

Design of Model-based VCU Software for Driving Performance Optimization of Electric Vehicle

Changkyu Lee¹ , Youngho Koo¹, Kwangnam Park¹, and Gwanhyung Kim^{2*} 

¹Hanwha Aerospace, Changwon 51561, Republic of Korea

²Department of Computer Engineering, Tongmyong University, Busan 48520, Republic of Korea

Abstract

This study designed a model-based Vehicle Control Unit (VCU) software for electric vehicles. Electric vehicles have transitioned from conventional powertrains (e.g., engines and transmissions) to electric powertrains. The primary role of the VCU is to determine the optimal torque for driving control. This decision is based on the driver's power request and current road conditions. The determined torque is then transmitted to the electric drive system, which includes motors and controllers. The VCU employs an Artificial Neural Network (ANN) and calibrated reference torque to enhance the electric vehicle's performance. The designed VCU software further refines the final reference torque by comparing the control logic with the torque calculation functions and ANN-generated reference torque. Vehicle tests confirmed the effective optimization of vehicle performance using the model-based VCU software, which includes an ANN.

Index Terms: VCU, ANN, Torque Control, Model-based Software, Electric Vehicle

I. INTRODUCTION

The Korean government announced an electric vehicle development plan in 2010 to solve environmental issues such as microdust. Despite the strong support of the Korean government for this plan, electric vehicles are being developed slowly. However, many small motor companies have been founded due to the low technical entrance level regarding powertrains, and they have aimed to change conventional powertrains to electric powertrains in vehicles. Therefore, this study began to develop Vehicle Control Unit (VCU) software to apply to such a powertrain [1].

The VCU designed using model-based software is divided into high- and low-level software layers. The high-level software layer consists of several software modules with a

torque-handling function according to the creep and normal modes. The reference torque generated by an Artificial Neural Network (ANN) was implemented as a lookup table in the software. This reference torque predicts the appropriate electric motor torque. The low-level software layer consists of several blocks in the target block set, which handles the input/output signals based on CAN. This CAN blockset interacts with the vehicle network toolbox of MATLAB/Simulink.

The goal of this study was to design model-based ECU software to optimize the performance of electric vehicles. The VCU software was evaluated through vehicle tests on flat and uphill roads by comparing the control logic and torque calculation functions with the reference torque using an ANN.

Received 6 September 2023, Revised 23 October 2023, Accepted 25 October 2023

*Corresponding Author Gwan-hyung Kim (E-mail: taichiboy1@gmail.com)

Department of Computer Engineering, Tongmyong University, Busan 48520, Republic of Korea

Open Access <https://doi.org/10.56977/jicce.2023.21.4.351>

print ISSN: 2234-8255 online ISSN: 2234-8883

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © The Korea Institute of Information and Communication Engineering

II. VEHICLE CONTROL UNIT

In Korea, VCU research for EVs and PHEVs is being performed under the leadership of vehicle manufacturers. In August 2011, 44 automobile manufacturers, research institutes, and parts companies, including the Hyundai Motor Company and Mando, formed a consortium and began a project to improve the performance of five key parts of electric vehicles: motors, air conditioning, vehicle weight reduction, batteries, and chargers. We are pursuing a plan to lay the foundation for the production of mid-sized electric vehicles that can drive more than 200 km on a single charge, have a maximum speed of 145 km/h, and can be quickly charged in 25 min. Currently, the Hyundai Motor Company/Kia Motors electric vehicle development plan for 2018 is to develop a single-charge electric vehicle. We are focusing on developing and researching an integrated vehicle control technology with plans to produce a vehicle capable of driving over 400 km. As leading electric vehicle companies promoted the mass production of electric vehicles in late 2010, Hyundai and Kia Motors Group promoted electric vehicle mass production in 2014 and expanded support for electric vehicle-related R&D investment expenses to catch up with leading companies. They account for 1.2% and 8% of China, respectively. If the domestic electric vehicle market expands, it will become dependent on foreign products. There are concerns regarding encroachment and technological dependence on core components. In Korea, hybrid drive system technology for passenger cars is being actively developed with the participation of automobile manufacturers and related companies with government support; however, heavy commercial vehicles such as buses and trucks, which account for a large portion of the air pollution as a means of public transportation, and research and technology development on duty-use hybrid drive systems are rarely conducted. Hybrid analysis technology is being actively developed, centering on universities, but vehicle-integrated control technology is judged to be very weak compared to that in advanced countries. Initially, most electric vehicle-related parts were dependent on imports owing to quality, reliability, and lack of domestic production, but now many small- and medium-sized companies, led by Hyundai Mobis and Mando, are supplying the core parts needed for electric vehicles and plug-in hybrid vehicles. Efforts are being made to bring the VCU, battery, BMS, drive motor, inverter, etc. to the level of technological stabilization. In the commercial vehicle sector, electric buses and parallel hybrid buses equipped with a VCU developed by Hyundai Bus have completed a pilot project, and technological advancement work to mass-produce them is being conducted in specific sections. In November 2016, Hyundai Motor Company announced plans for the mass production of electric buses in 2018.

In Korea, Hyundai Motor Company continues to perform

significant research and investment in the development of Hybrid Control Units (HCUs) and VCUs. However, the controllers of domestic electric vehicles and plug-in hybrid vehicles still largely use vehicle controllers from foreign companies. Overseas, companies such as Germany's Siemens and ZF began development in the 1990s, achieved mass production in the early 2000s, and are still devoted to research and development to optimize eco-friendly electric vehicle controllers. In addition, overseas, vehicle controller competition is fierce because of the presence of specialized manufacturing companies. Therefore, each company is developing its own EV/PHEV vehicle control system structure and establishing strategies to actively respond to the current market situation [2].

III. POWERTRAIN SYSTEM FOR ELECTRIC VEHICLE

The powertrain, which is combined with a gearbox, is mounted on an electric vehicle and is connected to the clutch in the gearbox. The powertrain of an electric vehicle consists of electric and mechanical traction systems. Fig. 1 shows the overall block diagram of an electric vehicle powertrain. The electric traction system is divided into a 3-Phase inverter and electric motor. The mechanical traction system consists of a gearbox, differential gear, and drive axle.

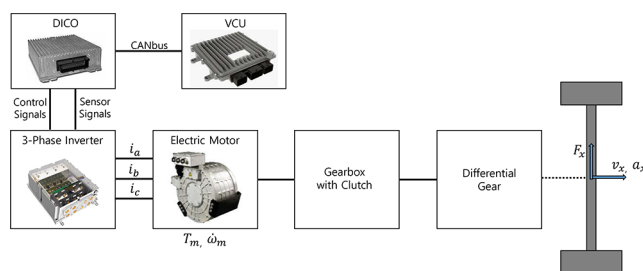


Fig. 1. Overall block diagram of electric vehicle powertrain

Fig. 2 shows that Reckon's electric motor has a maximum efficiency of 93% and maximum power of 118 kW. Siemens' DICO controls an electric motor's speed and torque through a 3-Phase inverter based on the VCU's torque demand.

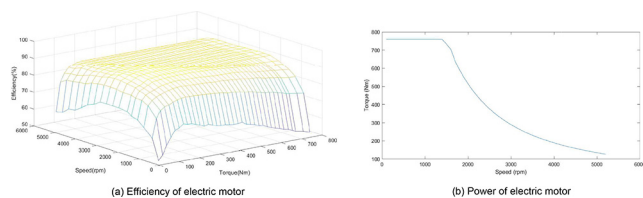


Fig. 2. Efficiency and power of electric motor

This electric traction system modeling formulated by (1) and (2). The electromagnetic torque is given by the following equation [3,4]:

$$T_m = 3p[\lambda_f i_q + (L_d - L_q)i_d i_q]/2 \quad (1)$$

The angular acceleration applied to the vehicle dynamics is

$$\dot{\omega}_m = \frac{G_t}{R_w(m+\Delta m)} \left(\frac{G_t \eta_t}{R_w(m+\Delta m)} T_m - mg \sin \phi - \mu_x mg \right) \quad (2)$$

where p is the number of pole pairs, and λ_f is the flux linkage due to the other rotor magnets linking the stator. i_d and i_q are the stator currents, and L_d and L_q are the stator inductances. G_t is the total gear ratio related to gearbox and differential gear, η_t is the total energy transfer efficiency related to gearbox and differential gear, R_w is the effective rolling radius, and μ_x is the coefficient of rolling resistance.

To represent vehicle velocity and acceleration as the control current for an electric motor, (3) and (4) are derived [4]:

$$a_x = f(i_a, i_b, \theta_e) \\ = \left[-\frac{P\lambda_f G_t \eta_t}{2R_w} \left((1.5 \sin \theta_e - \frac{\sqrt{3}}{2} \cos \theta_e) i_a - \sqrt{3} \cos \theta_e i_a \right) \right. \quad (3)$$

$$\left. -mg \sin \phi - \mu_x mg \right] \frac{1}{m+\Delta m} \\ v_x = f(i_a, i_b, \theta_e) \\ = \frac{1}{m+\Delta m} \int \left[-\frac{P\lambda_f G_t \eta_t}{2R_w} \left((1.5 \sin \theta_e - \frac{\sqrt{3}}{2} \cos \theta_e) i_a - \sqrt{3} \cos \theta_e i_a \right) - mg \sin \phi - \mu_x mg \right] d\tau \quad (4)$$

Here, a_x as acceleration and v_x as velocity can be applied to the plant modeling of electric vehicle simulations, such as SILS, to design the control logic or algorithm before the vehicle test.

IV. DESIGN OF VCU SOFTWARE FOR ELECTRIC VEHICLE

The VCU designed for model-based software is divided into high- and low-level software layers. Each software module is operated according to the execution time of the scheduler, and the input/output signals are processed by each software simultaneously. Figs. 3, 4, and 5 show the main functions in the top-level VCU software configuration of the software modules, including scheduler, drive control, ramp function, and creep function. The low-level software layer consists of a target blockset, such as a device driver, with input and output functions.

The main function, the drive control, handles the reference torque for the electric motor, which is calculated by the input signals of the accelerator and brake, and this reference

torque is adjusted by the speed and torque limitation. The final reference torque is then transmitted as the demand from the VCU to the DICO via CANbus. This function also includes an ANN with a two-layer feed-forward neural network. The ramp function controls the inclination of the torque demand by considering the response time according to vehicle acceleration and deceleration. The creep function controls the torque and speed of the electric motor in the case of vehicle departure, and in particular, it operates as an anti-slip function on a slope.

A. Reference torque calculation

Fig. 5 shows the drive control logic that creates a subsystem and mathematical blocks using MATLAB/Simulink. This core logic operates as the basic control strategy of the overall software modules for normal driving and consists of reference torque calculation, creep control, and torque regeneration by braking. This logic involves calculating the requested torque based on the driver's acceleration and brake pedal operations. The maximum torque is determined by the torque versus speed curve related to the efficiency and power of the electric motor.

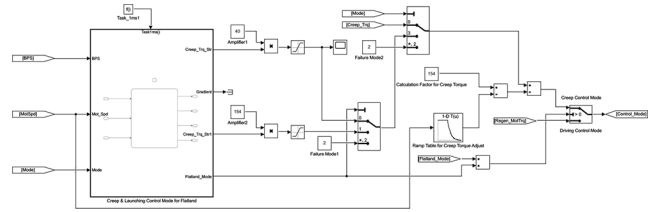


Fig. 3. Creep and launching control logic for flatland

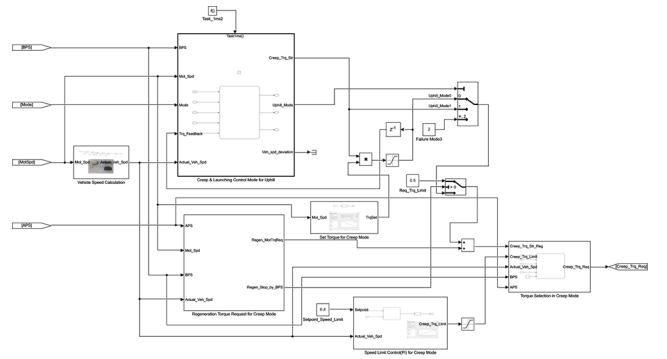


Fig. 4. Creep and launching control logic for uphill

The reference torque for normal driving, that is, the traction torque, can be obtained as

$$T_{ref} = C(A_G T_{mot} - B_C T_{reg}) \quad (5)$$

where T_{ref} , T_{mot} , T_{reg} , A_G , B_C , and C denote the reference

torque for traction, governor from the accelerator pedal, real motor torque, braking cutoff for regeneration, brake factor, and scale adjustment factor, respectively.

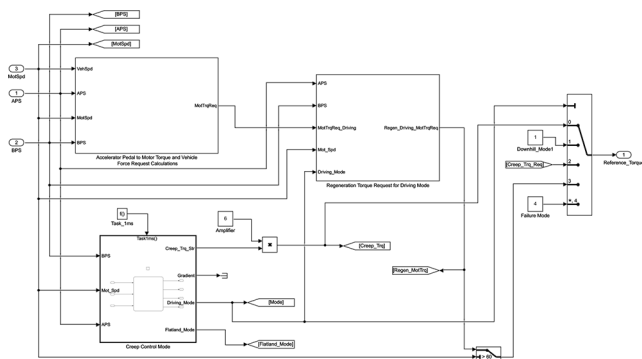


Fig. 5. Drive control logic for normal driving and torque handling

B. Reference torque compensation by ANN

The reference torque calculation in the drive-control logic is normally used as the tractive torque for a vehicle in flat land, uphill, and downhill conditions, but most engineers try to calibrate the parameters of this reference torque again because the performance of the vehicle is not sufficient. These calibrated parameters, lookup tables, factors, and amplifiers were applied to the software. The driving performance is optimized when calibration is increasingly performed by a test engineer; however, in this case, it is costly and time-consuming. Thus, most engineers complete the calibration at an acceptable level, and sometimes the responsibility and driving quality are not good owing to the result of this calibration.

This paper introduced that According to the input signals of the accelerator pedal, brake pedal, and motor speed, the reference torque is determined by the ANN to optimize the driving performance related to responsibility and driving quality. The ANN model is designed in the MATLAB/Simulink environment, as shown in Fig. 6(a), and creates the fitted reference torque, which serves to predict the optimized reference torque and compensate for the parameters of the typical reference torque in the drive control logic. The ANN model consists of subsystems for the normal driving and creep modes. The normal driving mode consists of a two-layer feed-forward based on a neural network with three neurons in the input layer, 150 neurons in the hidden layer, and one neuron in the output layer. The creep mode consists of a two-layer feed-forward based on a neural network with three neurons in the input layer, 100 neurons in the hidden layer, and one neuron in the output layer. All the inputs for the

neural network were the accelerator pedal, brake pedal, and motor speed, and the output was the reference torque with calibration, which was trained using the Bayesian Regularization algorithm.

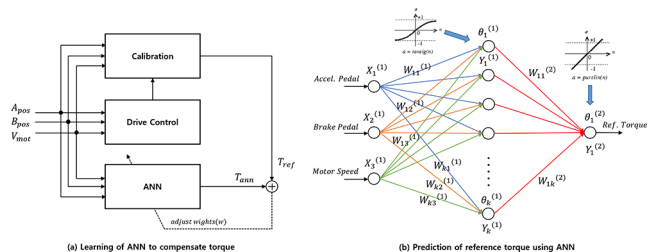


Fig. 6. Proposed ANN structure for reference torque compensation

Fig. 6(b) shows that a sigmoid activation function(tansig) was used in the hidden layer, and a linear function(purelin) was used in the output layer. Feedforward networks, which are a common architecture of ANN for static regression applications, can be represented and calculated as follows [4-7]:

$$Y_{jk} = F_k(\sum_{i=1}^{N_{k-1}} W_{ijk} Y_{i(k-1)} + b_{jk}) \quad (5)$$

where Y_{jk} is the output of neuron j from layer k , b_{jk} is the weight of neuron j in layer k , and W_{ijk} (the model-fitting parameters) are randomly selected connection weights. F_k is a nonlinear activation transfer function that is one of the main characteristic elements of an ANN with a common sigmoidal transfer function.

The predictive reference torque model, which is designed using the neural network toolbox in MATLAB/Simulink, is shown in Fig. 7. The ANN blocks in Fig. 7(b) are obtained from the gensim function to generate an embedded code through the Simulink coder. A Bayesian regularization algorithm was used to train the network to fit the inputs and target, Bayesian regularization algorithm is used [8,9]. This algorithm typically requires more time, but can result in good generalization in the case of difficult, small, or noisy datasets.

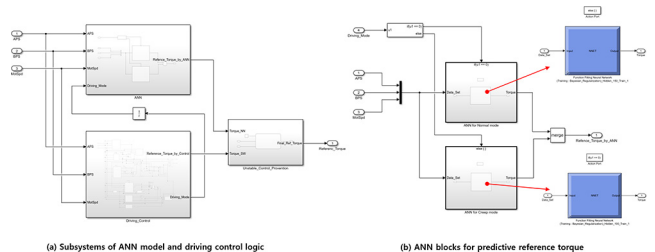


Fig. 7. Predictive reference torque model-based on ANN

The ANN model that predicted the reference torque for the two different datasets was determined after the vehicle test was completed. Figure 8 illustrates the results of the ANN models trained for the normal driving and creep modes. Fig. 8(a) shows a performance of 5.4633 at 1000 epochs and regression values of 0.99988. The inputs had a 3×45001 matrix with samples of three elements, and the target had a 1×45001 matrix with samples of one element. The validation and test data were divided into 27001 samples (60%), 9000 samples (20%), and 9000 samples (20%) samples for training, validation, and testing, respectively. Fig. 8(b) shows a performance of 12.3692 at epoch 1000 and a regression value of 0.99974. The inputs had a 3×70001 matrix with samples of three elements, and the target had a 1×70001 matrix with samples of one element. The validation and test data were divided into 42001 samples (60%) for training, 14000 samples (20%) for validation and 14000 samples (20%) for testing. According to the regression plots below, the fit was reasonably good for all datasets, with R-values of 0.99 or above.

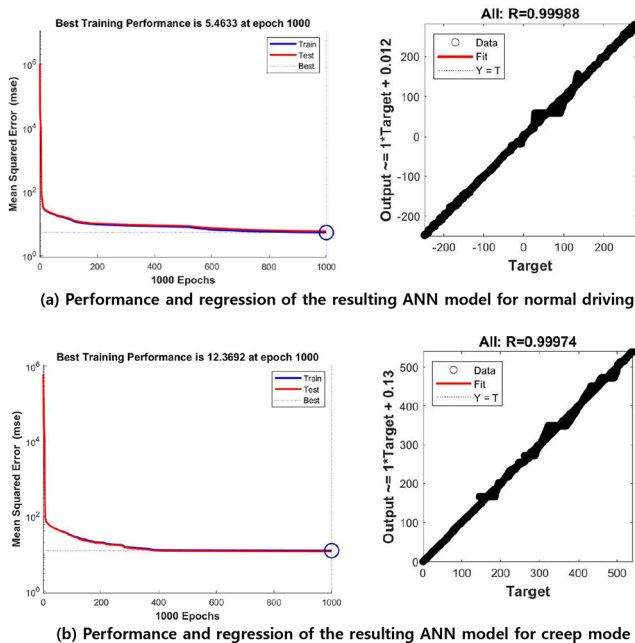


Fig. 8. Performance and regression of ANN models

V. TEST RESULTS OF VCU SOFTWARE

The purpose of this VCU software test is to evaluate the driving performance by comparing normal torque control with torque control based on an ANN through a vehicle test on straight roads and test hills. The test course of the vehicle was on a straight road for the normal driving mode and the gradient of the test hill was 12% for the creep mode and hill start.

The vehicle specifications for the test were a 1 ton weight, rear axle ratio of 3.727, and radial 195R14-6PR tires. The vehicle powertrain (see Table 1 and Fig. 9) consists of Reckon's electric motor and Simens's 3-Phase inverter, and it is mounted on the chassis frame.

The test was performed in two phases. First, the vehicle was tested by VCU, which was applied to the software of normal torque control in flatland and uphill conditions; second, the vehicle was tested by VCU, which was applied to the software of torque control based on ANN under the same conditions. For safety reasons, vehicle speed was limited to 65 km/h on flat land. To compare the driving performance with the VCU software, the signals of the accelerator and brake pedals using CAPL programming in the CANoe tool were overrode instead of operating the driver's pedal and brake. These signals were logged through a vehicle test under the normal driving and creep modes. CANoe also logs all datasets on CANbus, such as the vehicle speed, motor speed, and reference torque. At this time, the driver only controlled the steering of the vehicle and emergency stop using the brake pedal during the vehicle test.

Table 1. Test Conditions for Vehicle Driving

Proving Ground	Test Case	Operation by CANoe		Remark
		Accel Pedal (%)	Brake Pedal (%)	
Flatland (High speed circuit)	Torque control by normal driving	0%→25%	78%→0%	
	Torque control by ANN	0%→25%	78%→0%	
Uphill (Test hill 12%)	Torque control by normal driving	0%	78%→0%	Creep mode
	Torque control by ANN	0%	78%→0%	Creep mode

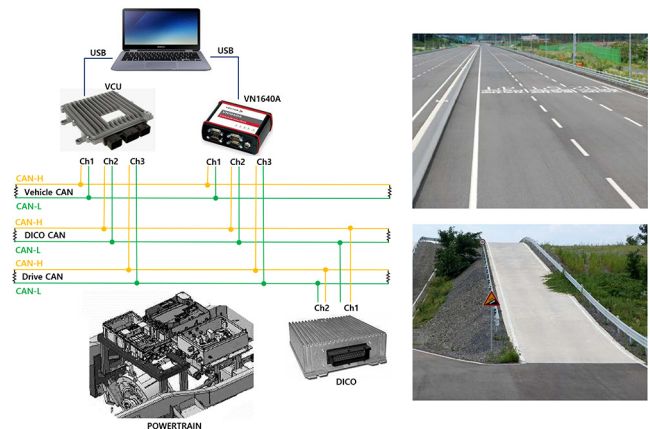


Fig. 9. Structure of powertrain and test environment

A. Test Results of VCU Software at Flatland

The purpose of this test was to evaluate whether VCU software with ANN models has an optimized driving performance that is better than that of VCU software with normal driving logic. Fig. 10(a) illustrates the test results of the VCU software with the ANN models trained for the normal driving mode. This paper does not present a comparison with the test results of the calibrated reference torque and the reference torque by ANN because the fitting results for the reference torque are already shown in Fig. 8, and it is known that the torque control profile between the calibrated reference torque and reference torque by ANN is the same through these fitting results.

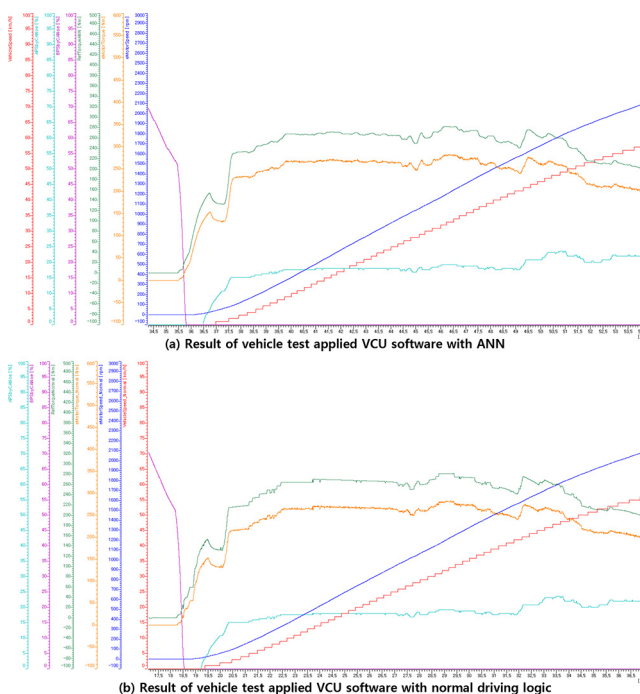


Fig. 10. Results of vehicle test comparing ANN model and normal driving logic on flatland

Fig. 10 shows the results of the vehicle test comparing the ANN model and normal driving logic on flatland. The reference torque was output from the accelerator pedal signal and brake pedal, which was automatically generated using the CANoe tool. Based on this reference torque, the motor and vehicle speeds increased.

The reference torque, which only consists of logics, was fluctuated and tough in the phase of vehicle departure during that time from 18.2 to 18.8 s in Fig. 10(b). However, the motor did not respond at the vehicle departure point at this time because the control function was operated in creep mode in this phase. In the other phase over 18.8 s, the motor

speed produced a faster and rougher response than the ANN model, owing to the reference torque. Calibration using the ANN model was required because of the vehicle response and unstable reference torque.

Fig. 10(a) shows that the reference torque with the ANN model was smooth in the phase of vehicle departure during the time from 35.4 to 36.3 s. Although the motor did not respond to the vehicle departure point, the reference torque exhibited a very stable output. In the other phase over 36.4 s, the motor speed produced an optimized response better than the normal driving logic according to the reference torque.

To compare the motor speed with the vehicle speed in the test results, each motor speed and vehicle speed were overlapped from the logged datasets using CANoe, as shown in Fig. 11. Although the deviation of motor speed is small in phase of vehicle departure, it showed that the vehicle speed started to make a difference due to the effects of each reference torque. The VCU software with normal driving logic had a launching performance in which the vehicle speed was 1 km/h faster than that of the VCU software with the ANN model at approximately 71.6 s. However, the VCU software with the ANN model showed better driving performance after approximately 76.8 s. This test showed that the vehicle speed increased gradually because the driving performance was properly optimized by the ANN model.

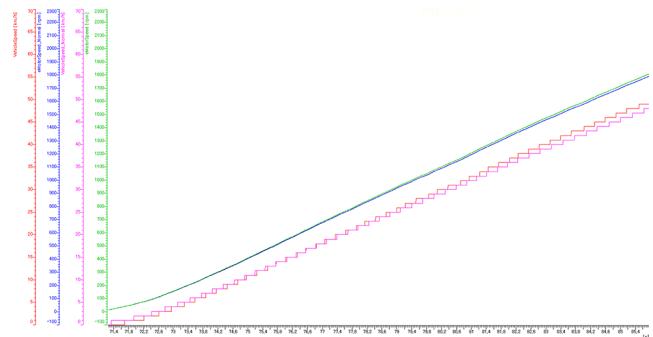


Fig. 11. Comparison of motor speed and vehicle speed results on flatland

B. Test Results of VCU Software on Uphill

This section presents the results of the vehicle test comparing the ANN model and the normal driving logic uphill, as shown in Fig. 12. The reference torque was output from the accelerator pedal signal and brake pedal, which is automatically generated by the CANoe tool, but the signal for the accelerator pedal was not generated to evaluate only the reference torque in the creep mode on uphill. According to this reference torque, the motor and vehicle speeds increased.

The reference torque was output in the vehicle departure phase from 33.6 to 36 s in Fig. 12(b), despite no accelerator

pedal operation. At this time, the VCU software calculated the slip ratio using the motor speed and direction owing to the uphill 12% when the brake pedal was released. The VCU, which was operated in creep mode, produced a proper response through the reference torque during vehicle departure uphill. In the phase over 36 s, the motor and vehicle speeds increased. The VCU software made an unexpected overshooting, as shown by the reference torque line, because the motor does not immediately respond to the torque demand having a dramatic increase of approximately 535 rpm for Δt of 0.34 s, as shown in Fig. 12(b). Thus, the calibration of the reference torque by the ANN model is required because of the vehicle response and unstable reference torque.

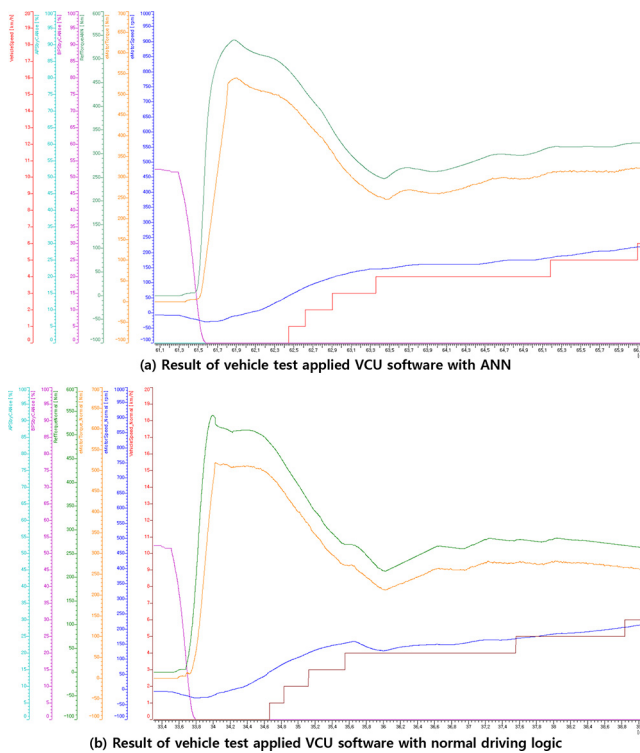


Fig. 12. Results of vehicle test comparing ANN model and normal driving logic on uphill

Fig. 12(a) shows that the reference torque with the ANN model was smooth in the phase of vehicle departure during the time from 61.3 to 63.5 s. Although a large amount of motor torque was required within a short time, the creep mode operated well uphill. In the phase over 62.4 s, the motor speed produced an optimized response compared with the normal driving logic according to the reference torque.

To compare the motor speed with the vehicle speed in the test results, each motor speed and vehicle speed were overlapped from the logged datasets using CANoe, as shown in Fig. 13. There was a slight deviation in motor speed in the

phase of vehicle departure, but it showed that the vehicle speed started to make a difference owing to the effects of the reference torque fitted by the ANN model. The VCU software with the ANN model had a launching and driving performance in which the vehicle speed was 1 km/h faster than that of the VCU software with normal driving logic at approximately 51.4 s, and as time passed, the interval of time between these two software packages increased. This test showed that the vehicle speed gradually increased during launching on uphill because the driving performance was properly optimized by the ANN model.

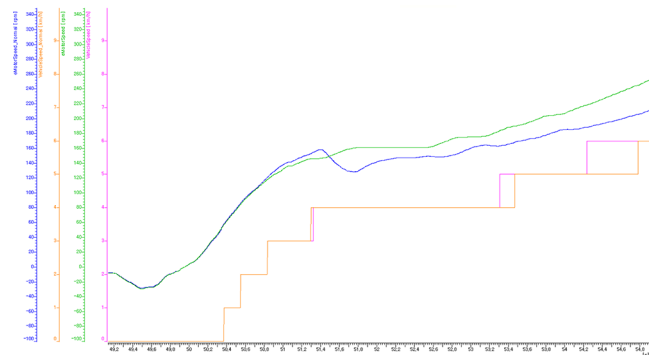


Fig. 13. Comparison of motor speed and vehicle speed test results on uphill

VI. CONCLUSIONS

The results presented in this paper indicate that VCU software improved the starting and driving performances due to the ANN models through a vehicle test on flatland and uphill. The VCU software with normal driving logic, which consists of lookup tables and torque calculation blocks, showed an unstable control status. Therefore, this can be considered the root cause of the calibration against the reference torque. On the other contrary, the ANN models provided smooth and stable control for the starting and driving of the vehicle.

The ANN model was generated as blocks for Simulink after it was designed using the neural networks toolbox in MATLAB/Simulink, and the VCU software with the ANN models was verified through a vehicle test. Although the development method of model-based software is still not generalized, ANN models are easily implemented and operate well as a specific function in VCU software.

In conclusion, the model-based VCU software with ANN models, which predicts the reference torque based on the accelerator pedal, brake pedal, and motor speed, optimized the driving performance of an electric vehicle. However, these ANN models should be checked for different test cases, and the driving performance under various road conditions should be evaluated.

REFERENCES

- [1] [1] S. K. Hwang, "Comparative study on electric vehicle policies between Korea and EU countries," *World Electric Vehicle Journal*, vol. 7, no. 4, pp. 692-702, Dec. 2015. DOI: 10.3390/wevj7040692.
- [2] [2] S. M. Lee, "Study on VCU development for EV/PHEV, majored in e-vehicle technology," M.S. Theses, University of Ulsan, Ulsan, Korea, 2017. [Online], Available: <https://oak.ulsan.ac.kr/handle/2021.oak/6184>.
- [3] [3] B. K. Powell, K. E. Bailey, and S. R. Cikanek, "Dynamic modeling and control of hybrid electric vehicle powertrain systems," *IEEE Control Systems Magazine*, vol. 18, no. 5, pp. 17-33, Oct. 1998. DOI: 10.1109/37.722250.
- [4] [4] G. M. Park, S. H. Lee, S. H. Jin, and S. S. Kwak, "Modeling and dynamic analysis for electric vehicle powertrain," *The Institute of Electronics Engineers of Korea – System and Control*, vol. 48, no. 6, pp. 71-81, Nov. 2011.
- [5] [5] E. E. Ekong, A. A. Adewale, A. Ben-Obaje, A. M. Alalade, and C. N. Ndujiuba, "Performance comparison of ANN training algorithms for hysteresis determination in LTE networks," *Journal of Physics Conference Series*, vol. 1378, no. 4, p. 042094, Dec. 2019. DOI: 10.1088/1742-6596/1378/4/042094.
- [6] [6] Y. Ryoo, J. Lee, Y. Lim, Y. Chang, E. Kim, and C. Moon, "Speed error compensation of electric differential system using neural network," *Journal of Institute of Control, Robotics and Systems*, vol. 7, no. 1, pp. 1205-1210, Jan. 2001.
- [7] [7] H.. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, *Neural Network Design*, 2nd ed. Stillwater, USA, Sep. 2014.
- [8] [8] A. M. Wefky, F. Espinosa, M. Mazo, J. A. Jiménez, E. Santiso, A. Gardel, and D. Pérez, "Gear predictor of manual transmission vehicles based on artificial neural network," *WIT Transactions on Modelling and Simulation*, vol. 48, pp. 37-46, May 2009. DOI: 10.2495/CMEM090041.
- [9] [9] A. P. Plumb, R. C. Rowe, P. York, and M. Brown, "Optimization of the predictive ability of artificial neural network (ANN) models: a comparison of three ANN programs and four classes of training algorithm," *European Journal of Pharmaceutical Sciences*, vol. 25, pp. 395-405, Jul. 2005. DOI: 10.1016/j.ejps.2005.04.010.
- [10] [10] J. T. Spingenberg, A. Klein, S. Falkner, and F. Hutter, "Bayesian optimization with robust bayesian neural networks," in *The Neural Information Processing Systems Conference*, Barcelona, Spain, pp. 1-9, 2016.



Changkyu Lee

Received his M.S. and Ph.D. degrees from the Department of Electronics and Communication Engineering of Korea Maritime and Ocean University, Busan, Korea, in 2003 and 2015, respectively. At present, he works as an engineer in the power propulsion technology team of Hanwha Aerospace, Korea. His main research interest areas are control algorithm, signal processing, and embedded software for powertrain regarding commercial and armored vehicle.



Youngho Koo

Received his B.S and M.S. degrees from the Pusan National University School of Mechanical Engineering, Busan, Korea, in 1987 and 2006, respectively. At present, he works as a team manager and principal engineer in the power propulsion tech. team of Hanwha Aerospace, Korea. His main research interest areas are power-pack system engineering including conventional and environment-friendly power system for armored vehicle.



Kwangnam Park

Received his B.S. degrees from the School of Mechatronics Engineering of Changwon National University, Changwon, Korea, in 1995. At present, he works as a center manager in amphibious systems R&D center of Hanwha Aerospace, Korea. His main research interest areas are vehicle system engineering, environment-friendly power-pack, and manned-unmanned teaming system based on AI for amphibious and next-generation armored vehicle.



Gwanhyung Kim

Received his Ph.D. degrees from the Department of Electronics and Communication Engineering of Korea Maritime and Ocean University, Busan, Korea. At present, he works as a professor in the Department of Computer Engineering at Tongmyong University. His main research interest areas are control algorithm, signal processing, and embedded software for powertrain regarding commercial and armored vehicle.