

Portable Real-Time Code from PTIDES Models

Patricia Derler, John Eidson, *Edward A. Lee*,
Slobodan Matic, Christos Stergiou,
Michael Zimmer
UC Berkeley

Invited Talk
*Workshop on Time Analysis and Model-Based Design,
from Functional Models to Distributed Deployments
(TiMoBD)*
ESWeek
Taipei, Taiwan, Oct. 9, 2011

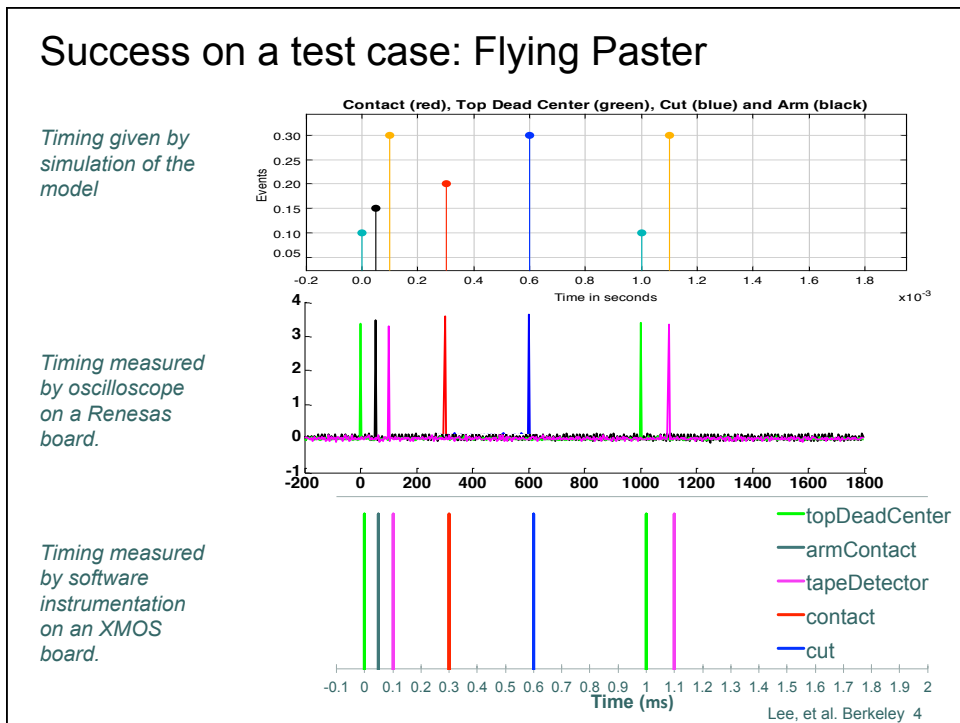
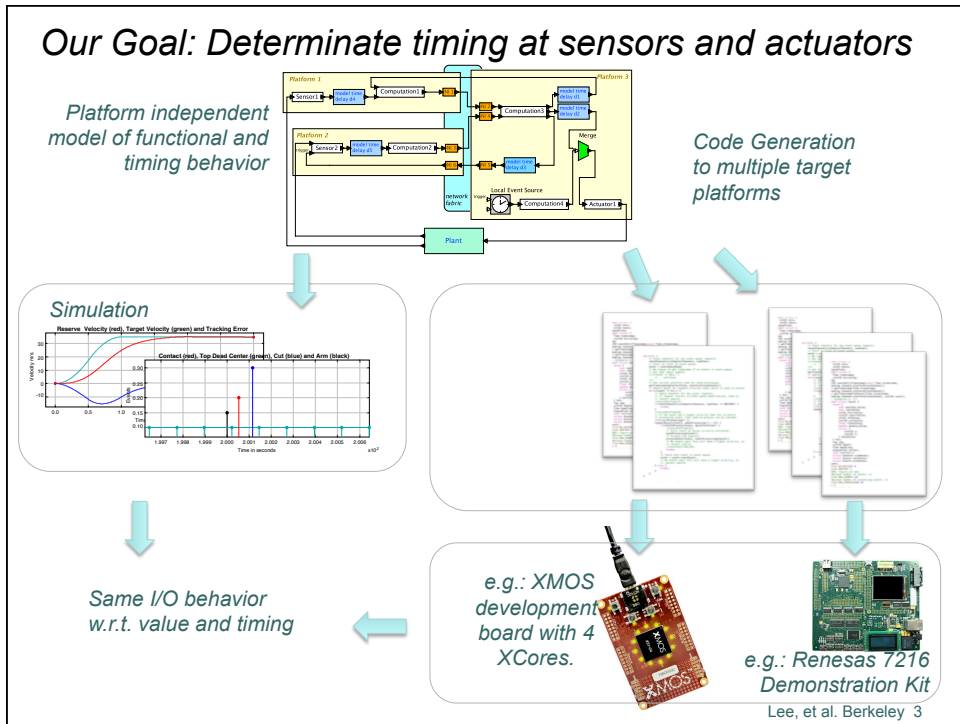
The Problem we are addressing

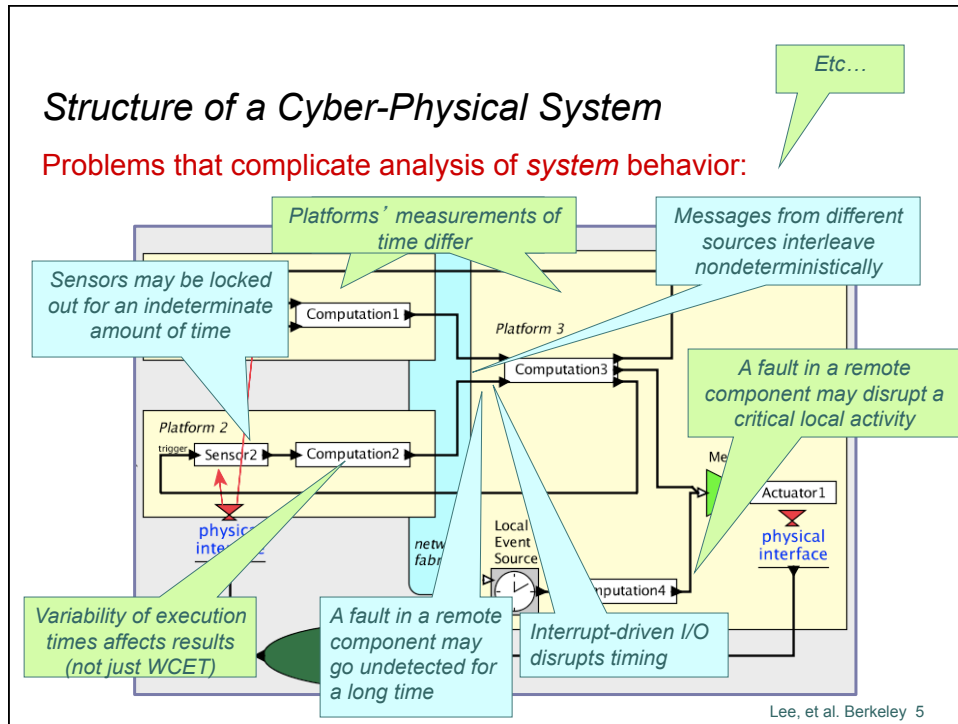
For cyber-physical systems, *programs do not adequately specify behavior.*

When interactions with physical processes are central, the temporal dynamics of software is a critical part of its behavior.

Today, temporal dynamics emerges from an implementation of the system, rather than being part of the model and/or program(s) specifying the system.

As a consequence, systems are *brittle and non-portable.*





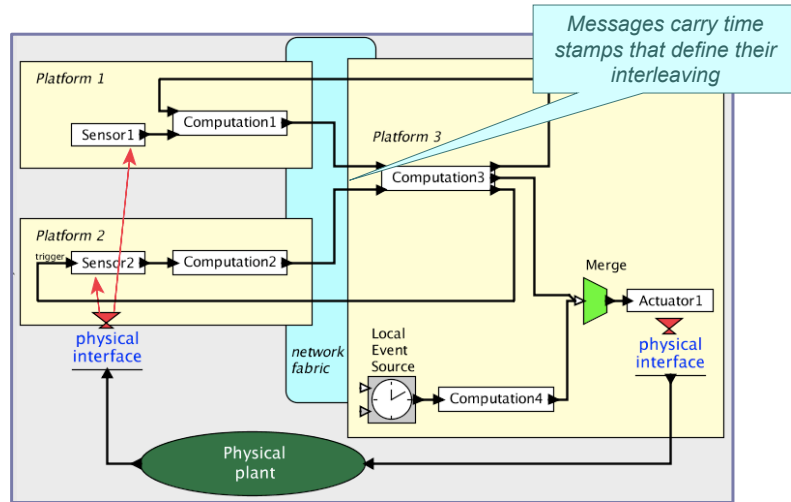
Our Programming Model

PTIDES:

Programming Temporally Integrated Distributed Embedded Systems

Based on a determinate discrete-event (DE) model of computation (MoC), originally developed for simulation.

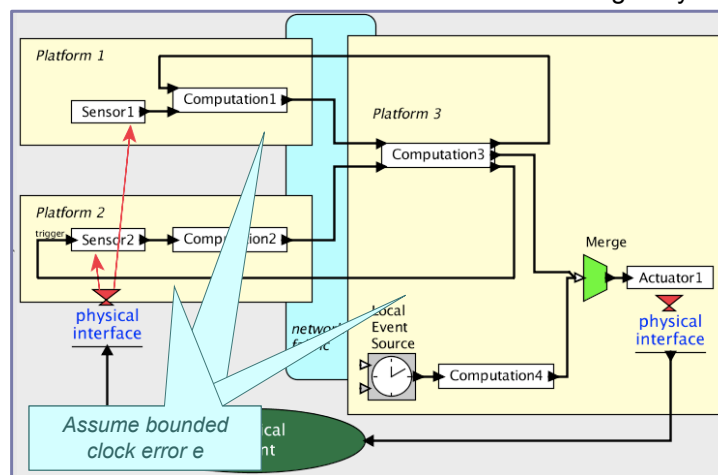
Ptides: First step:
Time-stamped messages.



Lee, et al. Berkeley 7

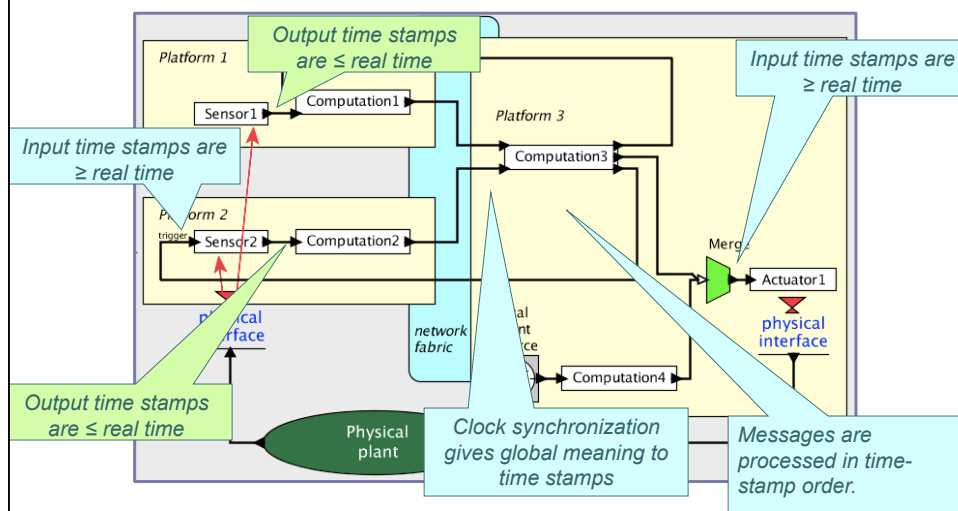
Ptides: Second step:
Network time synchronization

GPS, NTP, IEEE 1588, time-triggered busses, etc., all provide some form of common time base. These are becoming fairly common.



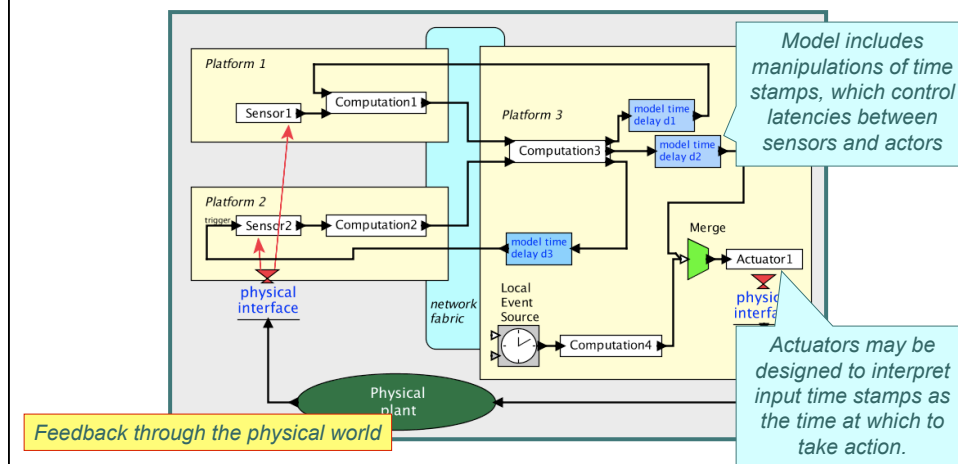
Lee, et al. Berkeley 8

Ptides: Third step:
Bind time stamps to real time at sensors and actuators



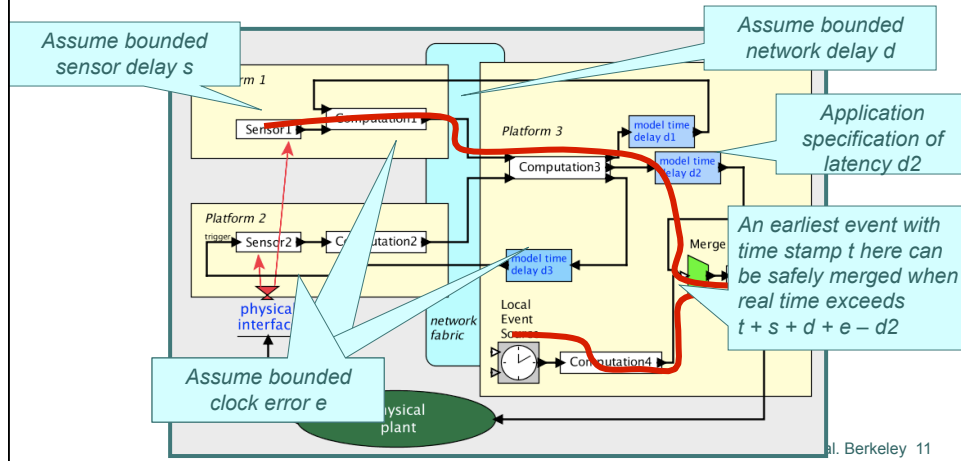
Ptides: Fourth step:
Specify latencies in the model

Global latencies between sensors and actuators become controllable, which enables analysis of system dynamics.



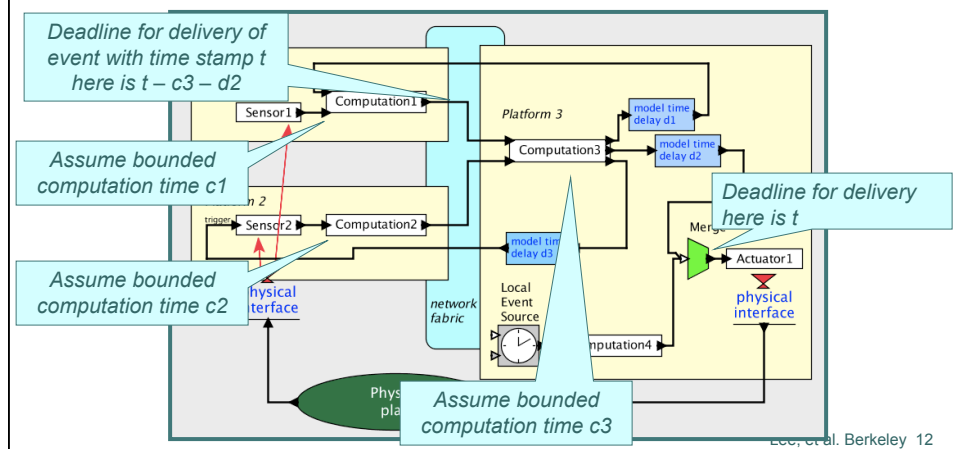
Ptides: Fifth step Safe-to-process analysis (ensures determinacy)

Safe-to-process analysis guarantees that the generated code obeys time-stamp semantics (events are processed in time-stamp order), given some assumptions.



Ptides Schedulability Analysis Determine whether *deadlines* can be met

Schedulability analysis incorporates computation times to determine whether we can guarantee that deadlines are met.



PtidyOS: A lightweight microkernel supporting Ptidies semantics

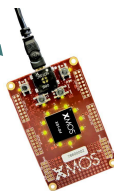
*An interesting property of PtidyOS is that despite being highly concurrent, preemptive, and EDF-based, it does not require threads.
A single stack is sufficient!*

PtidyOS runs on


- Arm (Luminary Micro)
- Renesas
- XMOS

Occupies about 16 kbytes of memory.

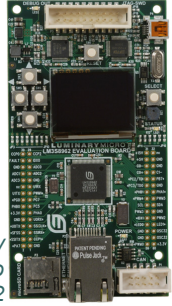
XMOS development board with 4 XCores.



Renesas 7216 Demonstration Kit

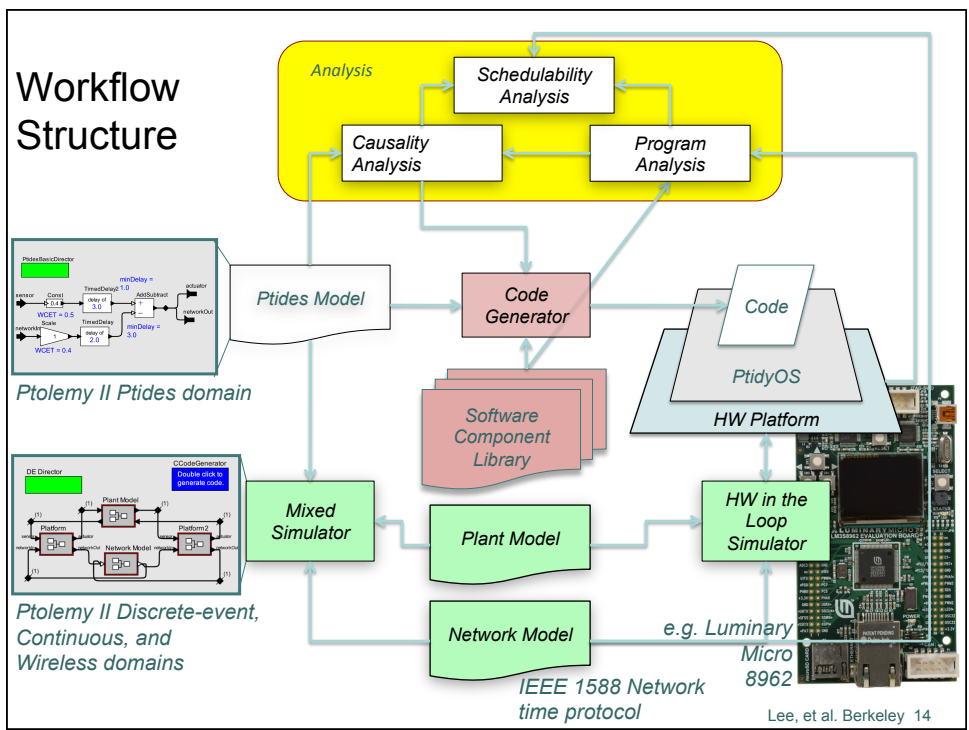


Luminary Micro 8962




The name "PtidyOS" is a bow to TinyOS, which is a similar style of runtime kernel.

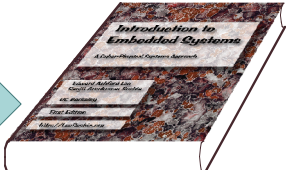
Lee, et al. Berkeley 13



A Typical Cyber-Physical System Printing Press



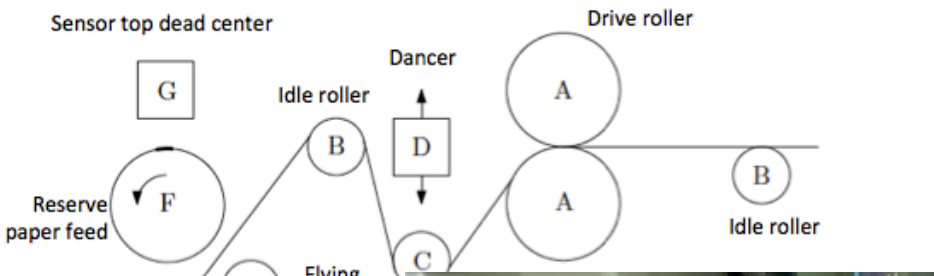
Bosch-Rexroth




- Application aspects
 - local (control)
 - distributed (coordination)
 - global (modes)
- Ethernet network
 - Synchronous, Time-Triggered
 - IEEE 1588 time-sync protocol
- High-speed, high precision
 - Speed: 1 inch/ms (~100km/hr)
 - Precision: 0.01 inch
 - > Time accuracy: 10us

15
Lee, et al. Berkeley 15

Example – Flying Paster



Source: <http://offsetpressman.blogspot.com/2011/03/how-flying-paster-works.html>



Lee, et al. Berkeley 16



Source: <http://offsetpressman.blogspot.com/2011/03/how-flying-paster-works.html>

Flying Paster

Printing Press – Model in Ptolemy II

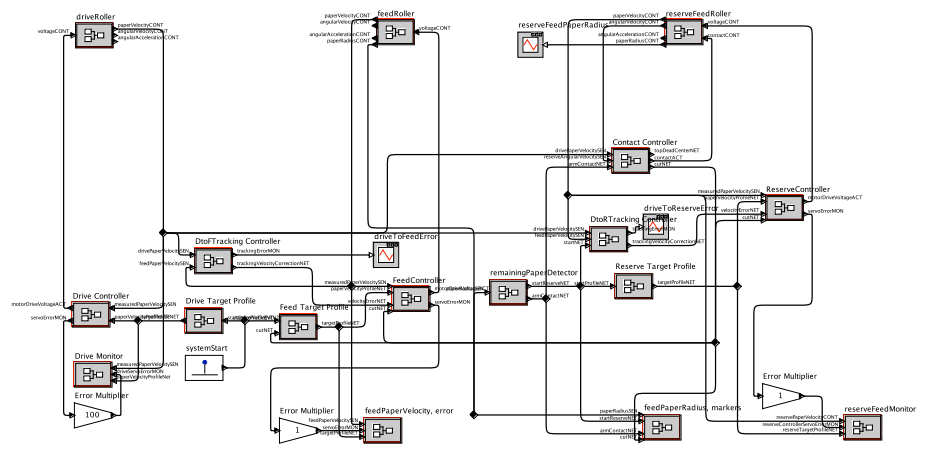
See talk Thursday by Patricia Derler at workshop on Model-Based Design with a Focus on Extra-Functional Properties

This design demonstrates DC motors driving a feed roller and a drive roller. The PID-based motor controllers minimize the error between the paper velocity produced by the roller and the target profile velocity produced by the Target Profile actor. The tracking error input allows one such roller to track the other to remove small differences in paper velocity.

The target profile is either a profile from 0 to maxPaperVelocity starting at time 0 and reaching the maximum value at time Interval seconds. The profile and its derivative are continuous.

SENSOR, ACTUATOR and NETWORK ACTORS STILL NEED TO BE ADDED

- maxPaperVelocity: 35.0
- startupInterval: 120.0
- systemSamplingInterval: 0.40
- systemStart: 0.0
- coreRadius: 0.07
- fullRollRadius: 0.7
- paperThickness: 0.000075



Printing Press – Model in Ptolemy II

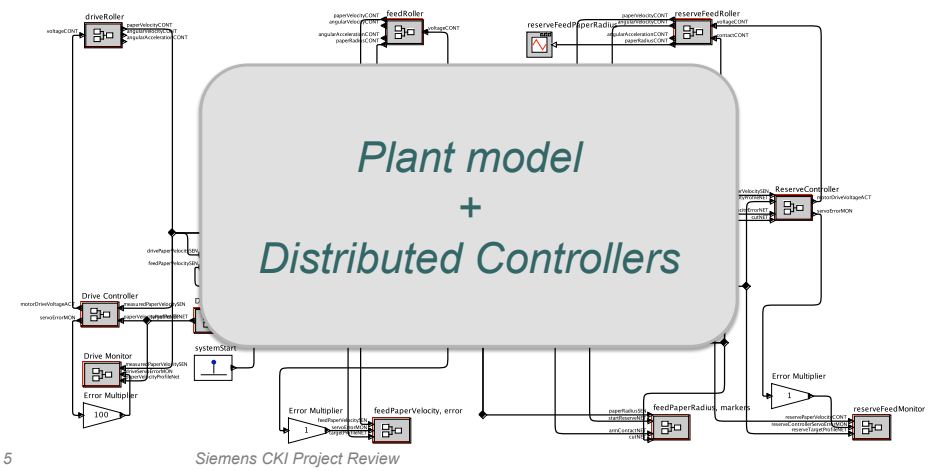
See talk Thursday by Patricia Derler at workshop on Model-Based Design with a Focus on Extra-Functional Properties

This design demonstrates DC motors driving a feed roller and a drive roller. The PID-based motor controllers minimize the error between the paper velocity produced by the roller and the target profile velocity produced by the Target Profile actor. The tracking error input allows one such roller to track the other to remove small differences in paper velocity.

The target profile is either a profile from 0 to maxPaperVelocity starting at time 0 and reaching the maximum value at time interval seconds. The profile and its derivative are continuous.

SENSOR, ACTUATOR and NETWORK ACTORS STILL NEED TO BE ADDED

- DE Director
 - maxPaperVelocity: 35.0
 - startInterval: 120.0
 - systemSamplingInterval: 0.40
 - systemStart: 0.0
- coreRadius: 0.07
- fullRollRadius: 0.7
- paperThickness: 0.000075



5

Siemens CKI Project Review

Printing Press – Model in Ptolemy II

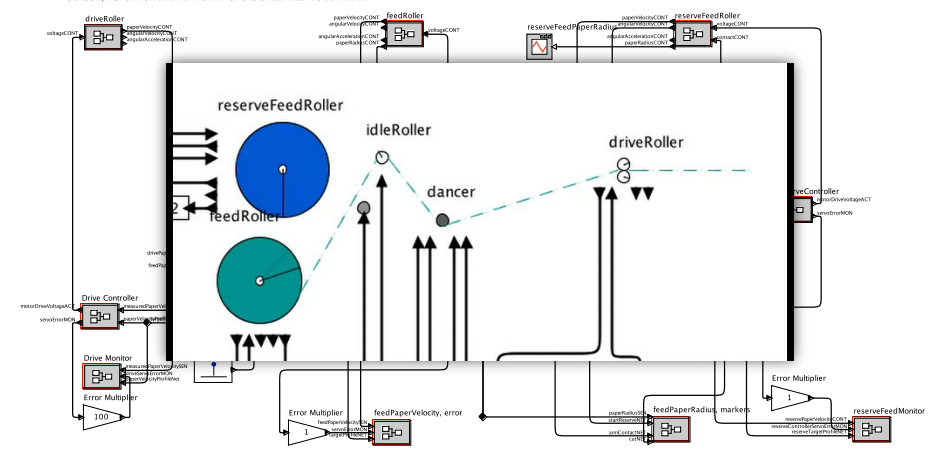
See talk Thursday by Patricia Derler at workshop on Model-Based Design with a Focus on Extra-Functional Properties

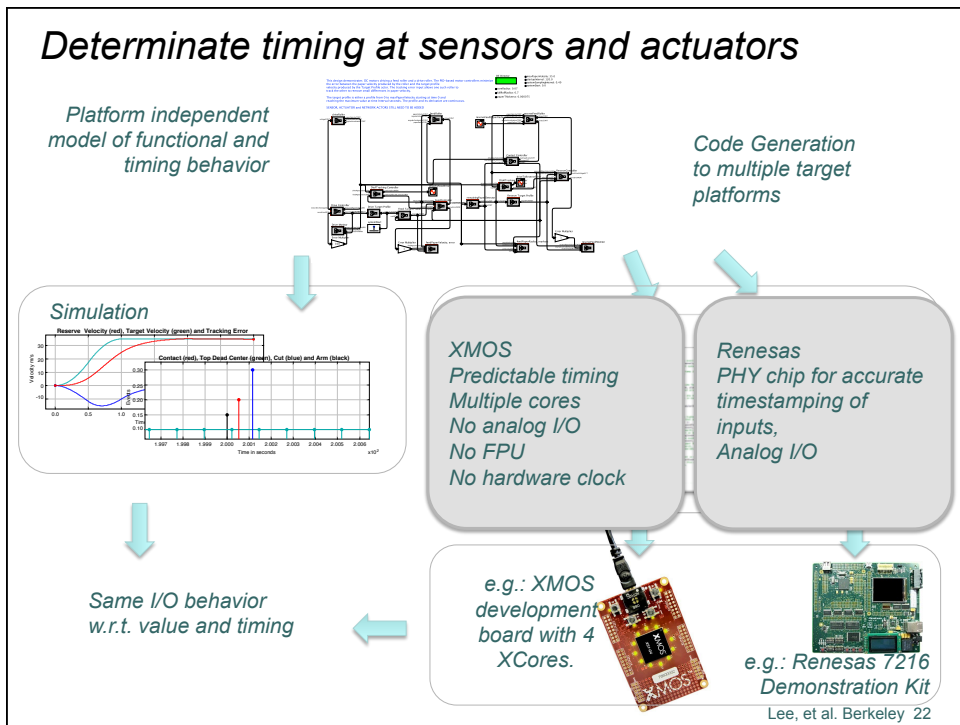
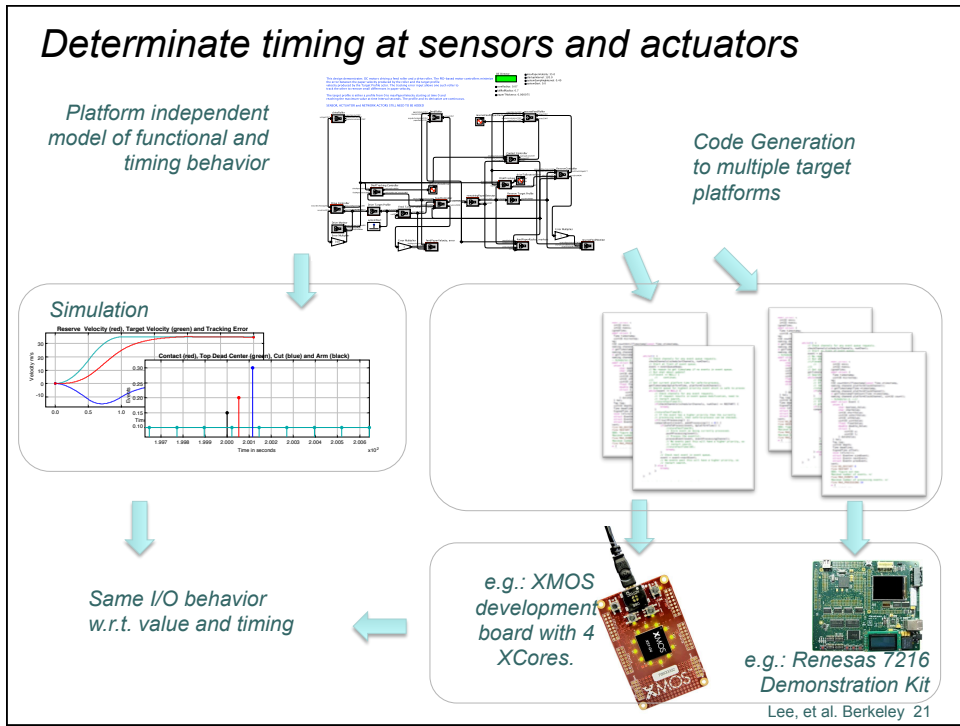
This design demonstrates DC motors driving a feed roller and a drive roller. The PID-based motor controllers minimize the error between the paper velocity produced by the roller and the target profile velocity produced by the Target Profile actor. The tracking error input allows one such roller to track the other to remove small differences in paper velocity.

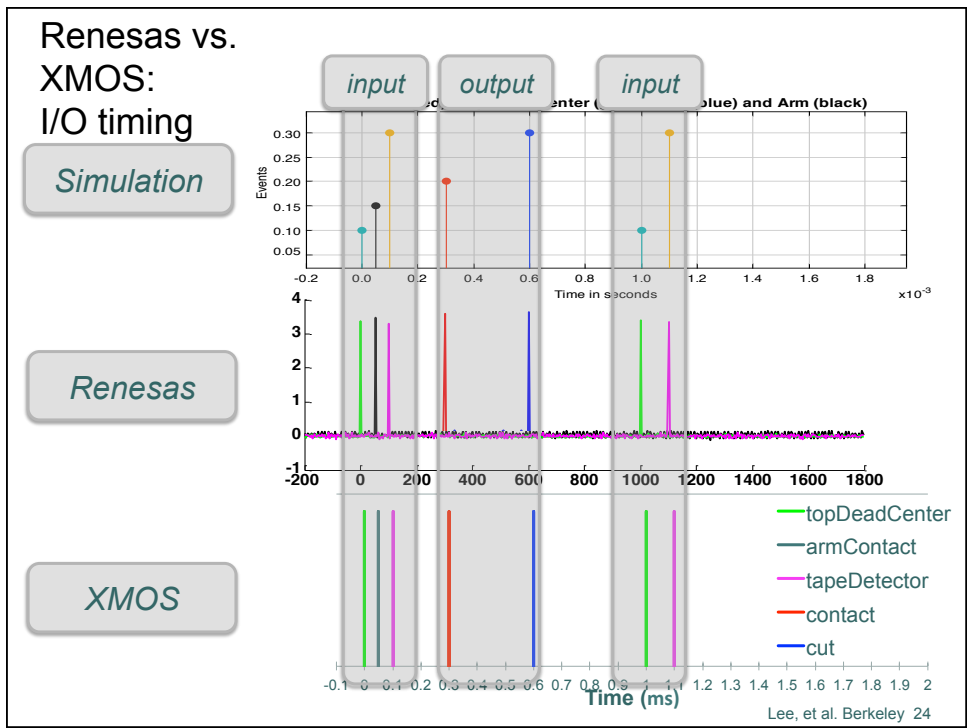
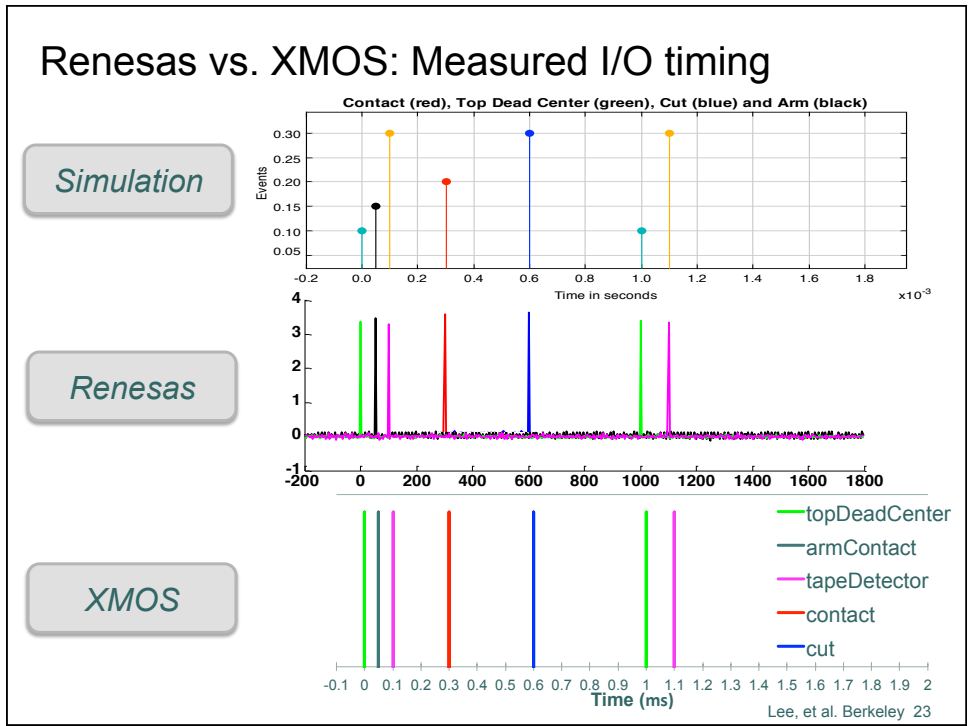
The target profile is either a profile from 0 to maxPaperVelocity starting at time 0 and reaching the maximum value at time interval seconds. The profile and its derivative are continuous.

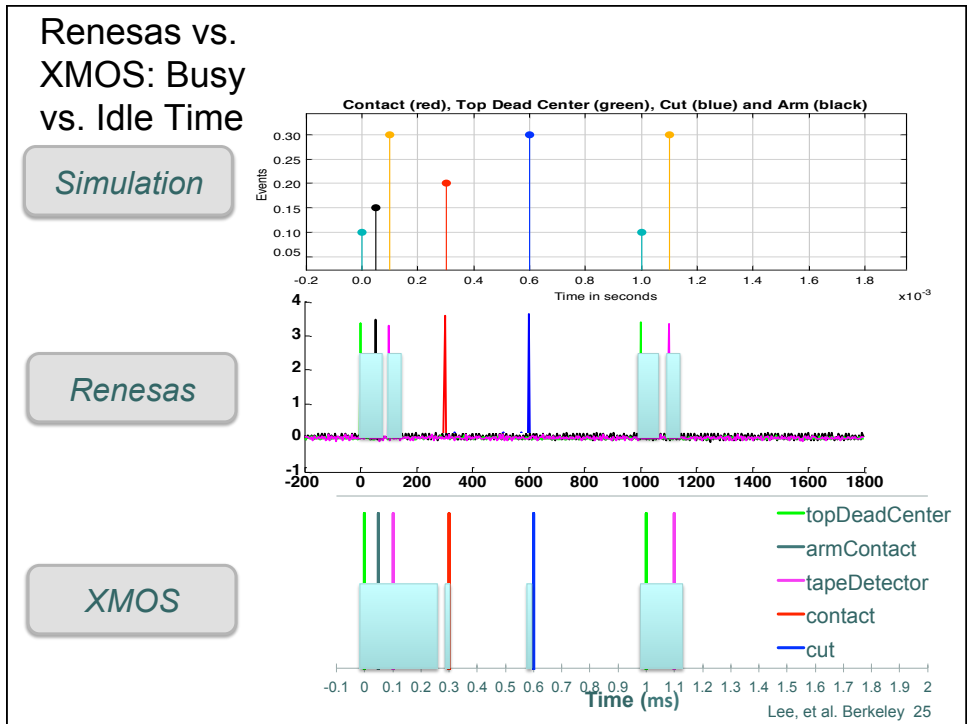
SENSOR, ACTUATOR and NETWORK ACTORS STILL NEED TO BE ADDED

- DE Director
 - maxPaperVelocity: 35.0
 - startInterval: 120.0
 - systemSamplingInterval: 0.40
 - systemStart: 0.0
- coreRadius: 0.07
- fullRollRadius: 0.7
- paperThickness: 0.000075









Ptides Publications

<http://chess.eecs.berkeley.edu/ptides/>

- Y. Zhao, J. Liu, E. A. Lee, “**A Programming Model for Time-Synchronized Distributed Real-Time Systems,**” RTAS 2007.
- T. H. Feng and E. A. Lee. “**Real-Time Distributed Discrete-Event Execution with Fault Tolerance,**” RTAS 2008.
- P. Derler, E. A. Lee, and S. Matic, “**Simulation and implementation of the ptides programming model,**” DS-RT 2008.
- J. Zou, S. Matic, E. A. Lee, T. H. Feng, and P. Derler, “**Execution strategies for Ptides, a programming model for distributed embedded systems,**” RTAS 2009.
- J. Zou, J. Auerbach, D. F. Bacon, E. A. Lee, “**PTIDES on Flexible Task Graph: Real-Time Embedded System Building from Theory to Practice,**” LCTES 2009.
- J. C. Eidson, E. A. Lee, S. Matic, S. A. Seshia and J. Zou, “**Time-centric Models For Designing Embedded Cyber-physical Systems,**” ACES-MB 2010.
- J. C. Eidson, E. A. Lee, S. Matic, S. A. Seshia, and J. Zou, **Distributed Real-Time Software for Cyber-Physical Systems,** To appear in *Proceedings of the IEEE* special issue on CPS, December, 2011.

Lee, et al. Berkeley 26

Conclusions

Overview References:

•Lee, *Computing needs time*. CACM, 52(5):70–79, 2009

•Derler, Lee, Sangiovanni-Vincentelli,

Modeling Cyber-Physical Systems,

To appear in Proc. of the IEEE December, 2011.

Today, timing behavior is a property only of *realizations* of software systems.

Tomorrow, timing behavior will be a semantic property of *programs and models*.

Raffaello Sanzio da Urbino – *The Athens School*

