Model-based System-Level Testing for Distributed Transportation CPS

Ratnesh Kumar, Prof. ECE, ISU, Fellow IEEE

In a distributed Cyber-Physical System (CPS), e.g., Figure 1, control units are distributed, collecting sensor measurements driven by the underlying physical dynamics, commanding target actuators, while interacting/communicating through an embedded bus/network. To formally specify the entire system architecture, AADL (standardized by SAE), can capture the architecture of software, computing/communication hardware/medium, and physical components, together with their behavior models, and other constraints.

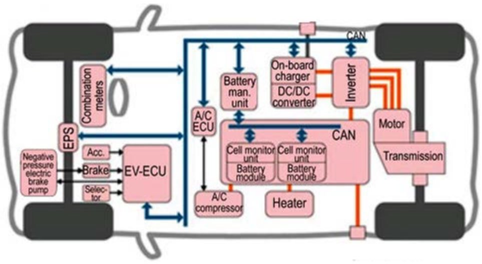

Figure 1. A Distributed CPS

To support model-based testing of *only* the software components, we have developed tools that automatically translate the control components in Simulink/Stateflow to Extended Automata, perform automatic test generation offering desired code/requirements coverage, execute tests and localize errors. Going forward, some of the research challenges of system-level testing are:

- **System component models:** For a model-based testing approach for the entire system, we need their formal *common* models, e.g., Extended/Timed Automata for discrete real-time behaviors in the software/computation/communication components, and Stochastic Automata for the physical components (those are subject to noise).
- **Interaction modeling:** This must capture the input/output connectivity of components and any interaction through shared variables for capturing *concurrency*, *interaction-latency/-errors*.
- **Worst-case computation/communication time analysis**: The models will need to be extended to incorporate worst-case *computation/communication delays* (obtained for example using MILP models of computation/communication components, e.g., we have developed such models for FlexRay/TTE). Models would also need to be augmented to capture *clock-drifts*, and light-weight synchronization mechanisms would be required to manage the drifts.
- **Robustness wrt computation imperfections**: Computing platforms are limited in *computational precision*, causing input-output signal perturbations, leading to the implemented computations being different from that of software. Model-based approach should capture such platform-level inaccuracies to ensure the control and data flows are preserved. Software models would need to be extended to capture computational inaccuracies.
- **Probabilistic issues**: Physical components are subject to noise, and so their models as well as requirements would be probabilistic. Testing approach would need to address the corresponding implications, e.g., there will be associated notions of *false negatives/positives*.
- **Verification/testing of large and complex distributed systems**: Tests achieving model coverage and/or requirements coverage need to be generated and validated at the system-level to ensure the correctness of the CPS design as well as implementation. A *compositional* approach combining component level testing, together with their interactions, inaccuracies, timing constraints, stochasticity, etc. would need to be developed. Goal would to be navigate from lower component levels to higher ones by having the components rely only on the *interface-behaviors* of the neighboring components (instead of their explicit full models).
- **Applications in Transportation CPS testbeds**: Transportation systems are safety-critical distributed CPS, e.g., cars at an intersection, planes at an airport, trains of a metro network, etc. A "system-of-systems" approach, addressing the above challenges, would be required.