Hands-On Field Operational Test Dataset of a Multi-Controller CPS: A Modeled Case Study on Autonomous Driving

Yong-Jun Shin, Esther Cho, Hansu Kim, Doo-Hwan Bae
Korea Advanced Institute of Science and Technology
Daejeon, Republic of Korea
{yjshin, esthercho, hansukim, bae}@se.kaist.ac.kr

2022 17th Annual System of Systems Engineering Conference (SOSE) | 978-1-6654-9623-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/SOSE55472.2022.9812701

***Abstract*—Cyber-physical systems (CPSs) observe operational environments and continuously decide on actions to achieve goals. In many cases, complex CPSs have several goals to satisfy simultaneously. To develop a CPS with multiple goals, engineers can create various control systems, each contributing to a goal based on a system-of-systems (SoS) engineering perspective. Engineers then conduct field operational tests (FOTs) to collect data for analyzing and optimizing the control systems. However, uncertainties in the physical environment and the emergent behavior of multiple controllers present several challenges in conducting FOTs. We have hands-on experience in performing massive FOTs of a multi-controller CPS to realize engineering challenges in the FOTs. We modeled and developed an autonomous robot vehicle consisting of a lane-keeping system and an adaptive cruise control system. To analyze and optimize autonomous driving, we conducted FOTs of 125 possible configurations of the control systems, each 50 times. This paper presents 1) the model, software, and hardware implementation manuals of our case study on autonomous driving, 2) an FOT log dataset obtained from about 100 hours of driving and its analysis results, 3) research challenges emerging in the multi-controller CPS FOT learned from our hands-on experience, and 4) possible applications of our dataset for future research.**

***Index Terms*—Cyber-physical systems, Multi-controller, System-of-systems, Autonomous vehicle, Field operational test dataset**

I. Introduction

Cyber-physical systems (CPSs) continuously adapt their actions to satisfy goals in physical environments [1]. A CPS has a feedback loop consisting of a controller that checks the goal achievement and manipulates physical components based on its decision-making strategy [2].

Developing a decision-making strategy is one of the primary purposes of CPS development. When there are many goals that a CPS is required to achieve simultaneously, it becomes more challenging to develop an effective strategy. One popular approach is to create dedicated controllers for each goal to divide the concern [3]–[5]. It views complex CPSs through the lens of system-of-systems (SoS) [6]–[8]. For example, both a lane-keeping system and an adaptive cruise control system operate together within an autonomous vehicle.

Engineers can conduct field operational tests (FOTs) [9] of a CPS under development to evaluate to what extent the CPS can achieve the given goals in the actual operational environment and optimize the configurations of the CPS controllers. However, conducting the CPS FOT has several engineering challenges. FOT results are stochastic because of uncertainty in the physical environment (e.g., sensor noise). It requires engineers to repeat many FOTs to obtain statistically significant results. In a multi-controller CPS, one controller may affect the performance of another controller during an FOT. A specific combination of controllers may trigger an emergent behavior that developers may not expect. Additionally, in many cases, the configuration space of the controllers under analysis is extensive and continuous, making the optimization of the controllers more exhaustive.

To realize these challenges, we have hands-on experience in developing a multi-controller CPS and conducting FOTs in the SoS perspective. We designed and modeled an autonomous robot vehicle consisting of a lane-keeping system and adaptive cruise control system. We then performed FOTs of 125 possible controller configurations each 50 times, and analyzed the results. This paper provides all materials and datasets related to this case study for future research and shares the lessons learned from our hands-on experience.

In summary, this paper contributes to the research on multi-controller CPS development by providing the following:

- A re-implementable case study of a multi-controller CPS, including its model, software, and hardware implementation manuals,
- An autonomous driving FOT log dataset of 125 controller configurations, each with 50 test results, obtained from about 100 hours of driving,
- Lessons learned from hands-on experience exposing research challenges emerging in the multi-controller CPS FOT,
- Possible applications of the FOT log dataset for future research.

The remainder of this paper is organized as follows: Section II provides a background on the control theoretic system design. Section III introduces our case study to develop a CPS, and Section IV presents our data collection method. Section V analyzes the collected FOT dataset. Section VI discusses the engineering challenges revealed in our experience. Section VII introduces some possible applications of our FOT log dataset. Section VIII addresses threats to validity, and Section IX

978-1-6654-9623-0/22/$31.00 ©2022 IEEE

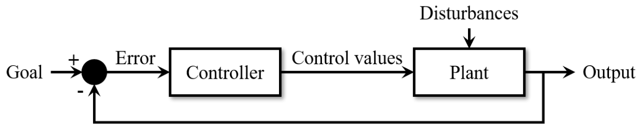


Fig. 1: A feedback loop from the control perspective [2]

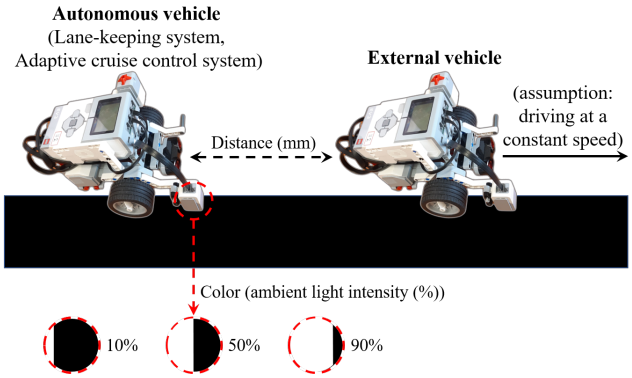


Fig. 2: An autonomous robot vehicle case study design

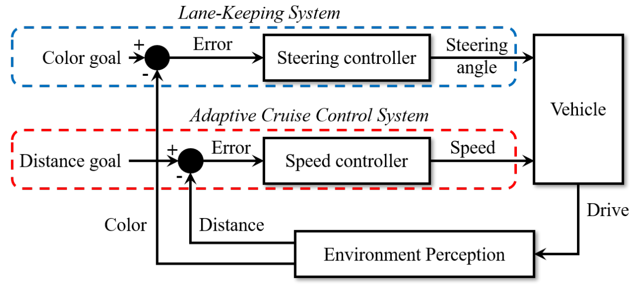


Fig. 3: Autonomous vehicle controllers

reviews the related work. Finally, Section X presents the conclusions and opportunities for future studies.

II. Background: Control-Theoretic Feedback Loop

Many CPSs have feedback loops that observe the uncertain and changing environments and make adaptive actions [10]. A popular approach to modeling the feedback loop is based on control theory [4], [11], [12]. Fig. 1 shows the feedback loop from a control perspective [2], [13]. The feedback loop consists of a controller and a plant. Control values generated by the controller manipulate the plant, and the plant's behavior depends on the control values. The behavior of the plant is measurable for the system goals. Using the measured behavior of the plant, the controller calculates the error associated with each goal and determines the control values of the plant to minimize the error. In addition to the control values, factors that affect the plant's behavior but are not under the direct control of the system are called disturbances or, sometimes, uncertainties in software engineering. The disturbances can make the behavior of the plant different from what the controller intended, so the controller should mitigate the effect of the disturbances. Many studies expect that the deep foundation of control theory will boost feedback loop design [14]–[17]. Therefore, this study also models and develops a multi-controller CPS using feedback loops based on this control perspective.

III. Autonomous Robot Vehicle Development

This section introduces the design of our case study to develop and analyze a multi-controller CPS. We developed an autonomous vehicle to provide a representative example of a multi-controller CPS. We utilized an open physical experiment environment *Platooning LEGOs* [18] that provides a programmable LEGO robot vehicle and an experimental track design[1]. Leveraging the physical experiment environment, we implemented our case study in Fig. 2. We developed an autonomous vehicle equipped with a lane-keeping system and an adaptive cruise control system. The vehicle observes its operational environment using a color sensor facing down (i.e., lane) and a distance sensor facing the front. The color sensor gives the light intensity value of the lane under the sensor. The value provides information about the vehicle's relative position from the lane center (i.e., the border between the white and black areas). In addition, there is an external vehicle in front of the autonomous vehicle, so the distance sensor gives the distance between the two vehicles. We assume that the external vehicle drives at a constant speed in this case study.

The autonomous vehicle has two explicit control systems and goals, as shown in Fig. 3. The control systems are modeled as decoupled feedback loops from the control perspective, and they operate together to achieve the two goals simultaneously. The first goal is to drive as smooth as possible following the center of the lane. The value of the lane center recognized by the color sensor accurately specifies this goal. The lane-keeping system observes the lane color of the current position. It calculates the error between the observed color value and the goal, and a steering controller decides the steering angle to keep the vehicle at the lane center. The second goal is to maintain the distance between the autonomous and the external vehicles to a set distance configured by the user. The adaptive cruise control system observes the distance and calculates the error from the goal. The speed controller sets the speed to minimize the error. Therefore, the steering angle and speed pair specify the vehicle's instant driving state.

We implemented the controllers as PID controllers [19]. A PID controller gets an observation value $o(t)$ (e.g., color or distance) from a sensor and calculates the error $e(t)$ for a goal. It returns a control value $y(t)$ (e.g., steering angle or speed) from the error $e(t)$ as follows: $y(t) = K_p e(t) + K_i \int_0^t e(\tau) \mathrm{d}\tau + K_d \frac{\mathrm{de}(t)}{\mathrm{dt}}$. In discrete system whose $t = 0, 1, 2, \ldots$, $y(t) = K_p e(t) + K_i \sum_{\tau=0}^{t} e(\tau) + K_d \Delta e(t)$, where $\Delta e(t) = e(t) - e(t-1)$. The three non-negative coefficients K_p, K_i, and K_d, each determines the degree of activation of different control mechanisms [19], configure a PID controller. We implemented

[1]Hardware implementation manuals of the robot vehicle and the FOT environment: https://github.com/KAIST-SE-Lab/Platooning-LEGOs

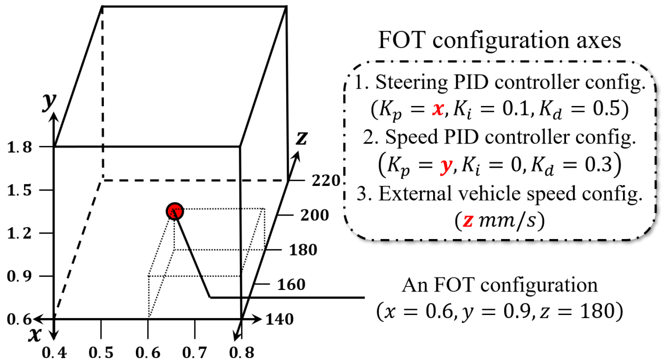


Fig. 4: Autonomous vehicle FOT configuration space

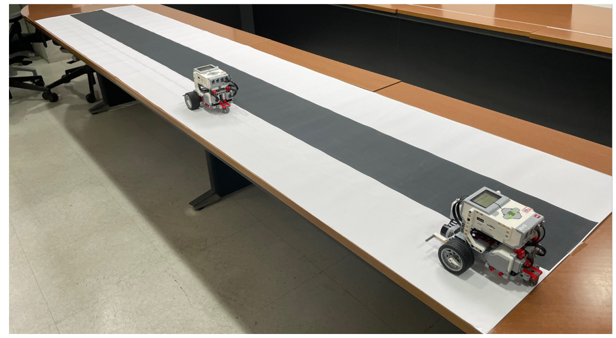


Fig. 5: Implemented robot vehicles and the FOT environment

the discrete PID controllers of the lane-keeping system and the adaptive cruise control system in Python embedded in the robot vehicle. The software iteratively calculates the steering angle and speed every 50 ms. It records $o(t)$, $e(t)$, $\sum_{\tau=0}^{t} e(\tau)$, $\Delta e(t)$, and $y(t)$ of both the steering and speed controllers. We released the controller software used in this case study.[2]

IV. Vehicle Field Operational Tests

To analyze autonomous driving, we conducted FOTs of the vehicle with numerous possible steering and speed controller configurations. We ran the vehicle FOT with varying independent variables that affect autonomous driving performance, as shown in Fig. 4. The configuration of the coefficients of the steering and speed PID controllers primarily affects driving performance. However, to limit the orthogonal configuration axes, we fix the K_i and K_d but only vary K_p of the controllers (x- and y-axes). In addition to the controller configurations, the environment is another factor that affects CPS goal achievement but is not under the direct control of the CPS. An external vehicle is a dynamic environment of an autonomous vehicle. Therefore, we also varied the constant speed of the external vehicle (z-axis) during FOTs. Based on a pre-experiment, we set our case study's configuration range and fixed coefficients, as shown in Fig. 4. Note that the configuration axes are continuous, so it is impossible to experiment with all possible configurations. We discretize the configuration range to five for each axis, so there are 125 ($= 5 \times 5 \times 5$) possible configurations of the autonomous vehicle FOT.

Fig. 5 shows the implemented robot vehicles and the FOT environment. The color goal was 33%, which is the value obtained when sensing the center of the lane in our experimental setting, and the safe distance goal between the two vehicles was 200 mm, which is longer than the length of a robot vehicle. The length of the lane is 3 m, and the distance between the tails of the two vehicles at the start of driving is 1 m. The two vehicles start driving simultaneously, and the experiment ends when the front vehicle arrives at the end of the lane. We keep the rest of the elements as consistent as possible, except for the independent variables under analysis (i.e., FOT configurations). However, since uncertainties may exist in the physical environment (e.g., sensor noise or non-uniform friction of the lane), we repeated the FOTs of all possible configurations in Fig. 4 50 times to obtain statistically significant results. The FOT dataset is available in our repository.[2]

[2]Controller software and FOT log data collected from this case study: https://github.com/est-cho/AV-FOT

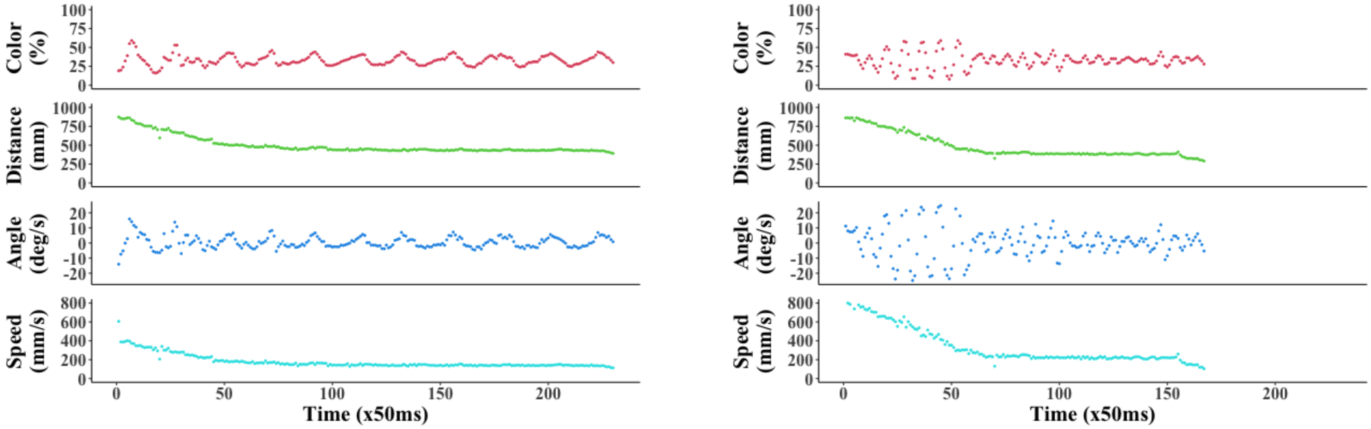
V. Field Operational Test Data Analysis

We conducted 50 FOTs for each configuration, taking about 100 hours, and collected 6,250 ($= 125 \times 50$) FOT logs. The volume of the dataset was about 80 MB. The raw data were released on our repository[2]. By analyzing the FOT logs collected by varying independent variables (i.e., configurations), engineers can understand the controllers of the CPS. This section describes the collected FOT logs by analyzing them from three viewpoints.

Viewpoint 1: Analyzing a single FOT result: The driving trace of an FOT is the time-series data of the variables described in Section III. Engineers may evaluate a vehicle's driving performance with a specific configuration by analyzing the time-series data. Fig. 6 visualizes two arbitrary FOT logs. It only visualizes the color and distance observation values, steering angle, and speed control values.

Since the FOT ended when the front vehicle arrived at the end of the lane, the lengths of the FOTs in Fig. 6 (a) and (b) differ depending on the external vehicle speed z. We also observed that the vehicle controllers continuously adapt the steering angle and speed during driving. Consequently, the observed values of lane color and front distance changed. In the log, we observed that the vehicle moves left and right to keep itself on the lane center as much as possible. In addition, after the autonomous vehicle caught up with the external vehicle, it drove while maintaining a safe distance from the external vehicle. We can observe that the change in configuration results in different shapes of time-series data. In addition, engineers can quantify the driving characteristics of a specific configuration, such as the time to catch up with the front vehicle and the amplitude of the fluctuation of the lane color [20].

Viewpoint 2: Analyzing the FOT results of a configuration: There are many FOT logs of the same configuration, so engineers can statistically evaluate the goal achievement



(a) Config. (x=0.4, y=0.6, z=140)

(b) Config. (x=0.8, y=1.2, z=220)

Fig. 6: Autonomous vehicle driving trace visualization

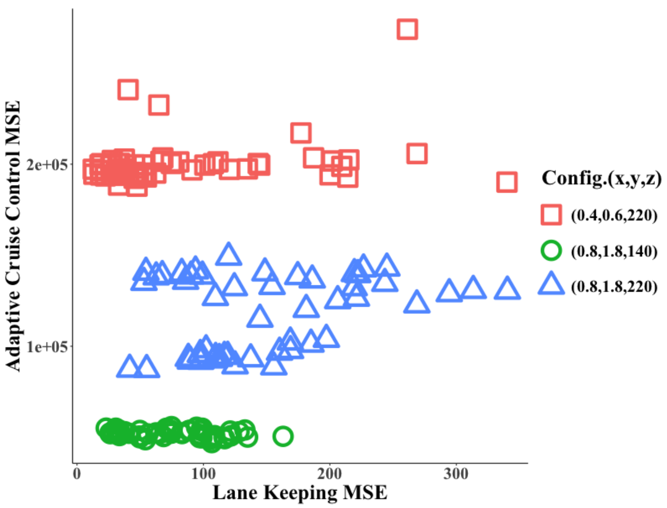


Fig. 7: Distribution of achievement of two autonomous driving goals obtained through repetitive FOTs

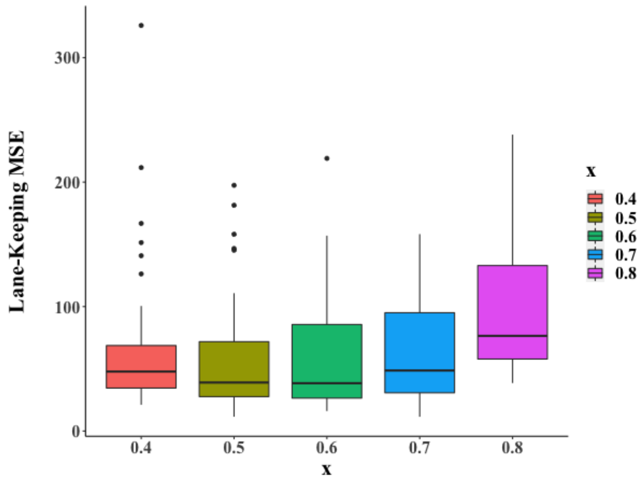
of the configuration. Fig. 7 shows the distribution of the two autonomous driving goal achievements (i.e., lane-keeping and adaptive cruise control) of three arbitrary configurations in terms of the mean squared error (MSE) of lane color and distance time-series data from the goals. A small lane-keeping MSE means driving close to the center of the lane, and a small adaptive cruise control MSE means catching up with the external vehicle quickly.

In Fig. 7, we can see that the FOT results were not always the same, even though engineers tested the same configuration. We tried to control other variables as much as possible, except configuration. However, uncertainties (e.g., sensor noise, the direction in which the vehicle was placed manually, or the remaining battery) still affected goal achievement. By analyzing this distribution, engineers could evaluate the consistency of many FOTs of the same configuration, thereby quantifying the degree of the uncertainties that affect the controllers. For example, configuration (0.8, 1.8, 140) appears to be less affected by such uncertainties than the other configurations shown in Fig. 7. In particular, we can see that the goal achievements are further dispersed by simply increasing the external vehicle speed while remaining in the other configurations. It shows that the degree of uncertainty of the FOT varies depending on not only the CPS's internal configurations but also the environmental configurations.

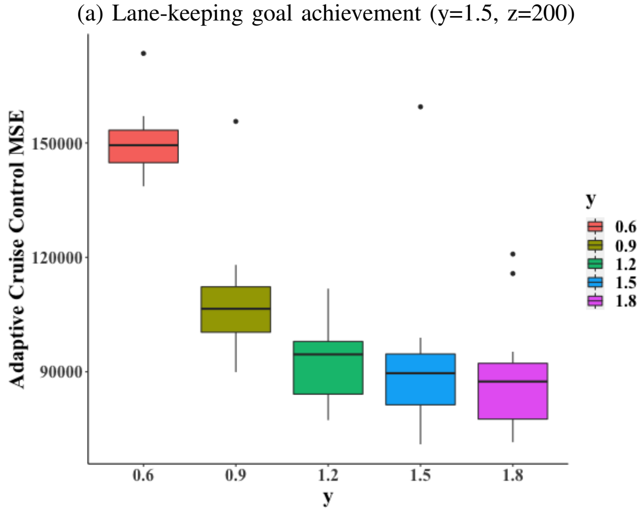
Viewpoint 3: Analyzing the FOT results of many configurations: Engineers can also explore changes in goal achievement by varying configurations to optimize the controllers of the autonomous vehicle. This allows engineers to understand how each configuration axis affects CPS's goal achievement. Fig. 8 shows how the goal achievements of the lane-keeping system and the adaptive cruise control system change with steering and speed controller configurations, respectively. Configuration axes that were not analyzed were arbitrarily fixed for visualization. Although many FOTs were not deterministic, we could statistically compare different configurations. In Fig. 8 (a), the steering controller whose K_p (x) was 0.6 performed the best on average when y was 1.5 and z was 200. In Fig. 8 (b), the adaptive cruise control system achieved its goal better as it increased the K_p of the speed controller (y) when x was 0.6 and z was 180.

Fig. 9 analyzes the errors of the two autonomous driving goals by simultaneously changing the two configuration axes. Subgraphs (a) and (b) show the MSEs for lane-keeping and adaptive cruise control, respectively. Additionally, the subgraphs also show when the speed of the external vehicle, which is an environmental factor, is 140, 180, and 220. A point in a 3D graph is the MSE average of 50 FOTs.

In Fig. 9 (a), as both x and y increase, the lane-keeping MSE generally increases. This means that the larger the K_ps of the steering and speed controllers are, the more the vehicle shakes left and right on the lane. Although y was a configuration variable of the adaptive cruise control system, it also affected the performance of the lane-keeping system. In



(a) Lane-keeping goal achievement (y=1.5, z=200)



(b) Adaptive cruise control goal achievement (x=0.6, z=180)

Fig. 8: Changes in the achievement of autonomous driving goals affected by configurations (one independent variable)

addition, shaking increased as the speed of the external vehicle increased. As shown in Fig. 9 (b), the adaptive cruise control MSE was primarily affected by the y configuration axis. The larger the y, the smaller the MSE. This finding shows that the autonomous vehicle could catch up to the external vehicle quickly and maintain the distance because the K_p of the speed controller was large. In addition, the faster the external vehicle, the harder it is to maintain the safe distance.

Based on this viewpoint, engineers can understand the trade-off between the goals of the autonomous vehicle and the goal achievements of each configuration. Finally, the controllers can be optimized based on this analysis and knowledge.

Engineers can analyze CPS's behavior and the controller configurations' impact on CPS goals with these various viewpoints. Through the analysis, the engineers obtain knowledge to understand CPS controllers. In addition, based on statistical analysis, many FOT results can provide statistically significant information to engineers.

VI. Lessons Learned

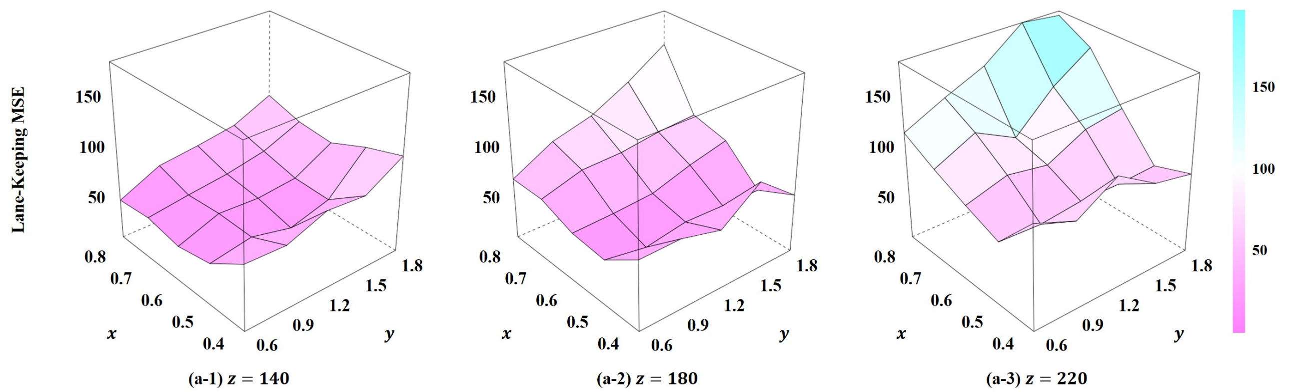
Although we brought the automotive domain to develop a multi-controller CPS and conduct FOTs, our experience in this paper revealed many domain-general engineering challenges. This section discusses the challenges for future research.

Expensive cost of FOTs: Running CPSs in actual operational environments is a time-consuming and exhausting task. Even though our robot vehicle was a simplified version of real vehicles, it took much time and effort to perform sufficient FOTs for multiple configurations of the vehicle controllers. The FOT cost of a CPS is proportional to the number of FOT configurations under test and the number of iterations required for statistical analysis. In addition, the FOT is sometimes dangerous. We also experienced accidents such as robots falling out of the track or external factors obstructing driving due to the tester's mistake. Although we can replace the FOT with high-fidelity simulation, the simulation relies on limited expert domain knowledge. In particular, we cannot omit FOTs of safety-critical CPSs, so methodologies for efficient and safe FOT are needed.

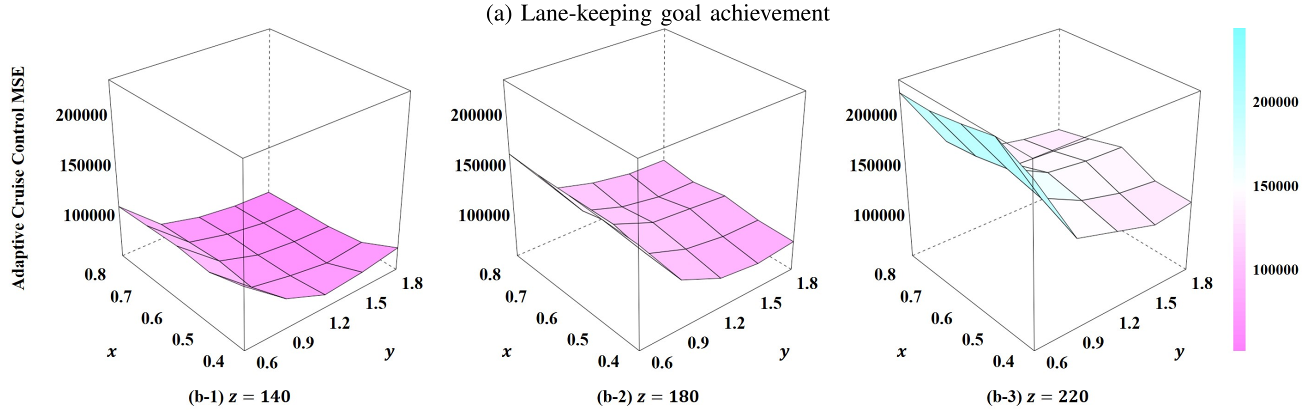
Uncertainties in multi-controller CPS FOTs: Engineers encounter two significant uncertainties in the FOTs of multi-controller CPSs. One is so-called environmental uncertainty in the physical environment [21]. The physical world itself is not stationary (e.g., changing weather, non-uniform friction of the road). The interaction between the CPS and the environment could be incorrect (e.g., sensor/actuator noise and failure). This degrades the reliability of a single FOT result and requires many FOTs for statistically significant results.

Another is uncertainty due to the unknown interdependence between different goals. CPS goals can influence each other in ways that engineers may not expect. A control decision for a specific goal may unintentionally contribute to or interfere with other goals. In our case study, when the adaptive cruise control system accelerates the vehicle to reduce the distance to the external vehicle, the error of the lane-keeping system increases. This makes it difficult to independently evaluate or optimize a particular controller through FOTs.

Difficulties of FOT log data analysis: Analysis of the FOT logs of the multi-controller CPSs has several challenges. Time-series data, which records CPS's operation frequently (e.g., every 50 ms in our vehicle), are substantial, so the technique should handle big data. Many features are also recorded in the FOT logs, so it is sometimes hard to find primary features related to a particular property of the CPS. In addition, because of the uncertainties of FOTs, the analysis results also vary depending on the given FOT logs. It also limits the reliability of the results to a certain confidence level. Finally, no matter how sufficient FOT data is collected, we should not misunderstand that all CPS behavior can be recorded in the data. The FOT log data only capture the observable features within the sensing capability of the CPS, so data analysis only provides partial information about CPS.



(a) Lane-keeping goal achievement



(b) Adaptive cruise control goal achievement

Fig. 9: Changes in the achievement of autonomous driving goals affected by configurations (two independent variables)

Therefore, information obtained by human inspectors is still meaningful and should not be overlooked.

VII. Possible Applications of Our FOT Dataset

We released the FOT logs collected from our case study[2] for future research on engineering for multi-controller CPS development and FOT engineering. This section introduces some possible applications of the FOT dataset.

Data-driven modeling of the CPS-environment interaction: Due to the interaction of the CPS and the environment, both CPS states and environmental states change over time. Accurate modeling of the interaction and its effect is the first step for an accurate CPS simulation to reduce the FOT cost [22], [23]. We can automatically extract valuable interaction models from our FOT log dataset [24], [25]. The FOT log shows sequential transitions of the CPS's internal data used for decision-making (e.g., speed and angle control values) and the environmental data observed by sensors (e.g., distance and color sensor values) every 50 ms. In addition, our dataset contains many FOT results of different CPS configurations, so it could also reveal the effect of the configurations on the interaction.

Quantifying uncertainties of multi-controller CPS: The uncertainties mentioned earlier stemming from CPS operation in a physical environment and multiple controllers' interdependence (emergent behavior) may cause the CPS to behave contrary to the engineers' expectations. To mitigate the uncertainties, the execution data of CPS may be analyzed further by quantifying uncertainties or extracting causes of variations in goal achievement within a configuration [26], [27]. To quantify uncertainty, enough sample data are needed to obtain statistically significant results from the analysis. Our FOT dataset presents 50 test results per 125 configurations, which provides expansive configuration space and ample test data.

CPS optimization based on data analysis: Although the CPS is expected to achieve its goals reliably, we have experienced that goal achievement significantly varies by the configurations of the internal controllers and the external environment. Unfortunately, engineers cannot accurately predict CPS behavior in the real world before runtime. Therefore, the runtime data can optimize the CPS for the operational environment [28], and related studies can use our dataset for this purpose. In particular, machine learning techniques for optimizing CPS configurations may use our dataset for training [29]. Real-time CPS monitoring and adaptation can also use our dataset by streaming the FOT logs [30], [31]. In addition, the FOT logs are actual multivariate time-series data, so there could be many possible applications [32].

Design of domain-specific FOT methodologies: Not limited to the specific applications described above, our dataset and hands-on experience in this paper can guide a domain-specific FOT methodology design [9]. In this paper, we focused on a scenario of repeating many FOTs over the configuration space of a multi-controller CPS. Although this scenario does not represent all situations, we believe that practitioners can design their own FOT methods based on the data and experiences described here.

VIII. Threats to Validity

The simplification of the autonomous vehicle as a robot threatens our case study. We tried to express the essential elements engineers would face in multi-controller CPS development as realistically as possible within a simplified experimental environment [18]. We developed two control systems (lane-keeping and adaptive cruise control systems) commonly installed in real cars. Our case study also includes an external vehicle that greatly influences safe driving in the real world. In addition, the format of the FOT log is the same as that of the actual digital tachograph [20]. Another threat is that the experimenters could bias the data we collected and disclosed. To reduce this threat, the authors repeated the FOTs individually for all possible configurations and aggregated the collected data.

IX. Related Work

As the complexity of the CPS increases, many CPSs are designed and developed as collections of multiple subsystems using SoS approaches. For example, an autonomous vehicle [33], [34], a smart factory [35], [36], and an industry robot [37], [38] are controlled by individual control systems or planners. However, it is hard to reproduce the case studies of the multi-controller CPS because of the sophisticated domain-specific knowledge required. This paper provides a re-implementable physical case study for the research community on the CPS from the SoS perspective. The implementation manuals of the physical part[1] and its software and the dataset[2] of our robot vehicle abstracting the actual car are available for future research. Researchers who do not have a multi-controller CPS experimental environment can consider using this case study and dataset.

This paper provides a real dataset obtained from a simplified robot vehicle. Many autonomous driving companies and research institutes have released datasets of their actual autonomous cars under development [39]–[42]. However, their datasets focus primarily on environmental sensor data, such as the camera, lidar, and radar data, for understanding the traffic situations an autonomous vehicle may encounter. They were released to build vast amounts of machine learning training data. Few datasets released vehicle bus data, such as steering angle and velocity [43], [44]. However, our dataset represents all sensor data, the internal data used for decision-making, and the actuator control data to show all of our hands-on experiences as much as possible.

X. Conclusion

We designed a case study to develop a multi-controller CPS and performed its FOTs. We implemented an autonomous robot vehicle equipped with a lane-keeping system and an adaptive cruise control system. Implementation manuals for the model, software, and hardware of the vehicle were provided. To analyze the various configurations of the two control systems, we discretized the continuous configuration space into 125 possible configurations and conducted 50 FOTs for each configuration. The collected FOT log dataset was analyzed and released. Based on our hands-on experience, we discussed the challenges of the multi-controller CPS FOTs and introduced how researchers can utilize our dataset for future research. In future studies, we also plan to provide an automated method to evaluate and optimize multi-controller CPS based on its FOT logs.

Acknowledgement

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2022-2020-0-01795) and (No. 2015-0-00250, (SW Star Lab) Software R&D for Model-based Analysis and Verification of Higher-order Large Complex System) supervised by the IITP(Institute of Information & Communications Technology Planning & Evaluation).

References

[1] R. Baheti and H. Gill, "Cyber-physical systems," *The impact of control technology*, vol. 12, no. 1, pp. 161–166, 2011.

[2] A. Filieri, M. Maggio, K. Angelopoulos, N. d'Ippolito, I. Gerostathopoulos, A. B. Hempel, H. Hoffmann, P. Jamshidi, E. Kalyvianaki, C. Klein *et al.*, "Software engineering meets control theory," in *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. IEEE, 2015, pp. 71–82.

[3] S. Shevtsov, D. Weyns, and M. Maggio, "Simca* a control-theoretic approach to handle uncertainty in self-adaptive systems with guarantees," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 13, no. 4, pp. 1–34, 2019.

[4] R. D. Caldas, A. Rodrigues, E. B. Gil, G. N. Rodrigues, T. Vogel, and P. Pelliccione, "A hybrid approach combining control theory and ai for engineering self-adaptive systems," in *Proceedings of the IEEE/ACM 15th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2020, pp. 9–19.

[5] M. Lee, K. Lee, C. Kim, and J. Lee, "Analytical design of multiloop pid controllers for desired closed-loop responses," *AIChE Journal*, vol. 50, no. 7, pp. 1631–1635, 2004.

[6] C. Guariniello, A. K. Raz, Z. Fang, and D. DeLaurentis, "System-of-systems tools and techniques for the analysis of cyber-physical systems," *Systems Engineering*, vol. 23, no. 4, pp. 480–491, 2020.

[7] M. J. de C Henshaw, "Systems of systems, cyber-physical systems, the internet-of-things… whatever next?" *Insight*, vol. 19, no. 3, pp. 51–54, 2016.

[8] L. Zhang, "Applying system of systems engineering approach to build complex cyber physical systems," in *Progress in Systems Engineering*. Springer, 2015, pp. 621–628.

[9] Y. Barnard and O. Carsten, "Field operational tests: challenges and methods," in *Proceedings of European Conference on Human Centred Design for Intelligent Transport Systems, Eds edn. HUMANIST publications, Lyon*, 2010, pp. 323–332.

[10] R. Alur, *Principles of cyber-physical systems*. MIT press, 2015.

[11] T. Patikirikorala, A. Colman, J. Han, and L. Wang, "A systematic survey on the design of self-adaptive software systems using control engineering approaches," in *2012 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. IEEE, 2012, pp. 33–42.

[12] S. Shevtsov and D. Weyns, "Keep it simplex: Satisfying multiple goals with guarantees in control-based self-adaptive systems," in *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*, 2016, pp. 229–241.

[13] R. De Lemos, H. Giese, H. A. Müller, M. Shaw, J. Andersson, M. Litoiu, B. Schmerl, G. Tamura, N. M. Villegas, T. Vogel *et al.*, "Software engineering for self-adaptive systems: A second research roadmap," in *Software Engineering for Self-Adaptive Systems II*. Springer, 2013, pp. 1–32.

[14] A. Filieri, M. Maggio, K. Angelopoulos, N. D'ippolito, I. Gerostathopoulos, A. B. Hempel, H. Hoffmann, P. Jamshidi, E. Kalyvianaki, C. Klein *et al.*, "Control strategies for self-adaptive software systems," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 11, no. 4, pp. 1–31, 2017.

[15] R. De Lemos, D. Garlan, C. Ghezzi, H. Giese, J. Andersson, M. Litoiu, B. Schmerl, D. Weyns, L. Baresi, N. Bencomo *et al.*, "Software engineering for self-adaptive systems: Research challenges in the provision of assurances," in *Software Engineering for Self-Adaptive Systems III. Assurances*. Springer, 2017, pp. 3–30.

[16] M. Litoiu, M. Shaw, G. Tamura, N. M. Villegas, H. A. Müller, H. Giese, R. Rouvoy, and E. Rutten, "What can control theory teach us about assurances in self-adaptive software systems?" in *Software Engineering for Self-Adaptive Systems III. Assurances*. Springer, 2017, pp. 90–134.

[17] J. Cámara, A. V. Papadopoulos, T. Vogel, D. Weyns, D. Garlan, S. Huang, and K. Tei, "Towards bridging the gap between control and self-adaptive system properties," in *Proceedings of the IEEE/ACM 15th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2020, pp. 78–84.

[18] Y.-J. Shin, L. Liu, S. Hyun, and D.-H. Bae, "Platooning legos: An open physical exemplar for engineering self-adaptive cyber-physical systems-of-systems," in *2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. IEEE, 2021, pp. 231–237.

[19] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum, *Feedback control theory*. Courier Corporation, 2013.

[20] T. Cherrett and D. Pitfield, "Extracting driving characteristics from heavy goods vehicle tachograph charts," *Transportation Planning and Technology*, vol. 24, no. 4, pp. 349–363, 2001.

[21] Y.-J. Shin, J.-Y. Bae, and D.-H. Bae, "Concepts and models of environment of self-adaptive systems: A systematic literature review," in *2021 28th Asia-Pacific Software Engineering Conference (APSEC)*, 2021, pp. 296–305.

[22] Y. Qin, C. Xu, P. Yu, and J. Lu, "Sit: Sampling-based interactive testing for self-adaptive apps," *Journal of Systems and Software*, vol. 120, pp. 70–88, 2016.

[23] W. Yang, C. Xu, Y. Liu, C. Cao, X. Ma, and J. Lu, "Verifying self-adaptive applications suffering uncertainty," in *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering*, 2014, pp. 199–210.

[24] D. Sykes, D. Corapi, J. Magee, J. Kramer, A. Russo, and K. Inoue, "Learning revised models for planning in adaptive systems," in *2013 35th International Conference on Software Engineering (ICSE)*. IEEE, 2013, pp. 63–71.

[25] Z. Ding, Y. Zhou, and M. Zhou, "Modeling self-adaptive software systems with learning petri nets," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 4, pp. 483–498, 2015.

[26] N. Bencomo, "Quantun: Quantification of uncertainty for the reassessment of requirements," in *2015 IEEE 23rd International Requirements Engineering Conference (RE)*. IEEE, 2015, pp. 236–240.

[27] N. Bencomo and A. Belaggoun, "A world full of surprises: Bayesian theory of surprise to quantify degrees of uncertainty," in *Companion Proceedings of the 36th International Conference on Software Engineering*, 2014, pp. 460–463.

[28] R. Al-Ali, L. Bulej, J. Kofroň, and T. Bureš, "A guide to design uncertainty-aware self-adaptive components in cyber–physical systems," *Future Generation Computer Systems*, vol. 128, pp. 466–489, 2022.

[29] S. Bandaru, A. H. Ng, and K. Deb, "Data mining methods for knowledge discovery in multi-objective optimization: Part a-survey," *Expert Systems with Applications*, vol. 70, pp. 139–159, 2017.

[30] X. Fei, N. Shah, N. Verba, K.-M. Chao, V. Sanchez-Anguix, J. Lewandowski, A. James, and Z. Usman, "Cps data streams analytics based on machine learning for cloud and fog computing: A survey," *Future generation computer systems*, vol. 90, pp. 435–450, 2019.

[31] Y.-J. Shin, E. Cho, and D.-H. Bae, "Pasta: An efficient proactive adaptation approach based on statistical model checking for self-adaptive systems," in *International Conference on Fundamental Approaches to Software Engineering*. Springer, Cham, 2021, pp. 292–312.

[32] W. W. Wei, *Multivariate time series analysis and applications*. John Wiley & Sons, 2018.

[33] R. B. Abdessalem, A. Panichella, S. Nejati, L. C. Briand, and T. Stifter, "Testing autonomous cars for feature interaction failures using many-objective search," in *2018 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2018, pp. 143–154.

[34] F. Camara, O. Giles, R. Madigan, M. Rothmüller, P. H. Rasmussen, S. Vendelbo-Larsen, G. Markkula, Y. M. Lee, L. Garach, N. Merat *et al.*, "Predicting pedestrian road-crossing assertiveness for autonomous vehicle control," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2098–2103.

[35] H. R. Faragardi, H. Fotouhi, T. Nolte, and R. Rahmani, "A cost efficient design of a multi-sink multi-controller wsn in a smart factory," in *2017 IEEE 19th international conference on high performance computing and communications; IEEE 15th international conference on smart city; IEEE 3rd international conference on data science and systems (HPCC/SmartCity/DSS)*. IEEE, 2017, pp. 594–602.

[36] M. Ciavotta, M. Alge, S. Menato, D. Rovere, and P. Pedrazzoli, "A microservice-based middleware for the digital factory," *Procedia manufacturing*, vol. 11, pp. 931–938, 2017.

[37] S. Zimmermann, G. Hakimifard, M. Zamora, R. Poranne, and S. Coros, "A multi-level optimization framework for simultaneous grasping and motion planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2966–2972, 2020.

[38] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Şucan, "Towards reliable grasping and manipulation in household environments," in *Experimental Robotics*. Springer, 2014, pp. 241–252.

[39] P. Wang, X. Huang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The apolloscape open dataset for autonomous driving and its application," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[40] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, L. Chen, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska, "One thousand and one hours: Self-driving motion prediction dataset," *arXiv preprint arXiv:2006.14480*, 2020.

[41] S. Agarwal, A. Vora, G. Pandey, W. Williams, H. Kourous, and J. McBride, "Ford multi-av seasonal dataset," *The International Journal of Robotics Research*, vol. 39, no. 12, p. 1367–1376, Sep 2020. [Online]. Available: http://dx.doi.org/10.1177/0278364920961451

[42] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 9710–9719.

[43] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth, "A2D2: Audi Autonomous Driving Dataset," 2020. [Online]. Available: https://www.a2d2.audi

[44] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.