Cyber-Physical Running Evaluation of a Tiny Driving Pod with Driving Schedule for a Driving System through the Tunnel

Min-Hwan Ok, Suyong Choi
Korea Railroad Research Institute/HTX Research Team, Uiwang, South Korea
Email: panflute00@naver.com

Kwan-sup Lee
KRRI/High-speed Rail-system Research Division, Uiwang, South Korea
Email: suchoi@krri.re.kr

Abstract— **For the cars driving on the tunnel-way in the future, a miniature-scale experiment is conducted using a driving robot with combinations of basic control-functions. In this work, the aim is to devise the driving control function for a driving robot, as the tiny driving pod machine, and the guidance function of a directing server, in a given route area. Acceleration/deceleration is performed with altering wheel rotations for the robot's drive on the route. The robot is driven by drive commands of a driving schedule within a predefined time variance along the scheduled route. The definite travel-time basis benefits from the control technique of the design of a cyber-physical vehicle system, and is surmised to be a fundamental property of car-driving on the tunnel-way.**

Index Terms— **Automatic driving, Driving schedule, Tiny driving pod, Driving through the tunnel, Miniature-scale experiment**

I. Introduction

The occurrence of unexpected phenomenon, such as fire, in the tunnel incurs dangerous situation for whom traversing through the tunnel. As there are tunnels with no sensors against fire, it is necessary to check the inside of the tunnels by tiny driving pod, an automatic driving vehicle, equipped with a set of sensors for detecting an outbreak of fire. To carry a set of sensors in the tunnel, the driving controller of a tiny driving pod is primarily required. In this work, the aim is to devise the driving control function for a driving robot, as the tiny driving pod machine, and the guidance function of a directing server, in a given route area. This work has a particular contribution under the premise of a promising future car-driving environment that many of cars are driving beneath the ground in urban regions [1]. The predetermined route of the driving robot is traveled on the unaltered route basis within the fixed travel time. The route calculation function of the directing server designates the predetermined travel route for each driving robot considering the merge of two routes. The robot is driven by drive commands of a driving schedule within a predefined time variance along the scheduled route. The current speed, position, and steering angle are reported in the driving, and this information is displayed on the directing server. There is a route / situation screen that can alter to abort the schedule against an abnormal situation in the current driving. The driving test performed in this work is a miniature-sized implementation of the driving controller of the tiny driving pod which can be realized in the future.

Since the actual automatic driving should be exact to the scheduled route, we adopted a control technique proposed in [2] on which vehicle is presupposed as Cyber-Physical System, CPS, for the purpose of automatic driving control. In a miniature-scale experiment the exactness is met with a requirement for the case of basic running test.

II. Cyber-Physical Vehicle System

For Cyber-Physical Vehicle Systems, CPVS, in the most general sense, control or regulation makes use of algorithms and feedback to calculate inputs for cyber and physical effectors that provide services, translocate or reorient the vehicle, ensure safety and achieve objectives. Holistic, integrated, tightly-coupled CPS modeling and control of both cyber and physical effectors are referred to as co-regulation [2]. Most of this section is referred to a survey of optimization and control of CPVS [2].

A. Feedback Scheduling

Feedback, as a principle, offers robustness to off-nominal conditions by using past measurements to compute future inputs to the system and has been applied in many domains. Most related to control of CPVS, and a departure from control of the physical system, is feedback scheduling. Feedback scheduling adjusts cyber resources based on the needs of the cyber system [3]. It accomplishes this by adapting traditional control theory

Manuscript received July 15, 2017; revised August 16, 2017.
Project number PK1701A; corresponding author is the first one.

doi: 10.18178/joace.5.2.53-56

to regulate the task schedule in the RTS. This, in turn, contributes to regulating the CPS as a whole. In this scheme, sampling periods of various control tasks are adjusted, and subtasks (parts of a task) are scheduled using feedback from execution time measurements and feedforward from workload changes [4].

Feedback scheduling algorithms can be computationally intense. A simple (i.e., linear) model that relates the cost of control performance to cyber resources would provide an excellent tool for feedback scheduling, which can then less expensively design task schedules [5].

B. Time-Varying Sampling

Uncertainty in sampling rate can be caused by transmission delays in a NCS, jitter and/or missed deadlines in the RTS, etc. Research investigating the design of controllers under uncertain delays has resulted in more robust systems. Typically, as in NCS research, these approaches consider a small range of possible sampling rates and stability, and robustness guarantees are given for that range under time-varying control schemes [6]. Successful optimal controllers under these circumstances using a linear matrix inequality (LMI) approach have been designed [7,8].

C. Coupled Cyber-Physical Co-regulation

Assuming a physical system modeled as:

$$\dot{x}_p = A_p x_p + B_p u_p. \tag{1}$$

where x˙_p is the physical state vector, Ap the system matrix, Bp the control matrix and up the physical control input, we seek a system of the form:

$$\dot{x}_c = A_c x_c + B_c u_c. \tag{2}$$

where components with subscript c are the cyber system analogs to the physical model components.

This allows us to write the coupled CPS as:

$$\Sigma_{CPS}\colon \left\{ \begin{bmatrix} \dot{x}_p \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} A_p & \\ & A_c \end{bmatrix} \begin{bmatrix} x_p \\ x_c \end{bmatrix} + \begin{bmatrix} B_p & \\ & B_c \end{bmatrix} \begin{bmatrix} u_p \\ u_c \end{bmatrix} \right. . \tag{3}$$

This co-regulation scheme has illustrated CPS tradeoffs possible over a series of domains including a spring-mass-damper system [9] and inverted pendulum [10], as well as the CubeSat domain [11].

In all cases, co-regulation consistently demonstrated good physical system tracking performance, while significantly reducing computational load. This abstraction approach to CPS co-regulation allows an engineer to leverage the wealth of traditional state-space control design techniques and to treat the scheduling of tasks as a control problem wherein interactions between cyber and physical states are represented in a common framework. It also provides the benefits of time-triggered control, such as ease of RTS scheduling and hard timing guarantees, while also offering the benefits of "on-demand" event-triggered control to reduce cyber resource utilization.

III. Composition of the Tiny Driving Pod

D. Driving Control Function for a Driving Robot

Stella B2 [12], a driving robot platform capable of running various driving algorithms in an indoor environment, is used and its driving is controlled by the driving speed and travel distance from the start position. The travel of the driving robot is carried out by the schedule based driving.

Figure 1. Driving robot as the tiny driving pod.

The functionalities of the entities are enlisted in Table I.

TABLE I. Entity Functionality List

Feature	Functionality Detail
Directing Server	Calculating Scheduled Route Telecommunication Channel with the Drive Controller Tracing the Driving Robot according to its Scheduled Route
Drive Controller	Sequencing commands along the Scheduled Route Telecommunication to the Directing Server
Definite Travel Time	Declaring Run Path Composing Scheduled Route
Driving in Hard Real-time	Start time, Stop time, Velocity Adjustment Keeping-up the Scheduled Route
Reporting Status	Driving states of the Driving Robot
Status Screen	Velocity, Position, Wheel Rotation-difference, etc.

E. Guidance Function of a Directing Server

A driving-control computer that controls the driving robot is placed on the top of the robot, and receives instructions from and sends reports to the directing server computer. As shown in Fig. 2, the two computers are connected to each other through a wireless network. In the case of instructing the directing server sends the instruction and receives the report in the case of reporting. The driving-control computer sends a driving command to the driving robot through RS232C communication, and the driving robot returns the result on the driving. Since there would be dozens of driving robots, the driving-control computer periodically reports the driving state so that the directing server manages the driving states of all the robots travelling. The driving-control computer executes the driving command sequences according to the driving schedule within admitted the time variance, by steering and forwarding with speed control of the driving robot. The directing server computer confirms whether each driving robot conforms to the driving schedule. The driving schedule is created by calculating the travel time of the route between the driving robots on the definite travel-time basis.

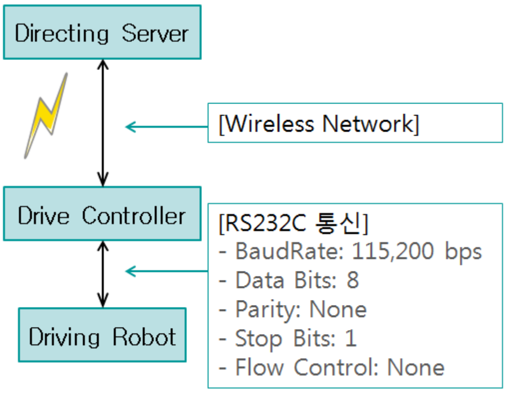

Figure 2. System structure.

F. Driving Control Function for a Driving Robot

1) Drive Commands.

These commands drive the robot straight-forward, stopping, accelerating, and turing.

- Speed command1 (F: Forward, B: Backward) (000 ~ 270 rad/sec)
- Speed command2 (F: Forward, B: Backward) (000 ~ 270 rad/sec)
- Speed command3 (F: Forward, B: Backward) (000 ~ 270 rad/sec)
- Speed command4 (F: Forward, B: Backward) (000 ~ 270 rad/sec)
- Position command (A: Angle [0~360 Degree], D: Distance [000 ~ 999cm], T: Time [0 ~ 255 Second])
- Stop command (1 ~ 3)
- Reset
- Initialization

2) State Commands.

These commands get states of the robot in its driving.

- State Value
- Current Speed
- Current Position
- Rotation Scale Factor (000~999)
- Driving Scale Factor (000~999)
- Acceleration (rad/sec2)
- p, i, d, il Value (000.000~999.999)
- CAN ID

3) Control Commands.

These commands set values for driving of the robot.

- Rotation Scale Factor (000~999)
- Driving Scale Factor (000~999)
- Acceleration (rad/sec2)
- p, i, d, il Value (000.000~999.999)
- CAN ID (001~255)

IV. EXPERIMENTATION

In the miniature-scale experiment, the driving-control computer commands the driving robot along the scheduled route. When the driving robot notifies the instruction server that it is ready to travel, the scheduled route as shown in Fig. 3 is transmitted. The driving-control computer loads the scheduled route and sends a driving command sequence to the driving robot in accordance with the driving schedule within the predefined time variance. The returned result is logged at the driving-control computer as shown in Fig. 4, and the state information of the driving robot is transmitted to the directing server. After the basic driving-control functions are verified, we tested driving along scheduled route on the definite travel-time basis as shown in Fig. 5. Continuous driving of the curves, speed change by the running section, stopping at the stop, and timely departure were executed according to the schedule.

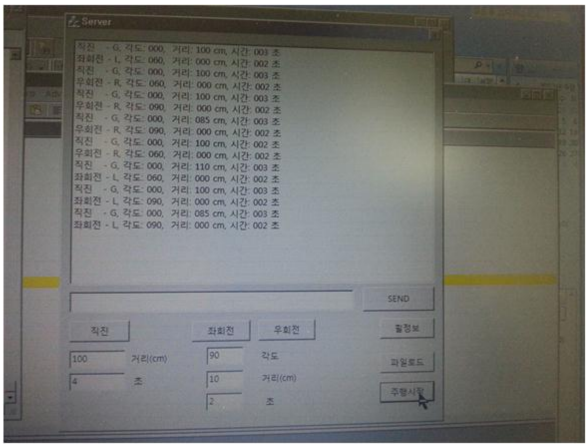

Figure 3. Scheduled route to be loaded.

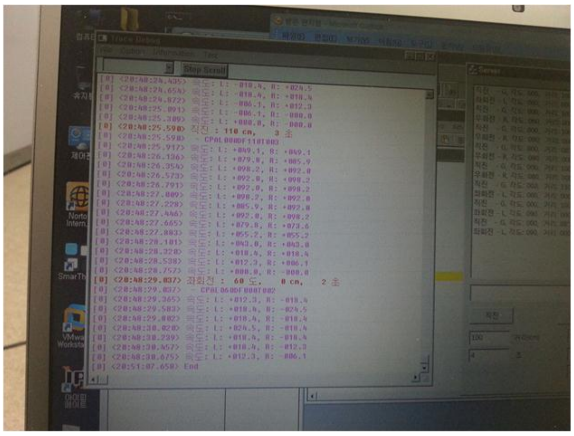

Figure 4. Results logged at the driving-control computer.

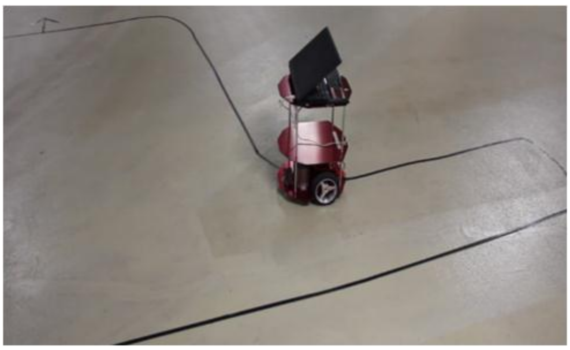

Figure 5. Driving test on the definite travel-time basis.

Throughout the basic running test, we confirmed the driving robot has run in the definite travel-time. The exact run time could be accomplished by non-delayed driving, velocity adjustment before a turn, and accurate turning.

V. Conclusion

For the cars driving on the tunnel-way in the future, a miniature-scale experiment is conducted using a driving robot with combinations of basic control-functions. A control technique of the design of a cyber-physical vehicle system is adopted in this basic run test, and we have confirmed the definite travel-time could be applied for driving on the tunnel-way. This definite travel-time basis benefits from the control technique of the design of a cyber-physical vehicle system, and is surmised to be a fundamental property of car-driving on the tunnel-way.

The company[1] has furthered its detail plans, which include an underground network of tunnels for both short and long-distance travel, similar to a modernised and higher-speed version of the London Underground and other city metro systems. Although in some important aspects such as safety and security, the tunnel-way is yet to be proven gradually in its steady advance. Despite its steady advance, it is a prominent transportation candidate in futuristics.

Acknowledgment

This research was supported by a grant from R&D Program of the Korea Railroad Research Institute, Republic of Korea.

References

[1] The Boring Company homepage. *FAQ.* [Online]. Available: https://www.boringcompany.com/.

[2] J. M. Bradley and E. M. Atkins, "Optimization and control of cyber-physical vehicle systems," *Sensors*, vol. 15, pp. 23020-23049, 2015.

[3] L. Sha, T. Abdelzaher, K. Årzén, A. Cervin, T. Baker, A. Burns, G. Buttazzo, M. Caccamo, and J. Lehoczky, "A. real-time scheduling theory: A historical perspective," *Real-time Syst*., vol. 28, pp. 101-155, 2004.

[4] A. Cervin, "Integrated control and real-time scheduling," Ph.D. dissertation, Lund Univ., Lund, Sweden, 2003.

[5] J. Eker, P. Hagander, and K. Årzén, "A feedback scheduler for real-time control tasks," *Control Eng. Pract*., vol. 8, pp. 1369-1378, 2000.

[6] E. Fridman, A. Seuret, and J. P. Richard, "Robust sampled-data stabilization of linear systems: An input delay approach," *Automatica*, vol. 40, pp. 1441-1446, 2004.

[7] A. Sala, "Computer control under time-varying sampling period: An LMI gridding approach.," *Automatica*, vol. 41, pp. 2077-2082, 2005.

[8] M. Schinkel, W. H. Chen, and A. Rantzer, "Optimal control for systems with varying sampling rate," in *Proc. 2002 American Control Conf.*, 2002, pp. 2979-1-2984.

[9] J. M. Bradley, E. M. Atkins, "Computational-physical state co-regulation in cyber-physical systems," presented at the ACM/IEEE Conference on Cyber-Physical Systems, Chicago, IL, Apr. 12-14, 2011.

[10] J. M. Bradley and E. M. Atkins, "Toward continuous state-space regulation of coupled cyber-physical systems," *IEEE Proc.* vol. 100, pp. 60-74, 2012.

[11] J. M. Bradley and E. M. Atkins, "Coupled cyber-physical system modeling and coregulation of a cubesat.," *IEEE Trans. Robot*, vol. 31, pp. 443-456, 2015.

[12] NTRex homepage. Mobile platform Stella B2. [Online]. Available: http://www.ntrexgo.com/archives/8692.

Min-Hwan Ok received his M.S. degree in Computer Science from Pusan National University, Pusan, Korea, in 1999, and Ph.D. degree in Computer Science at Korea University, Seoul, Korea in 2009. Since 2004, he has been a senior researcher in Korea Railroad Research Institute (KRRI). His current research interests include autonomous vehicle driving, autonomous drone flight, and hyperloop.

Suyong Choi received his B.S. degree in Mechanical Engineering from Pusan National University, Pusan, Korea, in 2011, and Integrated Master and Ph.D. degree in Nuclear and Quantum Engineering from KAIST, Daejeon, Korea in 2016. Since 2017, he has been a senior researcher of Hyper Tube eXpress Research Team at Korea Railroad Research Institute (KRRI). His current research interests include wireless power transfers, linear motors, and hyperloop.

Kwan-sup Lee received his B.S. degree in Mechanical Engineering from Hanyang University, Seoul, Korea, in 1982, and M.S. degree in Mechanical Engineering from KAIST, Daejeon, Korea in 1995. He received his Ph.D. degree in Control and Instrumentation Engineering from BMSTU in 2002. In 1997, he has joined in Korea Railroad Research Institute (KRRI). He has been a chief researcher of Hyper Tube eXpress Research Team since 2015. His current research interests include hyperloop and CBM.