Model-Based Design:

when you just can't hack it.

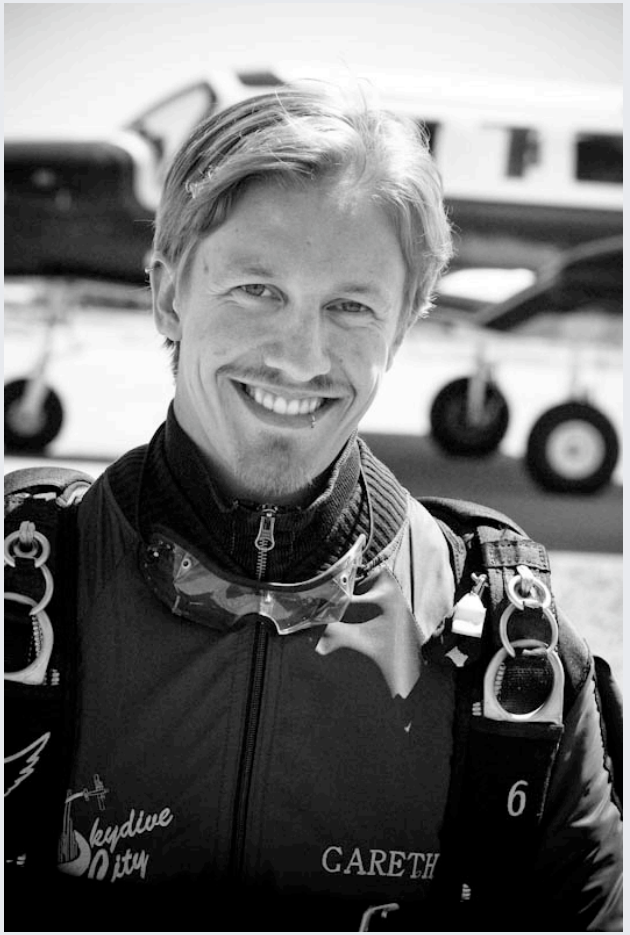
University of California, Berkeley
Department of Electrical Engineering & Computer Science

Edited and presented by

Jeff C. Jensen

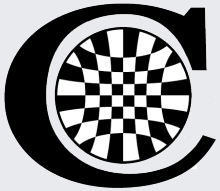
About the Presenter

- B.S. Electrical Engineering & Computer Science
  - University of California, Berkeley
- M.S. Electrical Engineering & Computer Science
  - University of California, Berkeley (Dec. 2009)
- Center for Hybrid and Embedded Software Systems (CHESS)
- Research: PtidyOS
  - Embedded RTOS for code generation
  - PTIDES model of computation
  - Advisor: Professor Edward A. Lee



Jeff C. Jensen

Cyber-Physical Systems

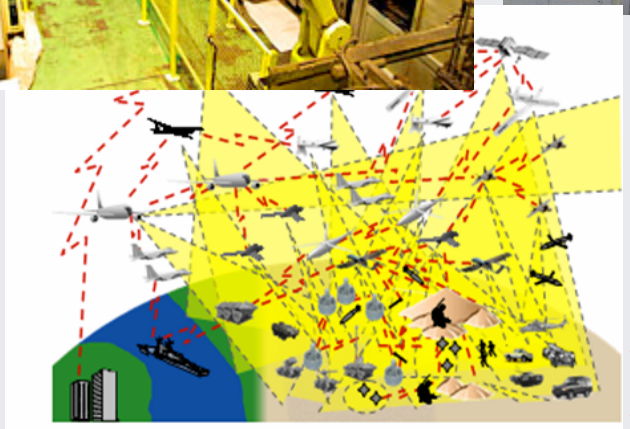


Cyber-physical systems enable seamless, fully synergistic integration of computational intelligence, communication, control, sensing, actuation, and adaptation with physical devices and information processes to routinely realize high-confidence, optimally performing systems that are essential for effectively operating life-, safety-, security-, and mission-critical applications. These systems must be capable of interacting correctly, safely, and securely with humans and the physical world in changing environments and unforeseen conditions.

Source: National Science Foundation, Presidential Budget Amendment; 2010

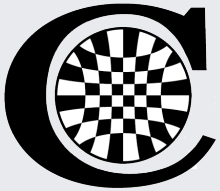
Distributed Real-Time Systems

Multiple computers, comprising of sensors and actuators, connected on a network that act and react on events to meet timing constraints.



Source: Y. Zhao, E. Lee, J. Lui, PTIDES: A Programming Model for Time-Synchronized Distributed Real-time Systems. University of California, Berkeley, department of Electrical Engineering & Computer Science, 2007.

The Problem with Threads



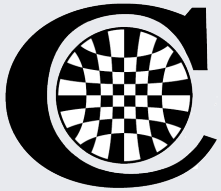
Nontrivial software written with threads is incomprehensible to humans. It cannot deliver repeatable and predictable timing, except in trivial cases.

-Edward A. Lee, University of California, Berkeley

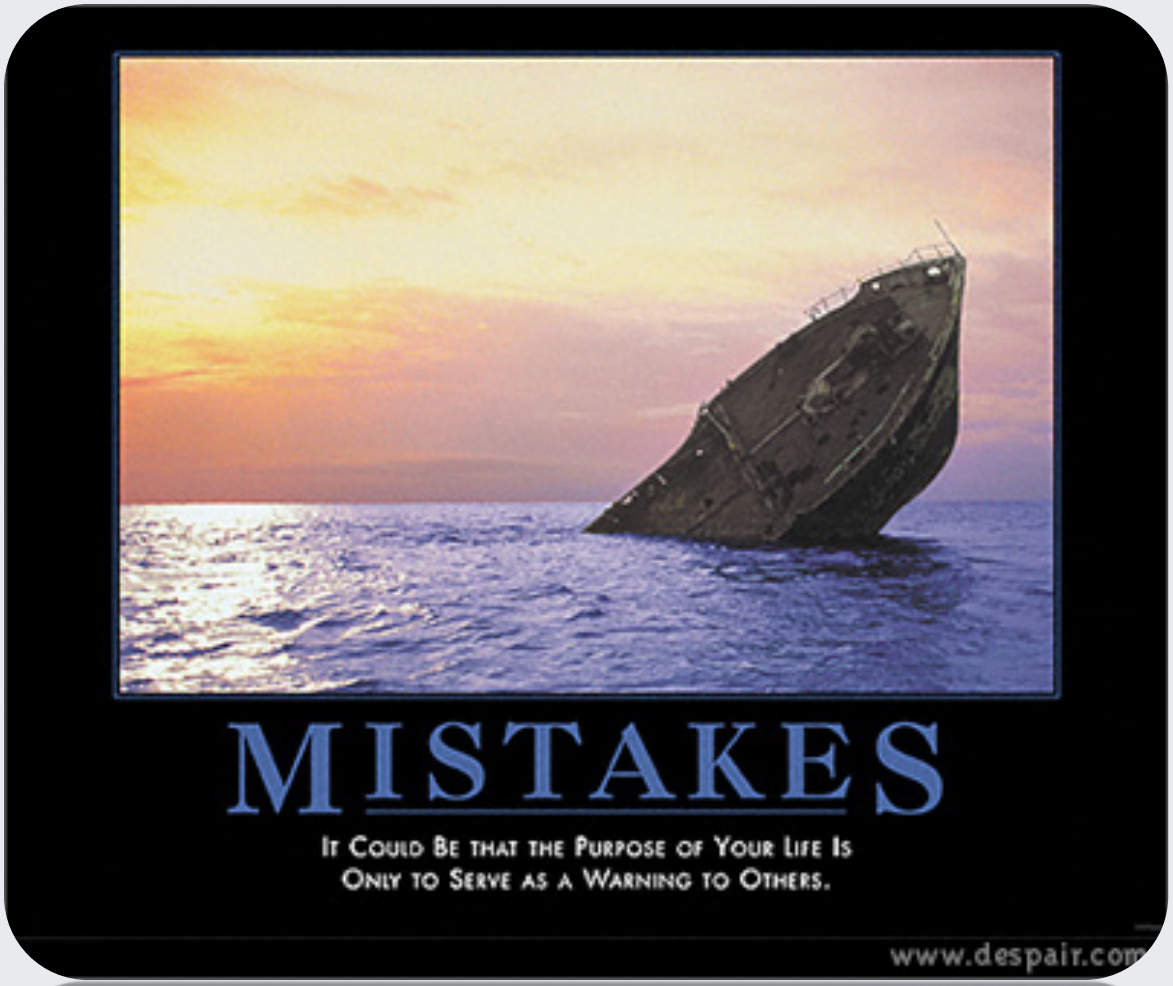
The Problem with Threads

- Wildly nondeterministic
- Incomprehensible
- Unverifiable
- Brittle designs
- Anomalous
- When are they successful?

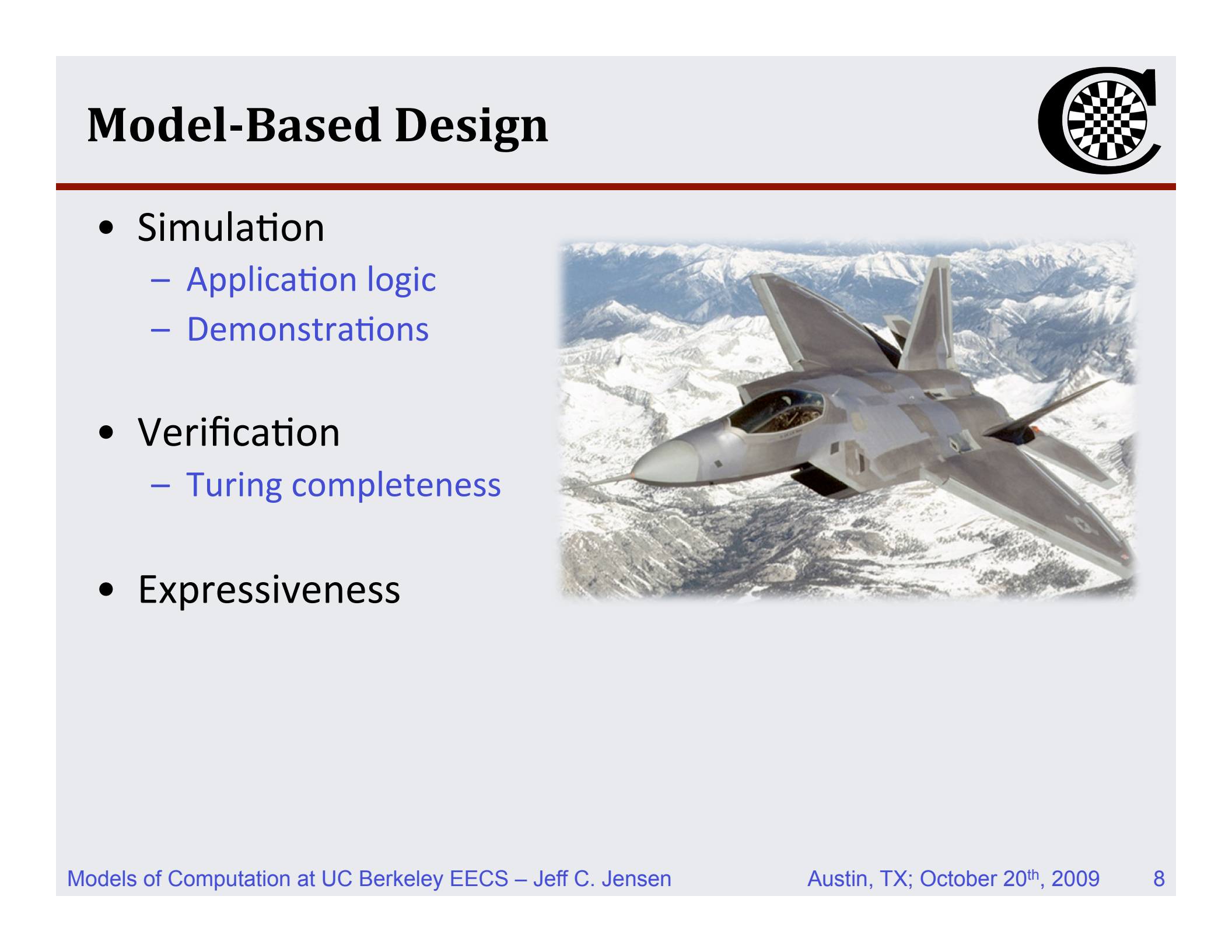
The Problem with Threads: Decidability



- Execution time
- Incomprehensible
- Unverifiable
- Brittle designs
- Anomalous
- Successful?

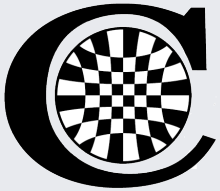


Model-Based Design

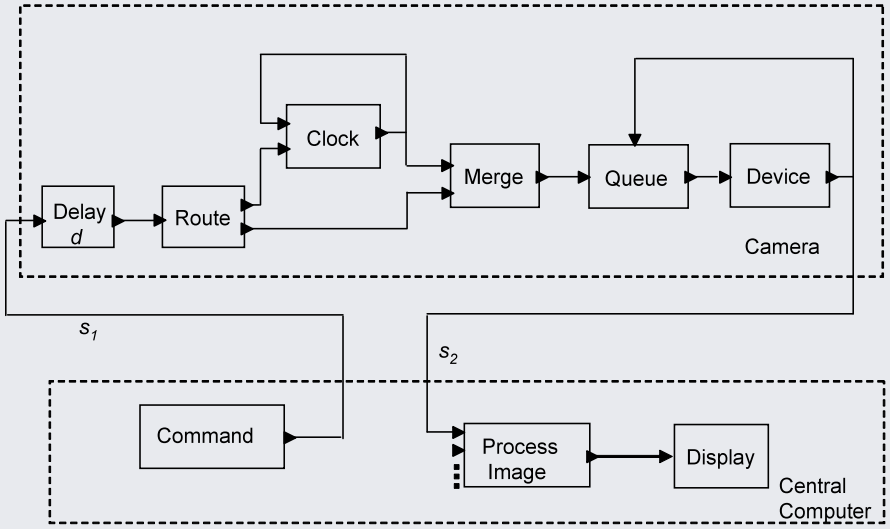


- Simulation
  - Application logic
  - Demonstrations
- Verification
  - Turing completeness
- Expressiveness

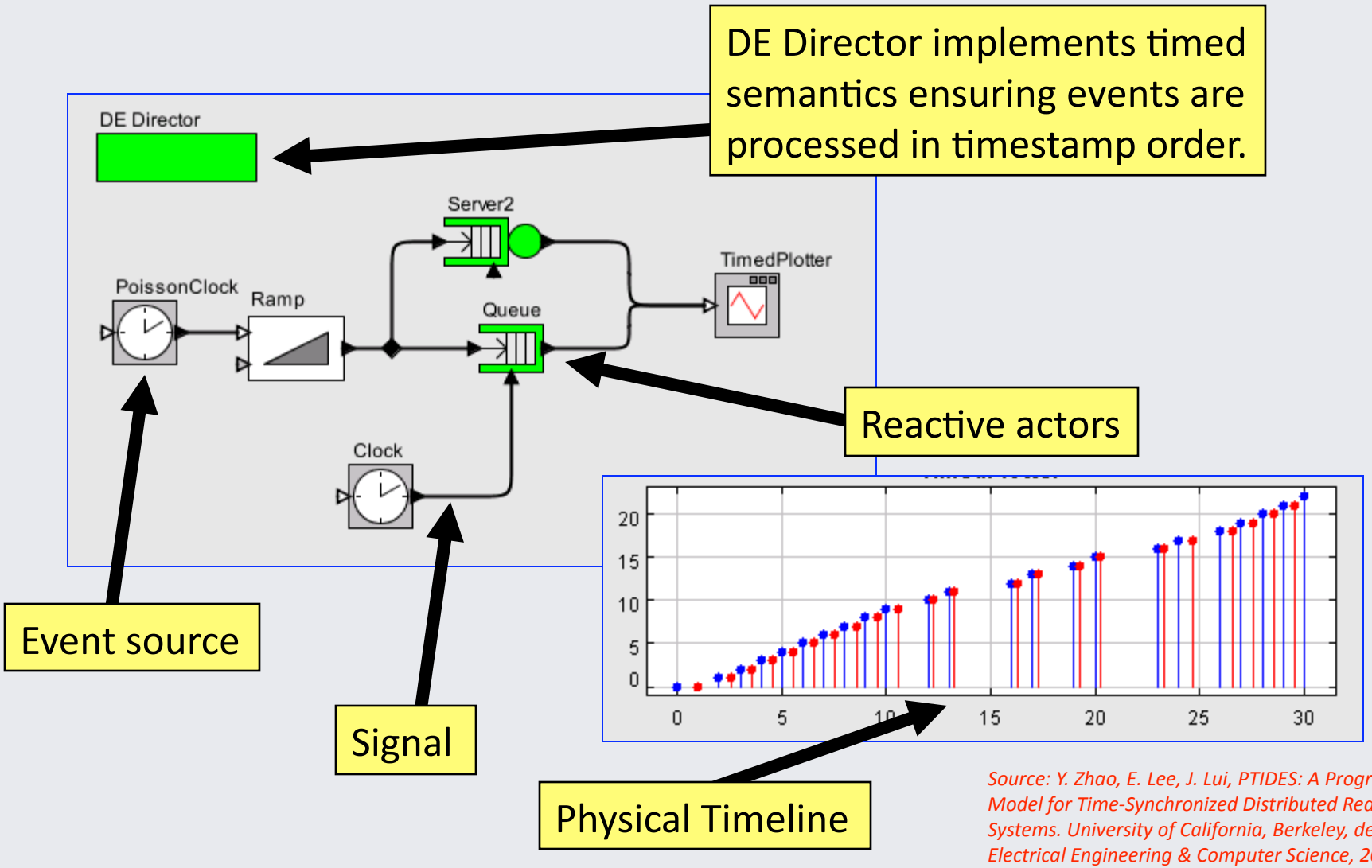
Discrete Event Model of Computation



- Primarily used in performance modeling and simulation:
  - Hardware systems (VHDL, Verilog)
  - Manufacturing systems
  - Communication networks (OPNET, NS-2)
  - Transportation systems
  - Stock market

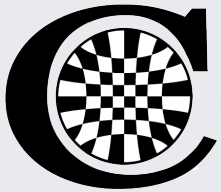


Discrete Event Modeling in Ptolemy II



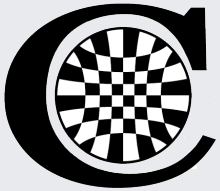
Source: Y. Zhao, E. Lee, J. Lui, PTIDES: A Programming Model for Time-Synchronized Distributed Real-time Systems. University of California, Berkeley, department of Electrical Engineering & Computer Science, 2007.

Discrete Event Model of Computation



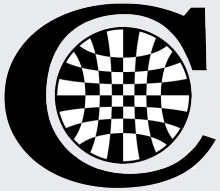
- Drawbacks
  - Centralized queue, sorted by timestamp
  - Difficulty in distributed networks
    - Conservative v. Optimistic (Chandy & Misra v. Jefferson)
  - Events processed in timestamp order regardless of dependencies

PTIDES

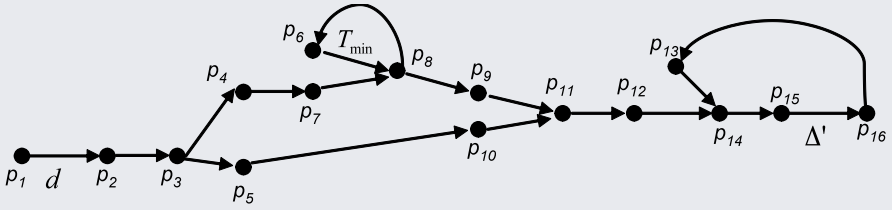


- **P**rogramming **T**emporally **I**ntegrated **D**istributed **E**mbedded **S**ystems
- Based on Discrete Event (DE)
- Binding model and physical time
- Contributor: Hugo Andrade, National Instruments
- Director: Professor Edward A. Lee

PTIDES – Static Analysis

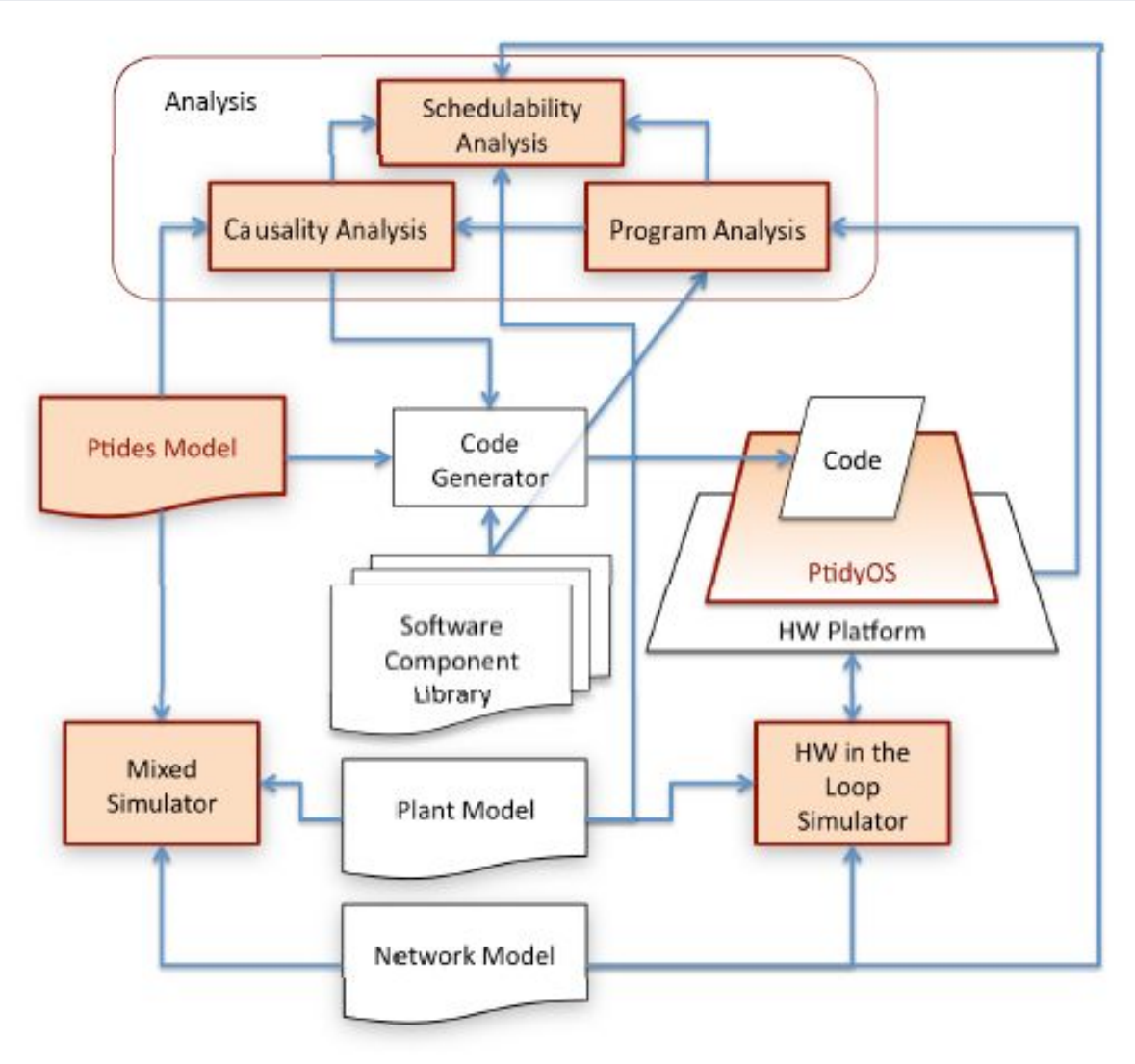


- Static causality analysis
  - Formal framework for dependency graphs
    - Establishes minimal ordering constraints
    - Timestamp ordering is considered only for causally related events
  - Executed at code-generation time
- Improves executability
- Improves reliability



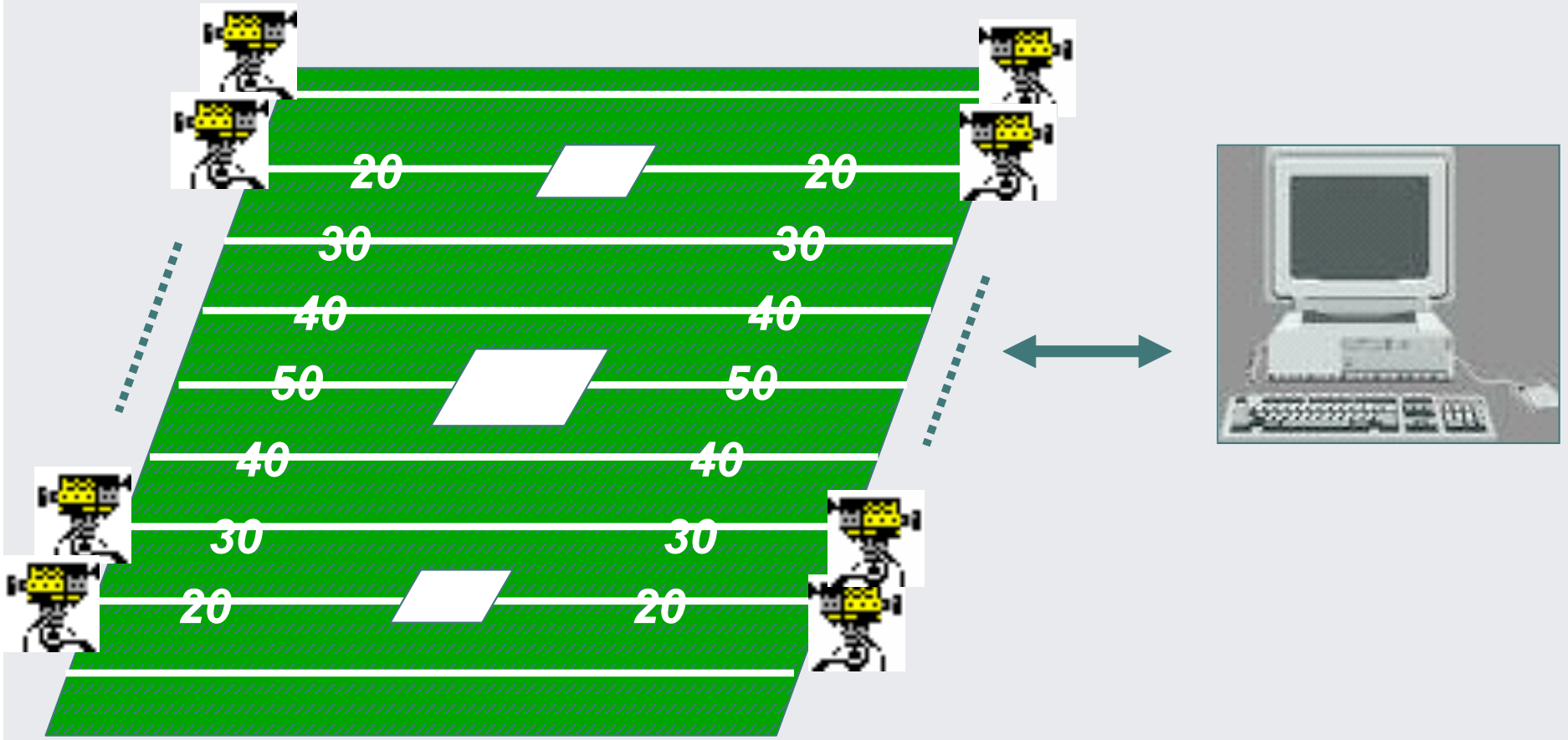
Source: Y. Zhao, J. Liu, and E. Lee, "A Programming Model for Time-Synchronized Distributed Real-Time Systems", Proceedings of the 13th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 07). Bellevue, WA, United States, April 3-6, 2007.

PTIDES – Development

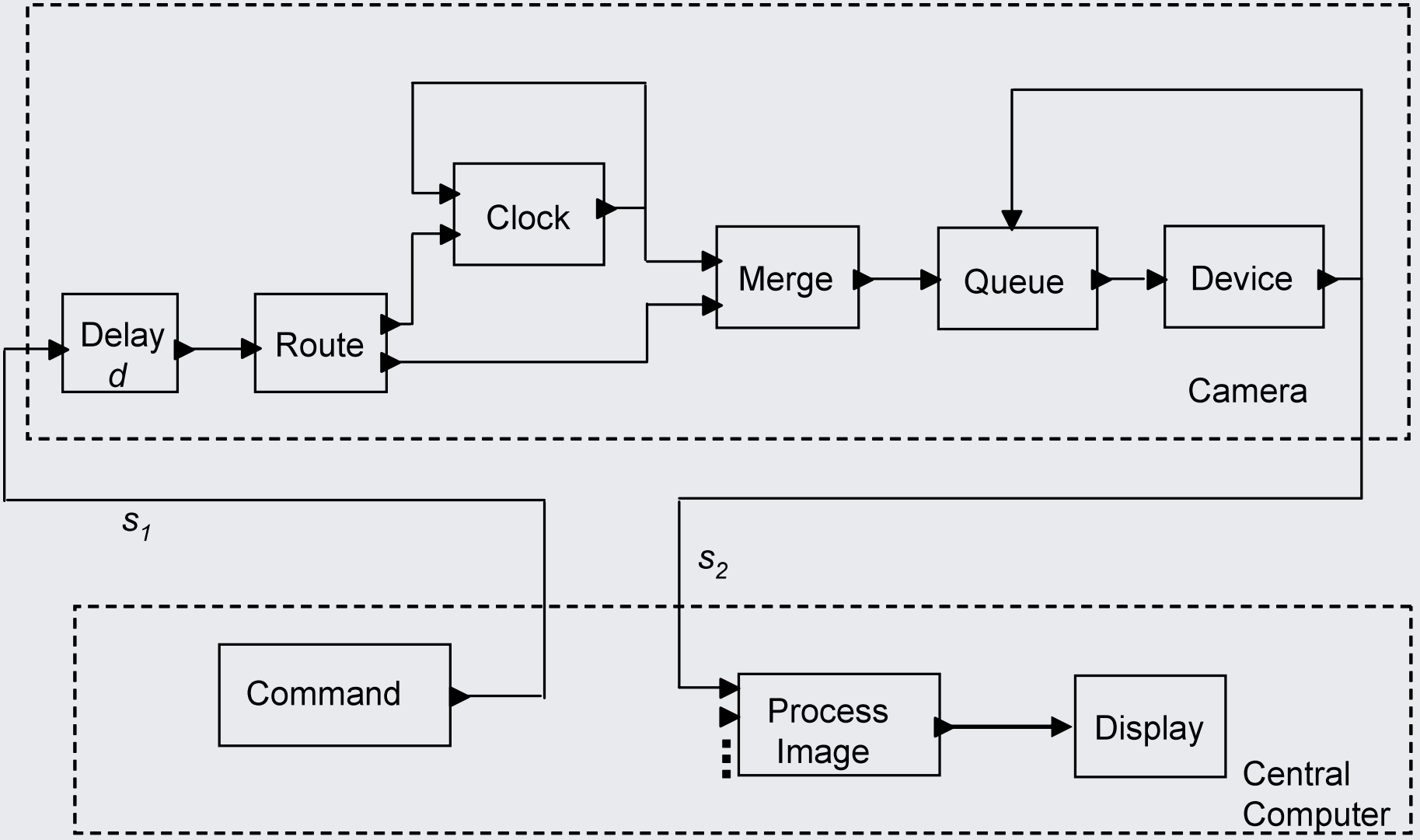


- Static analysis
  - Causality
  - Schedulability
- Code generation
- PtidyOS
- Regression Testing

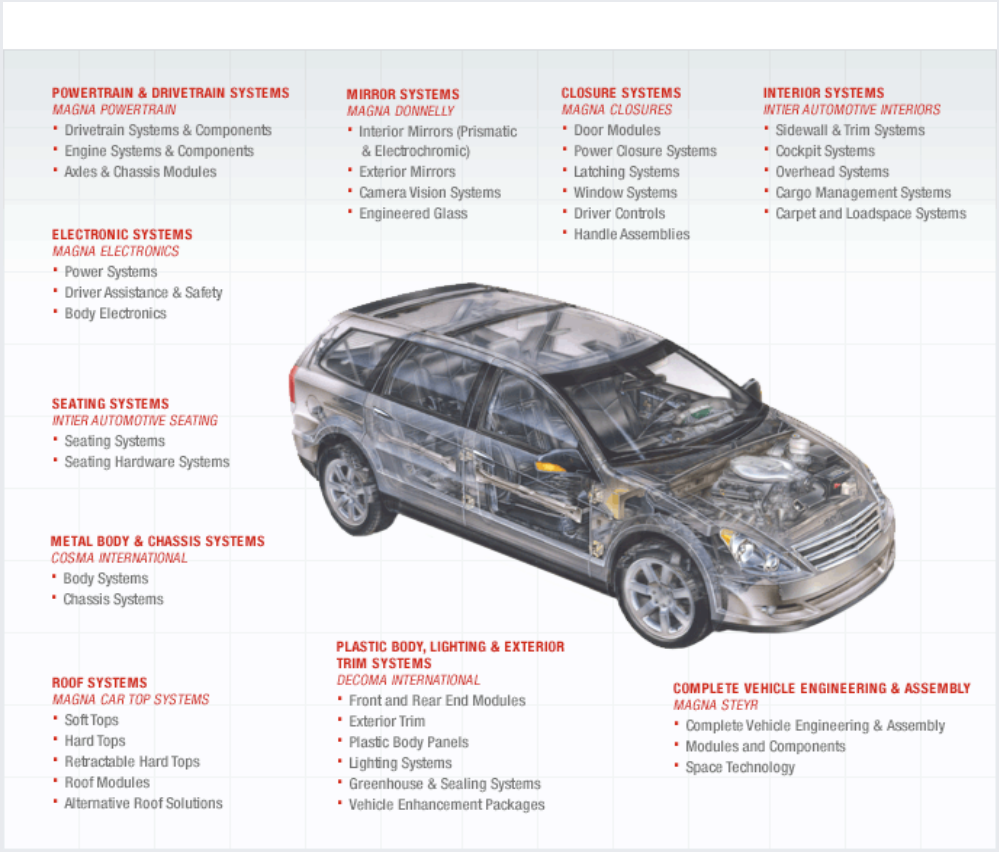
PTIDES Application: Superbowl



PTIDES Application: Superbowl



PTIDES Application: Tunneling Ball

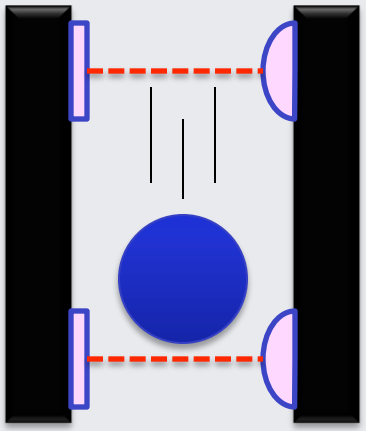


- Industry parallel: fuel injection
  - Piston position is sporadic
  - Fuel must quickly be injected in response to piston arrival

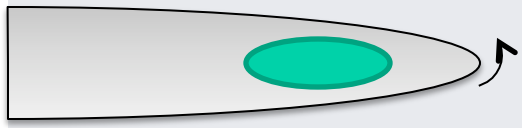
PTIDES Application: Tunneling Ball

- Industry parallel: automated production line
  - Object arrival is sporadic
  - Efficiency demands fast-response
  - Safety demands real-time response

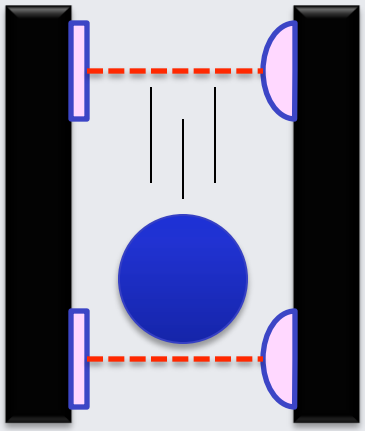
PTIDES Application: Tunneling Ball



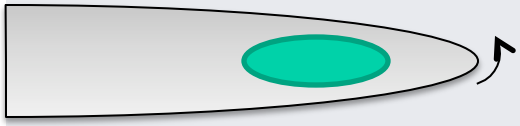
- Ball drops towards a spinning disc
  - Ball drops are random, sporadic
- Control to ensure ball tunnels through a door on one end of the disc
- Photodetector senses presence of ball
  - Timestamps presence at fixed altitude above disc
  - Determines initial velocity of ball
- Controller only slightly adjusts disc rate.
  - Disc never stops
  - Bounded acceleration



PTIDES Application: Tunneling Ball

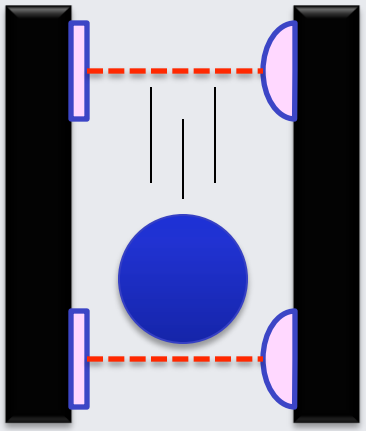


- Sporadic arrival times
  - Randomly spaced events with lower-bound on time gap between events
  - Application unaware of when a ball will arrive
  - Requires real-time response from the controller
- Concurrency
  - Motor decoding
  - Disc alignment
  - Ball drop detection
  - Trajectory planning
  - Disc control

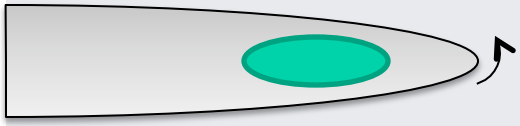


- Modeling necessary

PTIDES Application: Tunneling Ball

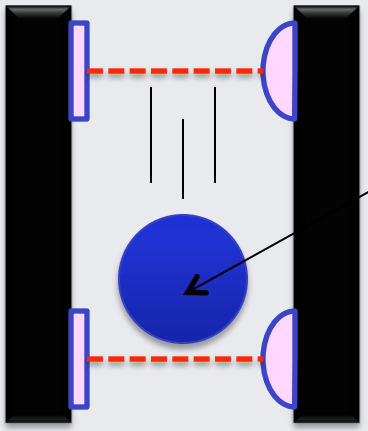


- Derivation of physical dynamics
  - Kinematics of ball in freefall
  - Kinematics of disc, tunnel
- Controller synthesis
  - Modeling of motor + inertial disc
  - Error tolerance
- Model-based design
  - PTIDES application
  - Simulations



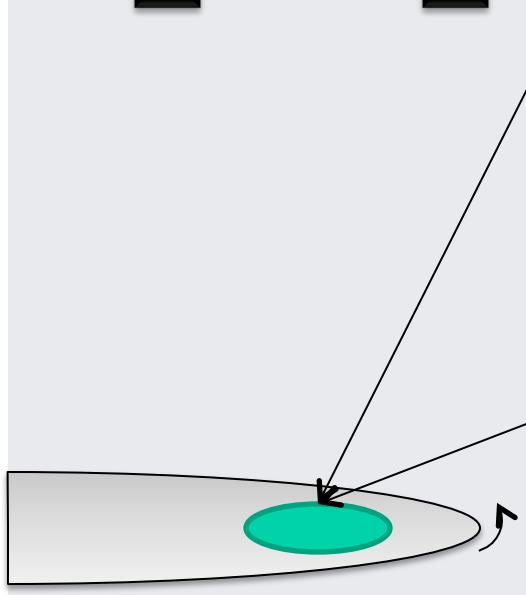
- Orthogonalization of concerns
  - Position control
  - Trajectory planning

PTIDES Application: Tunneling Ball



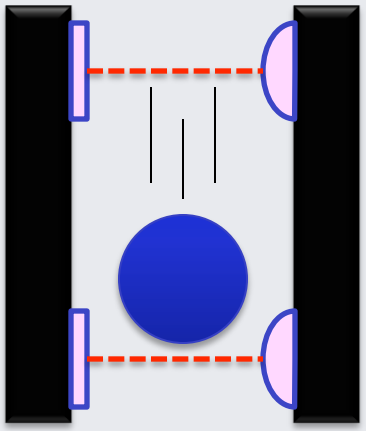
$$z(t) = z_0 - v_0 t - \tfrac{1}{2} g t^2$$

$$r_{\text{impact}}(t) = \begin{cases} \frac{1}{2} r_{\text{ball}} \left(1 + \cos\left(\frac{\pi z(t)}{r_{\text{ball}}} \right) \right) & \text{if } |z(t)| \le r_{\text{ball}} \\ 0 & \text{if } |z(t)| > r_{\text{ball}} \end{cases}$$



$$t_{arrive} = \frac{\sqrt{v_0^2 + 2g(z_0 - r_{ball})} - v_0}{g}$$

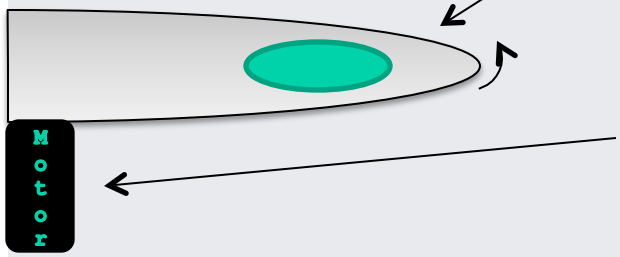
PTIDES Application: Tunneling Ball



$$K_A v(t) = Ri(t) + L\frac{di(t)}{dt} + K_B\frac{d\theta(t)}{dt}$$

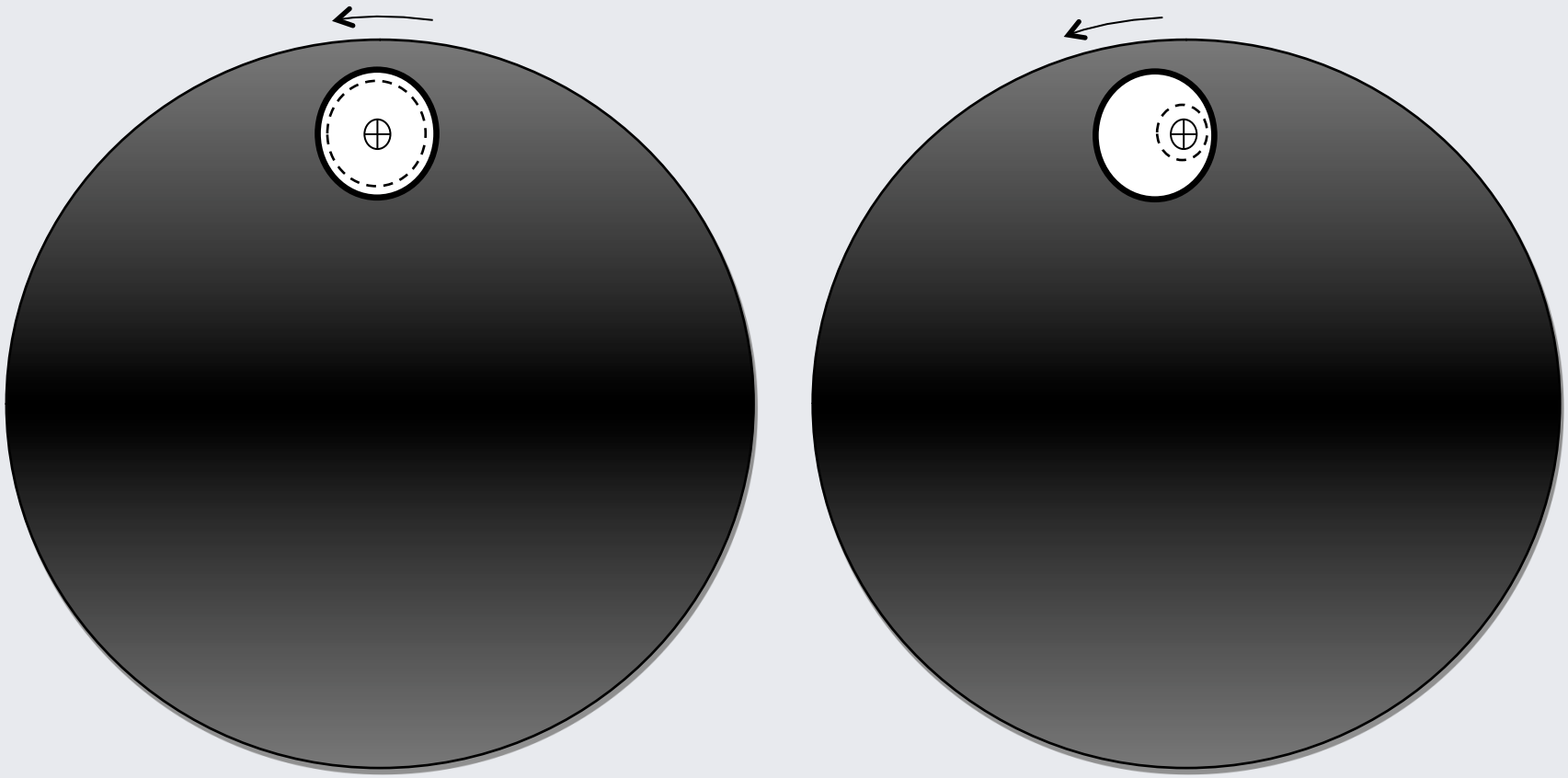
$$K_\tau i(t) = b\frac{d\theta(t)}{dt} + (J_A + J_D)\frac{d^2\theta(t)}{dt^2}$$

$$\theta(t) = \theta_0 + \omega_0 t$$



$$\frac{\Theta(s)}{V(s)} = \frac{K_A K_\tau}{JLs^3 + [R(J_A + J_L) + Lb]s^2 + (Rb + K_B K_\tau)s}$$

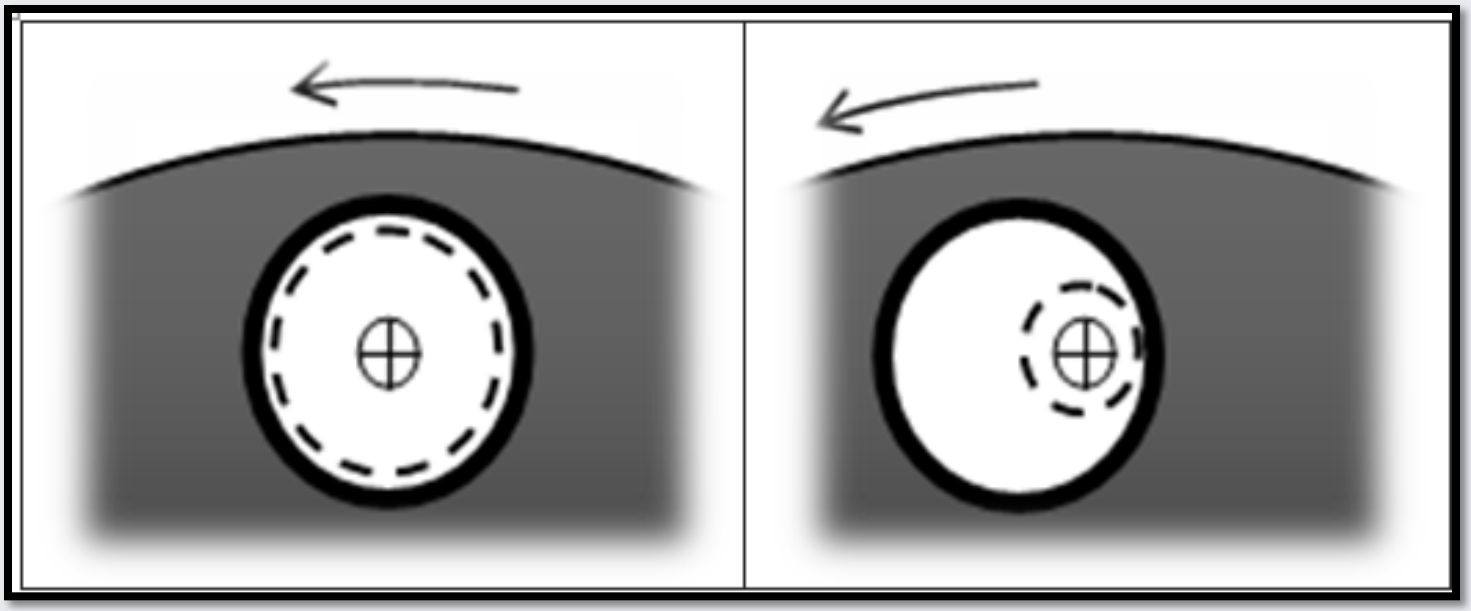
PTIDES Application: Tunneling Ball



The disc on the left is rotated so that a door is centered over the fixed drop target; the tunnel (indicated by the broken line) fills the entire door.

On the right, the disc has rotated, shrinking the tunnel.

PTIDES Application: Tunneling Ball

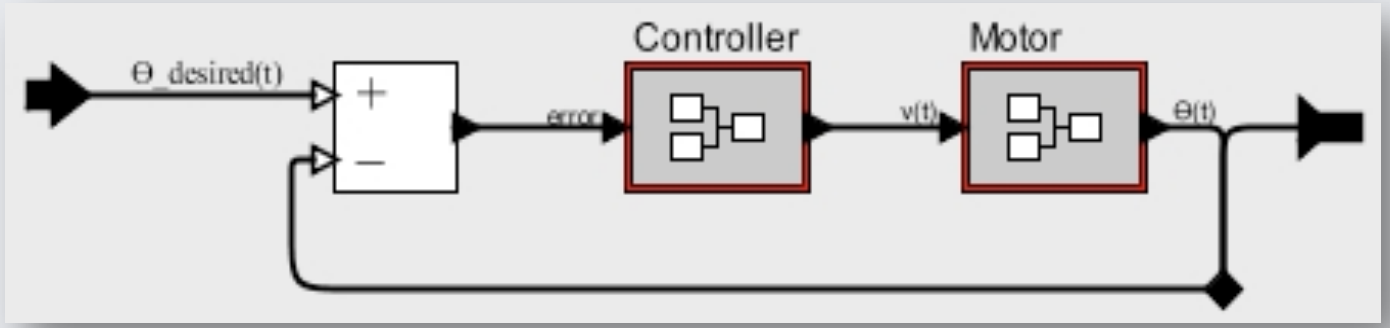


$$d(t) = \sqrt{2} r_{\text{drop}} \sqrt{1 - \cos(\theta(t))}$$

$$r_{\text{tunnel}}(t) = \begin{cases} r_{\text{door}} - d(t) & \text{if } d(t) \le r_{\text{door}} \\ 0 & \text{if } d(t) > r_{\text{door}} \end{cases}$$

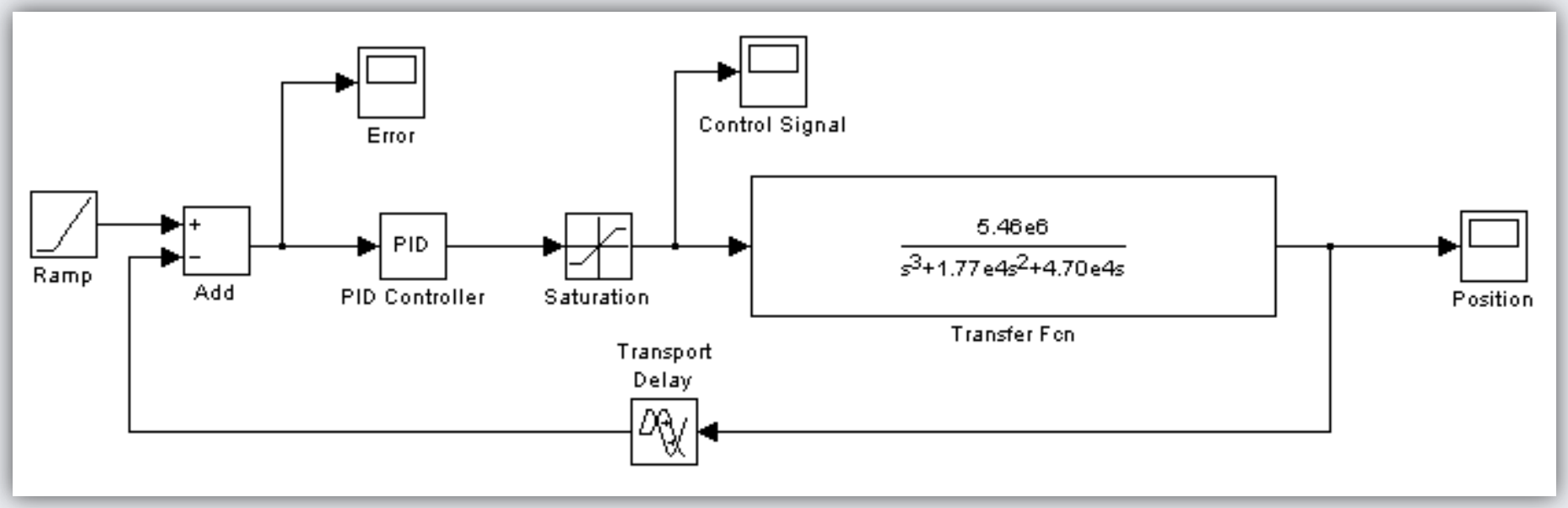
PTIDES Application: Tunneling Ball

- Modeling position control in Ptolemy II
  - PTIDES director
  - Custom discrete PID control actor
  - Motor continuous issues



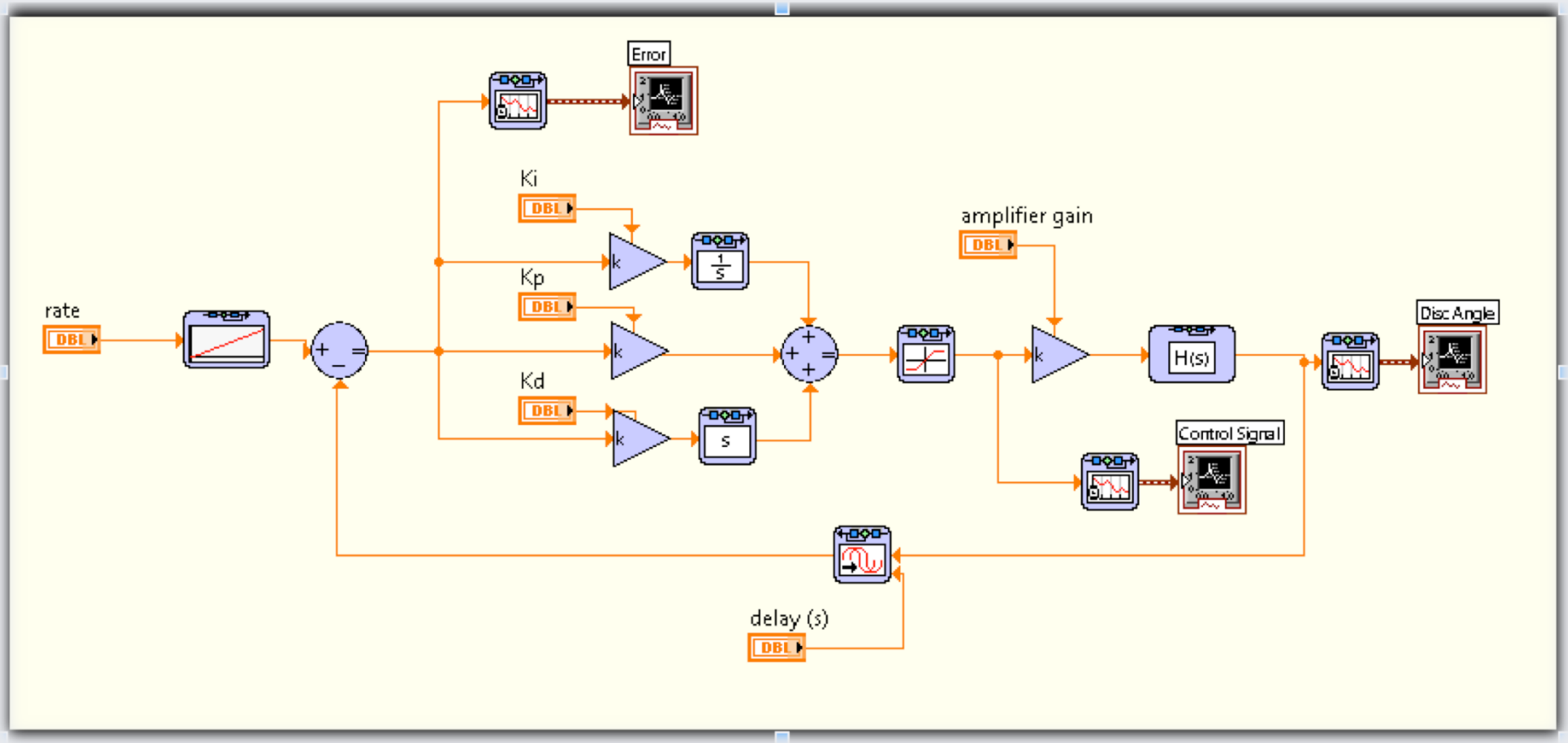
PTIDES Application: Tunneling Ball

- Modeling position control in MATLAB
  - Simulink
  - SISO toolkit
  - PID design
  - Optimization toolbox

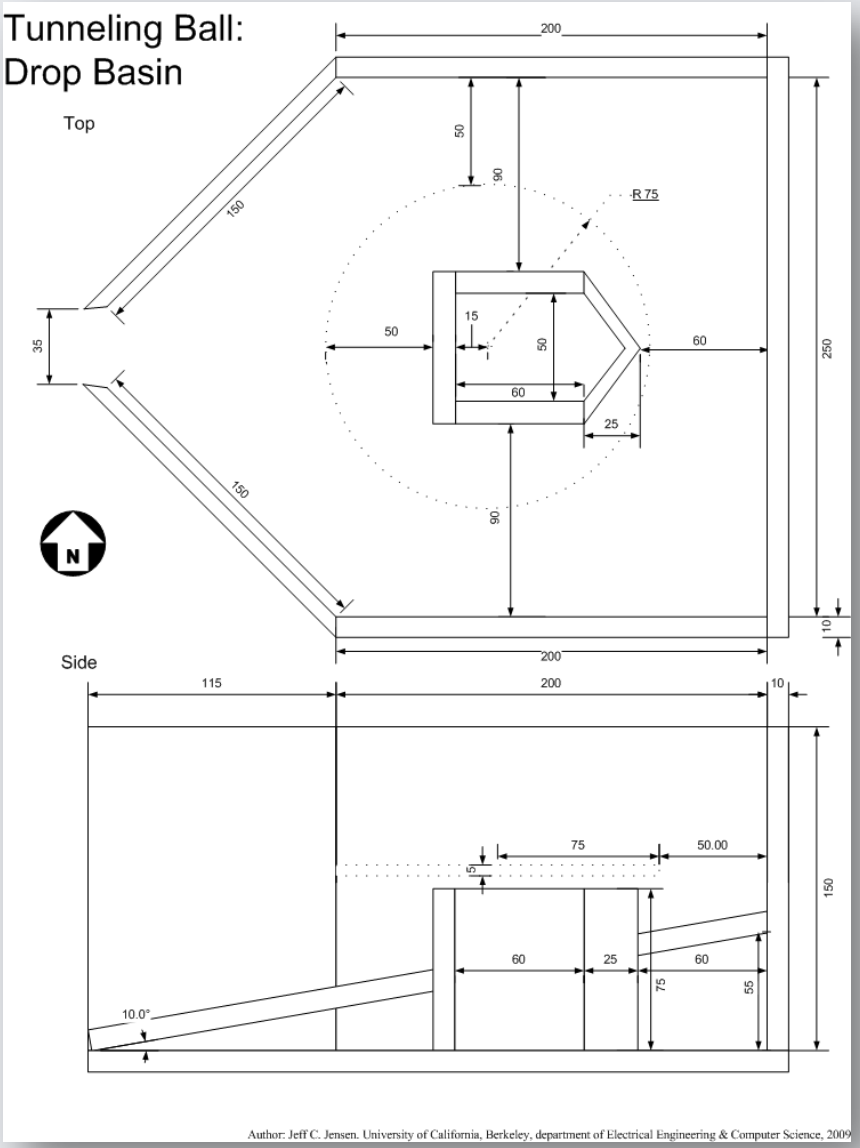


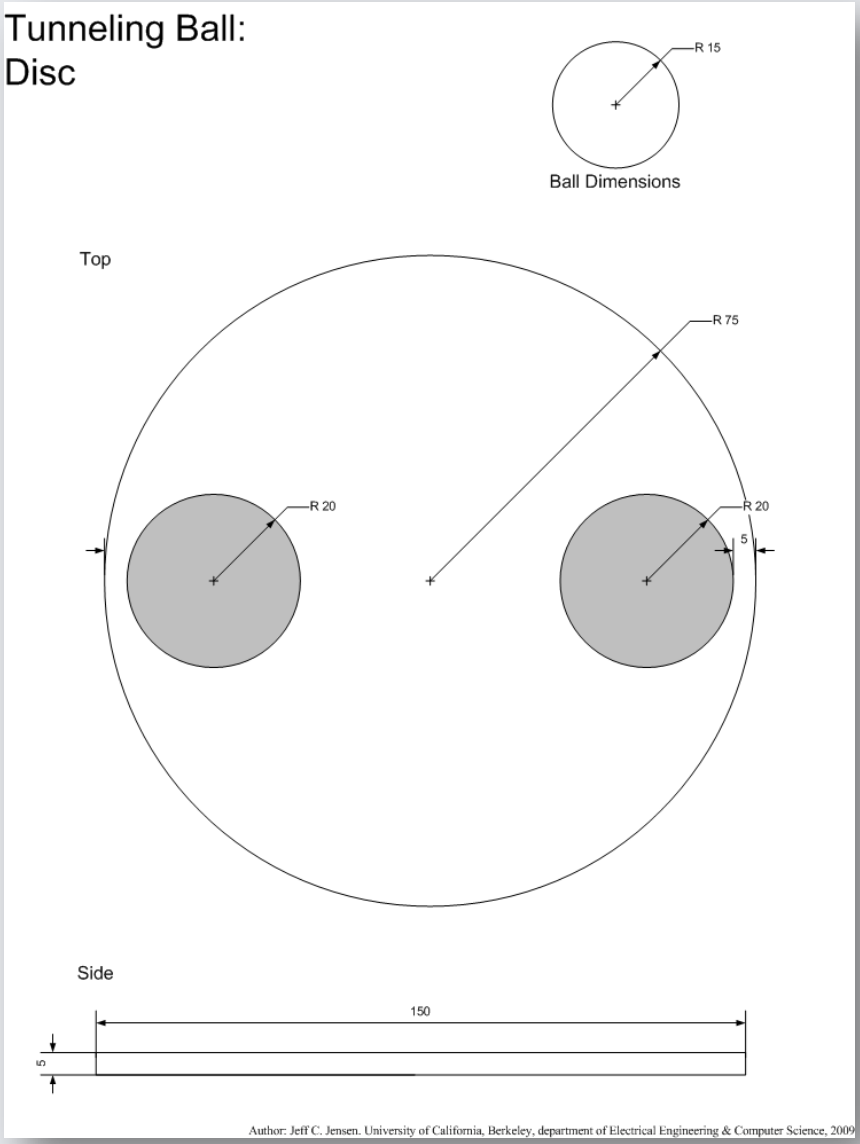
PTIDES Application: Tunneling Ball

- Modeling position control in LabVIEW
  - Control and Simulation Loop; linear SISO system representation

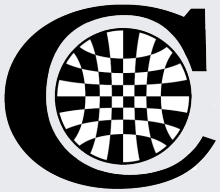


PTIDES Application: Tunneling Ball



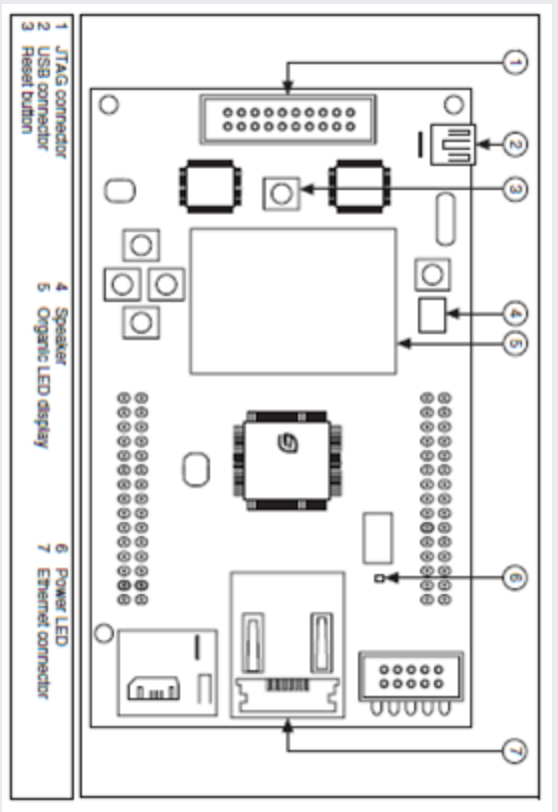


PTIDES Application: Tunneling Ball

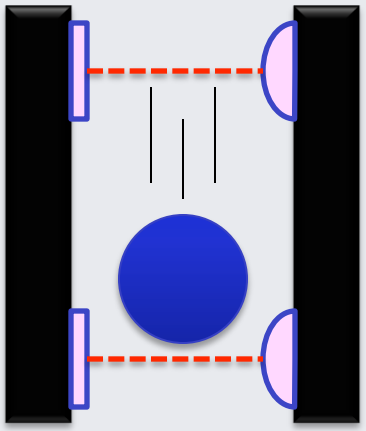


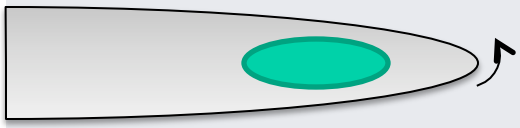
TI / Luminary Micro LM3s8962 Embedded Microcontroller

- ARM Cortex-M3
  - 50 MHz RISC
- 42 GPIO
  - 4 ADC @ 1 MHz
- IEEE 1588



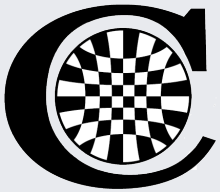
PTIDES Application: Tunneling Ball





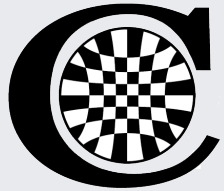
- HIL
  - Encoder
  - Plant model
- LabVIEW Co-Simulation
- Distributed Extensions
  - IEEE 1588
  - Global notion of time
  - Models many distributed networks

PtidyOS

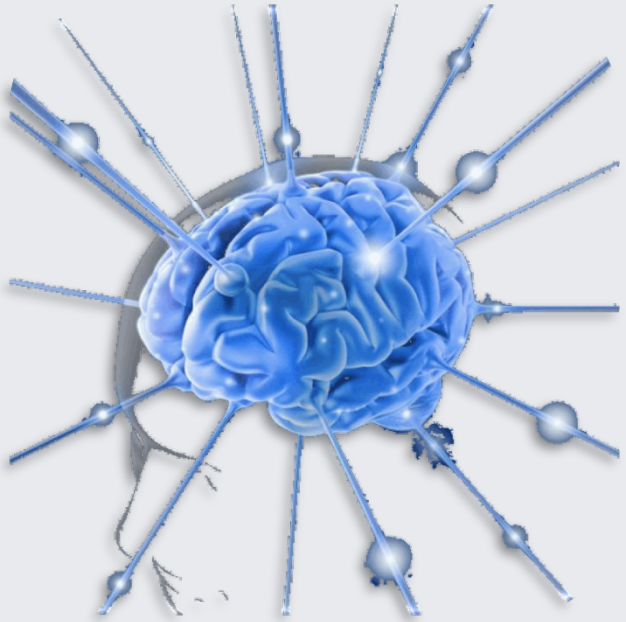


- Lightweight embedded OS
  - Small footprint
  - No kernel
- Implements PTIDES
  - Code generation (C)
  - Static analysis
- Context switching
  - Interrupt
  - SafeToProcess( )

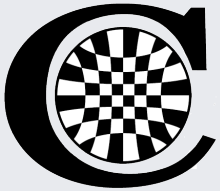
NI and Berkeley – Joint Research



- Curriculum
  - Controls, embedded, mechanical
- Models of computation
  - PTIDES
  - NextGEN
- Real-time operating systems
  - PtidyOS
- PRET



Berkeley Research – Ptolemy II

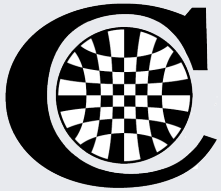


- Modeling, simulation, and design
  - Real-time distributed embedded systems
- Models of computation
  - Heterogeneous mixtures
- Code generation
  - C and Java
- Director: Professor Edward A. Lee

Berkeley Research – PRET

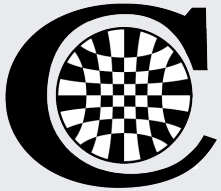
- **Pre**cision **T**imed Machines
  - Average vs. worst case
- Cycle-accurate simulator
- FPGA prototype
- Contributor: Hugo Andrade, National Instruments
- Director: Professor Edward A. Lee

Berkeley Research – Execution Time



- Game-theoretic approach
- Software space discovery
- “Black-box” perspective
- Applications: controls, event-driven models
- Director: Professor Sanjit A. Seshia

NI and U.C. Berkeley – Moving Forward



- IEEE 1588
  - Hard real-time systems
  - 5-10ns precision
  - Immediate PTIDES applications

- Continued R&D
  - Advancements in code generation
  - Concurrent models of computation
    - Distributed,
    - Deterministic
    - Timing from architecture to abstraction
  - Static analysis

Acknowledgments

- Edward A. Lee (U.C. Berkeley)
  - Robert S. Pepper Distinguished Professor
  - Chair, EECS (2007 - 2008)
- Sanjit A. Seshia (U.C. Berkeley)
  - Assistant Professor, EECS
- Hugo Andrade (NI)
  - R&D Site Manager (NI Berkeley)
- Jeannie Falcon (NI)
  - Principal Engineer, Control & Simulation

Edward A. Lee

Sanjit A. Seshia

Model-Based Design:

when you just can't hack it.

University of California, Berkeley
Department of Electrical Engineering & Computer Science

Edited and presented by

Jeff C. Jensen