Demo Abstract: A Co-Simulation Platform for Actuator Networks

Ahmad Al-Hammouri, Vincenzo Liberatore,
Huthaifa Al-Omari, Zakaria Al-Qudah,
Michael S. Branicky
Electrical Engineering and Computer Science
Case Western Reserve University
{ata5,vl,haa6,zma,mb}@case.edu

Deepak Agrawal
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Deepak.Agrawal@iitkgp.ac.in

Categories and Subject Descriptors

C.2 [**Computer System Organization**]: Computer-Communication Networks; I.6 [**Computing Methodologies**]: Simulation and Modeling; J.2 [**Computer Applications**]: Physical Sciences and Engineering

General Terms

Experimentation, Design, Performance

Keywords

Actuator networks, cyber-physical systems, simulation, modelica, ns-2, power grid

1 Introduction

Actuator networks will enable an unprecedented degree of distributed control of physical environments, and further progress will critically depend on the availability of a simulation platform that can capture both the physical and the communication dynamics.

An actuator network consists of interconnected sensors, controllers, and actuators (Fig. 1). Actuator networks are pervasive in a staggering variety of applications [6]. For example, an important future direction will involve their usage to monitor and control the power grid [7]. Actuator networks often differ from wireless sensor network because actuator networks are dominated by the energy and computational requirements of actuation. As a result, actuator networks often operate in environments where energy is readily available, communication can use wired media, extensive computation is feasible, but the physical control dynamics impose pressing real-time constraints [2, 5]. In general, actuator network performance critically affects the behavior, stability, and safety of the controlled physics, and conversely, the demands of the physics impact the design of an actuator network [3]. Therefore, future progress critically depends on tools for the combined simulation, emulation, and validation of actuator networks together with their surrounding physics.

We will demonstrate a tool that combines the ns-2 network simulator [4] with the *Modelica* framework. Modelica is a modeling language for large-scale physical systems and has been successfully applied to domains as varied as power grids, flight simulators, and racing cars [8]. Its integration with ns-2 enables us to simulate the physics jointly with a

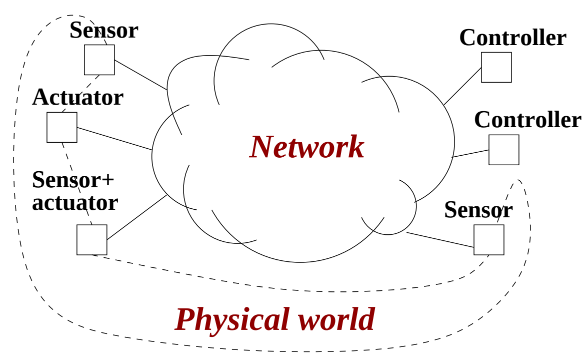

Figure 1. A conceptual representation of an actuator network consisting of sensors, actuators, and controllers.

controlling actuator network. In particular, we will demonstrate a power transmission application in which voltages depend on the levels of real-time service in the underlying actuator network.

A co-simulation platform must support the two-way communication between the simulators. For example, ns-2 would model the communication of sensor data obtained from the Modelica's simulation and, conversely, Modelica's actuators would use control signals that are being exchanged over the ns-2 network. The main technical difficulty consists in the two-way synchronization of the two simulators. On the contrary, it is easy to combine the simulators for a pure sensing application, in which sensor data flows solely from Modelica to ns-2, but no information flows in the reverse direction. We will demonstrate two-way communication of sensor and control information.

In the remainder of this abstract, we will provide a quick sketch of the underlying technical approach, and describe a demonstration in the context of power transmission. A preliminary version of the demonstration is available on-line (http://vorlon.case.edu/∼vxl11/NetBots/#modelicans2), and all of the software will be released shortly.

2 Ns-2-Modelica Integration

The integration is accomplished by creating communication elements in the ns-2 model and in Modelica. We have created a new ns-2 application agent that is responsible for communicating to a corresponding Modelica model. For example, the ns-2 agent can receive packets in the ns-2 simulation and use their payload to set a control signal in a corre-

SenSys'07, November 6–9, 2007, Sydney, Australia.
ACM 1-59593-763-6/07/0011

sponding Modelica actuator. On the Modelica side, we have written inter-simulator communication routines that we can link to generic sensor and actuator models. As a result, simulator communication is achieved by pairs of corresponding modules, one in each of the two simulators. The communication mechanics relies on Unix named pipes.

The major technical hurdle is the synchronization of the simulators' timing. Each simulation runs against simulated time, which is obviously different from a wall clock, and each simulator progresses at different speed. However, the two simulations must be synchronized so that events will propagate from one simulation to the other one and take effect at the appropriate time. Synchronization is achieved by enslaving each Modelica's element to its corresponding ns-2 agent. When the ns-2 agent is scheduled, say at time t, it instructs Modelica to run until time t, and to exchange data at that point. For example, if the ns-2 agent should deliver a control signal to an actuator at time t_1, it would instruct Modelica to continue the simulation until time t_1 and to read the new value of the control signal. Conversely, an ns-2 agent may sample a sensor at time t_2 so as to use its value as the payload of an ns-2 packet. Modelica would be instructed to progress until time t_2, at which point the sensor data is captured from Modelica and injected in the simulated network. In general, synchronization is achieved by suspending and restarting Modelica in such a way that the Modelica's simulation time never passes in front of that of ns-2. Furthermore, Modelica's simulation time catches up on demand with that of ns-2.

The integration would be simpler in a pure sensing simulation where information flows only from Modelica to ns-2. In pure sensing, Modelica can be executed first and generated sensor data collected or piped into ns-2. In particular, there would be no synchronization requirement for one simulation time to catch up with the other one

3 What Are You Going to See?

We will demonstrate the co-simulation of an actuator network that controls the voltage of a power transmission system. Actuator networks will enhance the power grid with better situational awareness and control, fault-tolerance, fault recovery, collaborative operations, and the incorporation of market dynamics to reduce peak prices and stabilize costs [1]. The demonstration will both show the capabilities of the co-simulation platform, as well as achieving a first step toward the co-simulation of an intelligent power grid.

The simulated scenario (Fig. 2) involves a generator (denoted as PMgen in the figure) that supplies electrical power to a remote load (zLoadDC) that is time-varying and subject to disturbances (transSig). To dynamically match the load with the energy supply, it is enough to keep the voltage to a constant value. The voltage is measured by a sensor (DCvoltage), its value sent through the network interface Y and received by a network interface U. The communication from Y to U is modeled via ns-2, and omitted from the figure. The netowrk consists of a simple dumbbell topology traversed by the sensor flow and by cross-traffic. The sensor data are then supplied to a PI controller (PI_vDC, gain) to dynamically change the armature voltage of the PMgen so that its

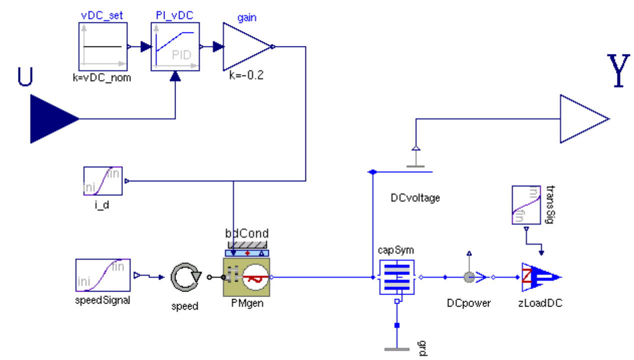

Figure 2. A model for a simple power system with controllers.

output matches the load. The main quantity of interest is the value of the voltage, which ideally should be a constant regardless of the variability in load. However, sensor data may be subject to delays and losses in the end-to-end path from Y to U, thereby jeopardizing the system ability to maintain a constant voltage value.

Acknowledgments

Research supported in part by NSF CCR-0329910, Department of Commerce TOP 39-60-04003, and Department of Energy DE-FC26-06NT42853. We thank Peter S. Trogos of MathPros, Inc. and Dr. Joseph Gauthier of Gauthier Simulation, Inc. for their invaluable help with Dymola, and Eugenio Villaseca for help with power systems.

4 References

[1] Final report on the August 14, 2003 blackout in the United States and Canada: Causes and recommendations. Technical report, North American Electric Reliability Corporation, Apr. 2004.

[2] A. Al-Hammouri, M. S. Branicky, V. Liberatore, and S. M. Phillips. Decentralized and dynamic bandwidth allocation in networked control systems. In *WPDRTS*, 2006.

[3] M. S. Branicky, V. Liberatore, and S. Phillips. Co-simulation for co-design of networked control systems. In *American Control Conference*, 2003.

[4] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu. Advances in network simulation. *Computer*, 33(5):59–68, May 2000.

[5] V. Liberatore. A play-back algorithm for networked control. In *INFOCOM*, 2006.

[6] V. Liberatore et al. Networked Control Systems Repository. `http://home.cwru.edu/ncs/`.

[7] J. Nutaro, P. T. Kuruganti, M. Shankar, L. Miller, and S. Mullen. Integrated modeling of electirc grid, communications, and control. In *ORMMES*, 2006.

[8] M. M. Tiller. *Introduction to Physical Modeling with Modelica*. Springer, 2001.