Project-Based CPS Education: A Case Study of an Autonomous Driving Student Project

Niklas Ueter, Kuan-Hsun Chen, and Jian-Jia Chen
Department of Informatics, Technical University of Dortmund, Germany
https://ls12-www.cs.tu-dortmund.de/

Citation: 10.1109/MDAT.2020.3012085

BibTeX:

```
@article{9149697,
    author={N. {Ueter} and K. {Chen} and J. {Chen}},
    journal={IEEE Design  and Test},
    title={Project-Based CPS Education: A Case Study of an Autonomous Driving Student Project},
    year={2020},
}
```

Project-Based CPS Education: A Case Study of an Autonomous Driving Student Project

Niklas Ueter, Kuan-Hsun Chen, and Jian-Jia Chen
Design Automation of Embedded Systems Group
Technische Universität Dortmund
Otto-Hahn-Straße 16, 44227 Dortmund, Germany
Telephone: +49(231) 755-6078, Fax: +49(231) 755-6116
{niklas.ueter, kuan-hsun.chen, jian-jia.chen}@tu-dortmund.de

***Abstract*—The classic lecture and exercise based teaching is the predominant way to educate computer science and engineering students at university. This is partly due to the time constraints resulting from the extensive curricula. We show that project-based cyber-physical systems (CPS) education (based on a relatively extensive and complex engineering problem) helps the students to learn transferring their acquired fundamental knowledge to real-world application, and learn how to handle non-idealized problems. In this paper, we explain the educational concepts, theoretic foundations, and report the students' results of our autonomous driving project-based course. We show the students achieved, e.g., management framework, simulation environment, navigation algorithms, etc. To evaluate the conjecture of our proposed concept, we review the anonymous ratings conducted by the university faculty and discuss the final results.**

***Index Terms*—Cyber-Physical Systems, Education, Engineering, Robotics, Autonomous Driving, Student Projects**

I. Introduction

Cyber-Physical Systems (CPS) are systems at the interface of the physical world and the cyber world, e.g., robots, aircrafts, factories or automobiles. In consequence, CPS engineers have to be familiar with general engineering methods and models in both domains, e.g., from mechanical engineering, electrical engineering and computer science. Notably, engineering is predominantly concerned with the methodological synthesis of systems, i.e., the design and realization of systems that adhere to specified functional and technical properties in a methodological and structured procedure. In order to design and synthesize complex systems, engineers have to decompose the holistic system into appropriate subsystems such that the properties of the integrated system can be synthesized.

We observed that structuring and decomposing the system into subsystems and dealing with the additional complexities of real-world implementations are the most challenging task for engineering students. This kind of learning experience is underrepresented in the curriculum of computer science and engineering. Moreover, when engineering students are confronted with technical challenges, they can often not address these by directly using solutions and methods that they have learned in university lectures. This is due to the additional complexity of large systems and the rapid technological changes in the state-of-the art. This forces the students to understand novel techniques and solutions as well as adapting and applying these techniques to their given problems or develop novel solutions based on their acquired fundamental knowledge. Furthermore, the students will have to work in larger teams to solve real-world engineering problems that encompass different engineering and business domains.

Besides introducing dynamic models describing physical environment [7], various approaches have been attempted in recent years to improve CPS education. That is for example the flipped (or inverted) classroom approach [8], educational CPS platforms [3], or shifting lectures to a first-year course [1]. However, all these approaches do not emphasize the learning of how to approach and solve a complex engineering problem. Project-based embedded system education has been studied by, e.g., [6], [11] with an emphasis on FPGA platforms. Recently, Kim et al. [4] proposed labs and projects by providing a series on application based courses for smart city CPS systems. Although [4] provides plenty of information about the used platforms and implementations, the educational principles that the lecture is development from are not discussed. Moreover, the design of this lecture is still based on the conventional split of lectures and hands-on lab assignments.

In contrast, we try to extend the aforementioned concepts, revisit the project-based learning concept and review our experience of a project-based course that was developed at our university.

Our Contribution: In this work, we propose a project-based CPS educational concept and present our designs and experiences.

- We elaborate on the education principles and theoretic foundations that are used to design this project course.
- Moreover, we report the supervisors' and students' observations, results and experiences. All results and more detailed reports can be found in the Github repository[1].
- We review the course evaluation by students and present the technical results of the project over two semesters.

II. Theoretic Foundations of our CPS Education

In this section, we shortly survey and summarize the skill-set that is considered to be fundamental to cyber-physical systems engineering and is used as a guideline for our project

[1] https://github.com/Autonomous-Racing-PG/ar-tu-do

Fig. 1: The remote-controlled car based on the Traxxas Fiesta ST platform after modifications and augmentation of sensors (without the chassis).

based CPS education program. In *A 21st Century Cyber-Physical Systems Education* [9], cyber-physical engineering is stated to include the following domains:

- Computing Theory
- Physical Modeling
- Discrete and Continuous Mathematics
- Sensing, Actuation, Control, Communication, Computing
- CPS System Development Principles

This implies that CPS engineers should be familiar with the fundamentals of computer architecture and computer organization. Moreover, basic knowledge in data structure, algorithms, and different models of computation, e.g., discrete event systems and automata theory are important to efficiently implement and engineer computing systems.

Foundations in computing are required to comprehend the properties and constraints that the physical world poses when interfacing both domains as well as to consider the resource limitations of the computing platforms in the software designs and implementations. Due to the fact that CPS inherently encompass continuous and discrete systems and the combinations thereof, knowledge in continuous as well as discrete mathematics are mandatory for all CPS engineers. Furthermore, CPS engineers have to understand the cross-cutting application of sensing, actuation, control, communication, and computing. This foundation is essential due to the cross-cutting focus of CPS between the physical and cyber aspects of systems, as well as control over communication networks, sensing,signal processing, and actuation with real-time constraints. Another important aspect in the education of engineers is the system development process, i.e., the steps from initial requirements to final system certification with an emphasis on safety-critical systems, robustness, and resiliency. That means, education in safety, resilience, security, and privacy, requirement development, hazard analysis, testing, formal verification and validation.

III. Project Design and Objective

In this section, we motivate the choice and design of our project as well as our educational intentions. This project allows the students to get in contact with all relevant areas of modern cyber-physical systems, e.g., real-time systems, automatic control, and complex data processing in integrated and constrained computing environments. The main objective of this project is to make a modified remote controlled car drive autonomously over a mock racing track as fast as possible and to be able to react to dynamic obstacles. Our long term vision is to use this project to interest the students in autonomous robotics and to initialize an F1/10 Autonomous Racing team from TU Dortmund to participate in the competition[2]. The students are provided all necessary hardware components such as a remote-controlled car in the scale 1:10 based on the *Traxxas Ford Fiesta ST* platform, sensors, racing tacks and an experimental environment.

The car is 535mm in length and 281mm wide with a net weight (without sensors) of 2.7 kg. In addition, the car features a four-wheel drive with open differential in the front and rear axis. Furthermore, the following components were augmented:

- NVIDIA Jetson TX2 System-on-Module
- Orbitty Carrier
- FOCBOX Motor Controller
- LIDAR Scanner (Hokuyo UST-10LX)
- Inertial Measurement Unit (Invensense MPU-9250)
- Traxxas TRX3350R Velineon BLDC
- Stereo Camera (ZED)

After all sensors were augmented, the chassis was customized to fit all sensors as is shown in Figure 1. Additionally, a new camera bracket was 3D printed in order to give the wheels more leeway and to and allow for easy clip attachment of the chassis. Based on this hardware platform, the students are supposed to design, build and test their autonomous driving strategy. More precisely, this entails to use the robot operating system (ROS) and integrate various available sensors, i.e., inertial-measurement units, accelerometer, cameras, and the LIDAR. The students should realize algorithms for localization, mapping, path planning, and automatic control.

A. Students & Prerequisites

The project team is made up of twelve graduate (master) students of computer science with different minors such as physics, mathematics, mechanical engineering or electrical engineering. Each student is supposed to be spending 14 hours per week for the project over a course of one year. On admission, we require the students to have the following skill-set that we consider mandatory such that the students are capable to further educate themselves on relevant topics and can be productive within the limited time frame. That is, we expect the students to have at least intermediate programming knowledge in any programming language, e.g., Python, C, C++, or Java. Moreover, the students should have basic knowledge in mathematics and in statistics and probability theory particularly. Lastly, we ask for basic knowledge in Linux.

We informed the applicants that knowledge in C++ programming, knowledge in Robot Operating System (ROS)

[2]The F1Tenth committee (http://f1tenth.org/) hosts an international competition every year, co-located with CPS-Week and ES-Week conferences.

programming, knowledge in sensor data and digital signal processing as well as knowledge in robotics and control are beneficial but not mandatory since we expect our students to be capable to learn required techniques and theories during the project phase. We asked applicants to inform us about their academic interests and specialized knowledge. Using this information, we assembled a team that consisted of students with expertise in different areas such that efficient collaboration in a heterogeneous group can be practiced as this is the common case in a real-world engineering environment.

B. Project Schedule

In order to give the students a guideline for the milestones and finished artifacts within a given time frame, we proposed the schedules shown in Table I and II for the first and second semester respectively. Nonetheless, the students are given the opportunity and are explicitly encouraged to create and manage their own project schedule in accordance with the organizers. In the following, we shortly explain our rationals behind this schedule design and elaborate on what we expect the students to do in every stage of the project.

Seminar Phase: The project is initialized with a seminar phase, which means that all participating students are asked to present a topic that is to be chosen from a provided list of topics. We proposed the following list of seminar topics for which we included literature, online coursework material, and lecture videos:

- *Project management*: The student should present the basics of project management for software projects. More specifically, the different roles of project participants, project stages, and organization of project based work should be presented. Moreover, the management of project resources, continuous replanning of project objectives, documentation and reporting should be explained. Lastly, the psychological component of project based work should be discussed to sensitize the students to potential conflicts that are rooted in different human characters.
- *Recursive State Estimation*: In this seminar topic, the problem of state estimation should be motivated and explained with respect to the context of this student project. In addition to that, the Bayes filter as well as the Kalman filter as a special case should be explained. Moreover, the assumptions, advantages and disadvantages of the proposed models should be explained.
- *Sensors and Actors of the remote-controlled car*: Explain how the sensors work and the frame of reference, e.g., gyroscope, accelerometer, and magnetometer. Furthermore, the actors, steering and drive train.
- *Simultaneous Localization and Mapping*: The SLAM problem should be explained with respect to the context of the project. In addition the Hector SLAM algorithm should be presented.
- *Kinematics and Dynamics*: Identify the kinematics of the remote controlled car and the physical forces and phenomena are relevant for this project.
- *Automatic Control & Model Predictive Control*: Introduction and motivation of automatic control theory. Especially, what are the models, objectives, and methods used. Moreover, the specialties of optimal and model predictive control should be introduced to the students.
- *Frameworks*: Given the NVIDIA Jetson TX-2 platform and the ROS environment, CUDA programming, development with ROS and Gazebo (http://gazebosim.org/) should be demoed and presented to the students (preferably by examples).
- *Machine Learning*: Some fundamental ideas and techniques of machine learning should be explained. Furthermore, some approaches should be identified to be useful for the project.
- *Coordinate Systems and Transformations*: In this seminar topic, coordinate systems and transformations are expected to be explained. Namely, the student should explore the different frames of reference, i.e., body and world frames. Furthermore, rotations, translations, homogeneous coordinates, and Euler angles should be presented. In addition, it should be explained how the description of dynamics relate to each other with respect to the different frames of reference, e.g., Coriolis force.

We notified the students of the required skills that we considered mandatory to understand the material and give students an easier chance to select a suitable topic. The intention of the seminar phase is twofold. Firstly, we want to make sure that all participating students get an overview of all engineering problems and first impressions of some solutions to those problems that are relevant to the successful completion of the project. Secondly, we want that each student becomes an *expert* on the seminar topic that they have presented and is thus able to direct and lead small teams in that problem area. To maximize the benefit for the group, we asked the students to present their work in front of the others as well as to prepare a paper for documentation.

Introductory Phase: During the *introductory phase*, all participants will work through ROS tutorials and study the existing software to familiarize themselves with the software architecture, workings of the provided sensors and the RC-car. In later phases of the project, smaller groups can be built to specialize on one or more of the above mentioned topics to solve the associated problems. With the finishing of this *introductory phase*, the students should choose a project management model, prepare a roadmap and a project schedule.

Analysis Phase: In the sixth week, the *analysis phase* starts, in which all relevant sub-problems to the overall goal of competitive autonomous driving are addressed and analyzed. These may include, local online trajectory planning, sensor data processing, control or machine learning approaches depending on the choices made by the project members.

Concept and Design Phase: After that, the students have six weeks to develop a prototype that addresses all previously analyzed requirements and is capable of autonomous driving.

Validation Phase: This prototype should be validated and tested in simulations as well as in real scenarios during a three

Task	Weeks
Introductory Phase	**6**
Seminar Stage	2
Outline a project schedule	1
Tutorials and Problem Identification	3
Analysis Phase	**20**
Identify and model relevant robot dynamics	5
Identify challenges of autonomous driving	5
Identify problems, usable approaches and algorithms	5
Analyze race tracks and other race relevant objectives	5
Concept and Design Phase	**24**
Control and Trajectory planning	6
SLAM	6
Sensor processing and Fusion	6
...	6
Validationphase	**3**
Demonstration and mock race	3
Documentation	**13**

TABLE I: Suggested project schedule for the first semester.

Task	Weeks
Consolidation phase	2
Implementation phase	10
Test phase	7
Demonstration	3
Documentation	16

TABLE II: Suggested project schedule for the second semester.

week period. At the end of the first semester, the students are obliged to prepare a midterm report to document their results, e.g., developments, concepts, implementations and prototype.

Consolidation Phase: The second semester mostly consists of refinements, improvements as well as further testing of the developed prototype. The project is finished if the final report is submitted and the final presentation and demonstration was given. During the whole project duration, the students should increasingly work independently of the supervisors and manage the internal groups themselves according to project management standards.

The aforementioned problem statement and solution approaches, e.g., motion planning, perception and control can be improved and refined freely. The students can for example experiment with machine learning techniques to improve lane control, autonomous behavior or obstacle detection.

IV. Achievements and Experiences

In this section, we report the students' project management, milestones, results and experiences. In order to achieve the abstract objective of *autonomous driving*, the students defined a set of concrete objectives that were adapted continuously according to internal processes. The main objectives that the group members decided on are stated below:

- Show navigation capabilities in a simulated environment, i.e., Gazebo (minimal requirement)
- Show autonomous driving capabilities in the real-world (minimal requirement)
- Participate in the official F1Tenth competition (optional requirement)
- The car should be able to be stopped at any time (additional requirement)
- Manual control to overwrite the autonomous driving should be possible any time (additional requirement)
- The car should be able to detect and detour static obstacles (additional requirement)

The preliminary structure that the students decided on is detailed in the following. The first initial objective was to implement and adapt the wall-following algorithm from the official tutorials to drive autonomously. All information that are gathered during the autonomous driving are then further used to refine the path planning algorithm. The students decided to implement and test different SLAM algorithms. With regards to the available sensors, the students decided to use the LIDAR only initially. In order to improve the performance of the wall-following algorithm, the data from the inertial-measurement unit and the stereo camera were used. In parallel with the implementation of the SLAM algorithms, the time-optimal trajectory planning is worked on. All the above objectives are solved for static racing tracks and afterwards for dynamic racing tracks with obstacles. In addition, a simulation environment has been developed and set up such that algorithms could be easily tested and damages of the car could be prevented.

A. Project management

Since the objective was intentionally weakly defined, it was unclear what the exact requirements for the software are in the beginning. For that reason, the students opted for an agile development approach in order to adapt to changes in requirements more easily. Finally, the students decided for weekly group meetings to discuss issues that are relevant for the overall project's success and for multiple meetings in individual smaller groups for problem specific discussions.

Every month, the team decided about the milestones, discussed and created work-packages. Then, every assignment is specified and articulated in a ticket in the Kanban board that every developer in the team has access to. Since all students are fully responsible for the project, i.e., there is no immediate supervisor to coordinate and manage the milestones and no formal hierarchy, the students have to emphasize teamwork and consent. Hence, the students decided that all developers should be able to work and communicate remotely.

To serve for the above purpose, online tools have been chosen. Namely, Slack (https://slack.com/) for communication. The students created dedicated chat rooms for different engineering problems, and to comment on reports, vote and get an automated reminder for meetings. Furthermore, the students decided to use Github (https://github.com/) for their sourcecode management. All sourcecode is open source but can only be edited by authorized developers. Github offers a ticket system, that allows for the creation and organization of tickets in different development stages using a kanban board.

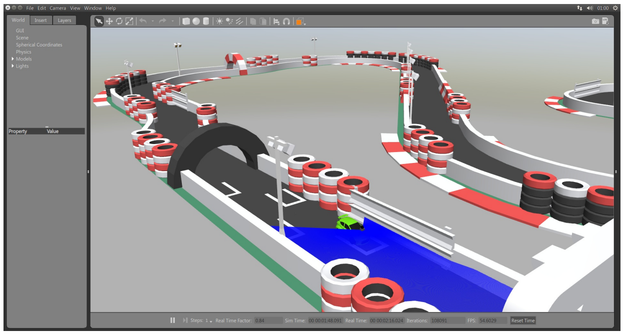

Fig. 2: Racing track model and the autonomous car in the Gazebo simulation environment.

The students have an overview of the pending workload and assign tickets to the developers directly. Whenever a student finished a feature, i.e., finished a ticket, the student can issue a pull request that is to be reviewed by the other students. Then, this feature is commented on and in consequence accepted or rejected for a pull. The students decided that this protocol increases the code quality, since the code is always inspected my multiple people. Another reason why the students chose Github is the easy possibility to create a wiki, in which the students decided to collect the documentation, general knowledge, helpful references and certain procedures. In addition, the students chose *Travis CI* for their continuous integration. Every time that the sourcecode in the master branch is changed on github, all created unit tests are run.

V. Evaluation and Results

In this section, we present the course evaluation provided by the *department of university development and organization* and review the results. Every semester at TU Dortmund University, the teaching quality is anonymously rated by the students and formally evaluated by the commission for Quality Assurance of Teaching (QSL). The questionnaire considers the rating of the *course content*, and the *course process* with a set of statements that have to be rated on a scale from 1 ($agree$) to 5 ($disagree$). Items concerning the *course content* of the project group (PG) are for example: *The course teaches formal, algorithmic, and mathematical competencies*; *The course teaches software analysis, design and project management skills*; *The course content is suitable to prepare for your professional future*; *The scope and expectations of the course are reasonable*. Questions concerning the *course process*, are for example *All required technical equipment is available*; *The supervision is frequent and sufficient enough*.

Twelve computer science master students in their final year of study participated in this questionnaire. All students strongly agreed that the PG teaches formal, algorithmic, and mathematical competencies. Furthermore, the students all agreed that analysis, design, realization, and project management skills for engineering projects are conveyed as well as the interplay of hardware and software, operating systems and intelligent systems. The students considered the contents PG suitable to prepare them for their future professional careers. It is to note that the students emphasized that they enjoyed the possibility to work fully independent with minimal supervision on this project. This overall positive reception by the students is shown in the award for the best PG offered by the faculty of computer science during winter semester 2018 and summer semester 2019.

Since October 2019, we have followed the same concepts in this paper to deploy a new project group with seven computer science master students, which will finish in September 2020. According to the intermediate feedback conducted in April 2020, they have enjoyed the lecture style, suggested guideline, and supplementary materials. We plan to file a report to analyze and compare the evaluations from these two project groups.

A. Summary of the Project Results

The students introduced a management framework and development processes in order to successfully complete the project. At the first step, the students created a simulation of the autonomous car and a racking track in Gazebo at a sufficient fidelity such that changes and algorithms could be tested in a virtual environment and thus avoid hardware damages. That is, the simulation does not only contain the mechanical models and dynamics of the car but also models for relevant internal components, i.e., the sensors or the VESC motor controller. After the simulation environment was set up, the students ensured that the manual control of the car is working at all times and is able to overwrite the autonomous control.

In order to implement the F1Tenth required halting ability, the students prepared a *deadman-switch*. A deadman-switch triggers a halting whenever the switch is not pressed. This version has the advantage that it will be triggered even when the connection of the controller to the car or server is broken. The team did test-drives with the SLAM algorithms, e.g., hector slam [5] and cartographer [2]. However, cartographer was the only SLAM algorithm that was able to generate a coherent map and to locate the car on that map by the students. Further test drives have shown that the speed of the car that could be achieved using SLAM algorithms was too slow to be further considered in a competitive racing environment. For that reason, SLAM based approaches have not been considered as an appropriate approach by the team anymore.

Machine learning approaches were also evaluated by the students. Namely *Q-Learning* [12] and a *Policy-Gradient* [10] variant as well as some evolutionary algorithms. Using *Q-Learning*, improvements of the autonomous driving capabilities could be achieved, but it was not possible for the students to create reliable learning agents. Furthermore, the speed control for different driving maneuvers could not be taught to the agents. In contrast, the *Policy-Gradient* variant performed better in controlling the speed of the car and thus lead to shorter achieved lap times. However, the variance in lap times was observed to be very high and collisions were very frequent. With the use of evolutionary algorithms, learning

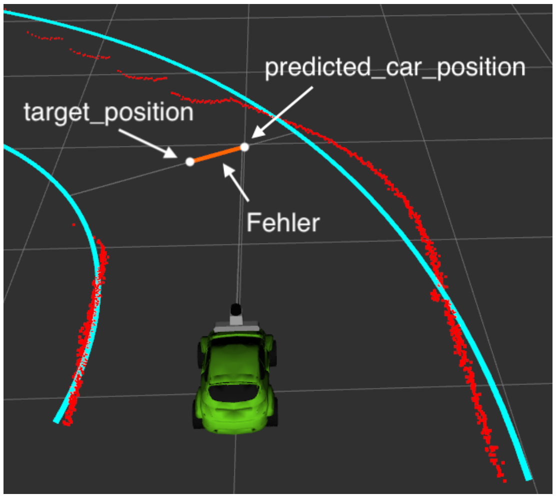

Fig. 3: Demonstration of the circle-fit wall-following algorithm within the Gazebo visualization environment.

agents could be generated in the simulations that lead to reasonable lap times with very little collisions. Nonetheless, it was not simple to transfer this approach to the real car. For future work, the simulation model must be further improved in order to allow for transferable results from evolutionary algorithms.

Finally, the students assessed that the improved circle-fit wall-following algorithm (see Figure 3) is the best for high-speeds, i.e., lead to the shortest lap times. The algorithm proved to work very well for a static racing tracks with no or few dynamic obstacles by just adjusting a few parameters for each given track. So far the students did not utilize the ZED camera, but propose to use it in the future since an improved sensing of the environment is beneficial for SLAM Algorithms. Especially for the detection of dynamic obstacles, the stereo camera is advantageous to the LIDAR.

VI. Conclusion and Outlook

In this work, we conducted a CPS education case study to educate graduate computer science and computer engineering students on cyber-physical systems. Our educational intention was to give the students the opportunity to learn how to solve complex engineering problems in a team and to give the students an opportunity to apply their acquired theoretical knowledge to an interesting and complex problem.

The studied project proved to be very successful both in the achieved results and the students' individual academic developments. The students have shown to organize themselves efficiently and to manage the project's milestones and progress well. Furthermore, the students have demonstrated to work in a very structured manner, i.e., they identified sub-problems and decomposed the overall engineering objective accordingly. The group also spent a lot of time on modeling the studied platform and on implementing these models in a simulation environment with sufficient model fidelity. The feedback from the students was that they very much liked to work and organize themselves independently without permanent supervision and virtual deadlines. However, we note that this amount of autonomy and style of working requires very capable, motivated and self-sufficient students in order for the project to succeed and for the students to develop academically.

Acknowledgements

We thank all PG 620 and PG 629 for their contributions to this paper.

References

[1] R. D. Chamberlain, R. K. Cytron, D. Shook, and B. Siever. Computers interacting with the physical world: A first-year course. In R. Chamberlain, W. Taha, and M. Törngren, editors, *Cyber Physical Systems. Model-Based Design*, pages 197–205, Cham, 2019. Springer International Publishing.

[2] W. Hess, D. Kohler, H. Rapp, and D. Andor. Real-time loop closure in 2d lidar slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1278, 2016.

[3] H. Isakovic, D. Ratasich, C. Hirsch, M. Platzer, B. Wally, T. Rausch, D. Nickovic, W. Krenn, G. Kappel, S. Dustdar, and R. Grosu. Cps/iot ecosystem: A platform for research and education. In R. Chamberlain, W. Taha, and M. Törngren, editors, *Cyber Physical Systems. Model-Based Design*, pages 206–213, Cham, 2019. Springer International Publishing.

[4] Y. Kim, A. Chen, S. Jadhav, C. S. Gloster, T. Le, and P. Hsu. Project Based Courses in Control Cyber Physical System Co-Design. In *Proceedings of the 2016 Workshop on Embedded and Cyber-Physical Systems Education*, WESE '16, New York, NY, USA, 2016. Association for Computing Machinery.

[5] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf. A Flexible and Scalable SLAM System with Full 3D Motion Estimation. In *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. IEEE, November 2011.

[6] A. Kumar, S. Fernando, and R. C. Panicker. Project-based learning in embedded systems education using an fpga platform. *IEEE Transactions on Education*, 56(4):407–415, 2013.

[7] E. A. Lee and S. A. Seshia. *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*. The MIT Press, 2nd edition, 2016.

[8] P. Marwedel and M. Engel. Flipped classroom teaching for a cyber-physical system course - an adequate presence-based learning approach in the internet age. In *10th European Workshop on Microelectronics Education (EWME), Tallinn, Estonia, May 14-16, 2014*, pages 11–15, 2014.

[9] J. A. Stankovic, J. W. Sturges, and J. Eisenberg. A 21st century cyber-physical systems education. *Computer*, 50(12):82–85, December 2017.

[10] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, page 1057–1063, Cambridge, MA, USA, 1999. MIT Press.

[11] Y. Ugurlu and T. Nagano. Project-based learning using labview and embedded hardware. In *2011 IEEE/SICE International Symposium on System Integration (SII)*, pages 561–566, 2011.

[12] C. J. C. H. Watkins and P. Dayan. Q-learning. In *Machine Learning*, pages 279–292, 1992.

Niklas Ueter received his master degree in computer science from TU Dortmund University, Germany, in 2018 and now is a PhD student at TU Dortmund University, supervised by Prof. Dr. Jian-Jia Chen. His research interests are in the area of embedded and real-time systems with a focus on real-time scheduling.

Kuan-Hsun Chen is a postdoc at Department of Informatics in TU Dortmund University, Germany. He received his Ph.D. in computer science from TU Dortmund University, Germany in 2019 and Best Student Paper Award at RTCSA 2018. His research interests include dependable computing, cyber-physical systems, and real-time operating systems.

Jian-Jia Chen is Professor at Department of Informatics in TU Dortmund University, Germany. He received his Ph.D. degree in from Department of Computer Science and Information Engineering, National Taiwan University, Taiwan in 2006. His research interests include real-time systems, embedded systems, power/energy-aware designs, and distributed computing. He received Best Paper Awards at CODES+ISSS 2014, RTCSA 2005 and 2013, and SAC 2009. He has involved in Technical Committees in many international conferences.