International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:8, No:2, 2014

UML Model for Double-Loop Control Self-Adaptive Braking System

Heung Sun Yoon, Jong Tae Kim

***Abstract*—**In this paper, we present an activity diagram model for double-loop control self-adaptive braking system. Since activity diagram helps to improve visibility of self-adaption. We can easily find where improvement is needed on double-loop control. Double-loop control is adopted since the design conditions and actual conditions can be different. The system is reconfigured in runtime by using double-loop control. We simulated to verify and validate our model by using MATLAB. We compared single-loop control model with double-loop control model. Simulation results show that double-loop control provides more consistent brake power control than single-loop control.

***Keywords*—**Activity diagram, automotive, braking system, double-loop, Self-adaptive, UML, vehicle.

I. Introduction

UNIFIED Modeling Language (UML) is a standardized modeling language in software engineering. The Unified Modeling Language includes a set of graphic notation techniques to create visual models. Use case diagram, sequence diagram, class diagram, and activity diagram are typical one of graphic notation techniques. UML is not a software development method, but UML is compatible with diverse development methods.

Activity diagrams of UML are representations of workflows which show stepwise activities and actions. It represents not only sequence control flow, but also performing parallel activities, so we can distinguish dependencies and non-dependencies explicitly to avoid premature design. Using Activity diagram is the best way to check dependencies between use cases.

Self-adaptive system has closed-loop feedback or feed forward to adapt changes that are occurring during runtime autonomously. The changes arise from internal cause or context such as external events. Self-adaptive system is required to monitor internal or external events, detect significant changes, decide how to react, and act to execute such decisions [1]. Self-adaption helps to reduce complexity, improve flexibility, and dependability of embedded systems [2].

Complexity of automotive embedded systems has been rapidly increasing since first software was introduced into vehicles in 1976 [3]. Automotive system is life-critical system which requires high safety and reliability. To improve reliability, there are many researches to make more flexible and increase reliability by applying self-adaption in automotive embedded system. One of those applies self-adaptive algorithm to adjust reference deceleration in time [4]. That paper designed system as single-loop control.

This paper models double-loop control self-adaptive braking system by using activity diagram which can represent explicit property of self-adaptation. We simulated our model. As a result we can verify and validate the model behavior even in early stages of the development process by using MATLAB. Furthermore, we compared single-loop control with double-loop control. We could verify an improvement in difference of braking distance.

Heung Sun Yoon is with the Department of Electrical and Computer Engineering, Sungkyunkwan University, Jangan-gu, Suwon, Republic of Korea (e-mail: y11010s@skku.edu).

Jong Tae Kim, is with the School of Electronics and Electrical Engineering, Sungkyunkwan University, Jangan-gu, Suwon, Republic of Korea (e-mail: jtkim@skku.edu).

II. Modeling Self-Adaptive Braking System

A. Self-Adaptive Braking System (SABS)

When a driver wants to decelerate, he/she expects a car to reduce its speed as much as they control the brake pedal. However, a driver might need to push the brake pedal more than they expected due to the various road conditions or brake pad abrasion.

Therefore, the brake control which provides stable speed control regardless of various conditions is required. SABS was designed to adapt to internal changes (brake pad) as well as external changes (road condition).

In order to achieve stable speed reduction, SABS calculates the coefficient SAB and controls the brake power (1). D represents the value of the brake pedal position sensor, and BP represents the brake power.

$$SAB \times D = BP \quad (1)$$

SAB which is used to control BP can be calculated as (2). Coefficient R and WO are used to make the system to adapt to the rapidly changing environments (road condition) and slowly changing environments (brake pad abrasion) respectively.

$$SAB = 1 + (R + WO), where\ |R| < 1, |WO| < 1 \quad (2)$$

B. System Architecture

State-of-the-art automotive embedded systems consist of many electronic control units (ECUs) and interconnection Network to control car and service multimedia. Typical ECUs consist of one or more CPUs, memories of various types, and a communication controller to access the interconnection network. ECU modules work to achieve dedicated objective as a part of vehicle. And they are connected to each other through automotive communication protocol, such as Controller Area

Network (CAN) [5], FlexRay [6] or Media Oriented Systems Transport (MOST) [7].

Self-adaption can be divided in to two categories external / internal adaptation. Internal adaptation is combined with application and adaptation logic. But in external adaptation, additional external adaptation manager exist. External approaches are more scalable than internal approaches. In addition, reusability of the adaptation manager is a significant advantage of the external approaches [1].

SABS architecture is designed as an additional module to send BP to brake control unit (BCU). SABS is external adaption manager for BCU. BCU and SABS are connected into FlexRay.

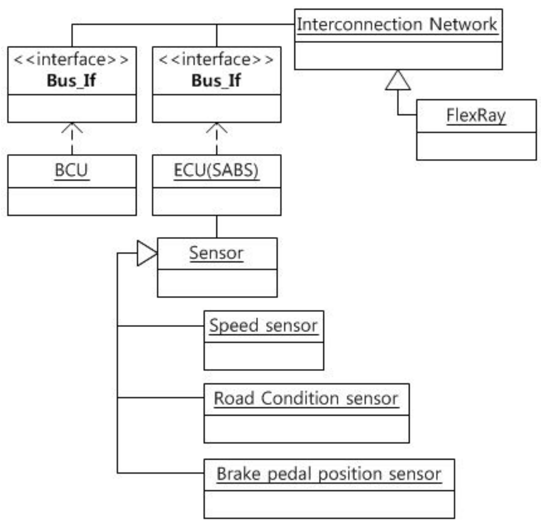

Fig. 1 Self-adaptive braking system architecture

Fig. 1 represents SABS architecture. The SABS monitors brake pedal position sensor and road condition sensor and calculates SAB to control brake power. The calculated BP is sent to BCU through FlexRayso that BCU operates actual braking. As a feedback, deceleration is obtained by speed sensor.

C.Activity Diagram to Calculate Coefficient R

Fig. 2 illustrates the activity diagram representation for calculating coefficient R. By means of the first loop, deceleration can be calculated by controlling the brake power for rapidly changing factors such as weather conditions.

When a driver pushes the brake pedal, the different R value is retrieved from the pre-defined mapping table based on the different road conditions (ice/moisture/ dry/temperature). SAB is determined based on the R value and transmitted to BCU (Brake Control Unit) so that the car can slow down its speed. The R value is updated in every loop by comparing actual speed reduction and reference speed reduction (3).

$$R_{t+1} = (\frac{Ref-ASR}{Ref} + 1) \times |R_t| \quad (3)$$

Ref represents the reference speed reduction and ASR represents the actual speed reduction acquired by the speed sensors in the car. By definition of R_{t+1}, desired speed reduction can be achieved through updating R_{t+1} by ratio of difference between reference speed reduction and actual speed reduction to reference speed reduction.

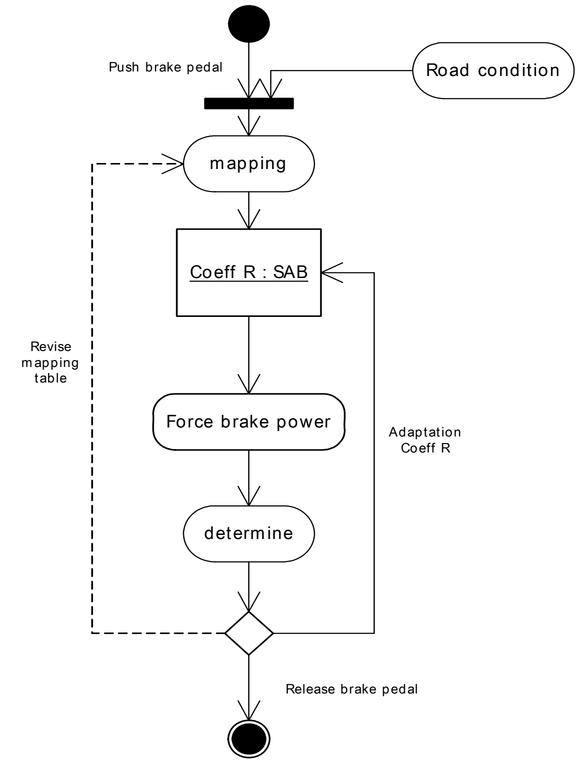

Fig. 2 Activity diagram for calculating coefficient R

Through the second loop, the pre-defined R values in the mapping table can be revised. When the pre-defined R value is not valid; the R value in the mapping table is modified. The designer implements the mapping table which contains the initial R values by using hardware, software co-simulation or prototyping. If the R value is appropriately pre-defined, the difference between the actual speed reduction and the reference speed reduction would be acceptable. However, inappropriately pre-defined R value would cause a significant difference. In the latter case, the desired speed reduction can be achieved by the first loop. However, the invalid initial value is continuously used.

Whereas the double-loop control in this paper can revise the pre-defined mapping table so that the R value can be adjusted appropriately. It is difficult to predefine the parameters correctly by considering all characteristics of vehicles and special possibilities. By means of the double-loop, the parameters can be revised even the current situation is different from the initially defined situation.

We need a threshold to decide when we revise invalid R. If difference between reference speed reduction and actual speed reduction is bigger than threshold, the R value will be revised to valid R which is value when actual speed reduction converges

International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:8, No:2, 2014

to reference speed reduction.

Activity diagram can visually illustrate the loop and consequently, it can be used effectively when designing the double-loop control system. Furthermore, we will verify functional behavior by simulation with MATLAB in next section.

D. Activity Diagram to Calculate Coefficient WO

Fig. 3 shows the activity diagram representation for calculating the coefficient WO. When the R value in the mapping table is revised, the changed value is accumulated as history data. When history data are accumulated enough; the tendency of coefficient R is generalized by linear regression.

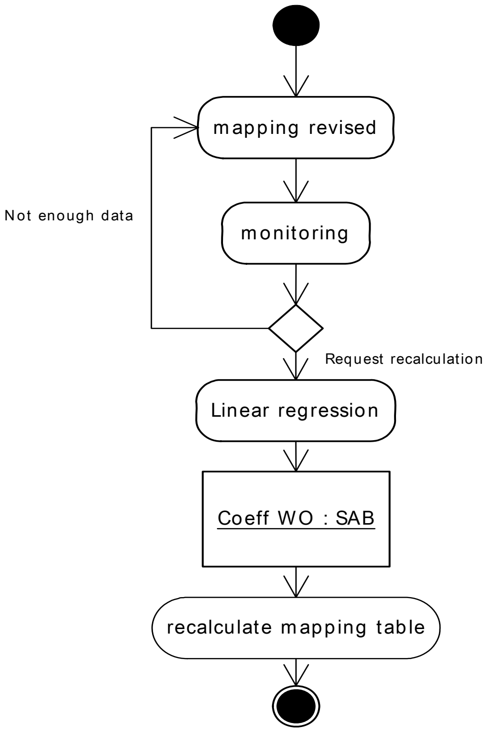

Fig. 3 Activity diagram for calculating coefficient WO

When the tendency of R value is assumed to represent the characteristics of the brake pad abrasion, WO can be calculated by linear regression. WO is a variable used to make the system adapt to the slowly changing factors. WO works as an offset and updates the R value in the mapping table (3).

E. Overhead

Over head of our model can consider various perspectives. In cost perspective, the SABS is designed in external adaptation. So it consumes an ECU module. But we can choose internal approach into BCU so that we can save cost. It's a trade-off cost against scalability and reusability.

Calculation overhead can be considered. The calculation time of R is very small, but calculation time of WO will be longer than R. Calculation time of linear regression depends on size of data. The linear regression can calculate simple equation of linear algebra. Furthermore, WO will be calculated during idle time in contrary to R is calculated when driver push brake pedal. WO is reflection of slow changes so that calculation of WO is not urgent.

F. Class Diagram of SABS

Based on activity diagram, we modeled SABS in class diagram which is illustrated in Fig. 4. Road condition manager, brake pedal position manager, and speed manager inherit sensor class. SABS class demands road condition and temperature to road condition manager when brake pedal position manager detect brake pedal has been pushed. SABS class searches for coefficient R by road condition and temperature from map manager class which have pre-defined R values. And calculate SAB and BP. The calculated BP is sent to BCU through FlexRay interface class, and after that SABS requires speed manager to get deceleration. So SABS can calculate R iteratively.

Linear regression class and map manager class are composition of SABS. Map manager has a mapping table which has pre-defined R depending on state of the road surface and temperature. Linear regression class generalizes tendency of R values when data of R have been accumulated enough.

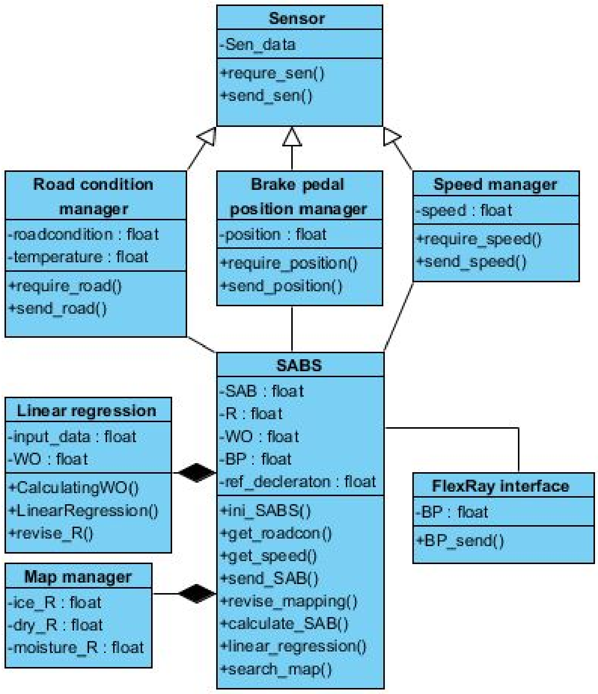

Fig. 4 Class diagram of SABS

III. Simulation

A. Calculation of Coefficient R

We simulated our model to verify functional behaviors with the MATLAB. The simulation is programmed based on activity diagram and class diagram. The value of reference speed reduction refers to [3]. Equations (1)-(3) simulated to verify those work correctly.

Fig. 5 represents reference speed reduction ($5.6574\text{m}/s^2$)

and actual speed reduction. We verified the actual speed reduction converge to reference speed reduction through (3). In this case, pre-defined R was smaller than converged R value by 24.12%. We can see actual speed reduction quickly achieved reference speed reduction. In our simulation, we set the threshold as 6%, so if difference between actual speed reduction and reference speed reduction is bigger than 0.3394 R will be revised. As a result, Fig. 5 is worst-case of double-loop control. As you can see, the maximum difference between the two speed reductions is 0.3394.

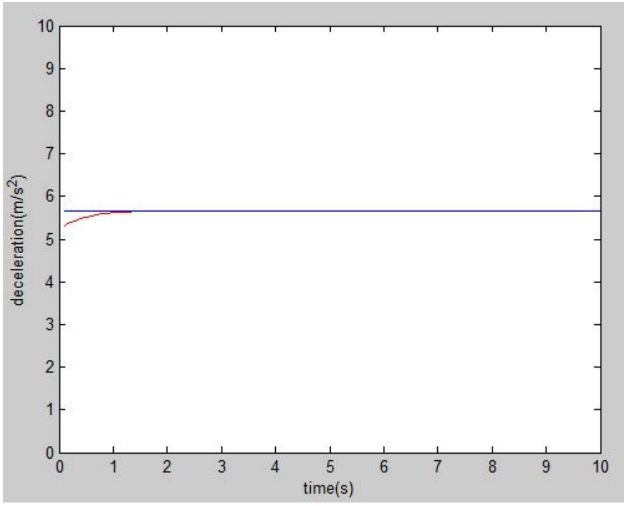

Fig. 5 Convergence of actual speed reduction (double-loop)

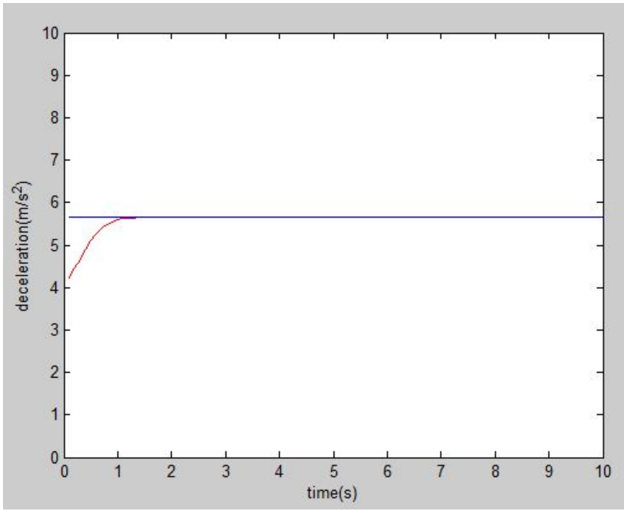

Fig. 6 Convergence of actual speed reduction (single-loop)

In our simulation, the worst-case of single-loop control represents as Fig. 6. Fig. 6 shows convergence of actual speed reduction when brake pads are worn out.

In single-loop control, there is no revising path to reconfigure pre-defined R. As time goes by, the maximum difference between actual speed reduction and reference speed reduction will be getting bigger and bigger. We simulated during constant time, so Fig. 6 represents worst-case of single-loop control in our simulation.

B. Calculation of Coefficient WO

We changed road condition randomly to verify calculation of coefficient WO. And we assumed characteristics of the brake pad occur slowly. The threshold was set for revising R value as 6%. When difference between reference speed reduction and actual speed reduction exceed 6% of reference speed reduction, we revised R value and accumulated R as history data.

WO is calculated by linear regression of history data R. After calculating WO, R should be subtracted as much WO. As a result, we can verify coefficient R is maintained as a similar level of initial value of R.

Fig. 7 represents linear regression of R values to generalize tendency of R. The ‘x’ marks are revised values of R. Through linear regression, we could see tendency of R and get coefficient WO. When we get generalized straight line by linear regression, WO is difference between first and last value on the straight line in axis history of R. In this simulation, WO was calculated as 0.1015.

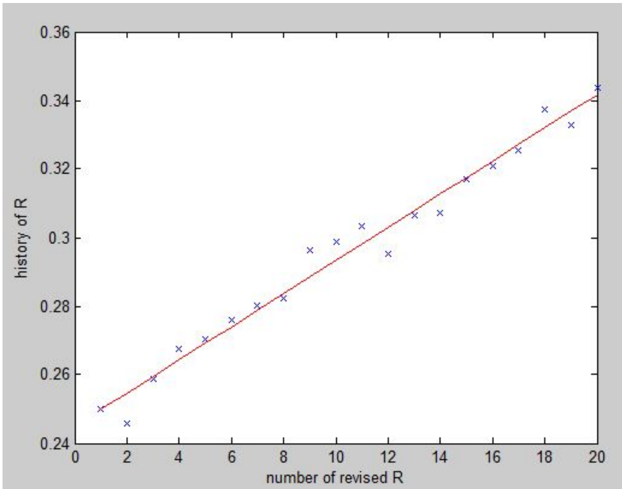

Fig. 7 Linear regression of R for calculating WO

IV. Conclusion

This paper presents the activity diagram representation of SABS architecture for robust brake power control. Since the visibility of self-adaptation can be enhanced by UML activity diagram, it was easy to design the double-loop control system by extending the single-loop control system.

Using the UML model and MATLAB simulation we verified that double-loop control SABS provides consistent brake power control even though brake pads have become worn out as time goes by. Four thousand data of braking distance are used to calculate mean. In single-loop control model, simulated braking distance was longer than reference braking distance by3.0216 m. But it was only 0.0644 m in double-loop control model. We verified that double-loop control provides more consistent brake power control than single-loop control.

REFERENCES

[1] M. Salehie and L. Tahvildari, “Self-adaptive software: Landscape and research challenges,” ACM Trans. Auton. Adapt. Syst., 4(2):1–42, 2009.

[2] M. Zeller, G. Weiss, D. Eilers, and R. Knorr, “Co-Simulation of Self-Adaptive Automotive Embedded Systems,” 8th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC), Dec. 2010.

[3] A. Pretschner, M. Broy, I. H. Kruger, and T. Stauner, "Software engineering for automotive systems: A roadmap." 2007 Future of Software Engineering. IEEE Computer Society, 2007.

[4] W. Weida, D.Nenggen, and X.Xiangyang, "An improved self-adaptive algorithm of vehicle reference speeds for ABS." Vehicular Electronics and Safety, 2006. ICVES 2006. IEEE International Conference on. IEEE, 2006.

[5] Robert Bosch GmbH, “CAN Specification Version 2.0,” 1991, http://www.semiconductors.bosch.de/pdf/can2spec.pdf.

[6] FlexRay Consortium. "FlexRay communications system protocol specification version 2.1." Request online: http://-www.flexray. com/specification request. php (2005).

[7] MOST Cooperation, “MOST Specification Rev. 3.0,” 2008, http://www.mostcooperation.com/.