Volume 2, Issue 2, August 01, 2017

TC-CCPS Newsletter

Technical Articles

- Jakaria Abdullah, Gaoyang Dai, Nan Guan, Morteza Mohaqeqi and Wang Yi: *“TIMES-Pro: A Tool for Modeling, Analysis, Simulation and Implementation of Cyber-Physical Systems”*.
- Wuling Huang, Ziyi Yin, Shuqi Zhang, Long Chen: *“Autonomous Vehicles Testing in Connected Proving Ground”*.
- Qi Zhu, Tianshu Wei: *“Co-design and Co-scheduling for Energy-efficient Buildings”*.

Summary of Activities

Call for Contributions

TIMES-Pro: A Tool for Modeling, Analysis, Simulation and Implementation of Cyber-Physical Systems

Jakaria Abdullah[1], Gaoyang Dai[1], Nan Guan[2], Morteza Mohaqeqi[1] and Wang Yi[1,3]

[1]Uppsala University, Sweden
[2]The Hong Kong Polytechnic University, Hong Kong SAR, China
[3]Northeastern University, China

1 Introduction

Cyber-Physical Systems are systems that contain both discrete components such as digital controllers that generate and react to discrete events according to control laws and continuous components such as physical plants whose behaviors change continuously according to natural laws. Existing design tools for designing such hybrid systems such as Simulink [1] and Modelica [2] have inherent limitation due to the lack of expressiveness in their underlying modeling language and ability for analysis. In this paper, we present an integrated system design tool TIMES-Pro which adopts an expressive yet analytically tractable modeling language based on the Digraph Real-Time (DRT) task model [3, 4, 5] to model discrete components and conditional differential equations to model continuous physical components (differential equations with mode switches). For analysis, the continuous components of a system will be abstracted according to a set of predicates of interests, controlling the interaction with the discrete components of the system. DRT models will be used to approximate the continuous components for automated analysis. Our goal is to develop a toolbox supporting modeling and abstraction of both discrete and continuous components, timing analysis and code generation for real-time simulation as well as final deployment on a given execution platform for the discrete components.

2 Design Decision

In this section we summarize design decisions concerning mainly the design of the modeling language as well as the architecture and the features of TIMES-Pro.

2.1 Trade-off between expressiveness and analysis efficiency

Ideally, the modeling language of a tool should be as expressive as possible to enable faithful modeling of complex system behaviors such as dynamic branching and looping. As the expressiveness of models grows, so grows the complexity of their analysis. The DRT task model [3] is a rather expressive model allowing large flexibility to express release patterns accurately by representing each computation task as a directed graph. It generalizes most existing models in real-time scheduling theory [7]. It is shown that the feasibility problem of DRT can be solved in pseudo-polynomial time [3]. Efficient techniques of exact response-time analysis for DRT task models, for both static-priority and EDF scheduling have been developed using over-approximation of workload abstraction and refinement methods [5]. Finally, DRT model is extended to support rendezvous-style synchronizations with efficient analysis using over-approximation and under-approximation of workload abstractions [8]. Based on availability of these efficient analysis methods we choose DRT as the modeling language of our tool.

2.2 Separation of communication and computation concerns

The two major aspects of computer systems embedded in a CPS are computation and communication. Computational elements of a system should be independently designed without adherence to any specific communication mechanism. This allows not only, separation of concerns, but also efficient analysis. From our previous work on task automata and scheduling analysis [6, 9], it is known that many decision problems are computationally hard (even undecidable) for systems where feedback is allowed. Allowing communication to occur during the execution of a computation task may easily bring the feedback effects and change the workload of the system dynamically [9]. Making communication independent of computation may also allow modularity in system design, and flexible and portable design. We impose this principle by allowing communication to occur only on the release of computational jobs.

2.3 Functional correctness independent of non-functional behavior

The functional correctness of a system should be maintained during design-space exploration for satisfying the non-functional requirements. For example, changing the execution time of a task should not change its output or its logical correctness. This sounds a simple principle to implement if only the functional correctness of a task is concerned. On the system level, this is a challenging problem. For example, a functionality of a system is implemented by the execution of a number of tasks. The system designer should make sure that the execution order of tasks by the scheduler will not change the functionality. Technically, this requires that the scheduling policies adopted by the scheduler should ensure the functional correctness implied by system-level global invariants.

2.4 System development in a simulated environment

A popular engineering technique to validate or certify CPS is emulation. However, in many cases the actual plant is not available for emulation. Firstly, the plant may be a hardware which is developed at the same time. Secondly, actual plant may be too sensitive and cannot tolerate an error during simulation (such as human organs). Finally, construction of the actual plant may be too expensive for the test of a prototype of concept. To encounter these deficiencies, we decide that our system design tool should provide simulated environment using realistic but approximated model of the actual plant. Here the challenge lies in modeling the continuous semantics of a physical process using discrete software so that all important plant behavior necessary for the simulation can be generated. Current practice of using numerical solvers for evaluating continuous state is either too slow or too complex for any real-time simulation. To counter this, we tend to explore computationally efficient approximation techniques for solving differential equations which we can model using only software components.

3 Tool Overview

In this section, we present the main features of TIMES-Pro, the tool architecture and the main components in the implementation. Architecture of our tool is shown in Figure 1.

The tool has the following features:

- **Editor** (see Figure 2) to graphically model a system and its associated timing, execution resource and synchronization requirements. A system description consists of either a DRT or SDRT task set. The list of all tasks with their assigned priorities is shown in the left side of the main graphical editor. Timing properties of the jobs of a selected task is presented in a table below the task set. All the properties (including the names) of both the task set and jobs properties table are editable. In the main graphical editor, a task is described by its directed cyclic graph structure. User can define a job type by assigning its WCET, relative deadline and associated execution code segment. Different job types are connected by edges where the user can specify the minimum inter-release time between the two jobs. As an incoming edge denotes release constraints of the job, a synchronization action relevant to this job is also specified as the edge property. In the first option, the system

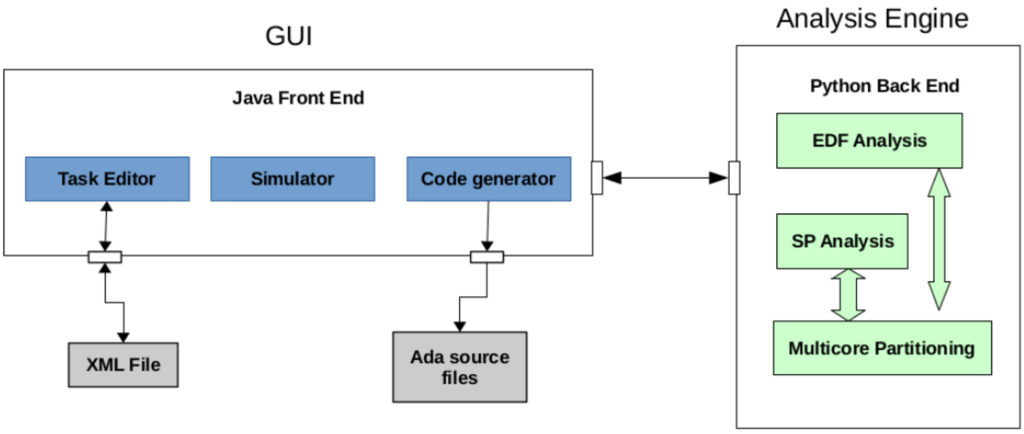
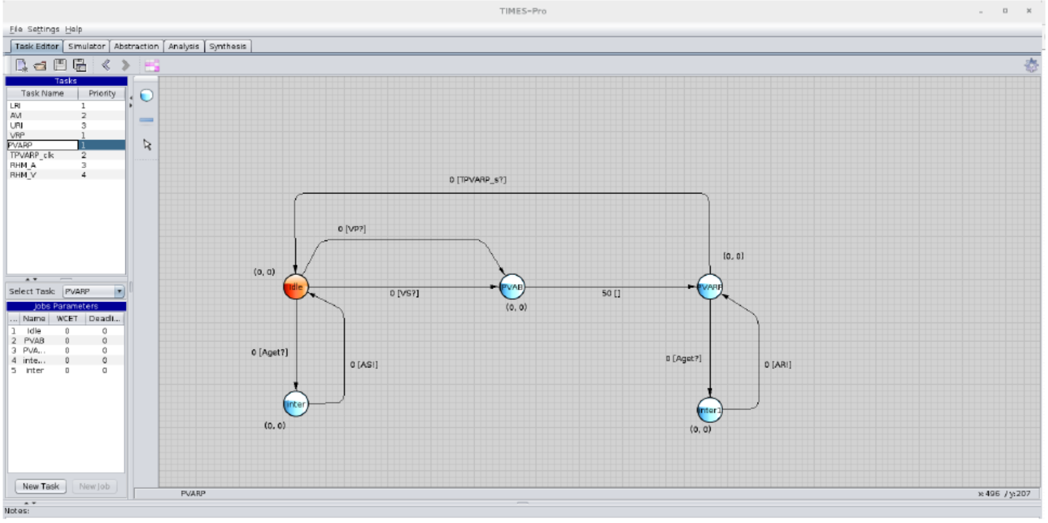


Figure 1: Tool architecture of TIMES-Pro.



Tasks

Task Name	Priority
LRI	1
AVI	2
URI	3
VRP	1
PVARP	1
TPVARP_clk	2
RHM_A	3
RHM_V	4

Select Task: PVARP

Jobs Parameters

...	Name	WCET	Deadli...
1	Idle	0	0
2	PVAB	0	0
3	PVA...	0	0
4	inte...	0	0
5	Inter	0	0

Figure 2: System modeling using SDRT tasks in TIMES-Pro editor.

designer explicitly states the branching condition variable together with the job code. Branching conditions are also allowed inside a job.

- **Simulator** (see Figure 3) to dynamically visualize the execution behavior and the resource utilization of a system model. The simulator generates possible execution traces with zero or random initial phase. This trace is displayed either stepwise or continuously up to the first deadline miss. It is possible to configure the speed of visual simulation within a scale of 1 to 10. System utilization is dynamically displayed below the main simulation. Currently the simulator supports fixed priority and EDF scheduling simulation on a uniprocessor.

- **Analyzer** to check that the tasks associated to a system model satisfy their timing requirements. The analysis suite includes schedulability analysis of tasks under Fixed Priority and EDF scheduling, computation of worst-case response times oand partitioning of workload on multiprocessors. To help testing the algorithms, analyzer has a configurable random task generator generating task sets of different size and utilization. Additionally, analyzer provides visualization data of different abstractions used for analysis like request functions (e.g., Figure 4).

- **Code Generator** to generate executable Ada code from task sets. The code generator realizes a subset of the behavior specified in the DRT/SDRT task model and assumes Ada runtime system will ensure proper execution of the generated code.

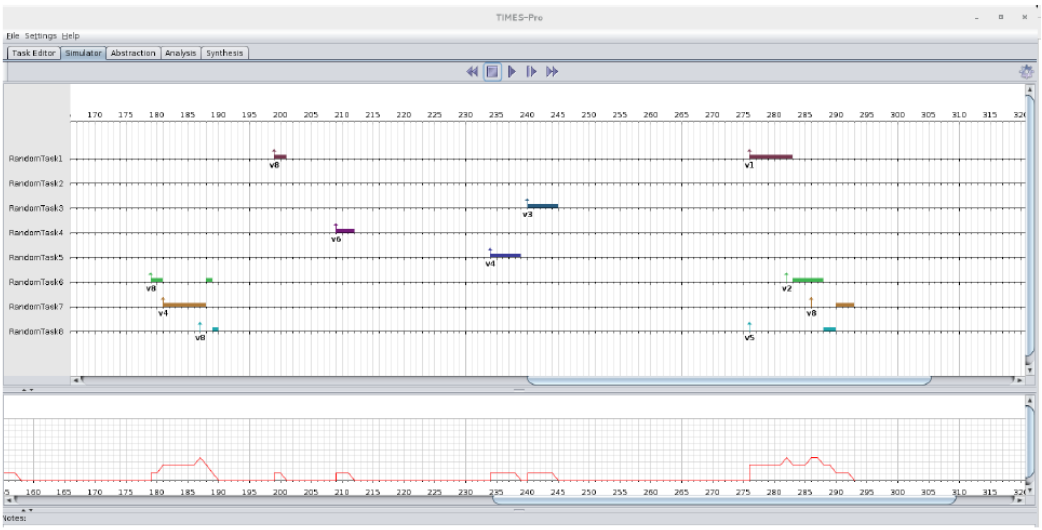
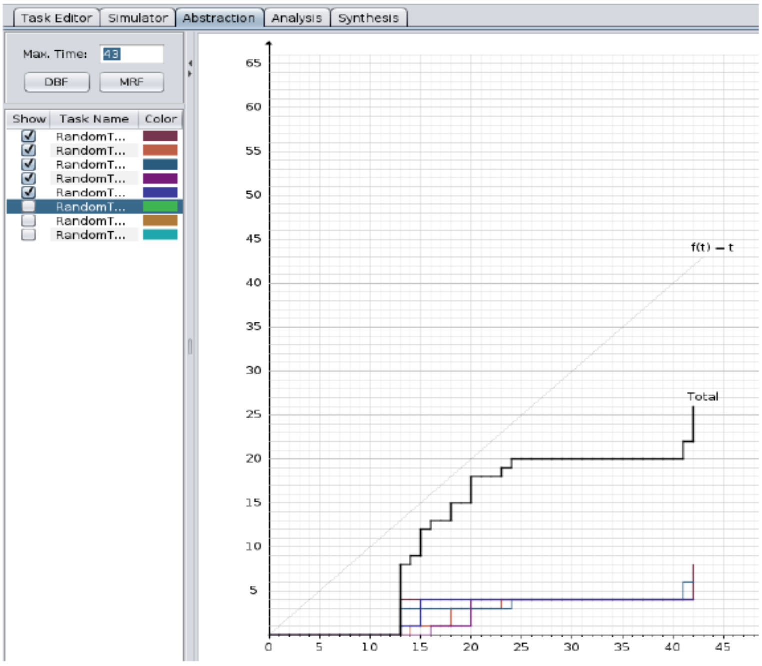


Figure 3: Visualization of job execution simulation in TIMES-Pro simulator.



Max. Time: 43

Show	Task Name	Color
☑	RandomT...	🟥
☑	RandomT...	🟧
☑	RandomT...	🟦
☑	RandomT...	🟪
☑	RandomT...	🟦
☐	RandomT...	🟩
☐	RandomT...	🟫
☐	RandomT...	🟦

Figure 4: Workload abstraction visualization in TIMES-Pro abstraction tab.

4 Conclusion and Future Work

This paper presents an integrated system design tool TIMES-Pro for the design and implementation of CPS. Different design decisions are explained and motivated; the tool architecture and the current state of implementation are presented. As future work, we will further develop the modeling language to support continuous components of CPS, and abstraction techniques for the analysis of combined behaviors by both types of components and simulate the behaviors in real-time.

References

[1] “Simulink”, `http://www.mathworks.com/products/simulink/`.

[2] “Modelica”, `http://modelica.org`.

[3] Stigge, M., Ekberg, P., Guan, N., Yi, W., “The Digraph Real-Time Task Model”. *Proc. of RTAS*, pp. 71–80, IEEE Press, New York (2011).

[4] Stigge, M., Yi., W., “Hardness Results for Static Priority Real-Time Scheduling”, *Proc. of ECRTS*, pp. 189–198 (2012).

[5] “Combinatorial Abstraction Refinement for Feasibility Analysis”, *Proc. Of RTSS*, pp. 340–349, IEEE Press, New York (2013).

[6] Amnell, T., Fersman, E., Mokrushin, L., Pettersson, P., Yi, W., “TIMES - A Tool for Modelling and Implementation of Embedded Systems”, *Proc. of TACAS*, pp. 460–464, Springer-Verlag (2002).

[7] Stigge, M., Yi.,W., “Models of Real-Time Workload: A Survey”, *Audsley, N., Baruah, S. (eds.) Real-Time Systems: the Past, the Present, and the Future*, pp. 133–160, (2013).

[8] Mohaqeqi, M., Abdullah, J., Guan, N., Yi, W., “Schedulability Analysis of Synchronous Digraph Real-Time Task”, *Proc. of ECRTS*, 2016, pp. 176–186.

[9] Fersman, E., Krcal, P., Pettersson, P., Yi, W., “Task automata: Schedulability, decidability and undecidability”, *Inf. Comput.*, vol. 205, no. 8, pp. 1149–1172, 2007.

Autonomous Vehicles Testing in Connected Proving Ground

Wuling Huang[1], Ziyi Yin[2], Shuqi Zhang[3], Long Chen[4]

[1]Institute of Automation, Chinese Academy of Sciences [2]Xi’an Jiaotong University
[3]Huazhong University of Science and Technology [4]Sun Yat-sen University

1 Introduction

Large-scale testing, including software simulation, Hardware in Loop (HIL) and real traffic field test, help to improve autonomous driving technologies [1, 2]. Based on these methods, the authors have proposed a parallel test method [7], which could accomplish most of the autonomous driving functions verification and evaluation through the simulation and field test parallel execution, which overcome the shortcoming of the common methods, could accomplish most of the autonomous driving functions testing [3, 4, 5]. As part of the parallel test, real traffic field test and interaction with real world infrastructure can improve the potential of the autonomous driving technologies with V2X support.

The open field testing need an environment with CPS based simulated traffic, to automatically carry out driving test. There are several autonomous vehicles (AVs) proving ground or test centers, such as M-City of MTC and iVPC [7]. These proving grounds facility consist of city, high-speed and rural roads, where scenario-based tests can be carried out in a repeatable and structured manner. This CPS based simulated traffic environment is the elemental support for parallel test, which includes V2X connection, connected testing robot and road side sensors etc. With the testing environment, tasks and methods, autonomous vehicles can be functional tested with detailed criteria obtained through V2X connection, such as vehicular and roadside V2X units, connected vision sensors, high-precision GPS positioning units, connected dummy cars, dummies and bikes etc.

This paper includes how to carry out the AVs parallel test in simulated traffic environment based on V2X connection, including the access of the vehicle and traffic information. The testing environment includes lots of CPS units with V2X connection and typical testing scenarios. With the testing methods and standardized testing procedures, with task evaluation criteria, we could compare the AVs and their algorithms and conduct automated testing. This work is a useful reference for AVs testing, and also for the next Intelligent Transportation System demonstration.

2 Autonomous Vehicles Testing Environment

2.1 Parallel Test of Autonomous Vehicles

The parallel test is an integration of simulation and real field test, including the driving simulation platform, with different sensors models, vehicle actuator and controller models, traffic scene definition, driving functions, which provides a novel method for autonomous driving perception, planning and control functional testing, as shown in Fig. 1. Besides the real field driving test, the simulation environment also provides a comprehensive testing solution, with vehicle sensors, communication system and function modules, tested with simulation, HIL, SIL and VeHIL methods.

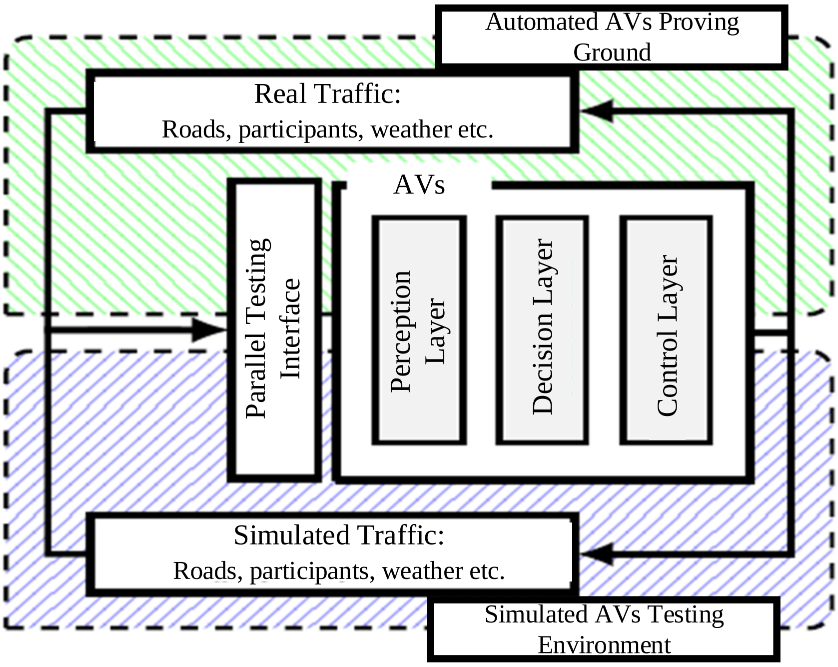


Figure 1: Simulation and Filed Testing Parallel Execution.

2.2 Autonomous Vehicles Testing Ground

Autonomous vehicles can be tested by driving a certain mileage in the typical real traffic environments with assessment of driving quality, as Fig. 2 shown. Autonomous driving test task Ti, along the test route with GPS road points, with different task complexity property and different environment complexities. Lists of simple function test cases are selected and assembled into different testing processes, which are further abstracted as driving tasks sets. By analyzing specific driving tasks sets, autonomous vehicle Perception Layer Functions, Decision Layer Functions, Navigation Layer Functions and Action Layer Functions can be full tested.

The driving tests can be deployed in a certain closed area with simulated highway, urban, rural roads autonomous driving, with V2X network, testing facilities, traffic scenes support. After the proving stage, the tests will be extended to a special testing area, with certain number of vehicles, and eventually to a large-scale urban demonstration.

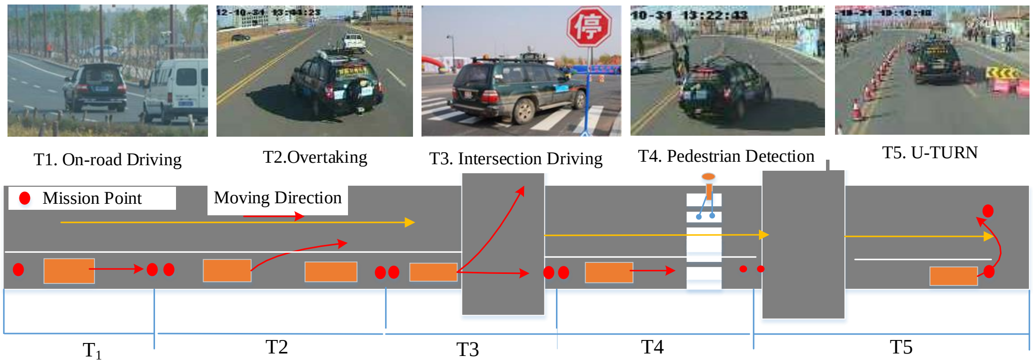


Figure 2: A Testing Example.

We did the first stage test in Intelligent Vehicle Testing Center of China (iVPC), located in Changshu, Jiangsu Province, including a simulated urban and suburban area traffic environment, with controlled traffic flow and testing facilities, shown in Fig. 3. The typical road section is with intersections, traffic signs and traffic lights, which is consistent with the common roads. The testing infrastructure and facilities includes GPS differential base station, LTE-V communication station and roadside units, DSRC on-board and roadside units, V2I supported traffic lights controllers, connected vision sensors, connected robotic pedestrians cones and dummy cars.

Figure 3: iVPC Testing Ground Overview.

V2X support the vehicle and road communication, makes the automated testing possible. As shown in Fig. 4, vehicular on-board units collect vehicle information through CAN bus and other sensors, such as vehicle attitude, high-precision position, speed and acceleration, throttle, braking and steering, perception information table, decision output, etc. then transmit these information to the roadside units and test data server application by V2X connection. The V2X network provides access to testing cloud with high precision digital map, wide scope traffic information, such as in the critical zones roadside facilities obtained the tested vehicles' location, speed and acceleration, relative position to the background vehicles etc.

3 AUTONOMOUS VEHICLE TESTING IN TYPICAL SCENARIOS

In AVs functional level, error and error rate of Perception Layer Functions, Decision Layer Functions, Navigation Layer Functions, Action Layer Functions are used for assessment criteria. In AVs performance level, the criteria

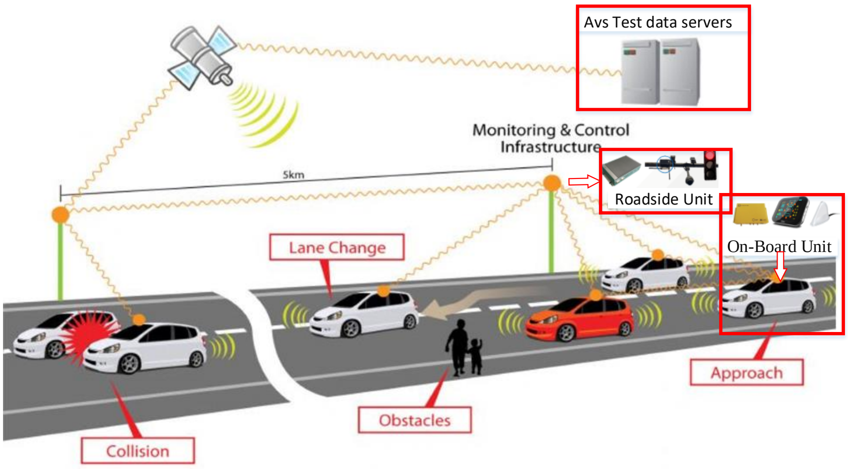


Figure 4: V2X support AVs Testing Scenarios.

used to measure vehicles' output, are classified into smoothness, safety, and smartness, which are synthesized from the AV functional assessment criteria, and used to determine how well the vehicle has successfully fulfilled each assigned task [6].

Fig. 5 shows AV A needs to make lane change with respect to Vehicle B. The safety score is characterized by the minimum distance between two vehicles during the lane change and the minimum distance to road boundary during the lane change. The closer is with the smaller safety score.

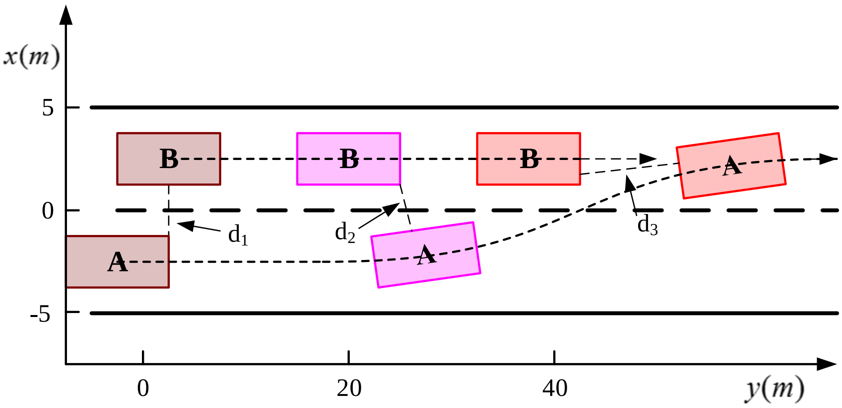


Figure 5: An illustration of how to evaluate the safety score in lane change scenario [6].

For a more complex traffic scenario, we need to track the relative distances between the testing vehicle and any another component (e.g. vehicle, bicycle, pedestrian, obstacle) in this traffic scenario and find the most dangerous situation to give a safety score. Fig. 6 are common testing scenarios with a single target or multiple targets, with the need to identify two or more vehicles in different lanes, the third car cut in or cut out in this scene.

The distance can be measured by the range detection Lidars or from the roadside vision sensors in typical scenarios. The other vehicle attitude, such as longitudinal ac/deceleration rates and the jerks can be obtained by the sensors mounted on the tested AVs. All this information can be sent over V2X networks, stored, processed and analyzed in the testing data servers. Based on the aforementioned standardized procedures and task evaluation criteria, we could compare the AVs and their algorithms proposed by different researchers and conduct automated testing.

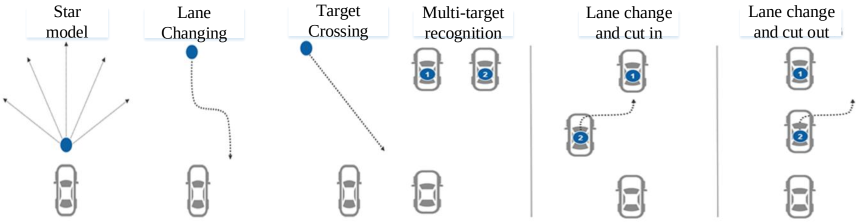


Figure 6: Single Target and Multiple Targets Testing Scenarios.

4 CONCLUSION

The CPS based simulated traffic environment is the elemental support for autonomous vehicles parallel test, which includes V2X connection and other connected testing facilities support. With the set-up of test environment, autonomous vehicles can be functional tested with detailed criteria values obtained by vehicular and roadside test units through V2X connection. The future work includes automated testing methods, automatic test scenario generation and configuration, automated tasks evaluation and full V2X support. This work is useful reference for autonomous vehicles automated testing, and also good reference for the next Intelligent Transportation System demonstration with autonomous vehicles.

References

[1] Fei-Yue Wang; Mirchandani, P.B.; Zhixue Wang, "The VISTA project and its applications," *Intelligent Systems, IEEE* , vol.17, no.6, pp.72–75, Nov/Dec 2002.

[2] Fei-Yue Wang; Xiaojing Wang; Li Li; Mirchandani, P.; Zhixue Wang; , "Digital and construction of a digital vehicle proving ground," *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE* , vol., no., pp. 533- 536, 9-11 June 2003

[3] W. Huang, D. Wen, J. Geng, N.-N. Zheng, "Task-Specific performance evaluation of UGVs: Case studies at the IVFC," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 1969-1979, 2014.

[4] Michael Montemerlo, Jan Becker, .., and Sebastian Thrun, "Junior: The Stanford Entry in the Urban Challenge," *J. Field Robot*. 25, 9 (September 2008), 569-597.

[5] Zhao D, et al., Accelerated Evaluation of Automated Vehicles Safety in Lane-Change Scenarios Based on Importance Sampling Techniques [J], *in IEEE Transactions on Intelligent Transportation Systems*, March 2017, vol. 18, no. 3, pp. 595-607.

[6] Li L, Huang W, Liu Y, Zheng N-N, Wang F, Intelligence testing for autonomous vehicles: A new approach [J], *IEEE Transactions on Intelligent Vehicles*, 2016, vol. 1, no. 2, pp. 158-166.

[7] Huang W, Wang K, Lv Y and Zhu F, Autonomous vehicles testing methods review [C], *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, 2016, pp. 163-168.

Co-design and Co-scheduling for Energy-efficient Buildings

Qi Zhu, Tianshu Wei, University of California, Riverside

Buildings account for nearly 40% of the U.S. primary energy consumption, 40% of the greenhouse gas emissions, and 70% of the electricity use [1]. Improving the energy efficiency of buildings therefore has critical impact on nation's economy, environment, and energy security. To this end, new energy-efficient technologies and techniques have been developed to address different aspects of buildings, e.g., from new insulation materials, advanced window controls, more efficient appliances and lighting systems, to integration of renewable energy sources, usage of battery storage, and more intelligent control of building HVAC (heating, ventilation and air conditioning) systems. In this article, we will discuss the approaches we developed in the past several years to address different aspects of buildings in integrated formulations and co-design/co-schedule them for improving energy efficiency and reducing energy cost.

Modern buildings are complex and classic cyber-physical systems. The cyber domain includes sensors collecting various environment and occupancy information, embedded controllers and central computers running control and diagnosis strategies, and wired or wireless networks connecting sensors, controllers and actuators. These cyber components closely interact with the physical domain that includes internal and external physical environment of the building, occupants activities, various appliances, energy storage and renewable sources, mechanical components of the HVAC system, etc. There are often strong inter-dependencies among these different cyber and physical components. To improve building energy efficiency, it is critical to *quantitatively* model, analyze and optimize the design and operation of these components in an *integrated* fashion.

Co-design: For building design, we observed that the accuracy of sensing data has a significant impact on the selection of optimal HVAC control strategy. In [2], we quantitatively analyzed such impact and formulated a framework for co-designing control algorithm and sensing platform (e.g., selection and placement of sensors). Intuitively, when a smaller number of low-cost sensors is selected due to budget constraint, the sensing data accuracy is lower. In this case, a more robust HVAC control algorithm is needed to maintain room temperature within the required range for occupancy comfort, which leads to higher energy consumption. When more and better sensors are selected, the sensing data accuracy is higher. In this case, more straightforward control algorithm is sufficient to satisfy constraints on room temperature, which consumes less energy. Such co-design framework should help building designers to trade off among sensor installation cost and energy consumption cost, and decide the best strategy for selecting sensors and HVAC control algorithm.

Co-scheduling: During building operation, a key aspect for energy efficiency is to leverage the scheduling flexibility provided by various energy demand loads in buildings, including HVAC, plug loads, and emerging loads such as EV (electric vehicle) charging, etc. Among these loads, HVAC consumes around 50% of the total building energy consumption. Its energy demand changes based on the dynamic physical environment (e.g., outside air temperature and sun radiation) and building occupancy activities, and needs to be carefully managed to satisfy the building temperature and air flow requirements. The building thermal flywheel effect allows temporarily unloading the HVAC systems without immediate impact on building occupants, and therefore provides significant flexibility in managing the demand. In addition, EV charging has become a significant energy load in many commercial and residential buildings, and will likely continue growing rapidly. In residential buildings, EV charging often occurs at night; while in commercial buildings with installed charging stations for tenants, EV charging demand concentrates at daytime and may coincide with the HVAC demand to cause spikes during peak hours. The peak demand and total energy consumption of the buildings could be reduced through intelligent scheduling of HVAC control (by turning on and off air conditioning and changing air flow volume at different times), EV charging (by charging EVs at different times and with different power levels), and other flexible demands.

Furthermore, on the energy supply side, utilizing multiple energy sources such as grid electricity, battery storage, and renewable sources provides more opportunities for reducing the peak demand and total energy cost. Note that battery storage is viewed as an energy source during discharging and an energy demand during charging. It has been increasingly used at building level to store energy during off-peak hours (or from renewable energy sources) and release energy at peak hours. This provides additional flexibility for building energy scheduling.

The demand side control (e.g., HVAC control and EV charging scheduling) depends on the availability of various energy sources, and the supply side energy sources scheduling (i.e., deciding which source to use and for how much at different times) requires the knowledge of energy demand. We believe that to maximize building energy efficiency, it is not only important to model and co-schedule different energy demands in an integrated formulation, it is also essential to *co-schedule energy demands with various supply sources*. In our early work [4], we propose an algorithm for co-scheduling HVAC control and battery usage, and demonstrate its effectiveness in reducing energy cost. In [5], we develop a model predictive control (MPC) based algorithm to *co-schedule HVAC control, EV charging and battery usage* for reducing the total building energy cost, including both the electricity consumption charge and the peak demand charge. More specifically, the demand side scheduling of HVAC control and EV charging depends on the availability of battery storage and the pricing of grid electricity, and the supply side scheduling of battery storage is based on the real-time knowledge of HVAC and EV charging demand. We address all these factors in an integrated formulation for MPC control, with constraints on room temperature and EV charging deadlines. In experiments, we demonstrate that such co-scheduling strategy can reduce building energy cost by 4% and peak demand by 15%.

In [3], we consider the energy efficiency issues in mixed-use datacenter buildings, where datacenters share common infrastructures and energy supplies with other operations such as non-IT offices and labs (the majority of datacenters are located in such mixed-use buildings). We develop an MPC formulation to co-schedule datacenter and HVAC loads, with consideration of solar energy and battery storage. The MPC formulation minimizes building energy cost while satisfying various requirements on room temperature, ventilation, and datacenter workload deadlines. Compared with separate scheduling strategy, our co-scheduling approach can reduce the building energy cost by 3% to 17%, and reduce the carbon footprint by 3%.

In [6], we leverage the co-scheduling strategies from [4, 5] and integrate such intelligent building energy management with grid optimization through a proactive demand response framework. We demonstrate that the building operation cost can be reduced by up to 20%, and power system generation cost is reduced by up to 10%. In [7], we address several security issues in such proactive demand response framework, and found the framework to be more resistant to guideline price manipulation attacks compared with traditional passive demand response strategy.

To conclude, we have developed several co-design and co-scheduling approaches to holistically address several major components in buildings, and seen promising results. Modern intelligent buildings have become complex cyber-physical-human systems. It is critical to develop such holistic formulations for maximizing energy efficiency while providing comfortable, convenient and secure services to occupants.

References

[1] Building Energy Data Book of DOE, available: https://catalog.data.gov/dataset/buildings-energy-data-book.

[2] M. Maasoumy, Q. Zhu, C. Li, F. Meggers, and A. Vincentelli. Co-Design of Control Algorithm and Embedded Platform for Building HVAC Systems. In *2013 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, April, 2013.

[3] T. Wei, M. A. Islam, Shaolei Ren, and Qi Zhu. Co-scheduling of Datacenter and HVAC Loads in Mixed-Use Buildings. In *2016 Seventh International Green and Sustainable Computing Conference (IGSC)*, November, 2016.

[4] T. Wei, T. Kim, S. Park, Q. Zhu, S.X.-D. Tan, N. Chang, S. Ula, and M. Maasoumy. Battery Management and Application for Energy-Efficient Buildings. In *2014 ACM/IEEE Design Automation Conference (DAC)*, June, 2014.

[5] T. Wei, Q. Zhu, and M. Maasoumy. Co-Scheduling of HVAC Control, EV Charging and Battery Usage for Building Energy Efficiency. In *2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, November, 2014.

[6] T. Wei, Q. Zhu, and N. Yu. Proactive Demand Participation of Smart Buildings in Smart Grid. *IEEE Transactions on Computers*, 65(5):1392–1406, May, 2016.

[7] Tianshu Wei, Bowen Zheng, Qi Zhu, and Shiyan Hu. Security Analysis of Proactive Participation of Smart Buildings in Smart Grid. In *2015 ACM/IEEE International Conference on Computer-Aided Design (ICCAD)*, November, 2015.

Technical Activities

1 Workshops

- The 2nd IEEE International Workshop on Design Automation for Cyber-Physical Systems (DACPS 2017)

2 Special Issues in Academic Journals

- Proceedings of the IEEE Special Issue on Design Automation for Cyber-Physical Systems
- IET Cyber-Physical Systems: Theory & Applications (IET-CPS) special issue on Cyber-Physical Systems for Medical and Life Sciences Applications
- IET Cyber-Physical Systems: Theory & Applications (IET-CPS) special issue on Cyber-Physical Systems in Smart Grids: Security and Operation
- IET Cyber-Physical Systems: Theory & Applications (IET-CPS) special issue on Cyber-Physical Aspects of EVs and HEVs
- IET Cyber-Physical Systems: Theory & Applications (IET-CPS) special issue on Recent Advances in Big Data Analytics and Cyber-physical Systems Security
- IET Cyber-Physical Systems: Theory & Applications (IET-CPS) special issue on Safety-Critical Cyber Physical Systems
- IEEE Transactions on Sustainable Computing (TSUSC) Special Issue on Sustainable Cyber-Physical Systems
- IEEE Transactions on Big Data Special Issue on Big Data for Cyber-Physical Systems

3 Book Publications

- Springer Book "Leveraging Big Data Techniques for Cyber-Physical Systems"

Call for Contributions

Newsletter of Technical Committee on Cyber-Physical Systems (IEEE Systems Council)

The newsletter of Technical Committee on Cyber-Physical Systems (TC-CPS) aims to provide timely updates on technologies, educations and opportunities in the field of cyber-physical systems (CPS). The letter will be published twice a year: one issue in February and the other issue in October. We are soliciting contributions to the newsletter. Topics of interest include (but are not limited to):

- Embedded system design for CPS
- Real-time system design and scheduling for CPS
- Distributed computing and control for CPS
- Resilient and robust system design for CPS
- Security issues for CPS
- Formal methods for modeling and verification of CPS
- Emerging applications such as automotive system, smart energy system, internet of things, biomedical device, etc.

Please directly contact the editors and/or associate editors by email to submit your contributions.

Submission Deadline:

All contributions must be submitted by **Jan. 1st, 2018** in order to be included in the February issue of the newsletter.

Editors:

- Helen Li, Duke University, USA, hai.li@duke.edu

Associate Editors:

- Long Chen, Sun Yat-Sen University, China, chenl46@mail.sysu.edu.cn
- Wuling Huang, Chinese Academy of Science, wuling.huang@ia.ac.cn
- Yier Jin, University of Florida, USA, yier.jin@ece.ufl.edu
- Rajiv Ranjan, Newcastle University, United Kingdom, raj.ranjan@ncl.ac.uk
- Yiyu Shi, University of Notre Dame, USA, yshi4@nd.edu
- Bei Yu, Chinese University of Hong Kong, Hong Kong, byu@cse.cuhk.edu.hk
- Qi Zhu, University of California at Riverside, USA, qzhu@ece.ucr.edu