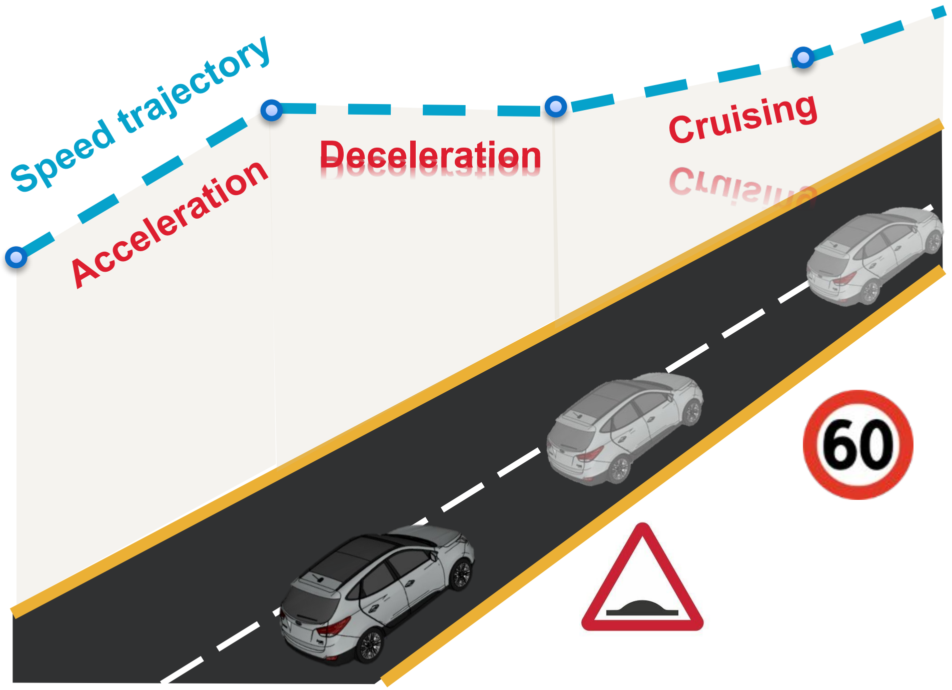
Department of Automotive Electronics & Control Engineering

Contents

Necessity of vehicle speed prediction

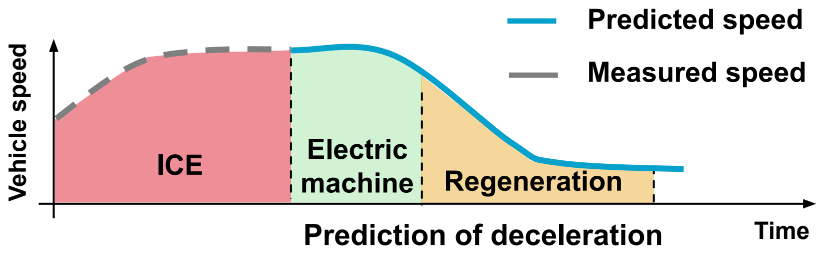
- Predicted speed based powertrain control strategies
  - Reduction of fuel consumption : energy management of Hybrid Electric Vehicles (HEVs)[1)]
  - Improvement of drivability: optimal gear shift strategies[2)]

Predicted speed profile

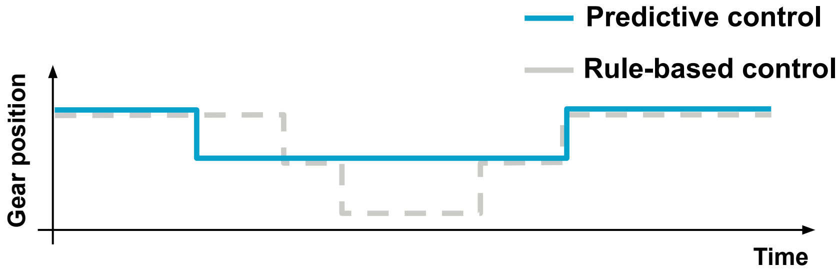


Powertrain control

- Energy management of the HEVs



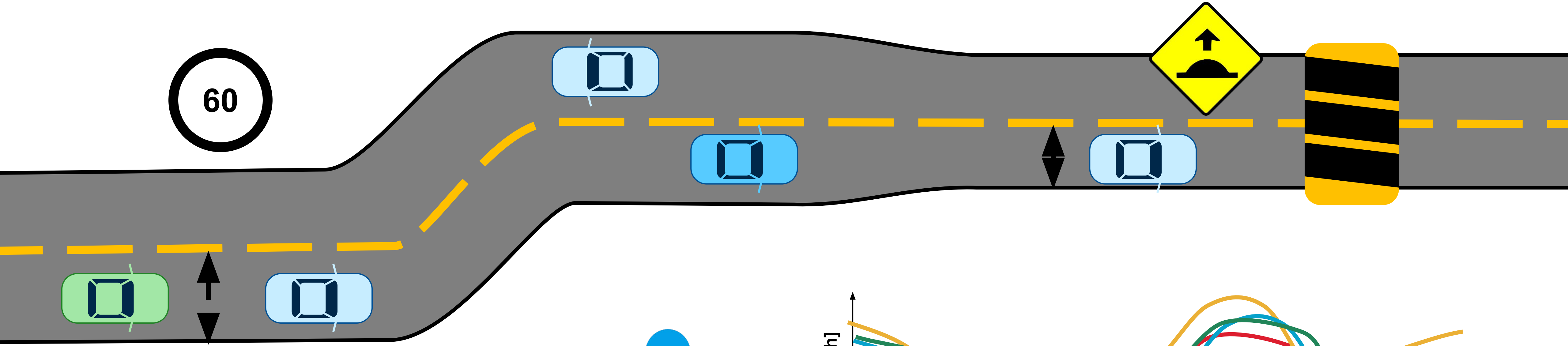
- Gear shift strategy



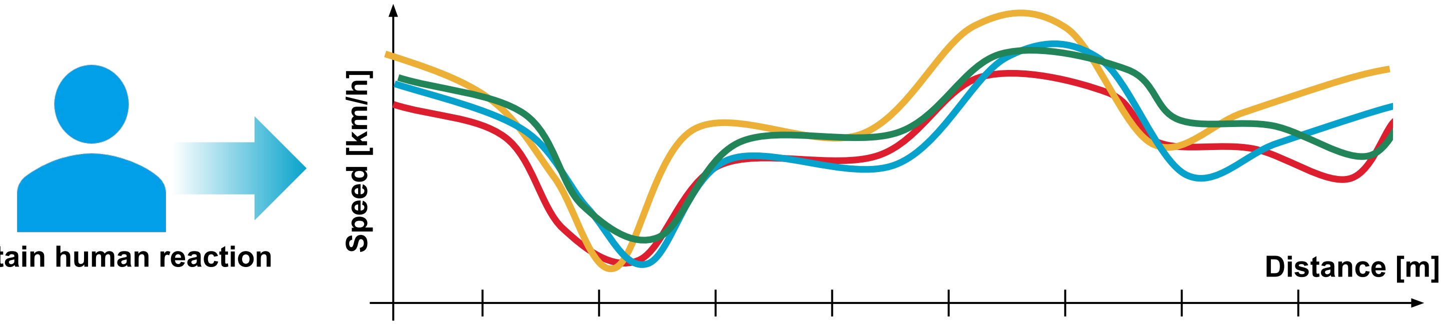
1) C. Sun, X. Hu, S. J. Moura, and F. Sun, "Velocity Predictors for Predictive Energy Management in Hybrid Electric Vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 3, pp. 1197–1204, 2015.
2) B. Wolfram and S. Thomas, "Anticipatory drivetrain management," *ATZ*, vol. 116, no. 01, pp. 30–33, 2014.

Research background (II)

- **Challenges of vehicle speed prediction**
  - Modeling of uncertain human reaction to arbitrary driving environment
    - Traffic condition, curve, speed limit, speed bump, road width, movement of vehicle ahead, etc.

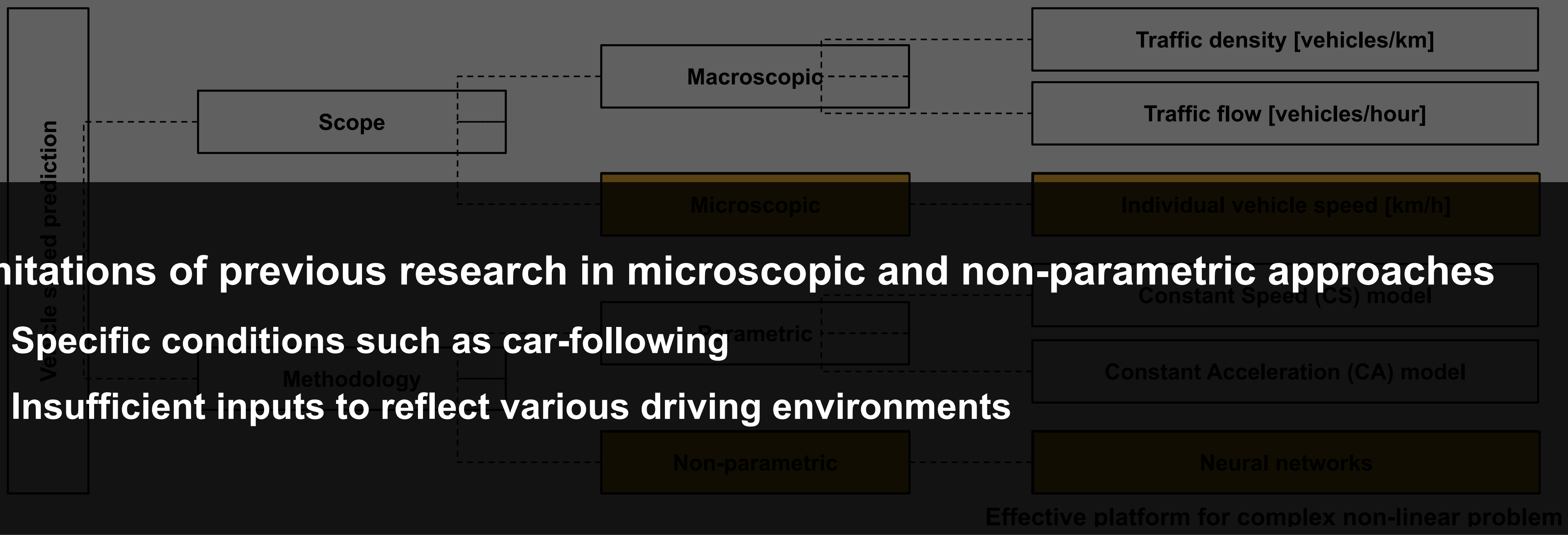


Arbitrary driving environment



Uncertain human reaction

■ Category of research on vehicle speed prediction*



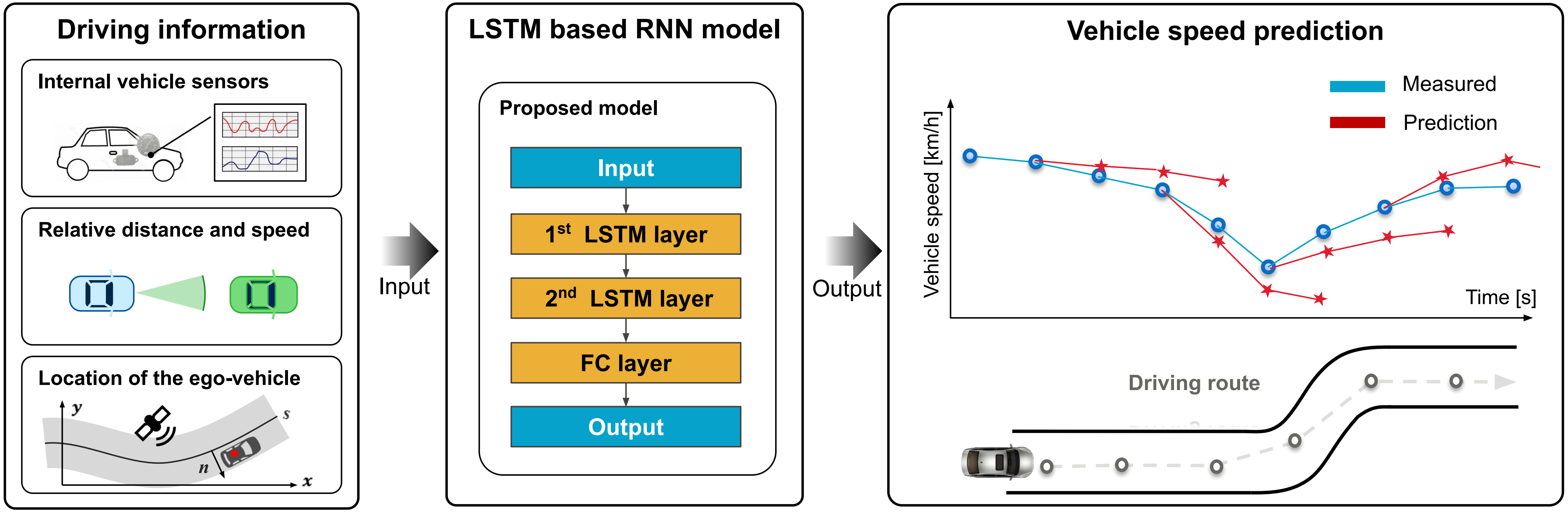
Limitations of previous research in microscopic and non-parametric approaches

- ❑ Specific conditions such as car-following
- ❑ Insufficient inputs to reflect various driving environments

* E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short-term traffic forecasting: Overview of objectives and methods," *Transp. Rev.*, vol. 24, no. 5, pp. 533–557, 2004.

Research objective

- **Design of the ego-vehicle speed prediction model using long short-term memory (LSTM) based recurrent neural networks (RNNs)**
  - Microscopic and non-parametric approach
  - Improvement of prediction accuracy on real urban roads by using various inputs
    - Internal vehicle information, relative distance and speed, location of the ego-vehicle
  - Implementation of the proposed model in the embedded system

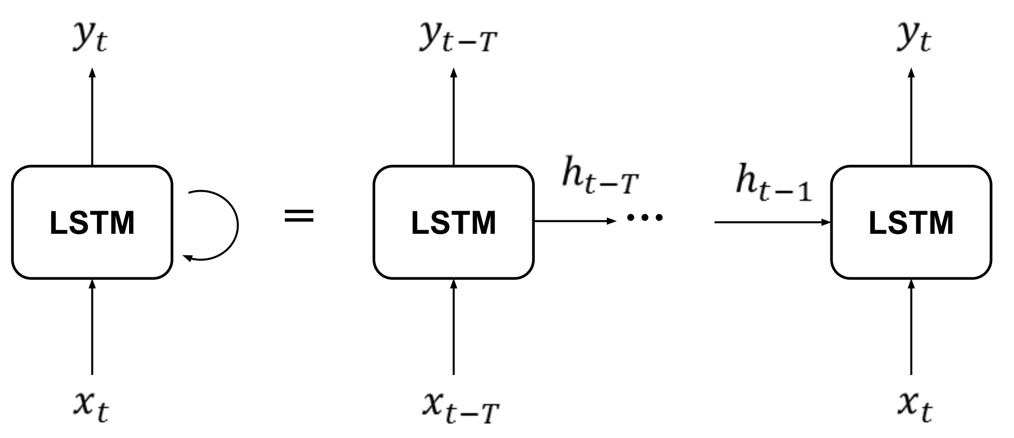


Methodology

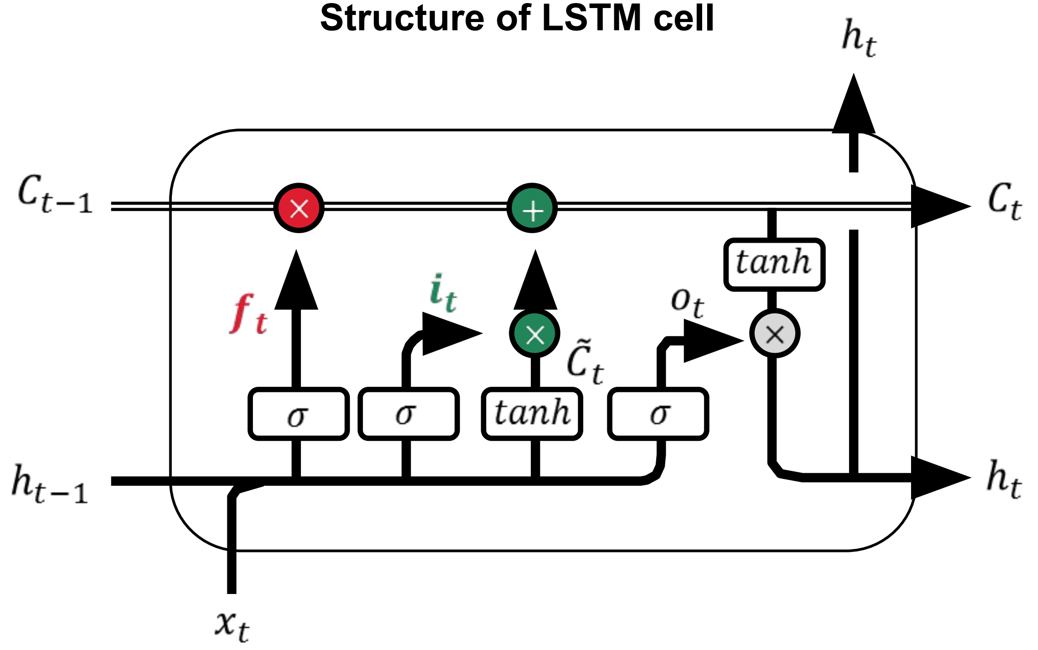
LSTM based RNNs*

- Strong prediction performance for sequential data due to feedback loop in RNNs
  - Temporal correlations in determination of the vehicle speed
- Effective conveyance of previous information of the inputs
  - Using cell states

Unfolded LSTM based RNNs



Structure of LSTM cell



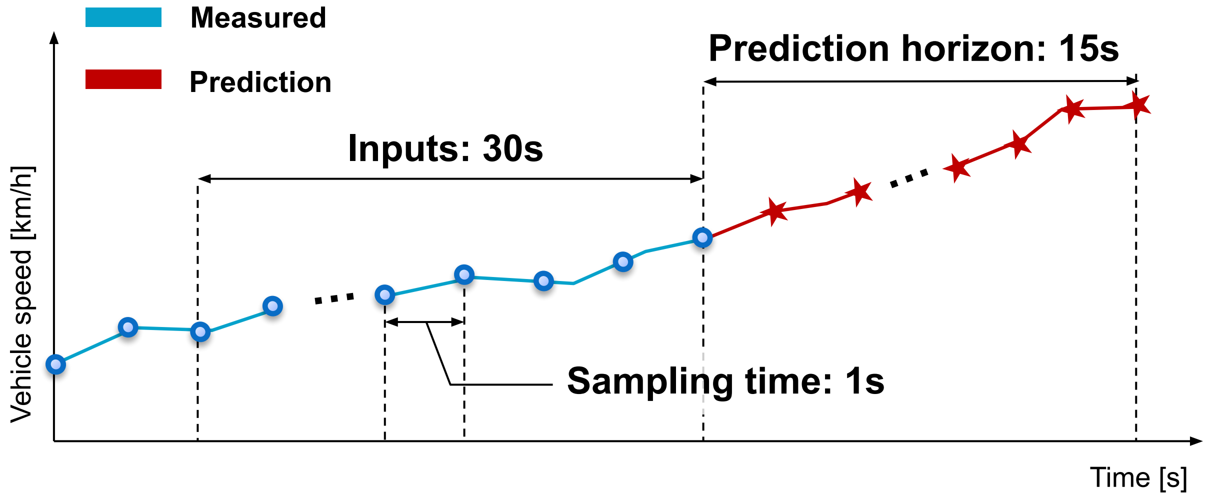
i_t: input gate, f_t: forget gate, o_t: output gate, h_t: hidden state, C_t: cell state, $\tilde{C}_t$: candiate cell state, σ: activation function, W: weight, b: bias, x: input state

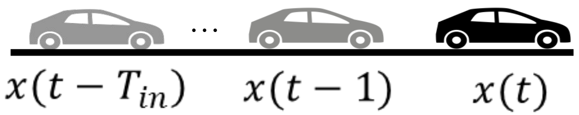
*S. Hochreiter and J. Urgen Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

Vehicle speed prediction model

Model specification

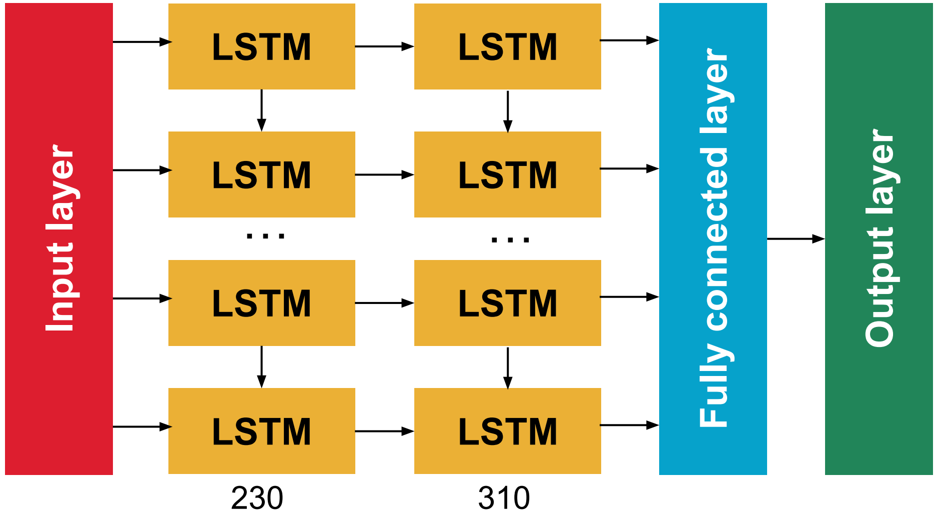
- Model structure
  - 2 LSTM layers + a fully connected layer
- Prediction horizon
  - T_{out}: 15 seconds
- Length of input states
  - T_{in}: 30 seconds
- Prediction every second

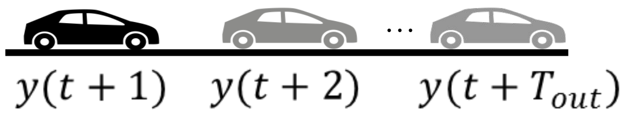




$x(t)$

1. Average speed of 4 wheels
2. Acceleration
3. Engine speed
4. Master cylinder pressure
5. Accel pedal position
6. Relative distance from the vehicle ahead
7. Relative speed from the vehicle ahead
8. Ego-vehicle location in the driving route



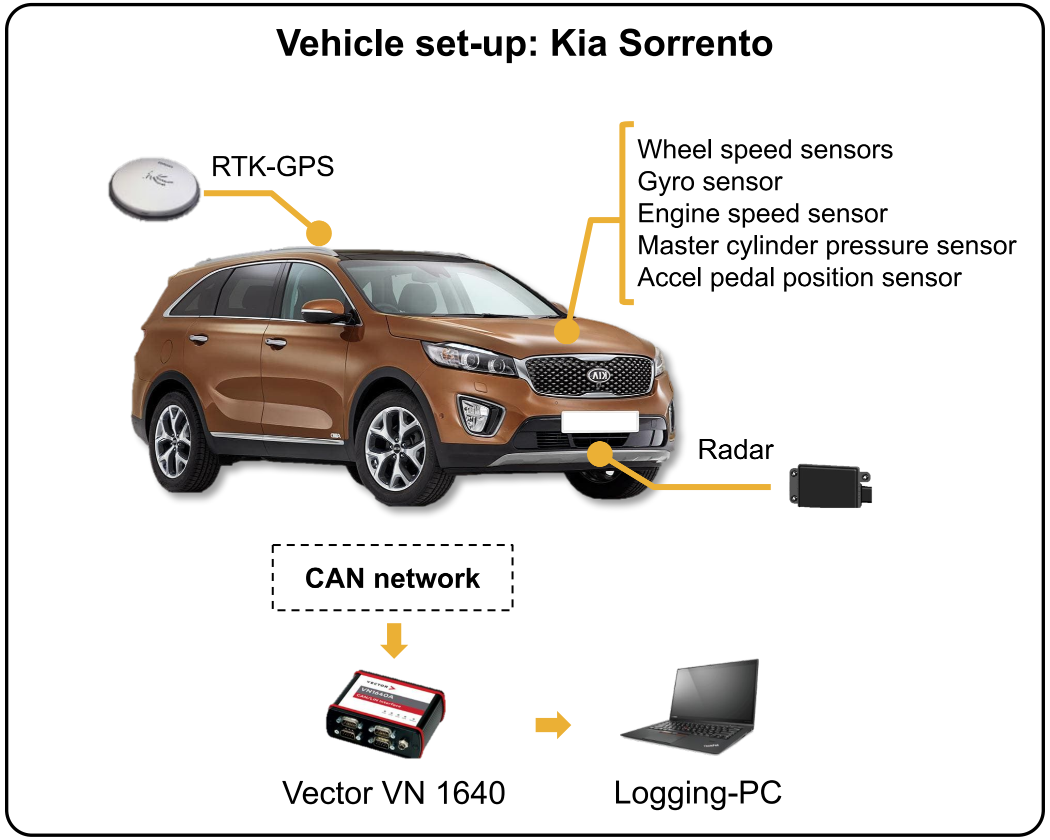


$y(t)$ Vehicle speed

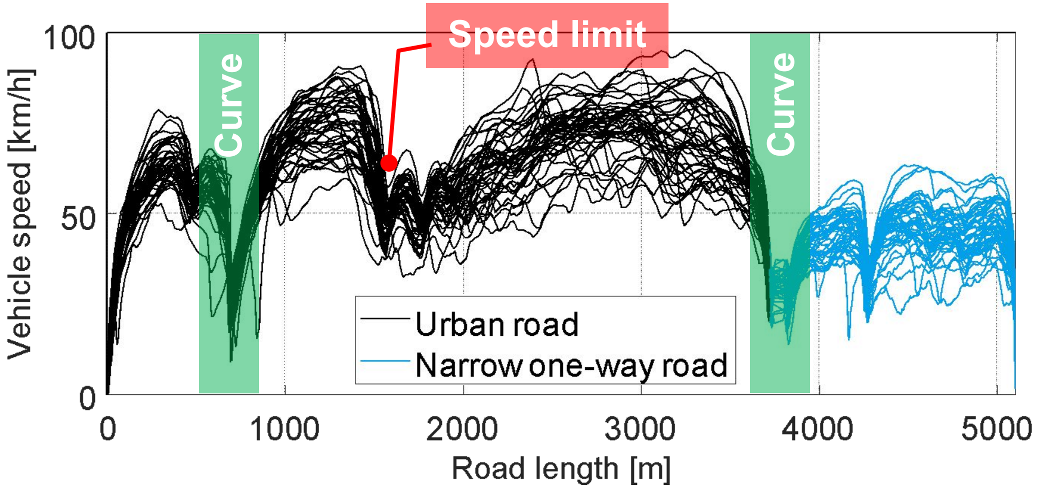
Data acquisition environment

■ Modeling data

- ► Training dataset: 34 cycles
- ► Validation dataset: 6 cycles
- ► Test dataset: 6 cycles



Driving route



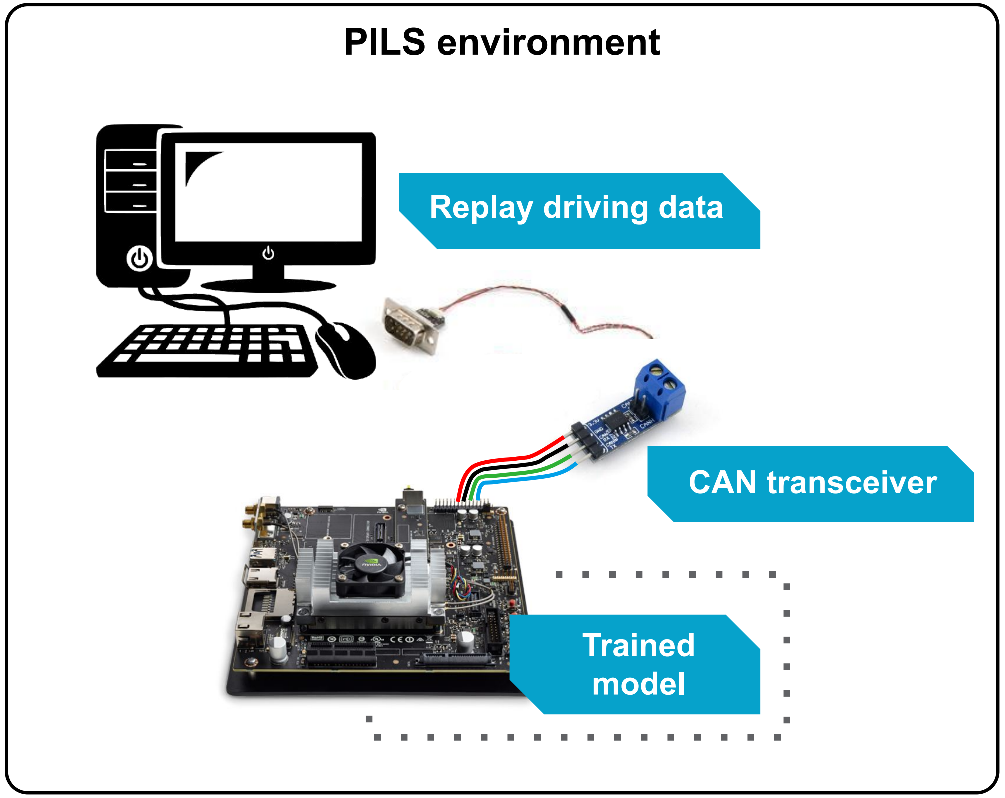
Speed profiles

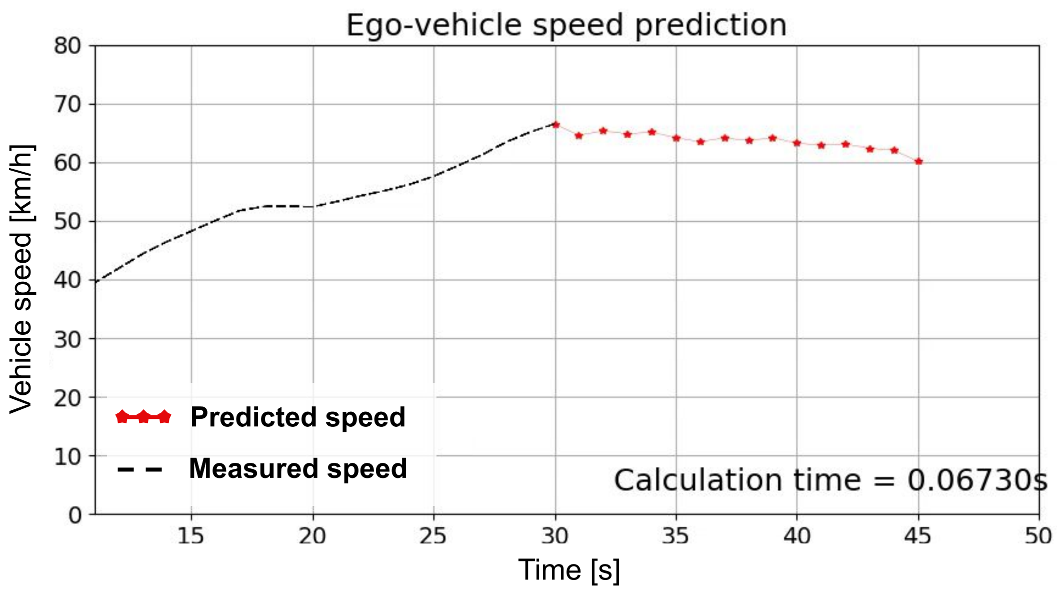
Results and analysis

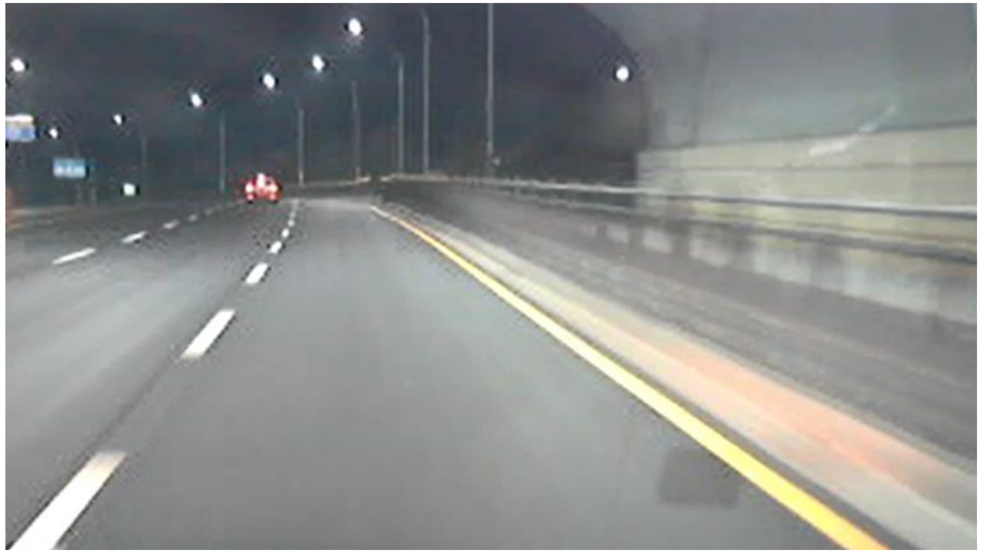
Processor in-the loop simulation (PILS)

Verification of real-time performance

- Integration to embedded system
- Nvidia Jetson TX2
  - GPU: 256 CUDA cores @ 1300MHz



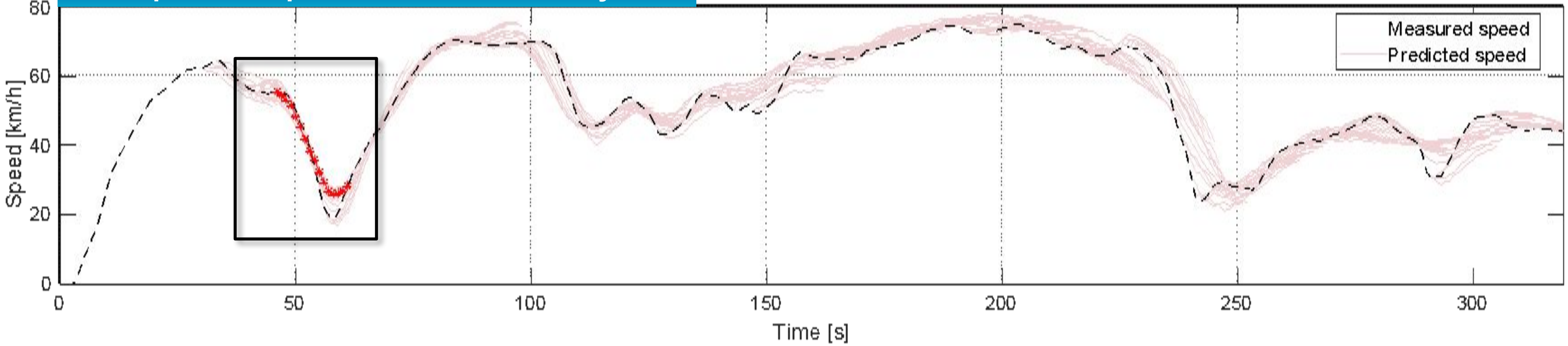




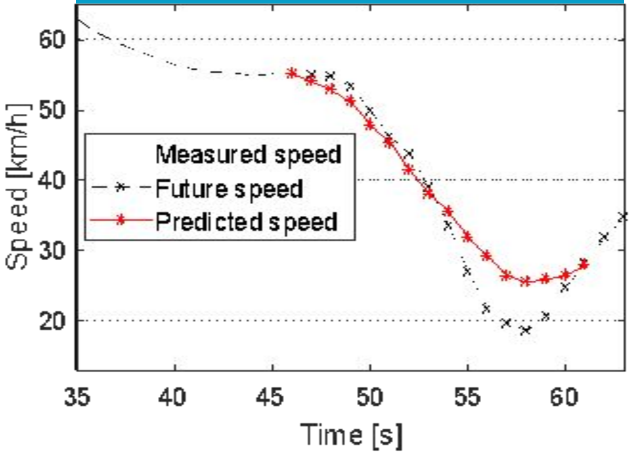
Prediction results

■ Evaluation of the proposed model along the prediction horizon

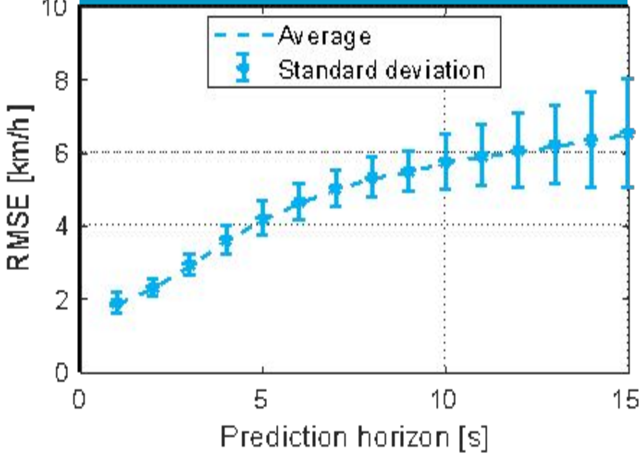
Example of the prediction result: test cycle 2



Detail view



Prediction horizon



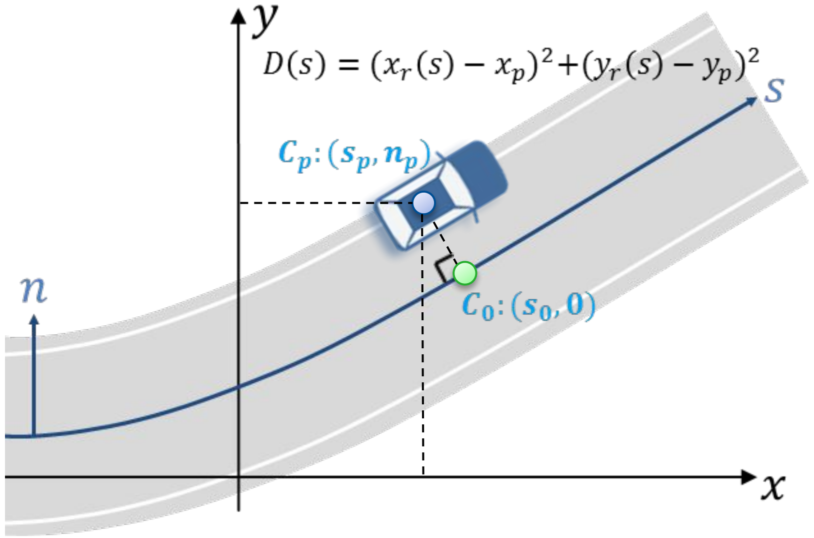
Index		RMSE [km/h]
Test 1	0.917	4.572
Test 2	0.918	4.030
Test 3	0.894	5.820
Test 4	0.885	4.725
Test 5	0.895	5.334
Test 6	0.875	5.660
Average	**0.897**	**5.023**

Curvilinear coordinate conversion (II)

- **Calculation of the closest point using a numerical technique that combines quadratic minimization and Newton's method**
  - Prevents divergence of the optimization and improves the rate of convergence for real-time implementation
  - **Quadratic Minimization**
    - $s_0^*[k] = \frac{1}{2} \cdot \frac{b_{23}D(s_1)+b_{31}D(s_2)+b_{12}D(s_3)}{b_{23}D(s_1)+b_{31}D(s_2)+b_{12}D(s_3)}, \quad k = 1,2,\ldots$

    where

    $a_{ij} = s_i - s_j$ and $b_{ij} = s_i^2 - s_j^2$
  - **Newton's method**
    - $s_0^{k+1} = s_0^k - \frac{D'(s_k^*)}{D''(s_k^*)}, \quad k = 1,2,\ldots$



$$\begin{cases} x_r(s) = a_x(s - s_i)^3 + b_x(s - s_i)^2 + c_x(s - s_i) + d_x \\ y_r(s) = a_y(s - s_i)^3 + b_y(s - s_i)^2 + c_y(s - s_i) + d_y \end{cases}$$

Conclusion

Conclusion

- **In order to reduce fuel consumption and improve drivability of a vehicle, vehicle speed prediction can be applied in the powertrain control strategies**

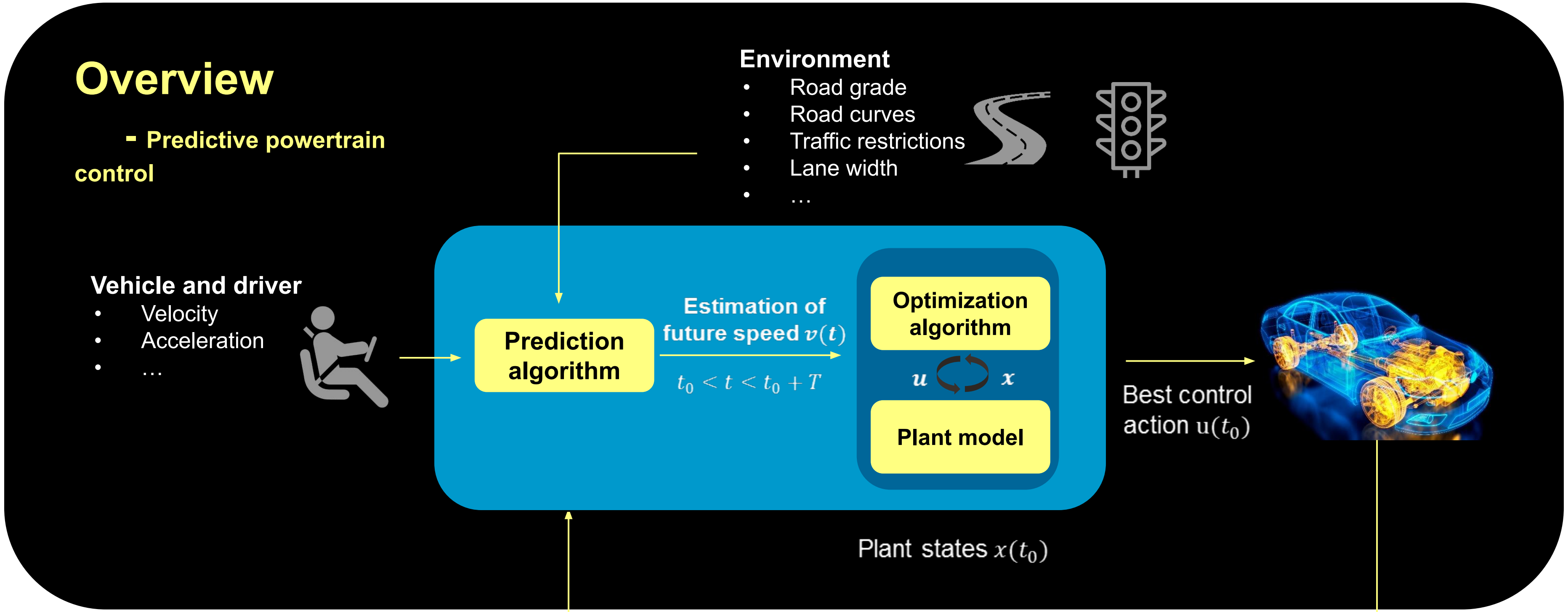
- **An ego-vehicle speed prediction model using LSTM based RNNs was proposed to improve the prediction performance in various driving conditions**
  - The proposed model uses several model inputs
    - Internal vehicle sensors, a radar sensor, and the location information of the ego-vehicle

- **The proposed model was validated by real-driving data from a vehicle equipped with the radar and RTK-GPS**
  - The RMSE of test data was 5.023 km/h
  - The maximum calculation time was 0.073 seconds in the embedded system

Thank you
for your attention!

Model predictive control in HEVs

- Minimize the fuel consumption and deviation of SOC



$$Minimize: J = \alpha \int_t^{t+T} \dot{m}_f dt + \beta[SOC(t+k) - SOC(t)]$$

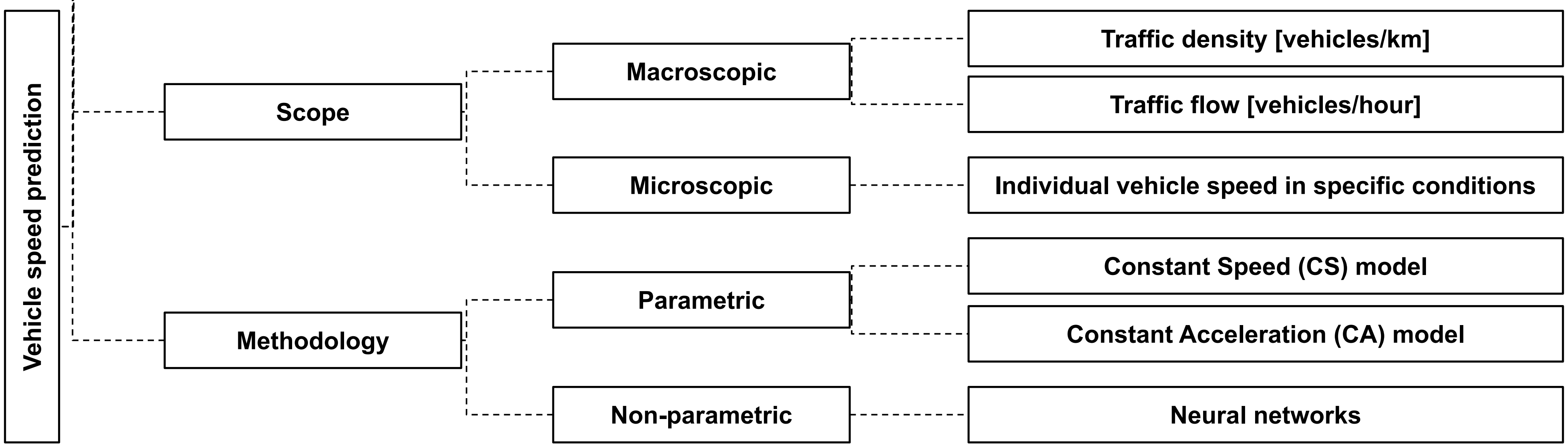
$\dot{m}_f$: fuel consumption, SOC: state of charge, T: prediction horizon

$$u = [P_{em}, gear\ position]$$

*C. Sun, X. Hu, S. J. Moura, and F. Sun, "Velocity Predictors for Predictive Energy Management in Hybrid Electric Vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 3, pp. 1197–1204, 2015.

Category of vehicle speed prediction research [3)]

- Scope
  - Macroscopic: forecasts traffic variables such as traffic density [vehicles/km]
  - Microscopic: forecasts individual vehicle speed in specific condition such as car-following
- Methodology
  - Parametric approaches: predicts a vehicle speed with a predetermined model structure
  - Non-parametric approaches: predicts a vehicle speed with a not pre-defined model structure



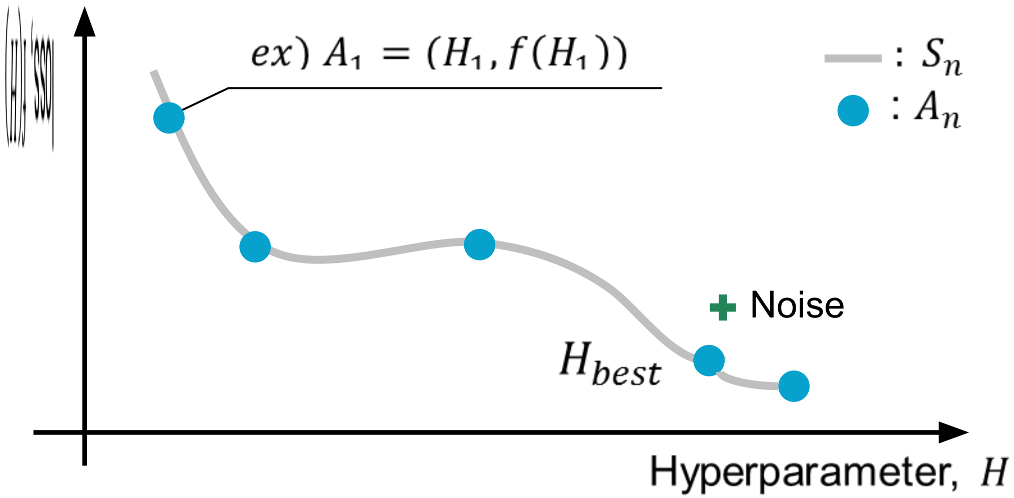
3) E. I. Vlahogianni, J. C. Golias, and M. G. Karlaftis, "Short-term traffic forecasting: Overview of objectives and methods," *Transp. Rev.*, vol. 24, no. 5, pp. 533–557, 2004.

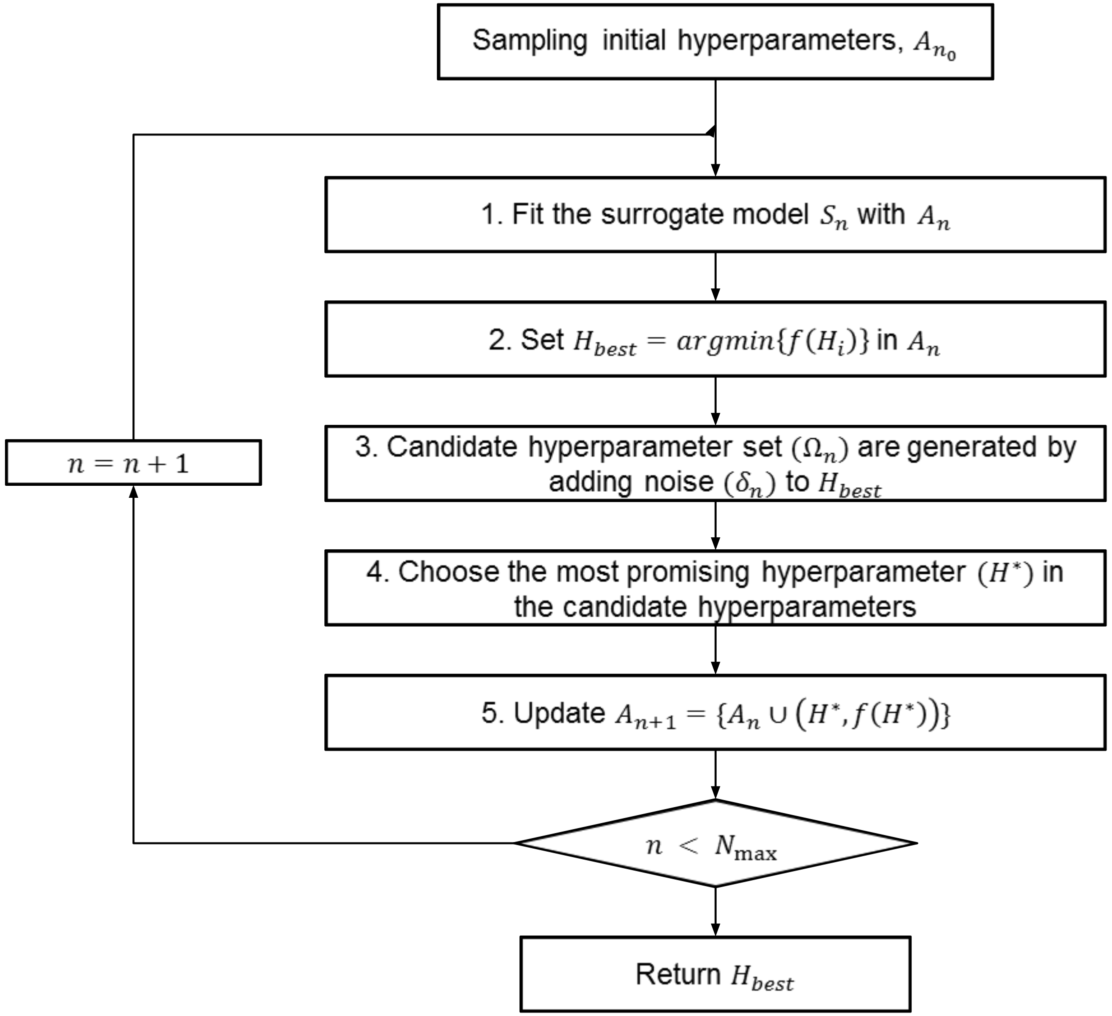
Hyperparameter Optimization via RBF and Dynamic coordinate search

Find best hyperparameter set which can minimize the loss in validation data set

- Objective: $\min_{H \in R^D} (f(H))$
- Sampled points: $A_n = \{(H_i, f_i(H_i))\}_{i=1}^{n}$
- Surrogate model: $S_n(H) = \sum_{i=1}^{n} \lambda_i \phi(\| H - H_i \|) + p(H)$

Surrogate model with sampled points

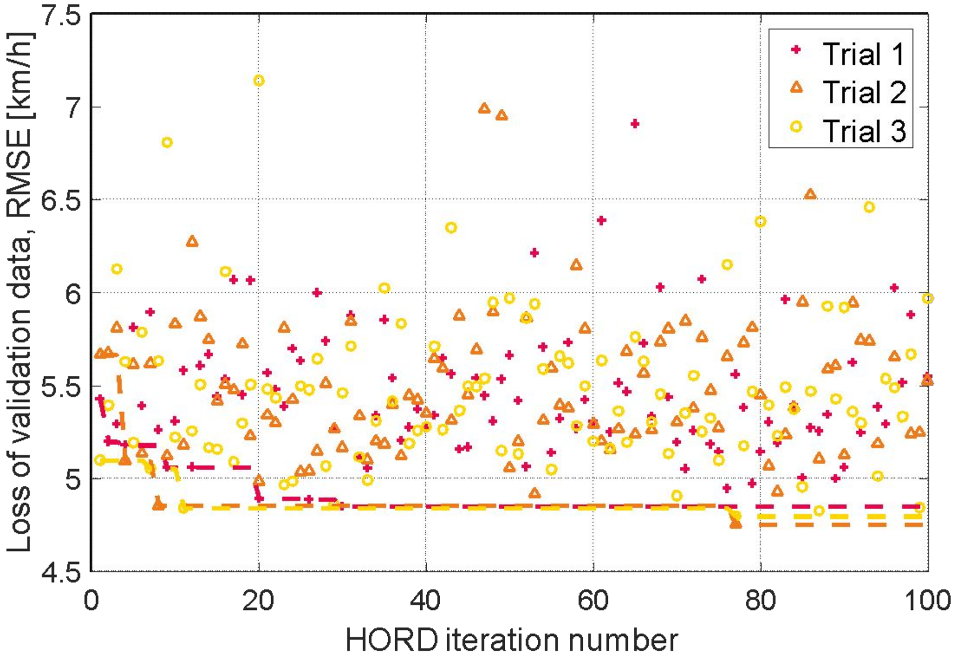




Model training

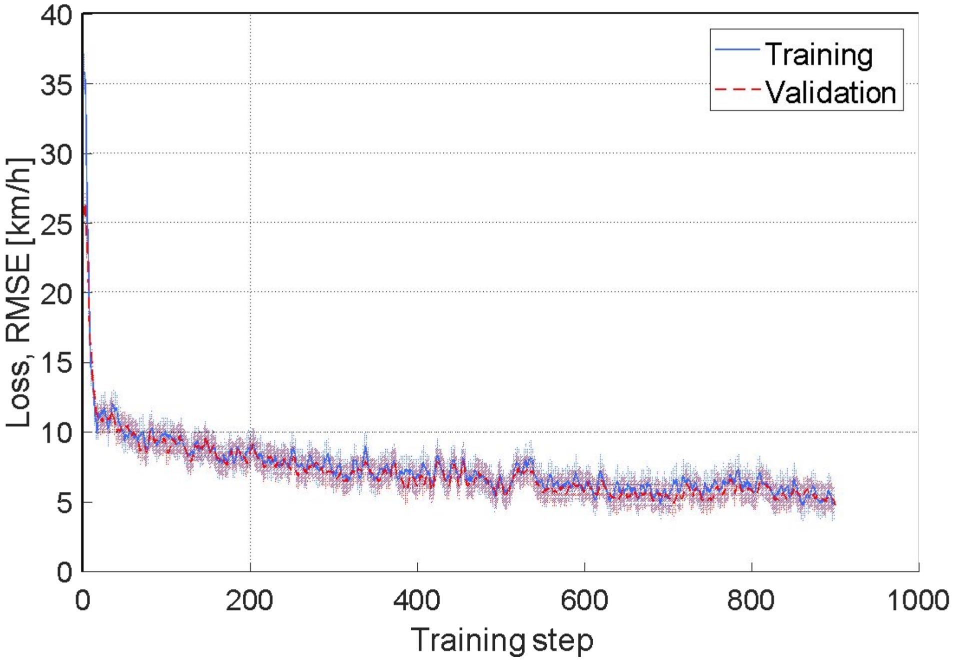
■ Model structure optimization

- ► Hyperparameter Optimization via Radial basis function and Dynamic coordinate search (HORD)*
  - – Search of the best hidden states number for the 1-LSTM layer and 2-LSTM layer



Index	Best iteration number	H1	H2	RMSE [km/h]
Trial 1	30	218	125	4.849
Trial 2	77	230	310	4.753
Trial 3	76	270	270	4.796

Trial 2, 77



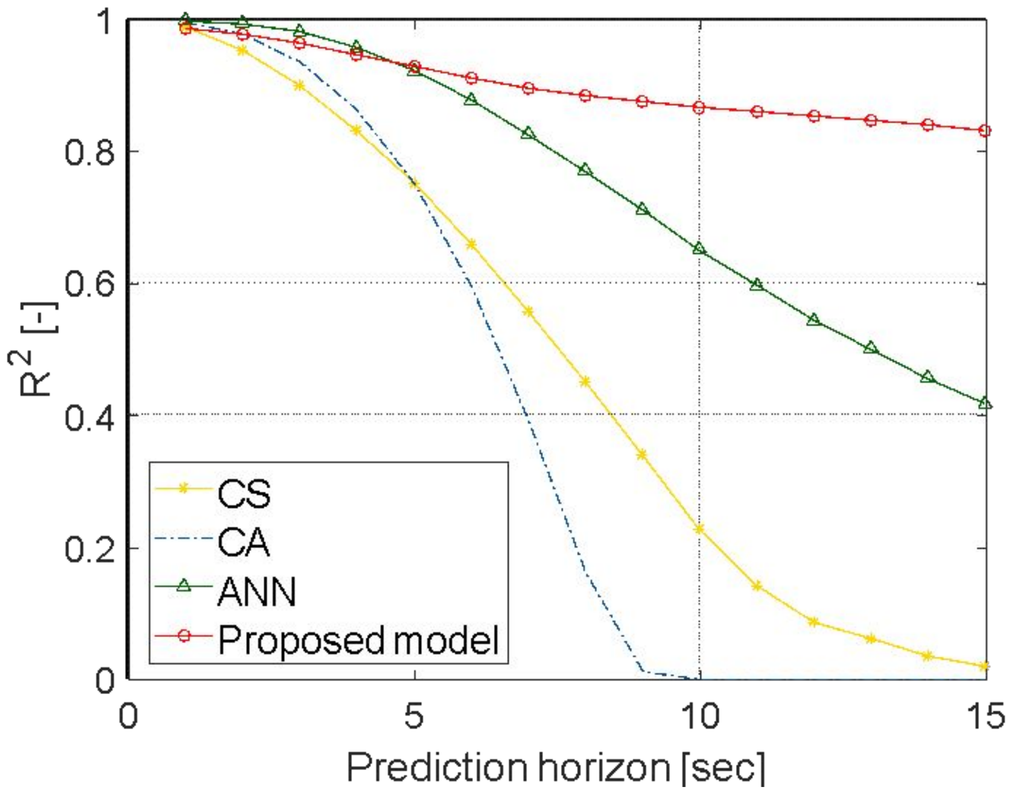
Index	Training data	Validation data
RMSE [km/h]	4.993	4.753

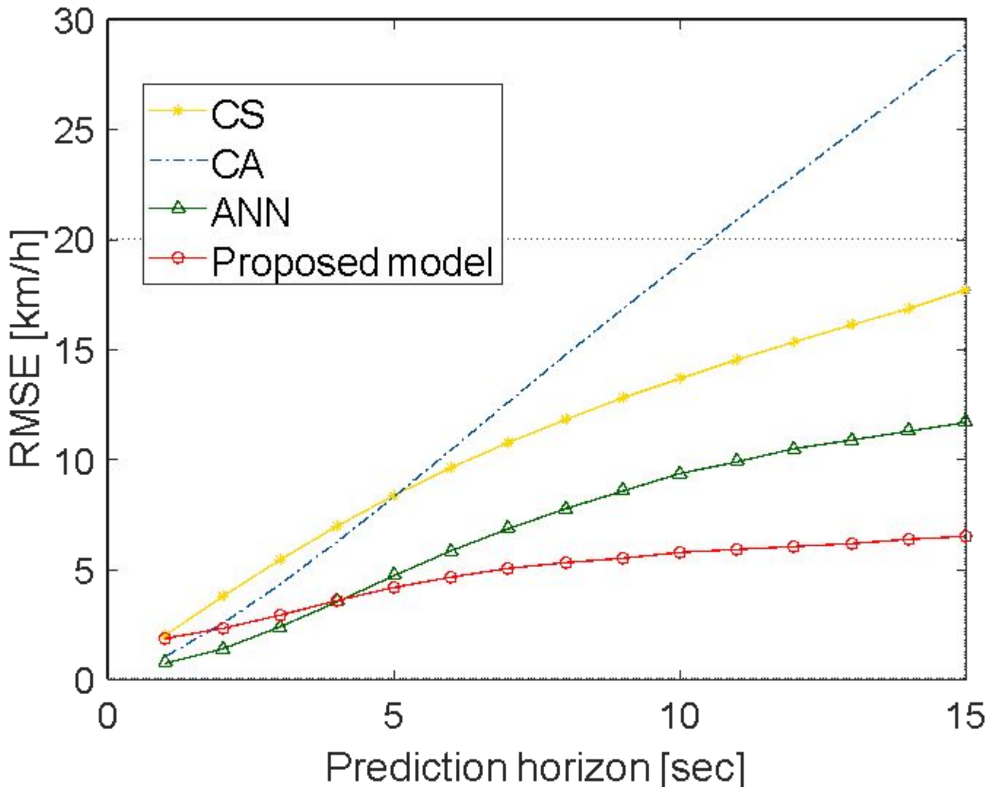
*I. Ilievski, T. Akhtar, J. Feng, and C. A. Shoemaker, "Efficient Hyperparameter Optimization of Deep Learning Algorithms Using Deterministic RBF Surrogates," 2016.

Comparison with previous studies

Model of previous studies

- Constant speed (CS) model
- Constant acceleration (CA) model
- Artificial neural network (2 MLP layers optimized with HORD algorithm)





Effect of the input combinations on the prediction performance

Group of inputs

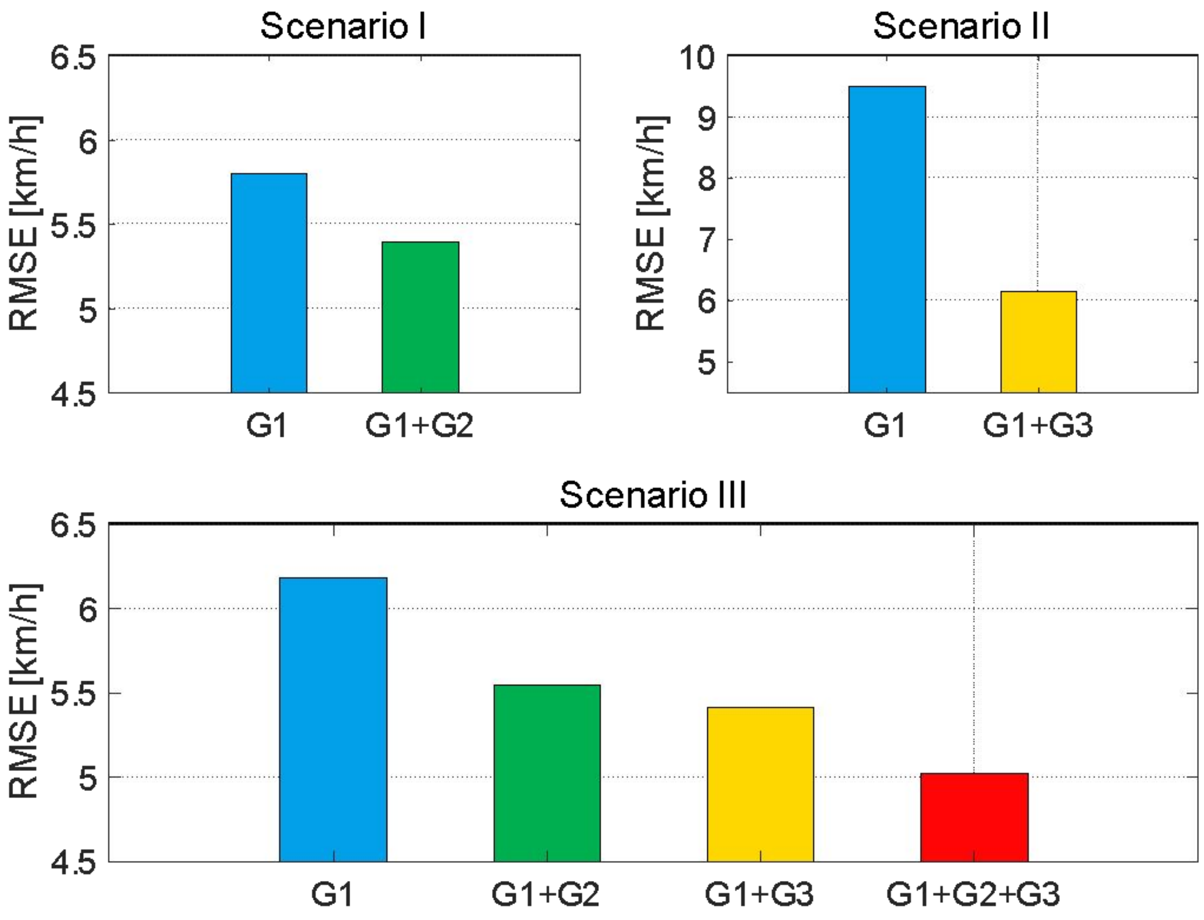
- Group1: $v(t), a(t), w(t), p_b(t), p_a(t)$
- Group2: $r_d(t), r_v(t)$
- Group3: $s(t)$

Scenarios

- Scenario I: Car-following
- Scenario II: On the sharp curve
- Scenario III: Full path

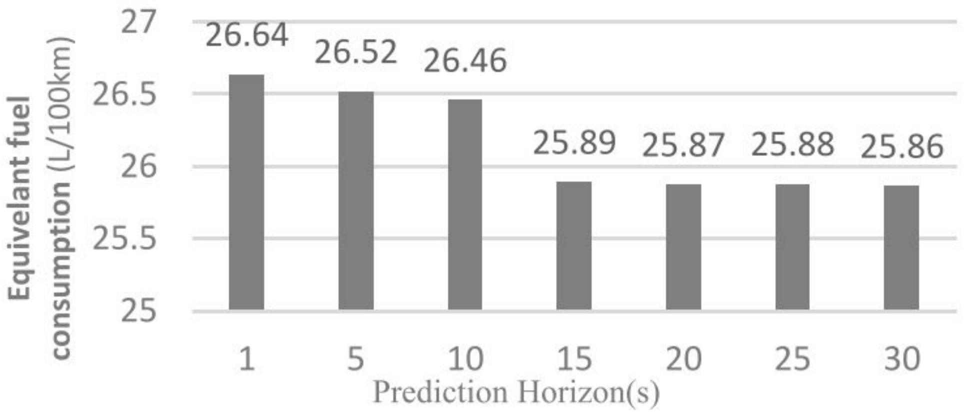
Prediction result (RMSE [km/h])

Inputs	Scenario I	Scenario II	Scenario III
G1	5.802	9.497	6.183
G1+G2	5.394	7.505	5.546
G1+G3	5.703	6.168	5.416
G1+G2+G3	4.843	6.923	**5.023**

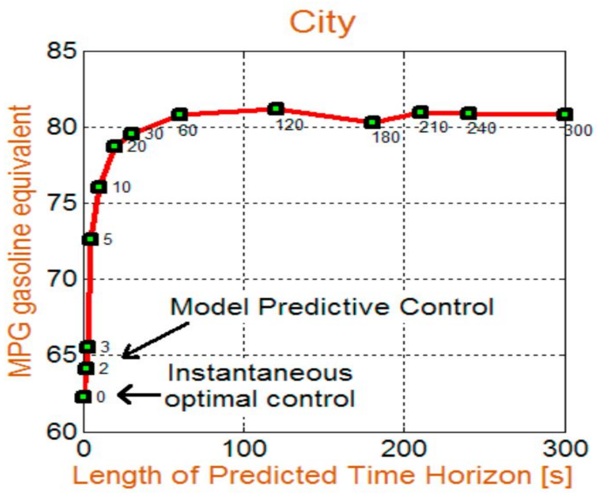


Effect of prediction horizon

- **HEVs application**
  - J. Lian *et al.*, "A mixed logical dynamical-model predictive control (MLD-MPC) energy management control strategy for plug-in hybrid electric vehicles (PHEVs)," *Energies*, vol. 10, no. 1, 2017.
  - A. Rezaei and J. B. Burl, "Effects of time horizon on Model Predictive Control for Hybrid Electric Vehicles," *IFAC-PapersOnLine*, vol. 28, no. 15, pp. 252–256, 2015.
- **Cooperative adaptive cruise control**
  - D. Lang, R. Schmied, and L. Del Re, "Prediction of Preceding Driver Behavior for Fuel Efficient Cooperative Adaptive Cruise Control," *SAE Int. J. Engines*, vol. 7, no. 1, pp. 2014-01–0298, 2014.



HEVs application [1]



HEVs application [2]

Table 2. Comparison of potential consumption reduction.

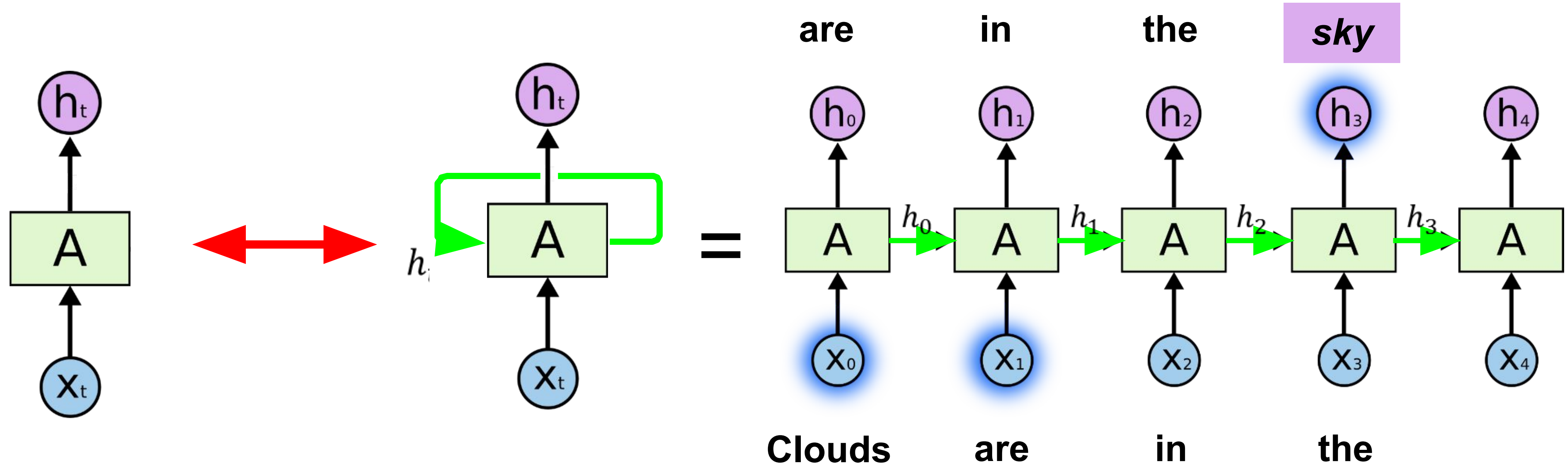
		prediction horizon			
margin	applied scenario	10s	15s	20s	25s
10m	perfect pred.	8.2%	8.5%	8.5%	8.6%
	Scenario 1	6.9%	7.1%	7.1%	6.9%
	Scenario 2	5.5%	5.7%	5.6%	5.3%
20m	perfect pred.	12.5%	14.3%	15%	15.1%
	Scenario 1	8.6%	11.1%	11.3%	10.5%
	Scenario 2	4.8%	7.2%	8%	6.8%
30m	perfect pred.	15.3%	19.4%	19.5%	19.8%
	Scenario 1	10.1%	14.5%	13.7%	13.1%
	Scenario 2	7%	11.5%	11.8%	10.4%

ACC application

Recurrent Neural Networks

Standard RNN structure (I)

- **Difference with the standard neural networks**
  - Using previous information when determining the current state

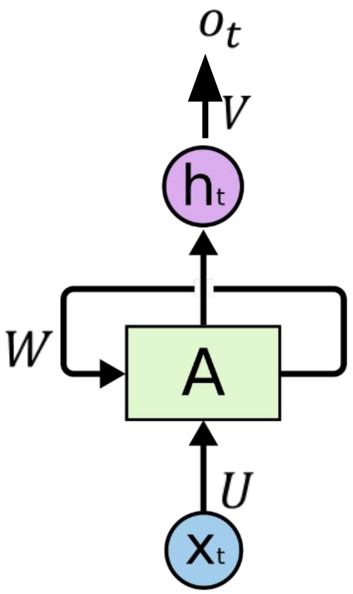


Standard neural network

Recurrent neural network

Standard RNN structure (II)

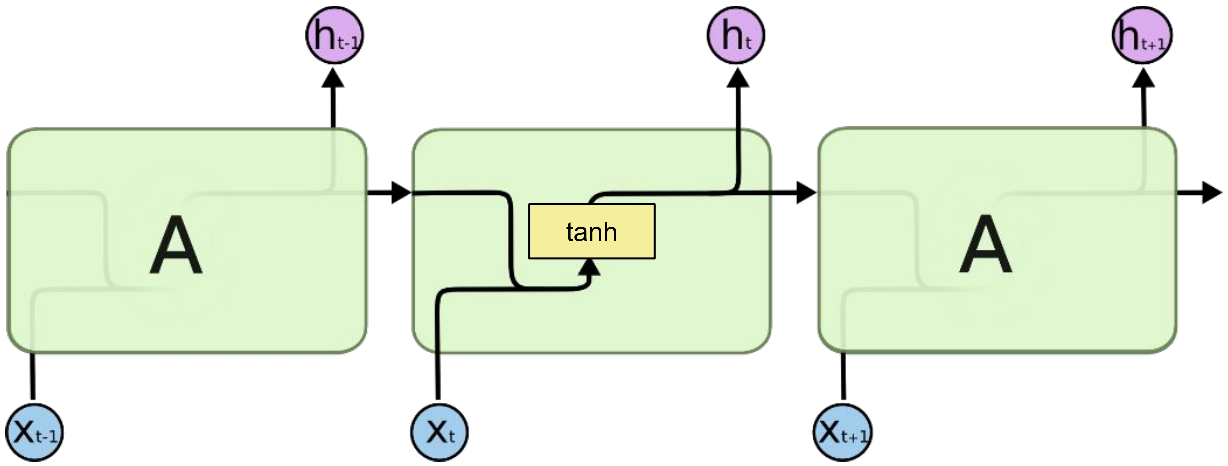
- **Structure of standard RNN cell: tanh layer**



$$h_t = \tanh(Ux_t + Wh_{t-1})$$

$$o_t = softmax(Vh_t)$$

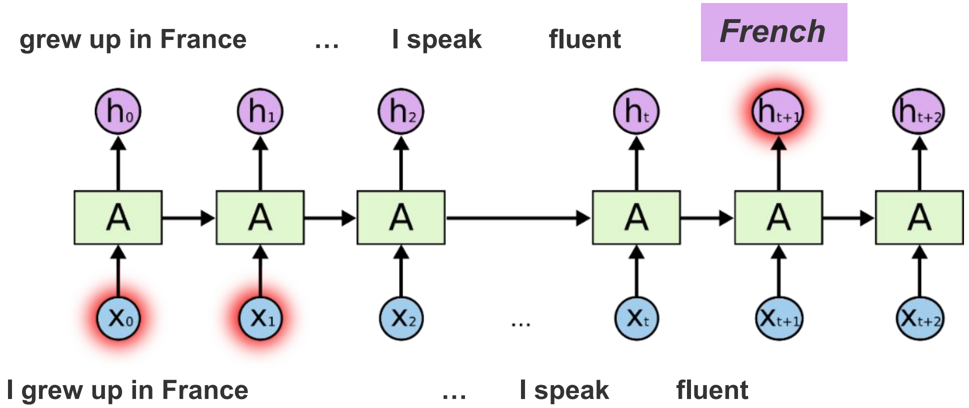
U: input weights matrix
W: Hidden weights matrix
V: output weights matrix

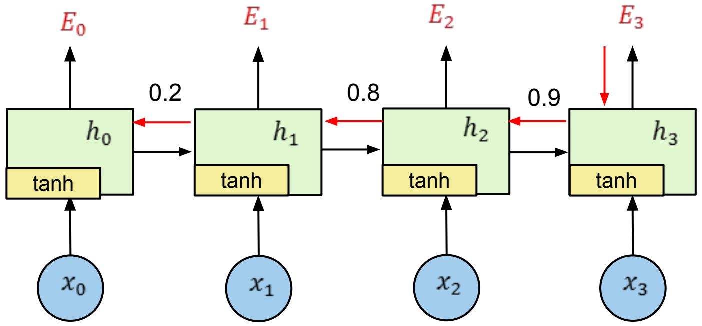


Standard RNN structure (III)

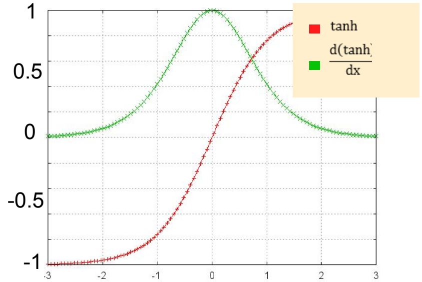
Vanishing gradient problem

- Standard RNN model easily forgets memory from long time ago because of backpropagation
- **One of the solutions prevents the vanishing gradient problem is LSTM**





$h_t = \tanh(Ux_t + Wh_{t-1})$



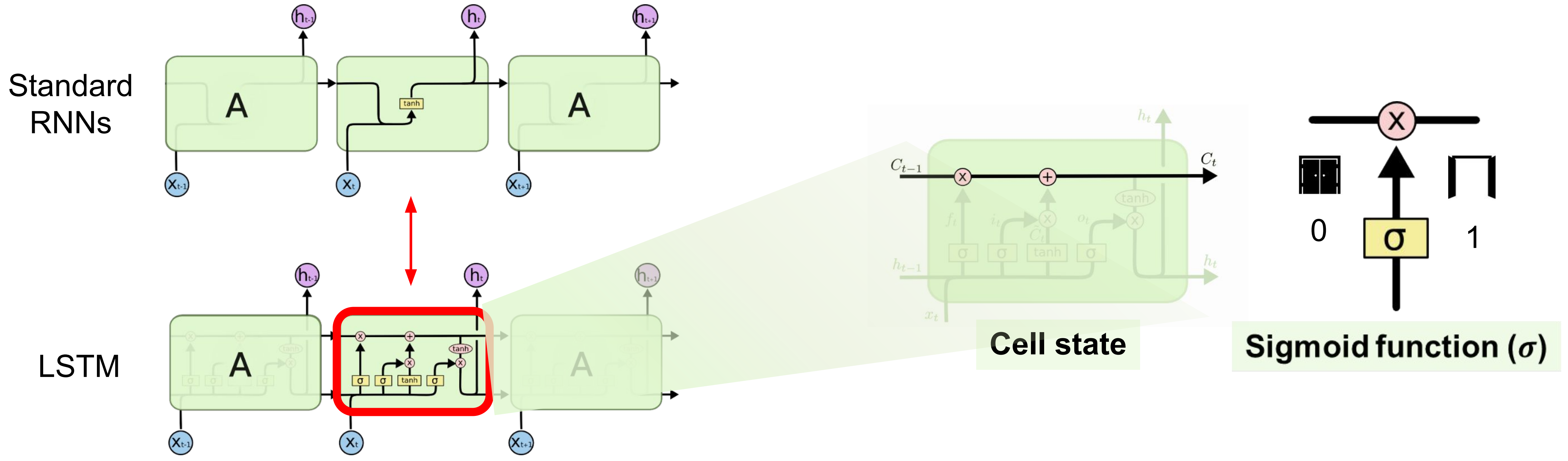
cf) From backpropagation theory: $w_i = w_i + \eta\delta x_i$

Error term : $\delta_0^3 = \frac{\delta E_3}{\delta h_3} \cdot \frac{\delta h_3}{\delta h_2} \cdot \frac{\delta h_2}{\delta h_1} \cdot \frac{\delta h_1}{\delta h_0}$

$0.9 \cdot 0.8 \cdot 0.2 \cdots \approx 0$

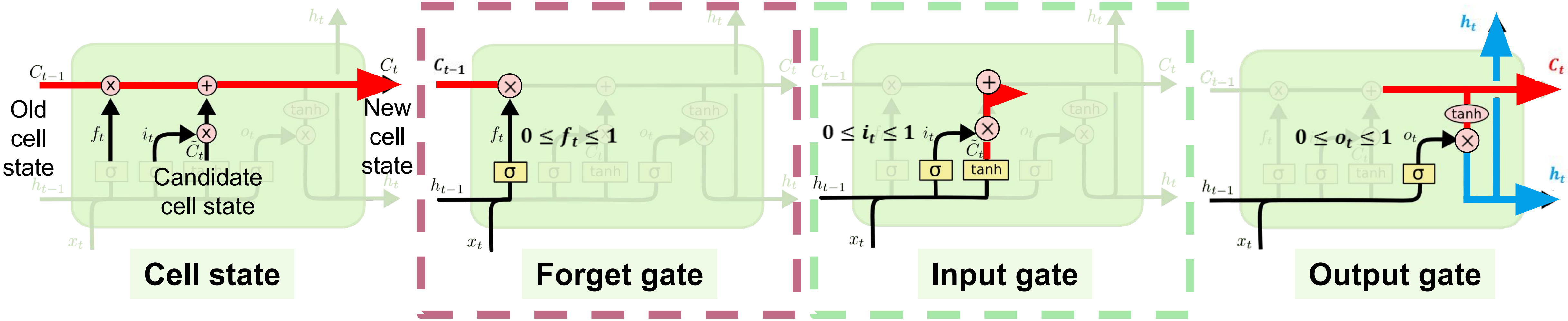
Long short term memory networks (I)

- **The core idea behind LSTM (Long Short Term Memory network)**
  - To relieve the vanishing gradient problem, LSTM uses cell state and gate(sigmoid function)
  - Cell state is a conveyor belt carrying information.
  - Gates(sigmoid function) determine to remove or add information to the cell state
    - Output of sigmoid function is from 0 to 1. It is used as gate (0: let nothing through, 1: let everything through)



LSTM process

- Forget gate(f_t) decides whether to keep old cell state(C_{t-1}) or not
- Input gate(i_t) determines how much candidate cell state($\tilde{C}_t$) is added to new cell state(C_t)
- Output gate(o_t) is multiplied by the cell state, allowing us to get output as we want



$$C_t = \boxed{C_{t-1} * f_t} + \boxed{\tilde{C}_t * i_t} \quad 0 \le f_t, i_t, o_t \le 1$$

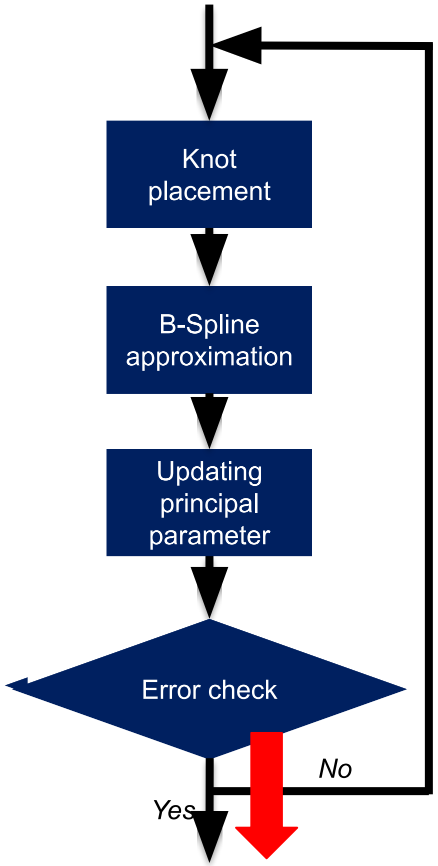
$$h_t = o_t * \tanh(C_t)$$

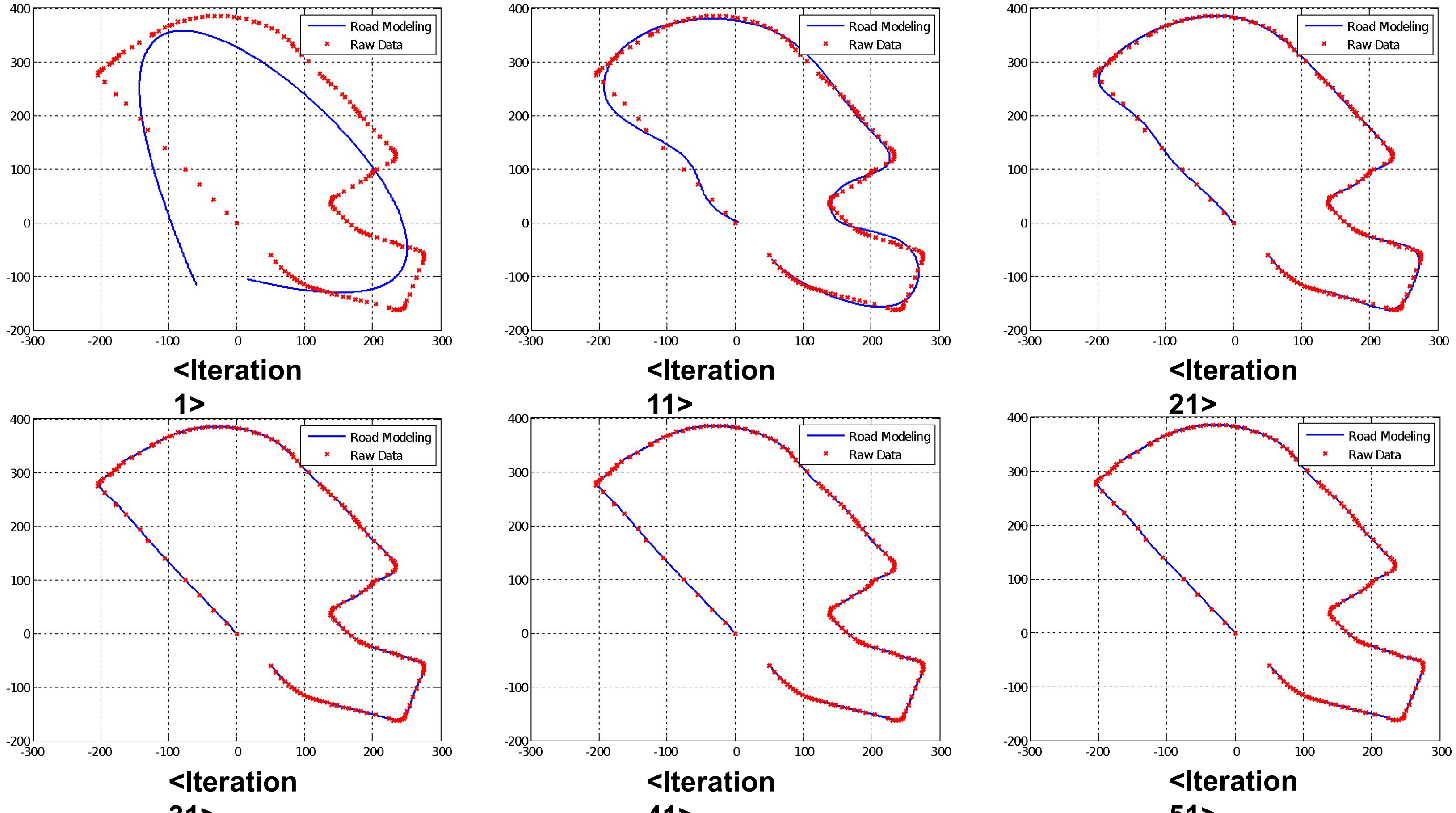
C_{t-1}: *Old cell state*

$\tilde{C}_t$: *Candidate cell state*

C_t: *New cell state*

- **Design the roadway model by using the B-spline***
  - Gradual correction
    - Repeated steps for B-spline creation and error correction



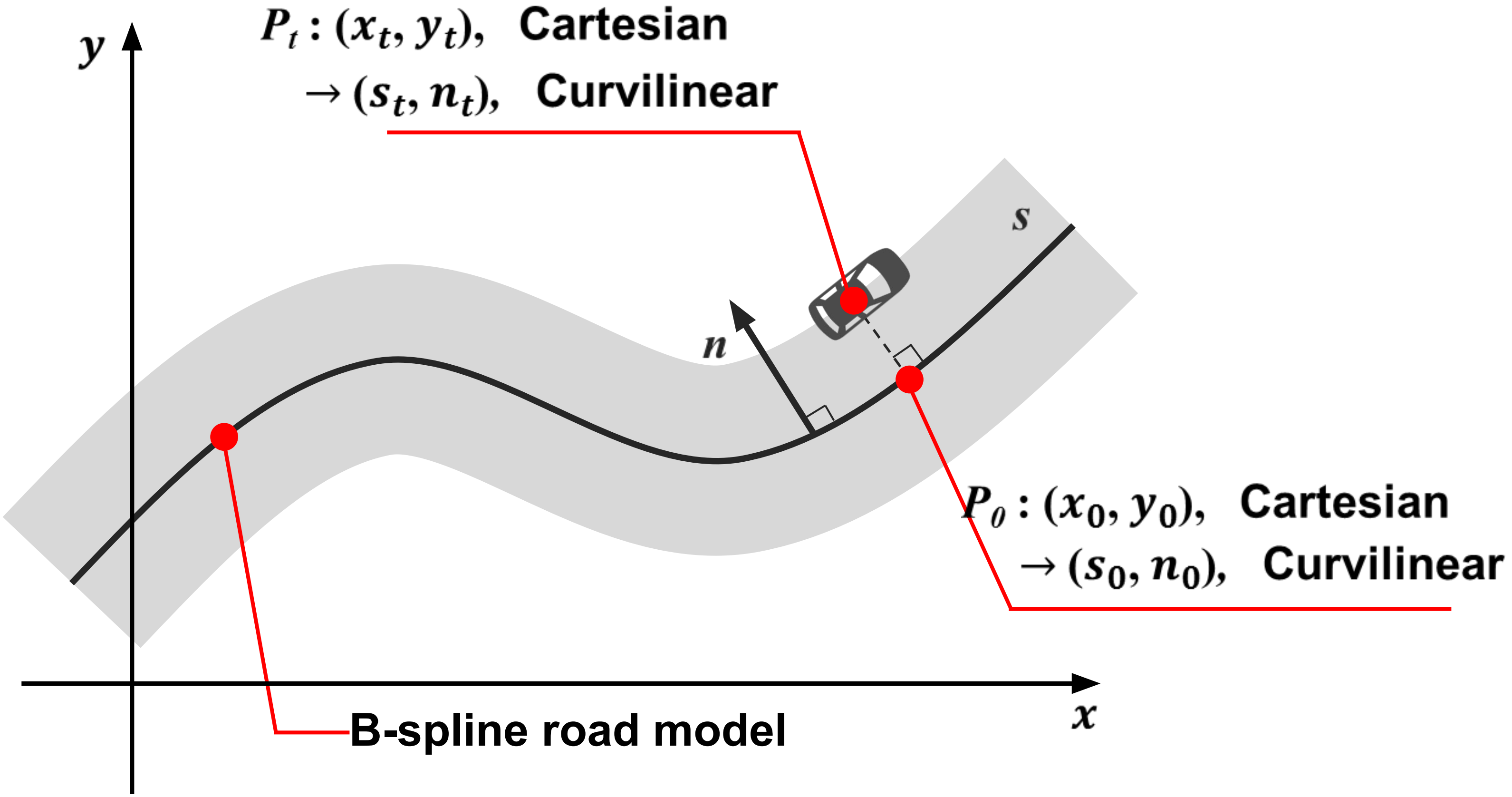


* K. Jo and M. Sunwoo, "Generation of a Precise Roadway Map for Autonomous Cars," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 3, pp. 1–13, 2013.

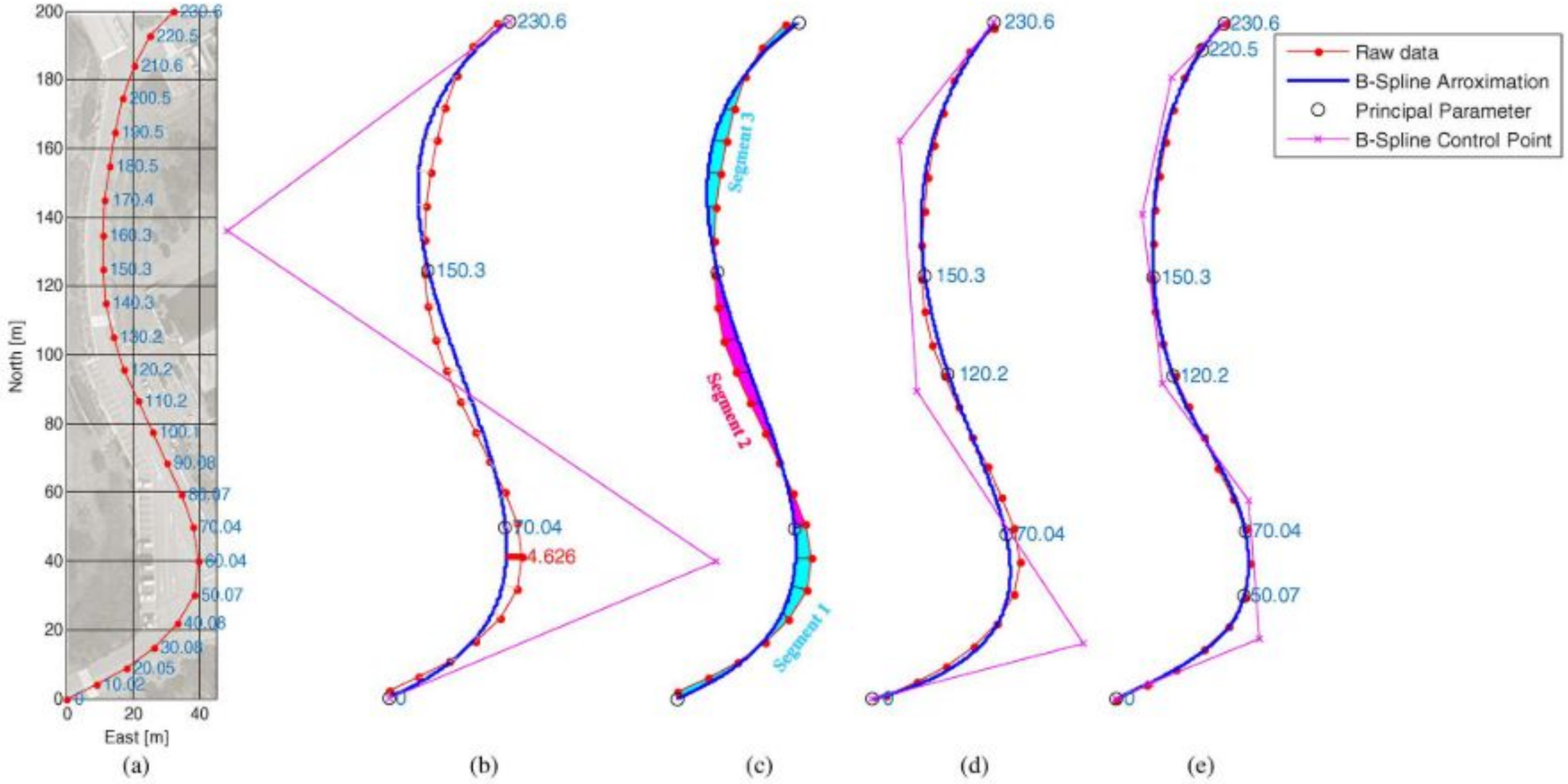
Ego-vehicle location on the route

Curvilinear coordinate system*

- Quadratic minimization
  - Initial guess of the P_0 by minimizing the distance between P_t and P_0
- Newton's method
  - Finding the final solution



* J. Kim, K. Jo, W. Lim, M. Lee, and M. Sunwoo, "Curvilinear-coordinate-based object and situation assessment for highly automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1559–1575, 2015


North [m]
East [m]
Raw data
B-Spline Arroximation
Principal Parameter
B-Spline Control Point
230.6
150.3
120.2
70.04
50.07
4.626
Segment 1
Segment 2
Segment 3
(a)
(b)
(c)
(d)
(e)