32nd Electric Vehicle Symposium (EVS32)
Lyon, France, May 19 - 22, 2019

Smart Regenerative Control Based on Reinforcement Learning Algorithm to Reflect Individual Driver Characteristics

Kyunghan Min[1)], Gyubin Sim[1)], Seongju Ahn[1)], Inseok Park[2)], Jeamyoung Yeon[2)], Myoungho Sunwoo[1)]

[1]*Department of Automotive Engineering, Hanyang University, Seoul, Korea, kyunghah.min@gmail.com*

[2]*Research & Development Division, Hyundai Motor Company, Hwaseong, Korea*

Summary

A smart regeneration system of electric vehicles automatically controls the regeneration torque of the electric motor to brake the vehicle by recognizing the deceleration conditions. In order to apply this assistance system without a sense of heterogeneity by autonomous braking, we propose a control management agent who determines the reference deceleration profile to reflect the individual driver characteristics. The agent is designed based on the reinforcement learning method to maximize future rewards to obtain the optimal value for the various driving condition. The proposed control algorithm is implemented in a python environment using various driving data.

Keywords: autonomous, EV, control system, modeling

1. Introduction

Intelligent transfer system dramatically affects advanced driver assistance systems (ADAS) which enhances not only driver convenience but also energy efficiency. For the ADAS implementation on the vehicle such as energy management of hybrid vehicles and electric vehicle, smart-cruise control system, autonomous emergency braking, the forecasting information such as predicted vehicle velocity has been widely used for the efficient operation [1]–[4]. As one of ADAS applications, a smart regeneration system of electric vehicles uses the forecasting information at braking situations [5], [6]. This system automatically controls the regeneration torque of the electric motor to brake the vehicle by recognizing the deceleration conditions. Thus, it leads the driver convenience due to excluding a driver's physical braking. In addition, avoidance of frequent braking through regeneration torque control harnesses energy that dissipates through a brake disk or drum.

To attain both convenience and energy efficiency through a regeneration torque control, the control system requires accurate prediction of vehicle deceleration profile for diverse deceleration conditions such as the stop condition in front of the traffic light, decelerating before the curvature load, speed limit or the condition when the preceding vehicle is decelerating. Furthermore, the driver can feel the heterogeneity because the automatic braking control is difficult to apply the driving style of the individual drivers [4], [6], [7].

In general, the smart regenerative control algorithm is applied as a rule-based control algorithm which adjusts the regeneration torque level according to the deceleration conditions [8]. However, this adjustable rule-based control algorithm cannot consider diverse deceleration conditions in urban driving. Furthermore, it is more difficult to consider the heterogeneity of individual drivers into account. To cope with these issues efficiently, we proposed a torque control management algorithm based on the reinforcement learning method because it can guarantee the real-time and robust performance with optimization of the control performance [9]. This torque control algorithm determines the weight factor for the desired deceleration value of the regenerative control. The desired deceleration value is generated based on both the individual model and general cruise control algorithm. Thus, we can obtain the safety regenerative control target by reflecting individual driver characteristics.

Paper is organized as the following. At chapter 2, the algorithm overview is introduced. The proposed algorithm consists of simulation models and torque control management. Chapter 3 describes simulation models that are vehicle model, battery model

2. Algorithm overview

Figure 1 shows the algorithm overview which contains the control algorithm, vehicle model and driver model. The control algorithm determines the regenerative torque according to the driving condition. When the driver takes his foot from the acceleration pedal, the algorithm recognizes the coasting conditions. Then, the control algorithm generates the regenerative torque of electric vehicle to decelerate the vehicle without the driver's brake pedal control. The vehicle model simulates the vehicle deceleration operation by the regenerative torque and the state of charge (SOC) of the battery. On the other hand, the preceding vehicle behaviour and the vehicle operation on acceleration conditions are demonstrated using the measured driving data.

In order to reflect the individual driver's characteristics, we apply the parametric driver model. The parametric driver model predicts the deceleration profile using the mathematical equation and model parameters. The model parameters are updated using the driving data of individual driver to represent the driver characteristics. The driver model is described in other papers about our previous researches. The control management agent determines the weight factor between the predicted deceleration profile and the general deceleration setpoint. This cruise control algorithm based on the linear quadratic regulation controller generates the general deceleration set-point. Consequently, the proposed control management agent can consider the driver characteristics while decelerating autonomously.

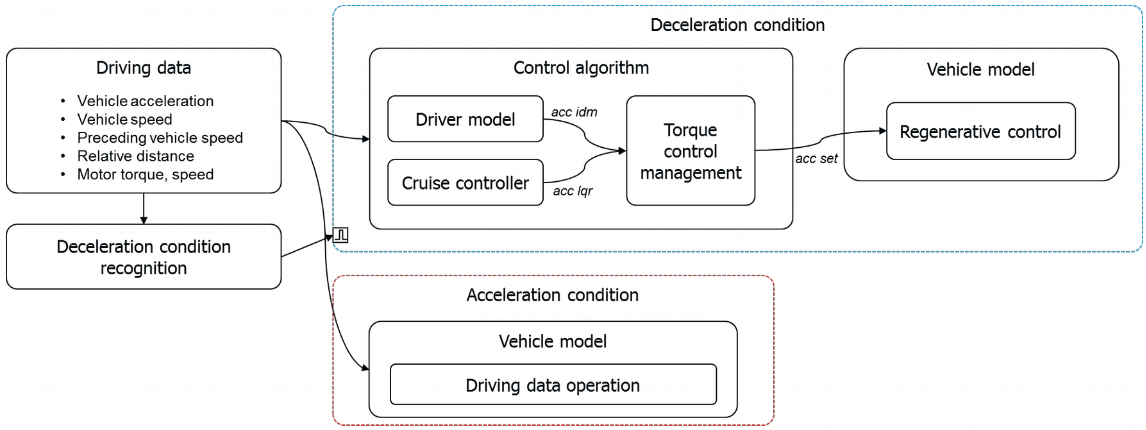


Figure 1Algorithm overview

3. Simulation environment

The algorithm is designed on a python environment because the python environment easily approaches to many deep learning libraries. In the python environment, we designed the electric vehicle model with a battery model and the intelligent driver model to simulate and learn the agent. The control management agent acts to maximize future rewards. To predict the future reward, the agent updates the reward function which is designed as the deep neural network.

3.1 Vehicle model

3.1.1 Model description

The vehicle model consists of the power source module, drive train module and vehicle module as shown in Figure 2. The vehicle model only simulates the longitudinal dynamics because the proposed smart regenerative control assists only the driver's braking behavior. We assume that the driver controls the acceleration pedal and steering.

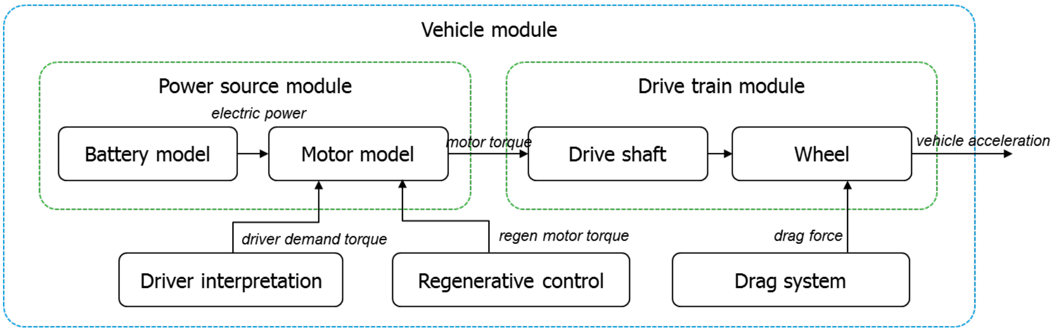


Figure 2 Vehicle model

The longitudinal vehicle behavior is modeled as the first order dynamic model based on Newton's second law as Equation (1) ~ (2). According to the dynamics, vehicle acceleration is calculated using the fraction force from the power source and drag force from the vehicle model. The power source generates the motor torque using the battery power. The drive train transfers the motor torque with the deceleration gear ratio, then, the traction force to tire wheel is determined. The drag force is calculated as the sum of the longitudinal resistances as Equation (3).

$$a_v = \left(\theta_s T_m \eta_s / r_w - F_d\right) / m_v \tag{1}$$

$$m_v = m_e + m_a + 4 I_w + \theta I_m + I_s \tag{2}$$

$$F_d = \left(0.75 c_d v_v^2 + c_a + c_b v_v^2\right) \tag{3}$$

The battery model is designed based on the well-known Chen model which is introduced in the paper [10]. This model can represent the battery characteristics especially according to the transient conditions. The model contains two RC circuit those represent long-term transient and short-term transient characteristics respectively as shown in Figure 3. Each electric device value is determined as the exponential equation with three model parameters depending on the battery SOC as Equation (4).

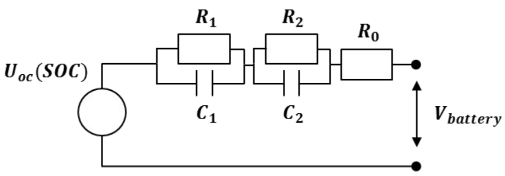


Figure 3 Chen battery model

$$R_i(SOC) = r_{ia} e^{-r_{ib} SOC} + r_{ic} \tag{4}$$

3.1.2 Parameter identification

The designed vehicle model can be configured as accustoming the model parameters such as the battery register, driver train inertia, vehicle air coefficient or vehicle mass. In order to simulate the vehicle operation as similarly with the real experiment vehicle, the model parameters are determined through the parameter identification process using the real vehicle driving data. To secure the driving data, KONA electric vehicle of Hyundai Motor Company is used. The vehicle experiments were conducted in various driving cases. The driving cases contain the car-following situation on the straight load, urban and highway driving and the uphill driving. Battery parameters are also identified using the driving data on various battery SOC range.

All parameters are identified based on the nonlinear least-square solver with the trust-region-reflective algorithm [11], [12]. Figure 4 shows the modeling results of vehicle and battery, respectively. As shown in the figure, the vehicle and battery model well represent the dynamic characteristics of the vehicle to use for longitudinal regenerative torque control. Table 1 and Table 2 describe parameter values.

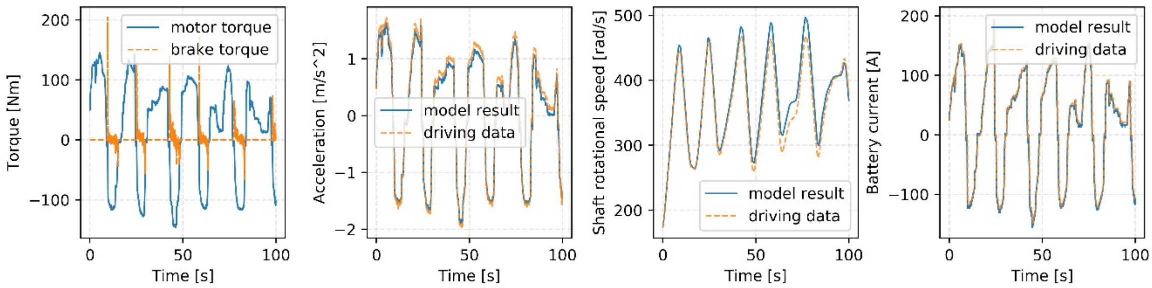


Figure 4 Vehicle modelling results

Table 1 Vehicle model parameters

	Description	Value [unit]		Description	Value [unit]
a_v	Vehicle acceleration	[m/s^2]	I_w	Inertia of wheel	0.14 [khm^2]
v_v	Vehicle velocity	[m/s]	I_m	Inertia of motor	0.028 [khm^2]
T_m	Motor torque	[Nm]	I_s	Inertia of shaft	0.75 [khm^2]
F_d	Drag force	[N]	c_d	Air drag coefficient	0.171 [Ns^2/m^2]
r_w	Wheel radius	0.318 [m]	c_a	Rolling coefficient	143 [N]
θ_s	Gear ratio of shaft	7.98 [-]	c_b	Rolling coefficient	0.389 [Ns^2/m^2]
η_s	Efficiency of shaft	0.99 [-]	m_a	Additional mass	100 [kg]
m_e	Empty vehicle mass	1685 [kg]			

Table 2 Battery model parameters

	Description	Value [unit]		Description	Value [unit]
R_0	Series register	[Ohm]	C_{1a}	Short capacitor param a	-649
R_{1a}	Short register param a	76.52	C_{1b}	Short capacitor param b	-64.3
R_{1b}	Short register param b	-7.95	C_{1c}	Short capacitor param c	12692
R_{1c}	Short register param c	23.83	C_{2a}	Long capacitor param a	-78409
R_{2a}	Long register param a	5.21	C_{2b}	Long capacitor param b	-0.013
R_{2b}	Long register param b	-35.23	C_{2c}	Long capacitor param c	30802
R_{2c}	Long register param c	124.9	V_{oc}	Open circuit voltage	356 [V]

3.1.3 Regenerative torque control

To regenerative control, we determine the regenerative motor torque. This regenerative motor torque is applied to the minus torque to the motor model in power source. Then this minus torque leads both the deceleration of the vehicle and the energy charge of the battery. Figure 5 shows the result of regenerative control using the simulation models.

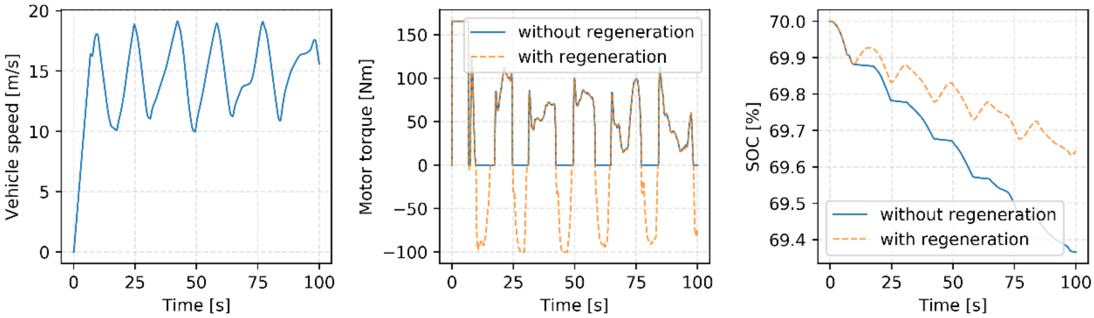


Figure 5 Regenerative control of vehicle model

3.2 Parametric deceleration model

3.2.1 Description of the prediction process

The parametric deceleration model was introduced in our previous research [13]. This model is designed based on the well-known intelligent driver model [3]. However, the model parameters of the existing intelligent driver model are designed as the parametric equations to represent the vehicle driving state under deceleration conditions. Furthermore, the driver model parameter affects the parametric equation to reflect the individual driver characteristics. Consequently, since the deceleration profile is determined as the parametric equation, the model parameter can reflect the individual driver characteristics to the deceleration prediction. The model parameters are described in Figure 6. When the driver decelerates, the braking timing, initial jerk, and specific acceleration values especially represent the driver characteristics. Thus, we determined these physical values as the model parameters. Also, when the deceleration is terminated, the termination relative velocity to the preceding vehicle also represents the driver characteristics.

Equation (5) describes the parametric deceleration model. The reference velocity(v_{ref}) and effective distance(d_{eff}) are designed as the parametric equations and the, parametric equations are determined according to the braking section and model parameters as shown in Figure 7. The braking section is a definition of braking interval meaning the deceleration characteristics. Its period is determined also by the driver model parameters. Each braking section means as follows. The coasting section means the driver's pedal shifting time from the acceleration pedal to the brake pedal. When the initial section, the driver pushes the brake pedal until the vehicle deceleration reaches to a specific value. The acceleration slope on the initial section is determined as the initial jerk parameter. After initial the section, the driver adjusts the brake pedal to converge to velocity condition for the preceding vehicle. Then, the driver controls the brake pedal to keep a safety distance. Using these braking sections, the model can represent the deceleration characteristics in detail.

$$a = a_m \left(1 - \left(\frac{v}{v_{ref}} \right)^{\delta} - \left(\frac{d}{d_{eff}} \right)^{2} \right) \tag{5}$$

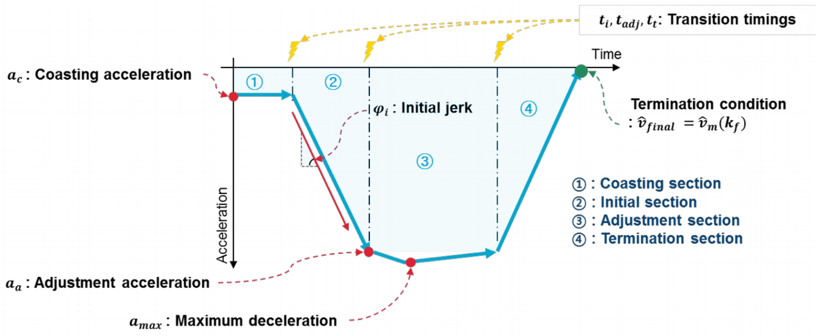


Figure 6 Driver model parameters and deceleration profile

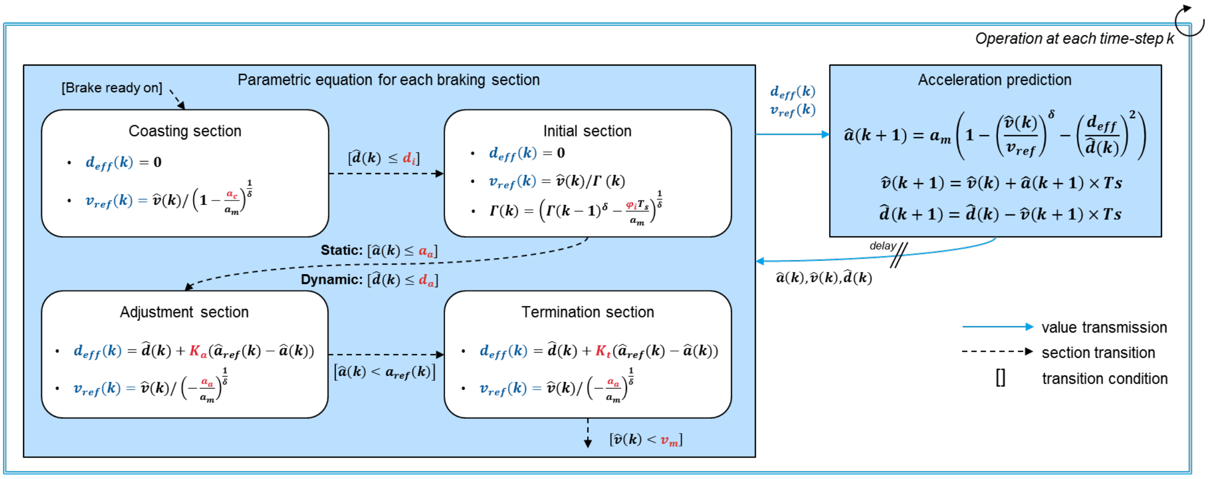


Figure 7 Model prediction process

3.2.2 Parameter learning algorithm

As mentioned in the introduction, the model parameter should be updated according to the driving data of each driver on the real-time driving condition. Thus, the learning algorithm which can be applied to the real-time embedded system is designed. The vector arrays for each model parameter are pre-determined for embedded suitable update algorithm. At first, the initial values of parameter vector are defined using the whole driving data of every driver. Second, the reference parameter values are calculated using the driving data of individual driver when every time a deceleration occurs. Then, using the reference parameter, the vector array of each parameter is updated. Consequently, the model parameters are updated for each deceleration driving according to individual driver characteristics. This update algorithm should determine the vector index for the model parameter vector to the vector index has a high correlation to its parameter value.

Figure 8 shows the parameter values and its index values for each parameter. As shown in the figure, the parameter initial jerk is correlated to the index parameter which means the initial vehicle states. It describes, when deceleration starts, the driver pushes the brake pedal more aggressively as the relative distance is small. The relative distance parameters are correlated of the relative distance at the previous braking section. The driver intends to keep the same time to collision of the previous braking section. Figure 8 also shows the base vector which is an initial value of the parameter vector and the learning results of three drivers. The result describes the driving characteristics; then, each updated parameter vector is used to agent learning.

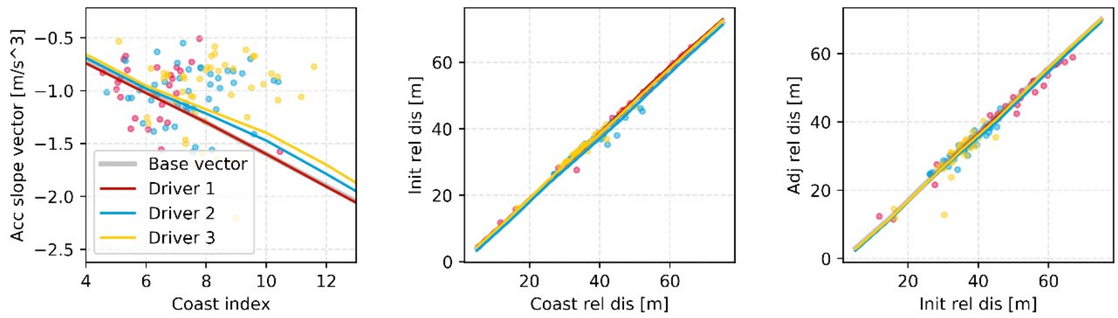


Figure 8 Model parameters and its updated vector for each driver

3.2.3 Acceleration set-point generation based on the driver model

Through the above parametric deceleration model of each driver, the predicted acceleration when the deceleration condition is determined. Figure 9 shows the prediction results for each driver; then we used this predicted acceleration profile as the acceleration set-point reflecting driver characteristics.

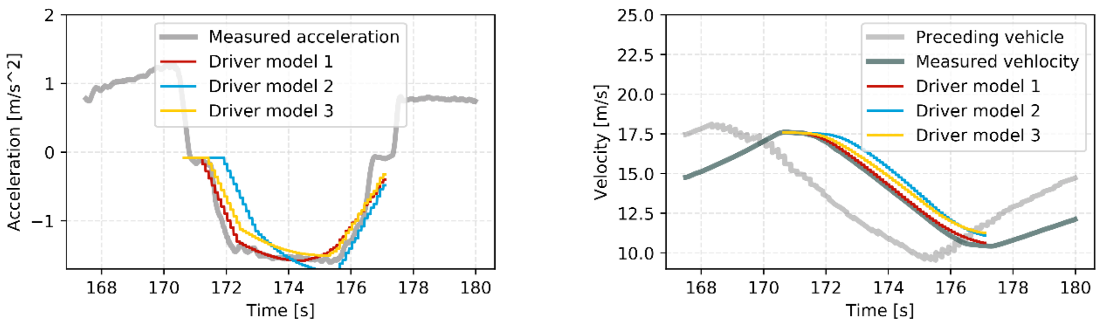


Figure 9 Deceleration states prediction results according to the individual driver

3.3 General cruise control algorithm

In the contrary, the general cruise control algorithm which determines the acceleration profile for cruise control is designed based on the linear quadratic regulation algorithm [14], [15]. The system model is designed with two regulated states that are relative velocity and safe distance. The relative velocity means the velocity difference between the preceding vehicle and ego vehicle. The safety distance state is a distance that should be controlled in a cruise situation. This parameter is determined depending on the preceding vehicle velocity and safety time gap. The cruise control algorithm generates the acceleration set-point to regulate these two states to keep the safety distance and control vehicle velocity to preceding vehicle velocity.

Figure 10 shows the cruise control results when the same deceleration case. When the deceleration starts, the cruise control algorithm generates large acceleration set-point value in comparing to the acceleration set-point from the driver model. Since the cruise control algorithm regulates states, it generates large control result according to initial state values when deceleration starts. However, the driver model considers the pedal shift time of human behavior when deceleration starts. As a result, the acceleration set-point from the driver model is small when the initial deceleration condition. In the other hand, generated acceleration set-point values show the opposite tendency when braking ends. The cruise control generates small deceleration value because the vehicle has already decelerated sufficiently. To determine the merged acceleration set-point, we designed the torque control management that defines the weight factor between these two set-points.

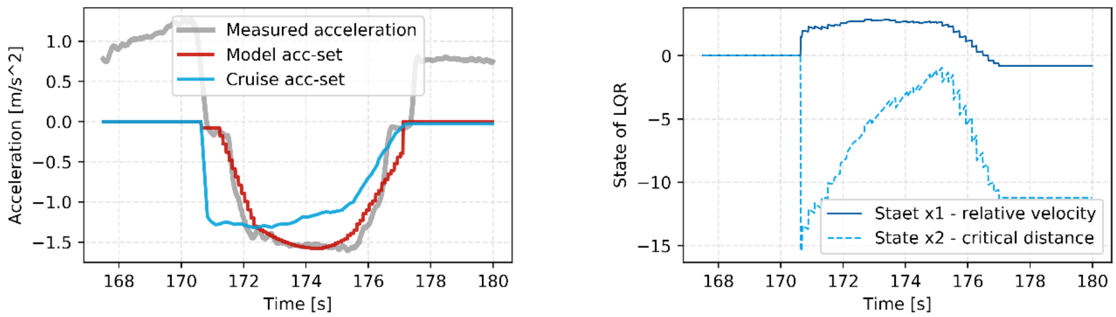


Figure 10 Cruise control results based on LQR algorithm

4. Torque control management

4.1 Overview of the reinforcement learning algorithm

We propose a torque control management that determines the weight factor of acceleration set-point of the torque controller. The algorithm should select the optimal weight factor between set-point from the deceleration model and cruise control set-point. However, it is difficult to determine the optimal value because the parametric model and cruise controller encounters diverse driving conditions. Thus, the model is designed based on a reinforcement learning algorithm since the reinforcement learning algorithm can handle the nonlinear and complex problem as learning by itself [16]–[18].

The reinforcement learning algorithm determines the optimal action to maximize future reward. Figure 11 describes the concept of reinforcement learning. As shown in the figure, the algorithm consists of an agent and an environment. The algorithm determines the optimal action based on the Markov Decision Process (MDP) by the interaction of the agent and environment. MDP deals with bellow features $[S, A, R, \Pi]$. S is a state that represent the current environment. A is the action of the agent. R is a reward function of the environment. The agent selects the action to makes the environment gives a maximum reward at the current state. Then, the state of the environment is changed according to the action and transition probability model Π. The environment generates the reward for each state and action.

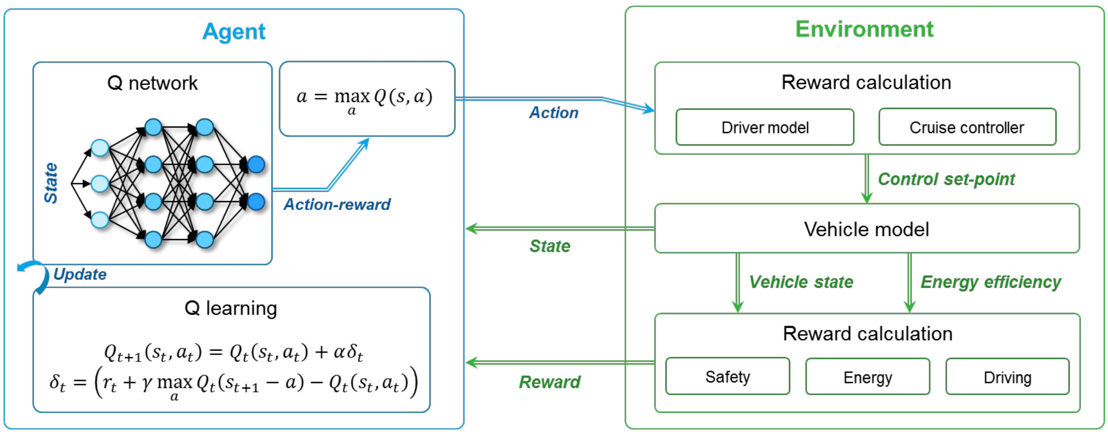


Figure 11 Reinforcement learning for torque control management

In order to determine the future reward according to the current state, the reinforcement learning algorithm defines the value function about the current state. As Equation (6), the value function $V(s)$ for each state means the expected sum of the reward. To focus on the current reward, the discounted factor γ is applied. The value function of the current states can be reformulated as a recursion expression by Bellman theory. As a result, the optimal policy of the agent is a selection of an action to maximize the value function at the current state.

The action-value function Q is defined as Equation (7). While the value function means the future reward at current state S, the action-value function q means the future reward by taking action A at current state S. Thus, the Q learning algorithm that optimal policy selects the maximum q action, is widely used. Since the determination of Q value function for all states and all action is impossible in various driving condition and control actions, the Q value function can be determined as a deep neural network. On the learning situation, the reinforcement learning algorithm updates this Q network using the simulation data. The learning algorithm and Q network are described in detail in the following chapter.

$$V(S)=E\left(R_{t+1}+R_{t+2}+R_{t+3}+\ldots+R_{end}\right) \tag{6}$$

$$Q(S,A)=E\left(R_{t+1}+\gamma Q\left(s_{t+1},A_{t+1}\right)\right) \tag{7}$$

4.2 Reinforcement learning algorithm application

On the proposed algorithm, the vehicle states, preceding vehicle speed and deceleration model states are defined as the state of the environment. The vehicle state and preceding vehicle speed are importance states in cruise control situations. Furthermore, the model states of parametric deceleration model are also defined due to reflect the individual driving characteristics. According to the states and vehicle simulation results, the environment model generates the reward. Since the agent selects the action to maximize the reward, the affordable definition of the reward function can determine the performance of the algorithm.

To consider safety, comfort and energy, we determine the three reward functions. The safety reward function prevents the vehicle collision. This reward function generates the high penalty value when the relative distance between vehicles is minus value. In addition, the relative distance is smaller than the criterion value, the reward function generates the normal penalty values to prepare for the collision. The comport reward function is related to the driver characteristics and driving data. The error value between the real-diving data and control result is defined as penalty values. To reduce this penalty, the agent selects the action as like human driving data. The comfort reward function also uses the predicted value from the parametric model of each driver because the driving data that is used to learning can only reflect that measured driving situations. By using the parametric driver model, the agent can consider the general driving characteristics of individual driver even in other driving cases. The last reward function is an energy regeneration reward function. The increasement of battery SOC by the regenerative control is determined as the positive reward of algorithm.

4.2 Q network design and learning algorithm

To estimate the optimal action value, the Q network is designed based on the deep neural network. At first, we proposed a sequential deep neural network. Since the sequential network is advantage to the time sequential data, it is suitable for torque control application. In addition, since the deceleration characteristics of the driver are affected by the braking section which is time dominant period, the proposed q network is effective.

The recurrent neural network with a long short-term memory is used to the sequence neural network for Q value approximation. The RNN is a representative sequence network because it takes the sequence input and predicts the sequential output. Using the hidden network, RNN extends the conventional feedforward neural network to handle the time sequential information. However, the RNN model has the vanishing problem and blowing up gradient cause the long-term dependency problem. LSTM architecture is introduced to solve this problem. The LSTM includes the memory cells in the hidden layer. This memory cell predicts the hidden state as like RNN, however, the cell state and gated structure can solve the limitation of RNN. The gate structure consists of the input gate, the output gate and the forget gate. The equations and Figure 12 describe the LSTM algorithm.

$$h_t=\sigma_h\left(W_{xh}x_t+W_{hh}h_{t-1}+b_h\right) \tag{8}$$

$$i_t=\sigma_g\left(W_{ig}\left[h_{t-1},x_t\right]+b_{ig}\right) \tag{9}$$

$$f_t=\sigma_g\left(W_{fg}\left[h_{t-1},x_t\right]+b_{fg}\right) \tag{10}$$

$$o_t = \sigma_g\left(W_{og}\left[h_{t-1}, x_t\right] + b_{og}\right) \tag{11}$$

$$\tilde{c}_t = \sigma_r\left(W_{cg}\left[h_{t-1}, x_t\right] + b_{cg}\right) \tag{12}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \tag{13}$$

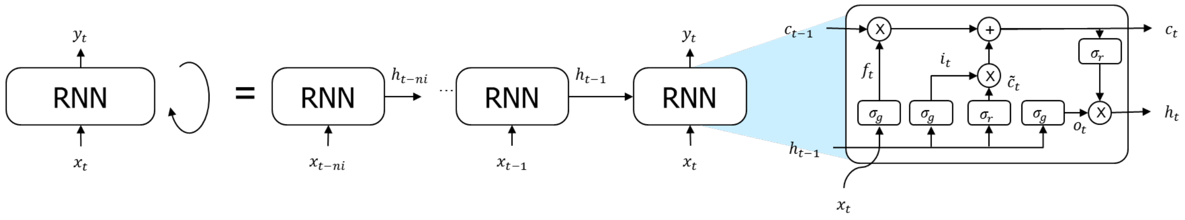


Figure 12 RNN architecture with LSTM cell

The Q networks are determined based on the sequential neural network, and they are updated through the Q learning algorithm with temporal difference learning method [19]. The Q learning updates the network after taking action A of state S. In this time, the environment generates the immediate reward and state transition. Using these action, state and reward, the action-reward value is calculated as Equation (7). Then, using this action-reward value, the target value for Q network parameter update is determined as Equation (14) ~ (15).

$$\theta_{t+1} = \theta_t + \alpha\left(Y_t^Q - Q\left(S_t, A_t; \theta_t\right)\right)\nabla_{\theta_t} Q\left(S_t, A_t; \theta_t\right) \tag{14}$$

$$Y_t^Q = R_{t+1} + \gamma \max_a Q\left(S_{t+1}, a; \theta_t\right) \tag{15}$$

5. Control results

Figure 13 shows learning results for one deceleration case. The left top figure is an acceleration of the vehicle. The driver model generates the model acceleration set-point which is a blue line, and the cruise controller generates the cruise acceleration set-point which is a yellow line. The red line is a control result by the merged acceleration set-point. The torque control management determines the merging ratio between the model and the cruise controller by selecting the action index. As shown in the left bottom figure, the smaller action index value is, the more acceleration set-point from the model is used. This system is learned iteratively to maximize the Q value as shown in the right bottom figure.

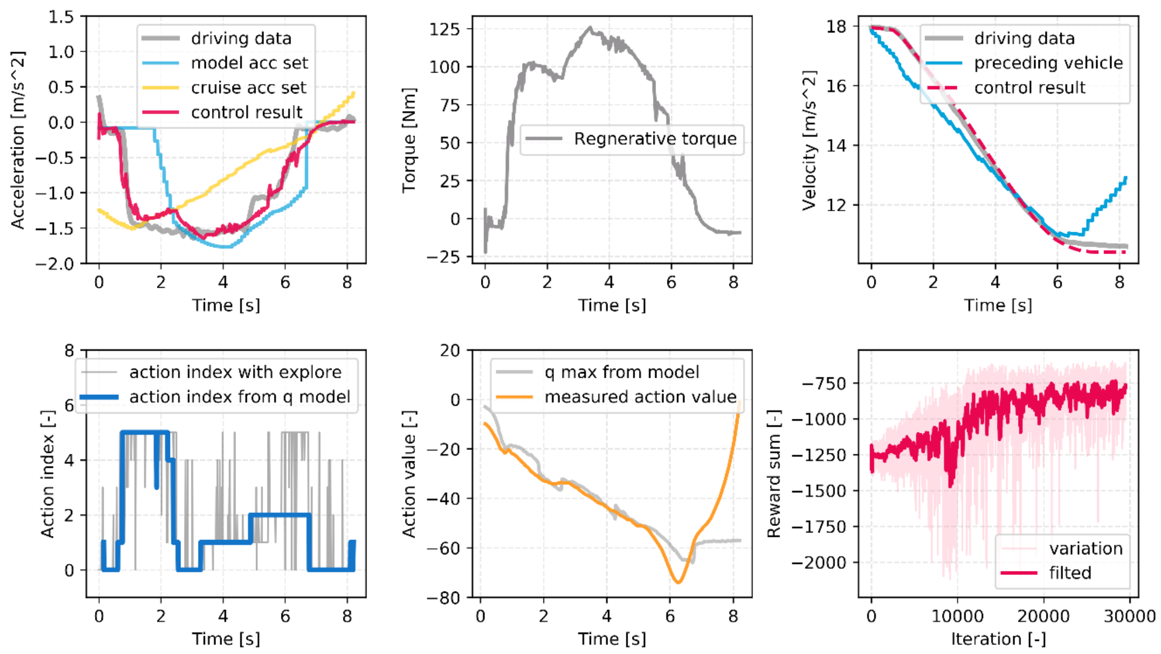


Figure 13 Learning and Regenerative control results for deceleration case

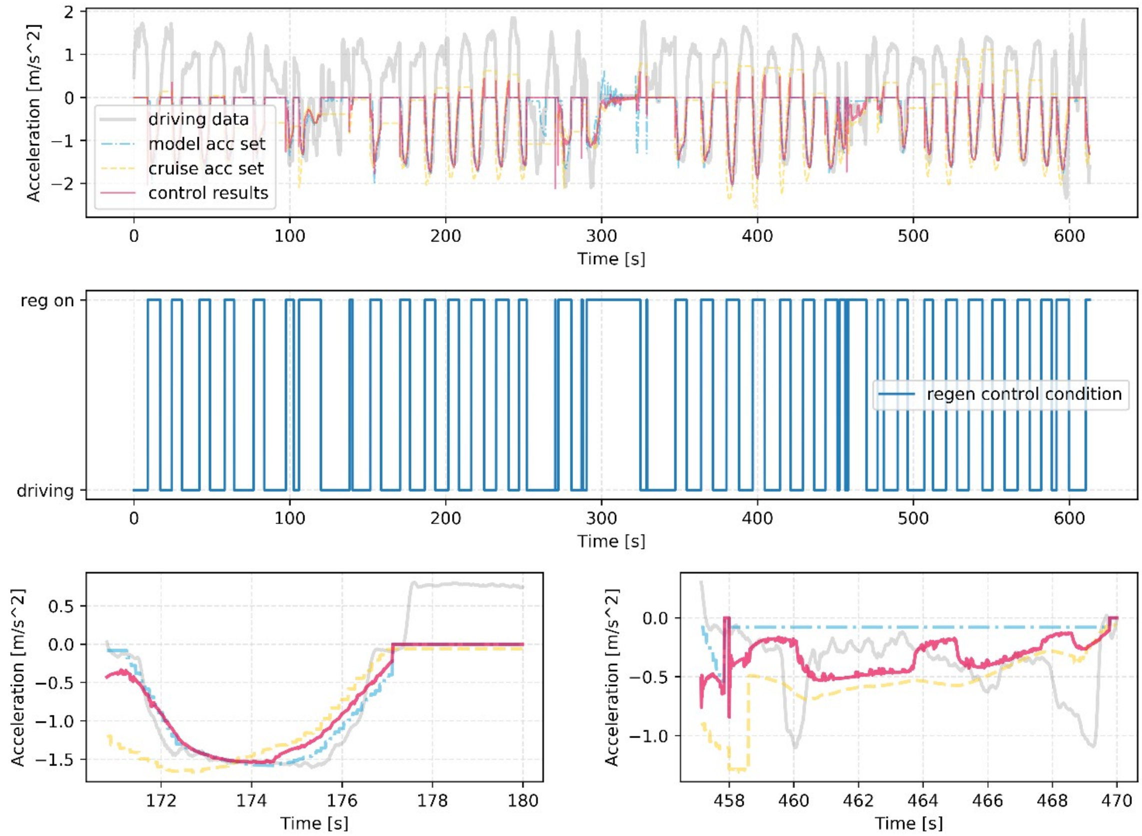


Figure 14 Regeneration control results using driving data

Figure 14 shows the control results for the various driving case. The proposed algorithm conducts the regenerative control when the driver releases the foot from the acceleration pedal. In each deceleration condition, the driver model and cruise control algorithm generates acceleration set-point, then the control management selects the merging ratio. As a control results about 170 seconds, the control management select the model-based acceleration set-point dominantly because this case the model well predicts the driver's behavior. On the other hand, the algorithm selects the cruise control set-point since the model does not work about 460 seconds.

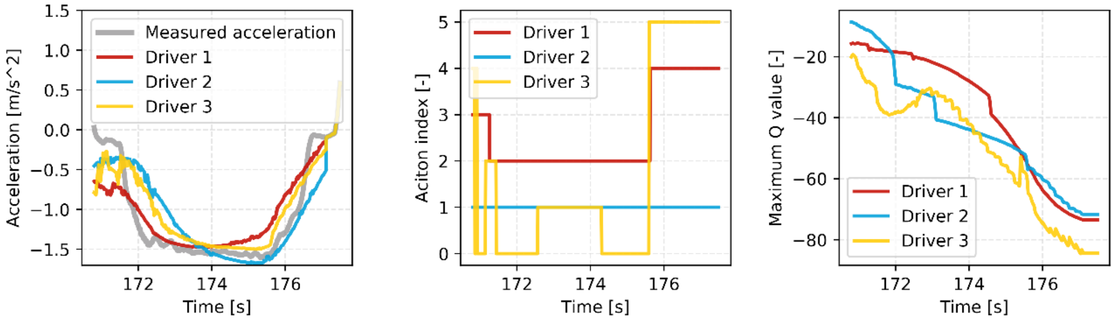


Figure 15 Regenerative control results depending on individual driver

Figure 15 shows the different control results according to individual driver model. To reflect the driver characteristics, the agent is learned using different driver parameters. As a result, the control results and action index of control management are different for the same deceleration condition.

Conclusion

In this paper, we proposed the smart regenerative control management based on the reinforcement learning algorithm to reflect the driver characteristics. The proposed algorithm recognizes the deceleration condition; then, the driver model generates the acceleration set-point. Since the driver model can represent the individual driver characteristics for deceleration behaving, the control algorithm controls the regenerative torque to trace the model generated set-point for reflecting the driver characteristics. Furthermore, the cruise controller generates acceleration set-point based on the LQR algorithm to prepare inaccurate prediction of the model. The control management is updated based on the Q learning algorithm to select the optimal merging ratio of acceleration set-point between the model and cruise controller. The learning algorithm is simulated using the vehicle model. Control results show the proposed algorithm selects optimal merging ratio with reflecting the driver characteristics.

Acknowledgments

This researched was financially supported by Hyundai Motor Company and the BK21 plus program (22A20130000045) under the Ministry of Education, Republic of Korea.

References

[1] M. Demir and A. Çavus, “A new driver behavior model to create realistic urban traffic environment,” 2012.

[2] Y. Qi, C. Xiang, W. Wang, B. Wen, and F. Ding, “MODEL PREDICTIVE COORDINATED CONTROL FOR DUAL-MODE POWER-SPLIT HYBRID ELECTRIC VEHICLE,” *Int J Automot Technol*, vol. 19, no. 2, pp. 345–358, Apr. 2018.

[3] R. Malinauskas, “The Intelligent Driver Model : Analysis and Application to Adaptive Cruise Control The Intelligent Driver Model : Analysis and Application,” 2014.

[4] 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, Apr. 2014.

[5] B. Balasubramanian and A. C. Huzefa, “Development of regeneration braking model for electric vehicle range improvement,” in *2017 IEEE Transportation Electrification Conference (ITEC-India)*, 2017, pp. 1–5.

[6] J. C. McCall and M. M. Trivedi, “Driver Behavior and Situation-aware Brake Assistance for Intelligent Vehicles,” *Proc. IEEE*, vol. 95, no. 2, pp. 374–387, 2007.

[7] V. A. Butakov and P. Ioannou, “Personalized Driver Assistance for Signalized Intersections Using V2I Communication,” *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1910–1919, 2016.

[8] “Discover the New Hyundai Kona Electric - Electric SUV | Hyundai UK.” [Online]. Available: https://www.hyundai.co.uk/new-cars/kona-electric. [Accessed: 26-Sep-2018].

[9] R. Xiong, J. Cao, and Q. Yu, “Reinforcement learning-based real-time power management for hybrid energy storage system in the plug-in hybrid electric vehicle,” *Appl. Energy*, vol. 211, pp. 538–548, Feb. 2018.

[10] M. Chen and G. A. Rincon-Mora, “Accurate Electrical Battery Model Capable of Predicting Runtime and I–V Performance,” *IEEE Trans. Energy Convers.*, vol. 21, no. 2, pp. 504–511, Jun. 2006.

[11] T. F. Coleman and Y. Li, “On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds,” *Math. Program.*, vol. 67, no. 1–3, pp. 189–224, Oct. 1994.

[12] T. F. Coleman and Y. Li, “An Interior Trust Region Approach for Nonlinear Minimization Subject to Bounds,” *SIAM J. Optim.*, vol. 6, no. 2, pp. 418–445, May 1996.

[13] K. Min, K. Yeon, G. Sim, and P. M. Sunwoo, “Prediction Algorithm for Decelerating Driving States Based on Driver Characteristics for Smart Regenerative Control of Electric Vehicles,” in *Aachen Colloquium China*, 2018.

[14] M. Chou and X. Xia, “Optimal cruise control of heavy-haul trains equipped with electronically controlled pneumatic brake systems,” *Control Eng. Pract.*, vol. 15, no. 5, pp. 511–519, May 2007.

[15] M. J. Khan, S. Wang, U. Keerio, and N. Rao, “LQR Autonomous Longitudinal Cruise Control with a Minimum Order State Observer,” *Intelligent Systems and Control*. ACTA Press.

[16] H. Chae, C. M. Kang, B. Kim, J. Kim, C. C. Chung, and J. W. Choi, “Autonomous Braking System via Deep Reinforcement Learning,” 2017.

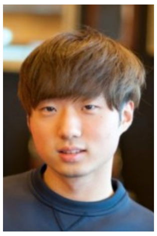
[17] A. Nair *et al.*, “Massively Parallel Methods for Deep Reinforcement Learning,” Jul. 2015.

[18] R. Xiong, J. Cao, and Q. Yu, “Reinforcement learning-based real-time power management for hybrid energy storage system in the plug-in hybrid electric vehicle,” *Appl. Energy*, vol. 211, pp. 538–548, Feb. 2018.

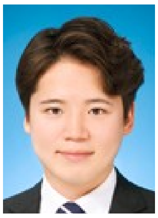
[19] C. J. C. H. Watkins and P. Dayan, “Q-learning,” *Mach. Learn.*, vol. 8, no. 3–4, pp. 279–292, May 1992.

Authors

Kyunghan Min received a B.S. in Mechanical Engineering in 2011, and a Ph.D. degree in Automotive Engineering in 2017 from Hanyang University, Seoul, Korea. His main fields of research are algorithm design of powertrain management system, real-time embedded system for control algorithm. His current research activities include prediction of vehicle states, regenerative braking control for electric vehicles based on driver characteristics.



Gyubin Sim received a B.S. in Automotive Engineering from Hanyang University in 2018, and he is currently pursuing a master's degree in the Department of Automotive Electronic Control at the same university. He has researched the state estimation of the engine and powertrain control of electric vehicles as a member of Automotive Control and Electronics Laboratory. His current research activities include prediction of vehicle states, regenerative braking control and analysis of drivers' characteristics.



Seongju Ahn received a B.S in Automotive Engineering from Hanyang University in 2019, and he is currently pursuing a master's degree in the Department of Automotive Engineering at the same university. He has researched the powertrain control of electric vehicles as a member of Automotive Control and Electronics Laboratory. His current research activities include the state recognition and regenerative braking control of electric vehicles.

Inseok Park received the B.S. degree of Electronics and Computer engineering from Hanyang University in 2005. M.S. and Ph. D. degree of Automotive Engineering from Hanyang University, Seoul, Korea, in 2007 and 2013, respectively. Currently, he is working on a department of eco-vehicle control development of Hyundai Motor Group R&D Center, Korea. His research activities include automotive software engineering and model-based software design. He has also done some work on nonlinear, robust and MIMO controller design for automotive powertrain systems such as spark ignition and turbocharged diesel engines. Currently, he is working on motion control and regenerative coasting strategies of electric vehicles.

Jeamyoung Yeon received the B.S. degree of Mechanical engineering from Hanyang University in 2002. M.S. and Ph. D. degree of Automotive Engineering from Hanyang University, Seoul, Korea, in 2004 and 2008, respectively. Currently, he is working on a department of eco-vehicle control development of Hyundai Motor Group R&D Center, Korea. His research activities include automotive software engineering and model-based software design. He has also done some work on holistic optimization strategy for ECU integration. Currently, he is working on a control system architecture for electric vehicles and many others.

Myoungho Sunwoo received a B.S. in Electrical Engineering from Hanyang University in 1979, and M.S. in Electrical Engineering from the University of Texas at Austin in 1983, and a Ph.D. in System Engineering from Oakland University in 1990. He joined General Motors Research (GMR) Laboratories, Warren, MI, in 1985 and has worked in the area of automotive electronics and control for 30 years. During his nine-year tenure at GMR, he worked on the design and development of various electronic control systems for powertrains and chassis. Since 1993, he has led research activities as a Professor with the Department of Automotive Engineering at Hanyang University. His work has focused on automotive electronics and controls (such as modeling and control of internal combustion engines, design of automotive distributed real-time control systems, intelligent autonomous vehicles, and automotive education programs).