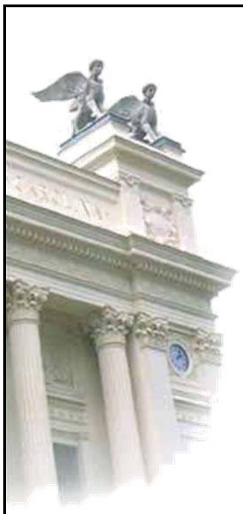


Simulation of Cyber-Physical Control Systems

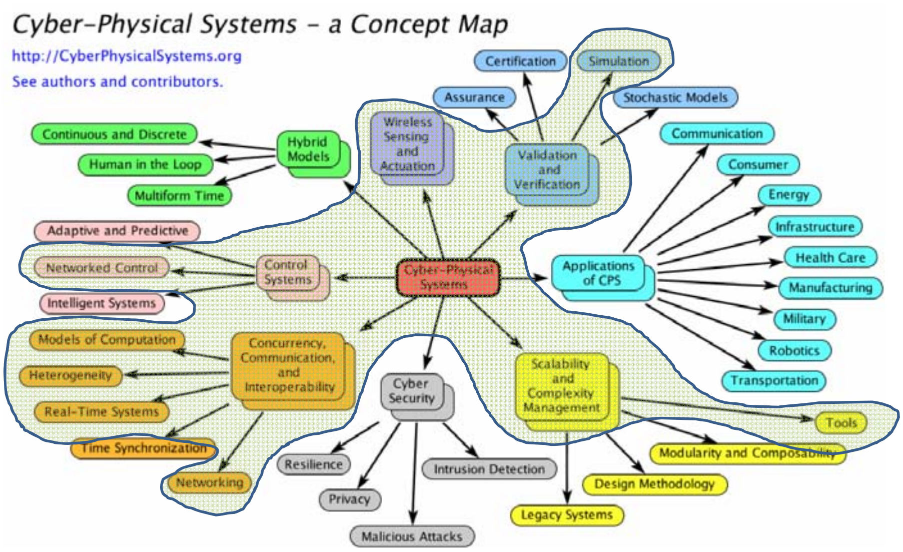
Karl-Erik Årzén
Lund University

Notre Dame Workshop on Control of Cyber-Physical Systems, London, Oct 20-21, 2012

Contents

- Simulation of CPS
- TrueTime
  - New features
- Truetime in Simulink
  - Mobile robotics demo
- Truetime in Modelica
  - Demo

CPS Taxonomy



CPS and Control

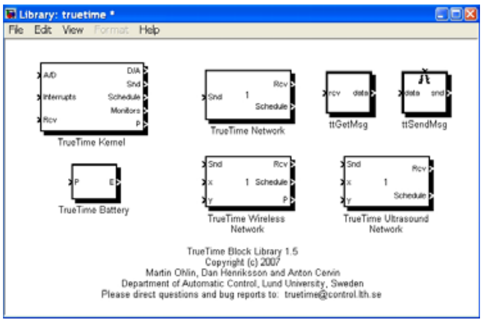
- Large-scale distributed and decentralized control
  - Large scale
  - Decentralized analysis, synthesis and optimization
  - Utilize sparsity
- Resource-constrained implementation platforms
  - Embedded and networked control
  - Temporal nondeterminism and temporal robustness
  - Delays, jitter, lost packets, quantization, ….
  - Codesign, cross/layer design, …..

Simulation of CPS

- Complement to formal methods
- When formal methods are not applicable
- Need simulation tools to cover both the cyber part and and the physical part
- Holistic co-simulation of
  - computations inside the computers
    - tasks and interrupt handlers
  - wireless and wired communication
  - dynamics of the physical parts
  - sensor and actuator dynamics
  - the power consumption in the nodes

TrueTime

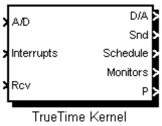
- Simulator for the cyber parts of CPS
- Embedded in physical system simulators (Simulink, Modelica)
- Simulation of
  - Real-time kernels
  - Wired and wireless networks
- Developed in Lund since 1999
  - Version 2.0
  - Large userbase
  - GPL

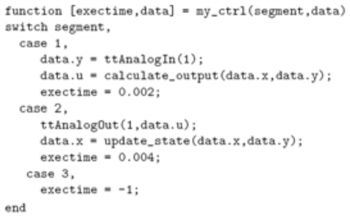


TrueTime Block Library 1.5
Copyright (c) 2007
Martin Ohlin, Dan Henriksson and Anton Cervin
Department of Automatic Control, Lund University, Sweden
Please direct questions and bug reports to: truetime@control.lth.se

Modeling of Computations

- Simulates an event-based real-time kernel
- Executes user-defined tasks and interrupt handlers
  - C/C++ or M-files
- Arbitrary user-defined scheduling policies
- Real-time primitives
- Code structured into code segments
  - emulate multithreading

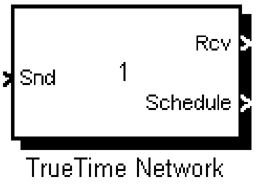




```
function [exectime,data] = my_ctrl(segment,data)
switch segment,
  case 1,
     data.y = ttAnalogIn(1);
     data.u = calculate_output(data.x,data.y);
     exectime = 0.002;
  case 2,
     ttAnalogOut(1,data.u);
     data.x = update_state(data.x,data.y);
     exectime = 0.004;
   case 3,
     exectime = -1;
end
```

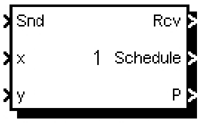
Modeling of Wired Networks

- Models the medium access delay and the transmission delay
- A number of pre-defined data-link layer protocols
  - Switched Ethernet
  - CAN
  - Round Robin
  - FDMA
  - TDMA
  - CSMA/CD (Shared Ethernet)
  - Flexray
  - PROFINET IO



Modeling of Wireless Networks

- Supports two common MAC layer policies:
  - IEEE 802.11 b/g (WLAN)
  - IEEE 802.15.4 (“ZigBee”)
- x and y inputs for node locations (2D)
- Radio models:
  - Exponential path loss (default)
  - User-defined models to model fading etc

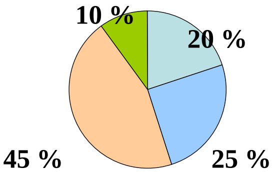


TrueTime Wireless Network

9

New Features

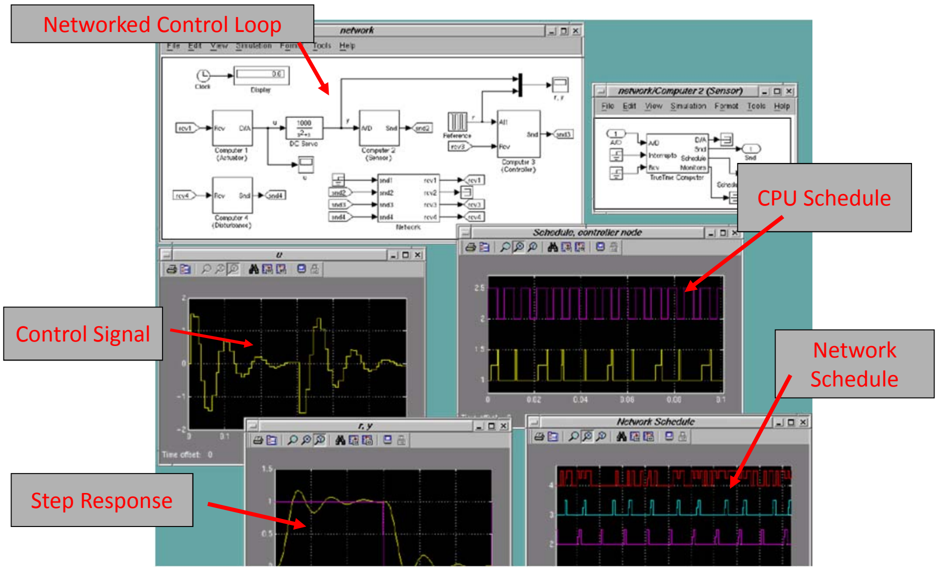
- Multicore kernels
  - Each TrueTime kernel may have multiple cores
  - Partitioned scheduling
  - `ttSetNumberOfCPUs(no)`
  - `ttSetCPUAffinity(task,cpu)`
- Constant bandwidth servers (CBS)
  - Virtual processors
  - Temporal isolation
  - `ttCreateCBS(budget,period)`
  - `ttAttachCBS(task,CBS)`
  - `ttSetCBSParameters(budget, period)`



TrueTime for Simulink

- S-function interface
  - Kernels
  - Networks
- Task code
  - C/C++
  - M-file script language

TrueTime: Networked Embedded Control



Networked Control Loop

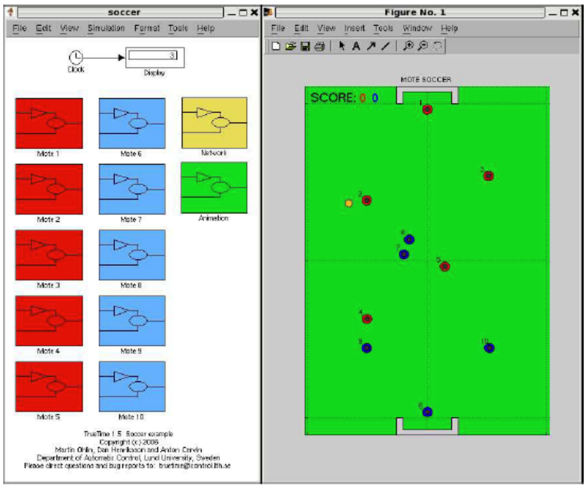
CPU Schedule

Control Signal

Network Schedule

Step Response

TrueTime: Mobile Wireless Robotics



TrueTime 1.5 Soccer example
Copyright (c) 2008
Martin Ohlin, Dan Henriksson and Anton Cervin
Department of Automatic Control, Lund University, Sweden
Please direct questions and bug reports to: truetime@control.lth.se

TrueTime: Mobile Robotics

- Tunnel road safety scenario in RUNES
  - EU FP6 IP (2004-2007)
  - Coordinated by Ericsson
- Stationary sensor network in a road tunnel
- Mobile robots as mobile gateways for restoring connectivity among isolated subislands of the network

Localization

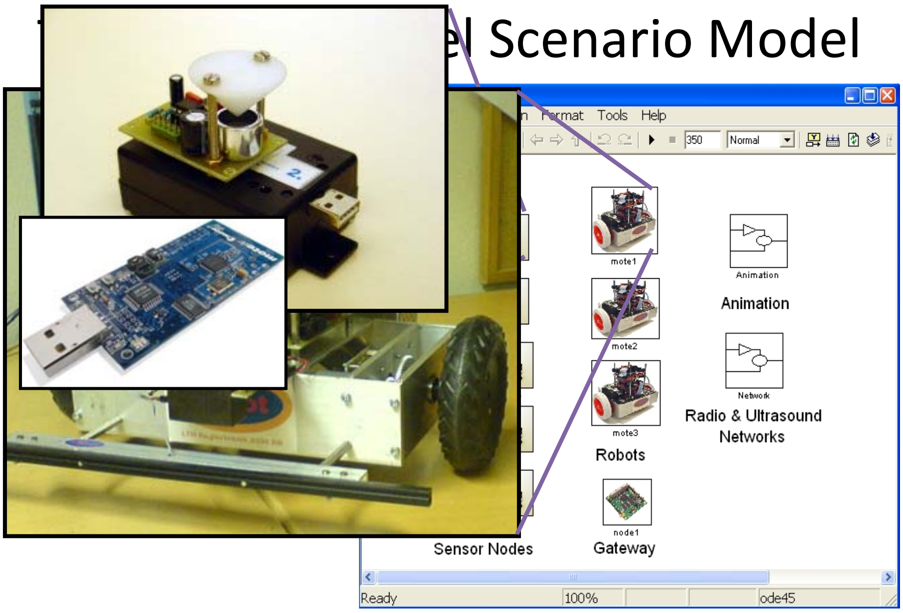
- Ultrasound-based
  - Active mobile robots
  - Passive stationary nodes
- Robot broadcasts radio packet and ultrasound pulse ”simultaneously”
- Difference in time-of-arrival allows each reachable node to calculate its distance to the robot
- Each node sends its distance measurement back to the robot
- Extended Kalman Filter fuses distance measurements with wheel encoders

Verification Problem

- Robot with several microprocessors, I2C bus communication
- Sensor network radio communication
  - IEEE 802.11 b/g (WLAN)
  - AODV routing protocol
- Ultrasound localization
- IR-based obstacle avoidance
- Control and estimation

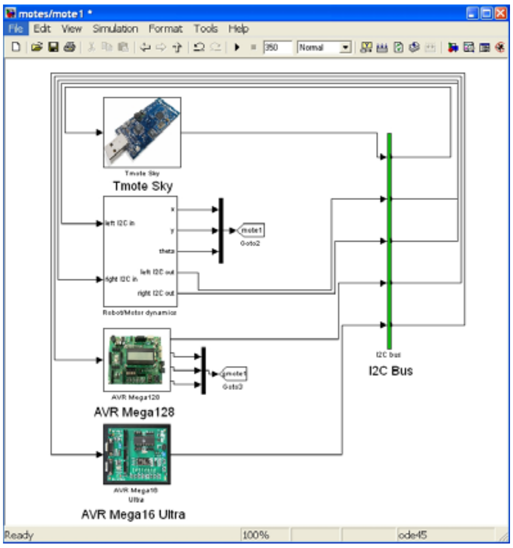
➡ How verify the functionality and timeliness of this??
  - TrueTime used for developing a simulator in parallel with the real physical implementation
  - Proof of concept and verification

l Scenario Model

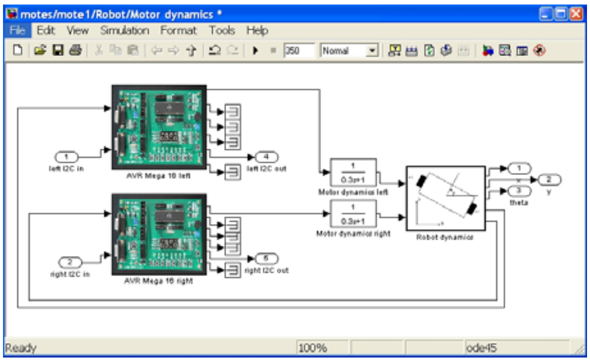


Robot Submodel

- Tmote Sky
  - Radio interface & bus master
  - Robot controller
- AVR Mega128
  - Compute engine
  - IR interface
  - EKF, navigation, and obstacle avoidance
- AVR Mega16
  - Ultrasound interface
- I2C bus
- Wheel and motor submodel

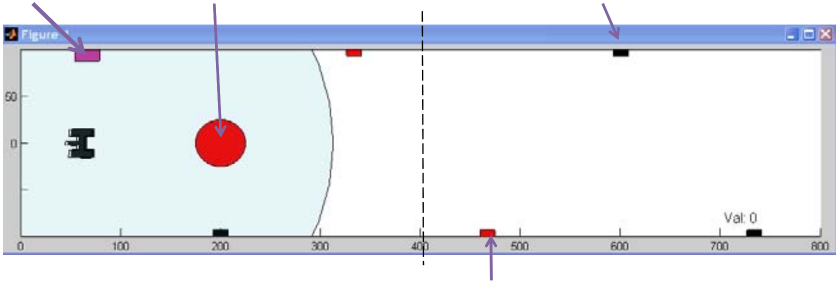


Wheel and Motor Submodel



- One AVR Mega16 for each wheel/motor
- Simple motor models
- Dual-drive unicycle robot dynamics model

Animation



Gateway

Obstacle

Partition

Active sensor

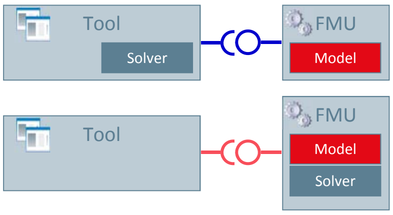
Inactive sensor

- Both the true position of the robots and their internal estimate of their position are shown
- A sensor node that is turned off will not participate in the message routing and in the ultrasound localization

Demo

TrueTime for Modelica

- Network part
  - Native Modelica version
  - External C code version for Dymola
- Full TrueTime
  - Flexible Mockup Interface (FMI)
    - Open source non-proprietary model exchange format
    - Model Exchange
    - Co-Simulation



TrueTime for FMI

- Kernels and Networks are Flexible Mockup Units (FMUs)
  - Modelica simulation tools:
    - Dymola
    - Open-source tools: OpenModelica, JModelica
  - Non-Modelica tools that embrace FMI
- Task code written in C
- Work in progress
  - Vanderbilt University
  - DARPA Adaptive Vehicle Make (AVM) programme
  - TrueTime a part of the Meta toolchain for CPS

Demo

Simulation of CPS

- Even more important than in conventional control
- Co-simulation of cyber parts and physical parts
- TrueTime embeds models of the cyber parts within physical system simulators
  - Simulink
  - Modelica simulators

References

- Anton Cervin, Dan Henriksson, Bo Lincoln, Johan Eker, Karl-Erik Årzén, How Does Control Timing Affect Performance? Analysis and Simulation of Timing Using Jitterbug and TrueTime, *IEEE Control Systems Magazine,* **23:**3, pp. 16–30, June 2003.
- Karl-Erik Årzén, Martin Ohlin, Anton Cervin, Peter Alriksson, Dan Henriksson, Holistic Simulation of Mobile Robot and Sensor Network Applications Using TrueTime, In *Proceedings of the European Control Conference*, Kos, Greece, July 2007.
- Johan Åkesson, Karl-Erik Årzén, Magnus Gäfvert, Tove Bergdahl, Hubertus Tummescheit Modeling and Optimization with Optimica and JModelica.org—Languages and Tools for Solving Large-Scale Dynamic Optimization Problem, *Computers and Chemical Engineering,* **34:**11, pp. 1737-1749, November 2010.