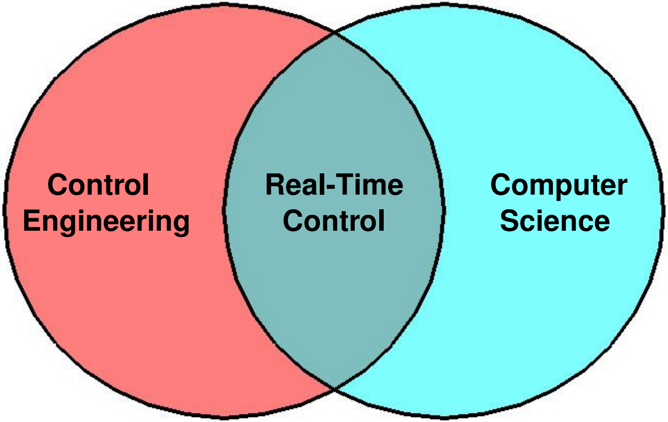
Truetime for Simulation of Networked Embedded Control Systems

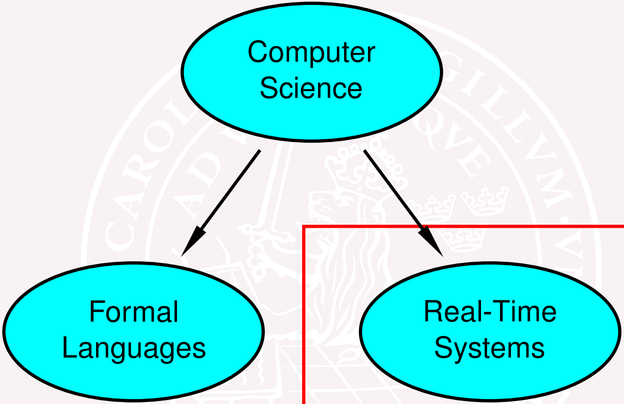
Anton Cervin, Dan Henriksson, Martin Ohlin, Karl-Erik Årzén

Department of Automatic Control
Lund University
Sweden

Positioning of TrueTime



Positioning of TrueTime



State Machines
Petri Nets
Timed Automata

RT Operating Systems
RT Scheduling Algorithms
Network Protocols

TrueTime Main Idea

Co-simulation of control task execution,
network communication, and plant dynamics.

- Simulink blocks that model real-time kernels and communication networks
- The kernels execute user code (tasks and interrupt handlers) written in C++ or MATLAB code
- The simulated application is programmed in much the same way as a real application

TrueTime Main Idea

Co-simulation of control task execution,
network communication, and plant dynamics.

- Simulink blocks that model real-time kernels and communication networks
- The kernels execute user code (tasks and interrupt handlers) written in C++ or MATLAB code
- The simulated application is programmed in much the same way as a real application

Why Co-Simulation?

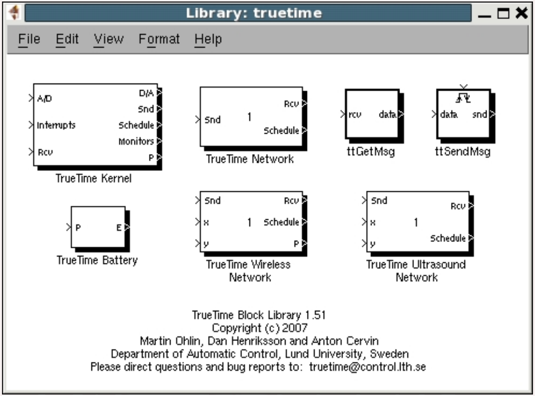
- Networked embedded systems are very complex systems
- Temporal nondeterminism
  - preemption by higher-priority tasks, blocking, varying computation times, kernel overhead, . . .
  - network interface delays, queuing delays, (re)transmission delays, lost packets, . . .

A Very Brief History

- 1999 – first release
- 2005 – version 1.3
  - Wireless networks
  - Battery-powered devices
  - Dynamic voltage scaling
  - Local clocks
- 2006 – version 1.4
  - User-defined pathloss function
  - AODV routing
- 2007 – version 1.5
  - Stand-alone network interface blocks

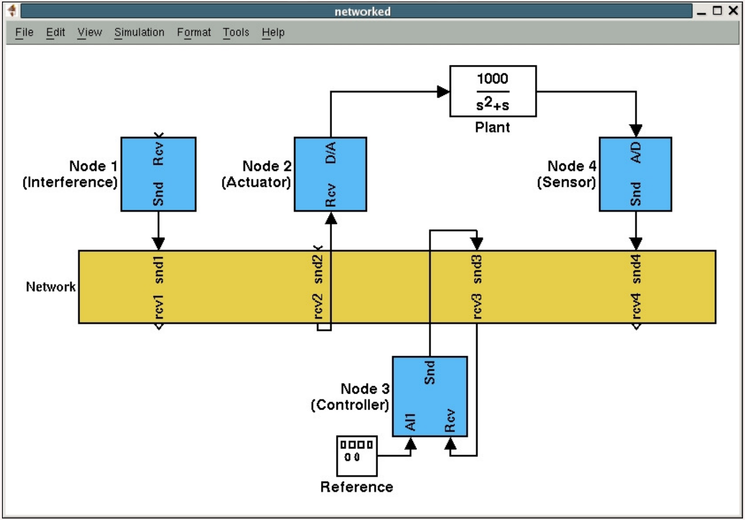
Contributors: J. Eker, A. Cervin, D. Henriksson, M. Ohlin

The TrueTime block library

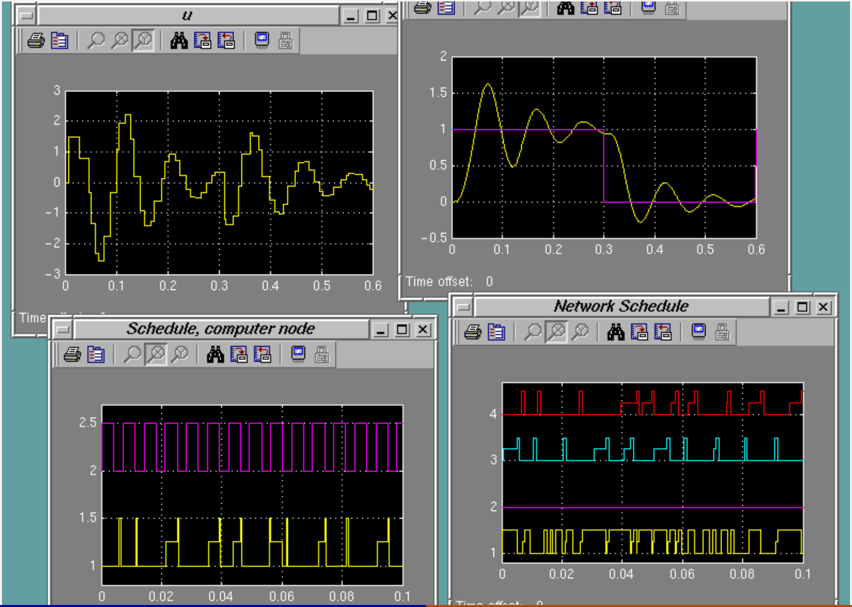


- A Kernel block, three Network blocks, and a Battery block
  - Simulink S-functions written in C++
  - Event-based execution using zero-crossing functions
  - Portable to other simulation environments

Example – Networked Control Loop

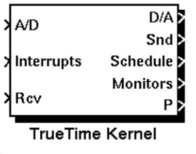


Example – Networked Control Loop



The Kernel Block

- Simulates a generic real-time kernel with A/D-D/A and network interfaces
- Executes user-defined tasks and interrupt handlers
- Supports various scheduling policies
- Supports all common real-time primitives (sleepUntil, setPriority, wait/notify, . . .)
- More features: context switch overheads, overrun handlers, data logging, . . .



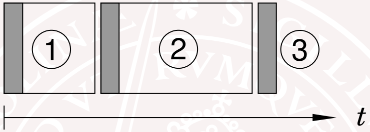
Example of Kernel Initialization Script

```
nbrInputs = 3;
nbrOutputs = 3;
ttInitKernel(nbrInputs, nbrOutputs, 'prioFP');
periods = [0.01 0.02 0.04];
code = 'my_ctrl';
for k = 1:3
   data.u = 0;
   taskname = ['Task ' num2str(k)];
   offset = 0;
   period = periods(k);
   prio = k;
   ttCreatePeriodicTask(taskname,offset,period,prio,code,data);
end
```

Code Functions

- Each task or interrupt handler in the user application must be implemented in a code function
- The code function is called repeatedly by the kernel during the simulation
- The simulated execution time of the code is returned by the code function
- Three options for the implementation:
  - C++ code (fast)
  - MATLAB code (medium)
  - Simulink block diagram (slow)

Code Segments



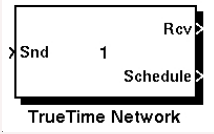
- The code is divided into one or several segments
- The code in each segment is executed nonpreemptively
- Multiple segments are used to simulate
  - input-output delays
  - self-suspensions
  - waiting (for events, semaphores, monitors, or mailboxes)
  - loops or branches

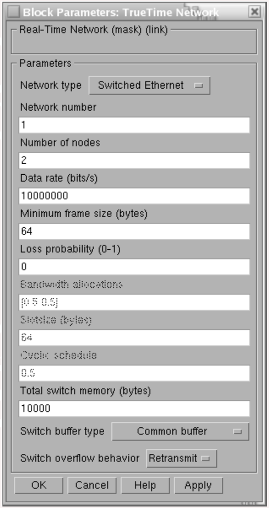
Example of a MATLAB Code Function

```
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
```

The Wired Network Block

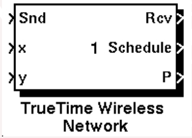
- Supports six common MAC layer policies:
  - CSMA/CD (Ethernet)
  - CSMA/AMP (CAN)
  - Round Robin (Token bus)
  - FDMA
  - TDMA
  - Switched Ethernet
- Policy-dependent network parameters
- Generates a transmission schedule

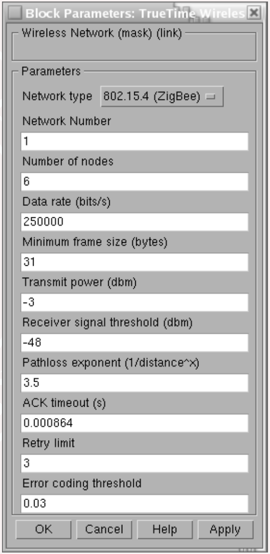




The Wireless Network Block

- Used in the same way as the wired network block
- Supports two common MAC layer policies:
  - 802.11b/g (WLAN)
  - 802.15.4 (ZigBee)
- Variable network parameters
- x and y inputs for node locations
- Generates a transmission schedule





The Wireless Network Model

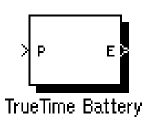
- Isotropic antennas
- Default path-loss formula:

$$\frac{1}{d^a}$$

  - d – distance between nodes $\left(= \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}\right)$
  - a – environment parameter (e.g., 2–4)
- The SIR in the receiver is calculated, and a probabilistic measure is used to determine the number of bit errors
- An error coding threshold is used to determine whether the package can be reconstructed

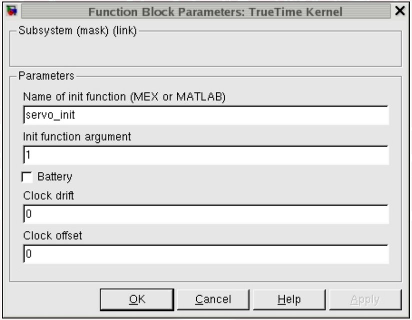
The Battery Block / Dynamic Voltage Scaling

- Simulation of battery-powered devices
- Simple integrator model
  - Discharged or charged
- Energy sinks:
  - Computations, radio transmissions, usage of sensors and actuators, ...
- Dynamic Voltage Scaling
  - The kernel CPU speed can be changed from the application to consume less power



Local Clocks with Offset and Drift

- Distributed systems with local clocks
- Sensor networks are based on cheap hardware:
  - Low manufacturing accuracy ⇒ large clock drift
- Simulate clock synchronization protocols



A Real-World Application

- Multiple processors and networks
- Based on VxWorks and IBM Rational Rose RT
- Used TrueTime to describe the timing behavior
- Has ported TrueTime to a mechatronics simulation environment

“We found TrueTime to be a great tool for describing the timing behavior in a straightforward way.”

More Real-World Applications

Bosch AG

- Extended the network block with support for TTCAN and Flexray
- Used for investigating the impact of time-triggered communication on a distributed vehicle dynamics control system

Haldex AB

- Simulation of CAN-based distributed control systems

TrueTime – Summary

Co-simulation of

- computations inside computer nodes
- wired and wireless communication between nodes
- the dynamics of the physical plant under control
- sensor and actuator dynamics
- the dynamics of mobile robots/nodes
  - position-dependent communication conditions
- the dynamics of the environment
- energy consumption in the nodes

Some Work in Progress

- Multi-threaded C++ version
  - No code segments
- Scilab/Scicos version

Download

TrueTime is freeware and can be downloaded from

`http://www.control.lth.se/truetime`