Composition in Heterogeneous Systems

Janos Sztipanovits
ISIS, Vanderbilt University

12 October, 2009

Outline

- Effects of heterogeneity
- Passivity-based design
- Toward a high-confidence model-based design tool chain

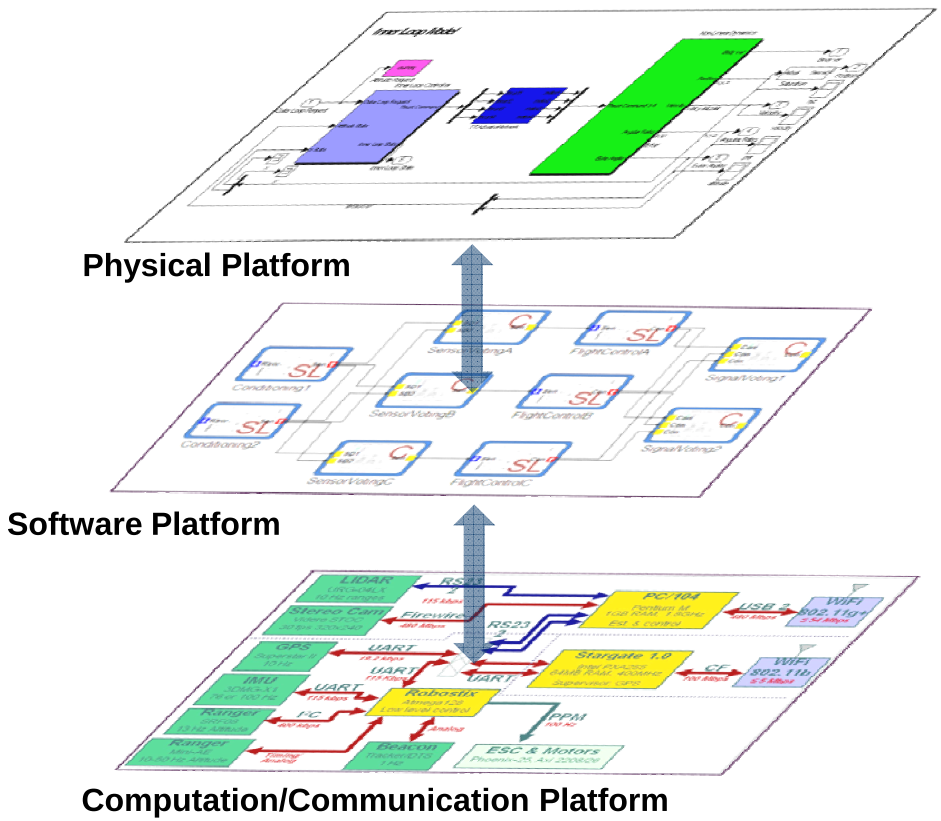
Model-Based Design & Platforms

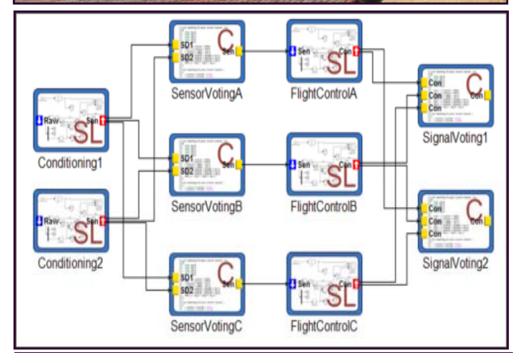
Key Idea: Manage design complexity by creating abstraction layers in the design flow.

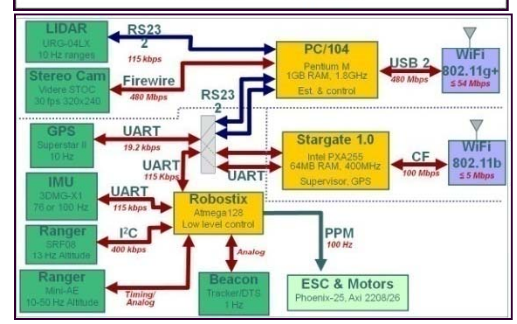
Abstraction layers define platforms.

Abstractions are linked through mapping.

Abstraction layers allow the verification of different properties .





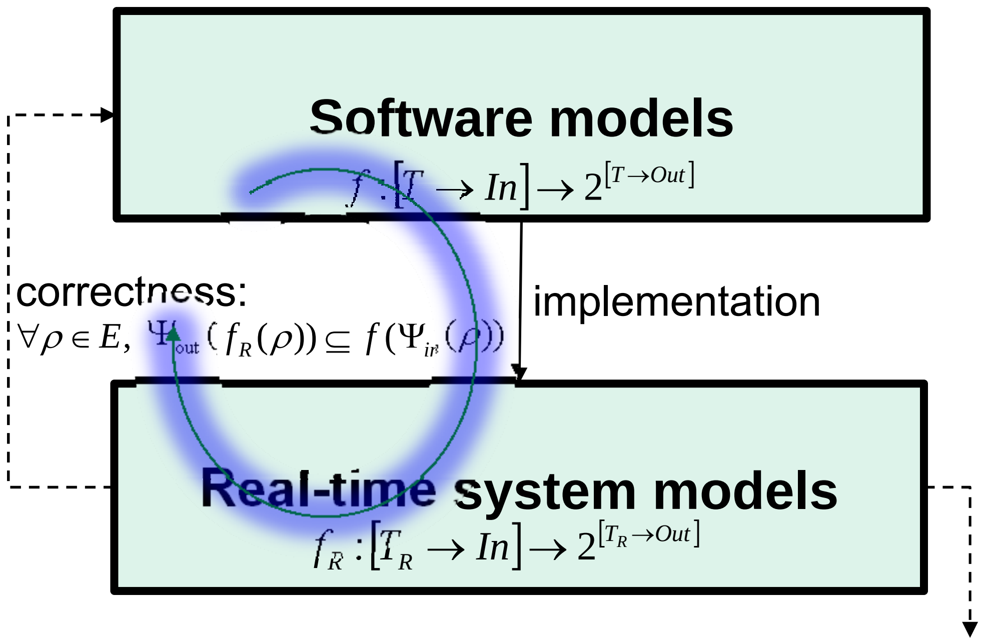


Claire Tomlin, UC Berkeley

Abstraction layers: SW-RTS

Sifakis at al: "Building Models of Real-Time Systems from Application Software," *Proceedings of the IEEE* Vol. 91, No. 1. pp. 100-111, January 2003

In CPS, essential system properties such as stability, safety, performance are expressed in terms of physical behavior

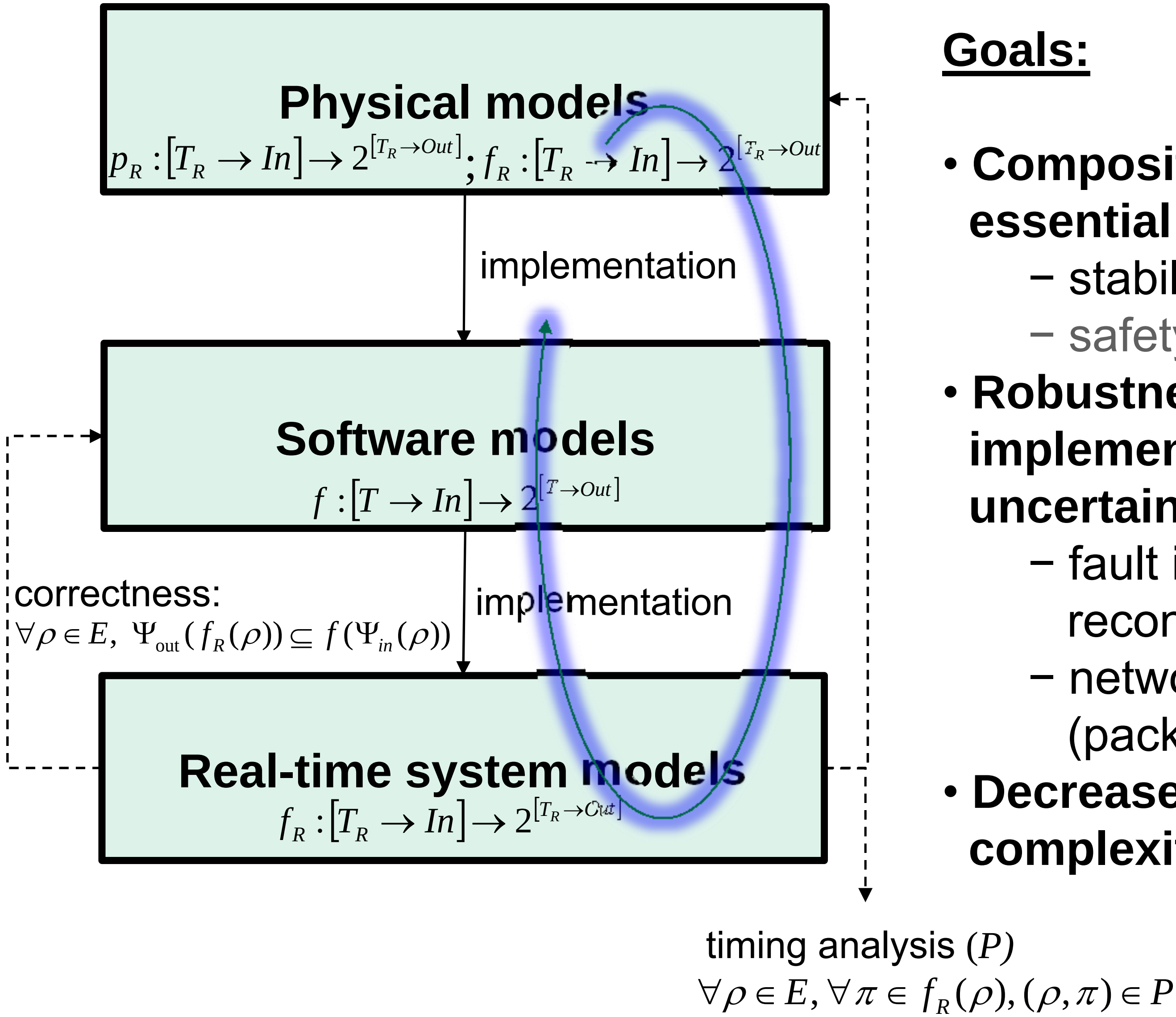


- f : reactive program. Program execution creates a mapping between logical-time inputs and outputs.
- f_R : real-time system. Programs are packaged into interacting components. Scheduler control access to computation and communication resources according to time constraints P

timing analysis (P)

$\forall \rho \in E, \forall \pi \in f_R(\rho), (\rho, \pi) \in P$

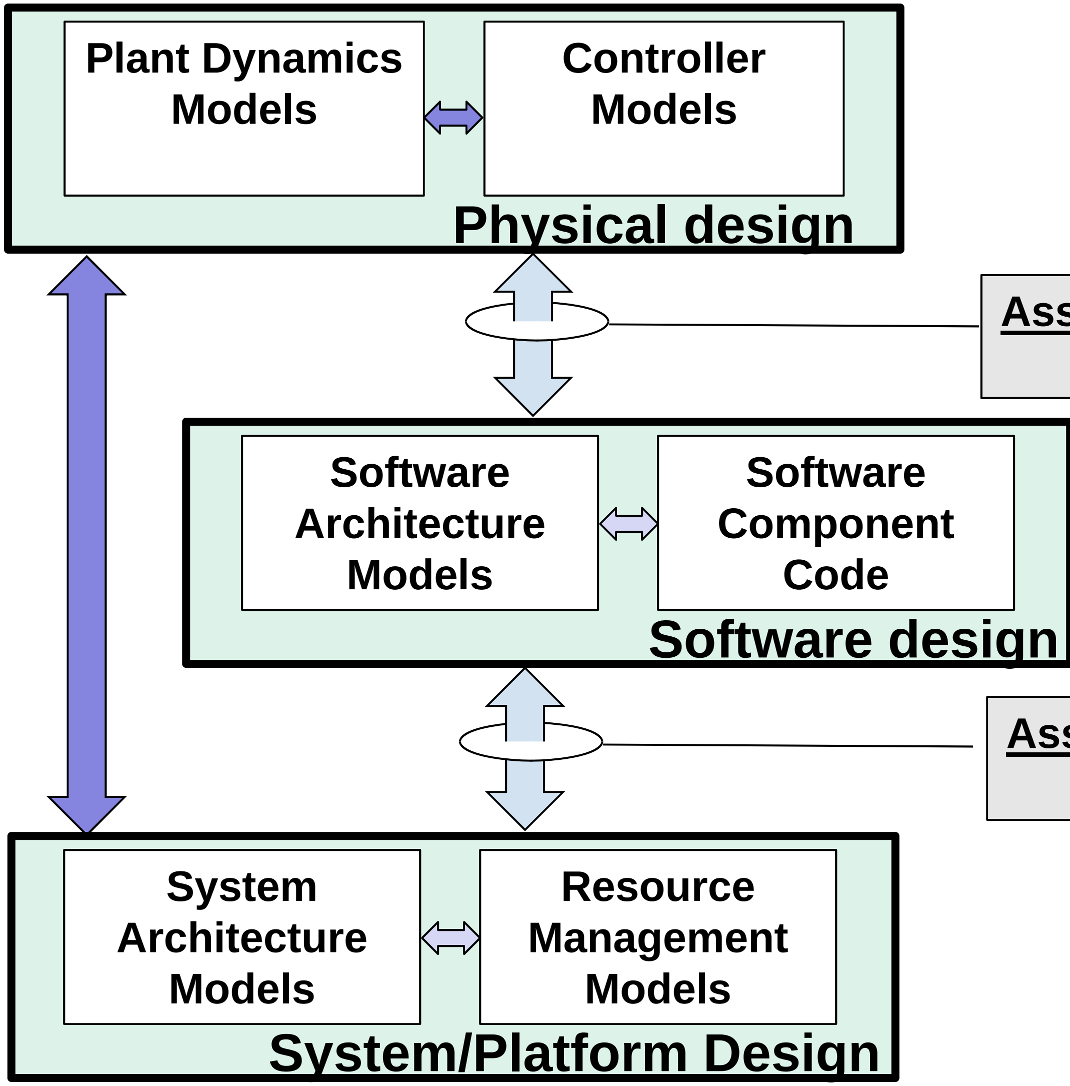
Abstraction layers: PHY-SW-RTS



Goals:

- **Compositional verification of essential dynamic properties**
  - stability
  - safety
- **Robustness against implementation changes and uncertainties**
  - fault induced reconfiguration of SW/HW
  - network uncertainties (packet drops, delays)
- **Decreased verification complexity**

Composition Inside Abstraction Layers



Dynamics: $B(t) = \kappa_p(B_1(t),\ldots,B_j(t))$

- *Properties*: stability, safety, performance
- *Abstractions*: continuous time, functions, signals, flows,…

Assumption: Effects of digital implementation can be neglected

Software : $B(i) = \kappa_c(B_1(i),\ldots,B_k(i))$

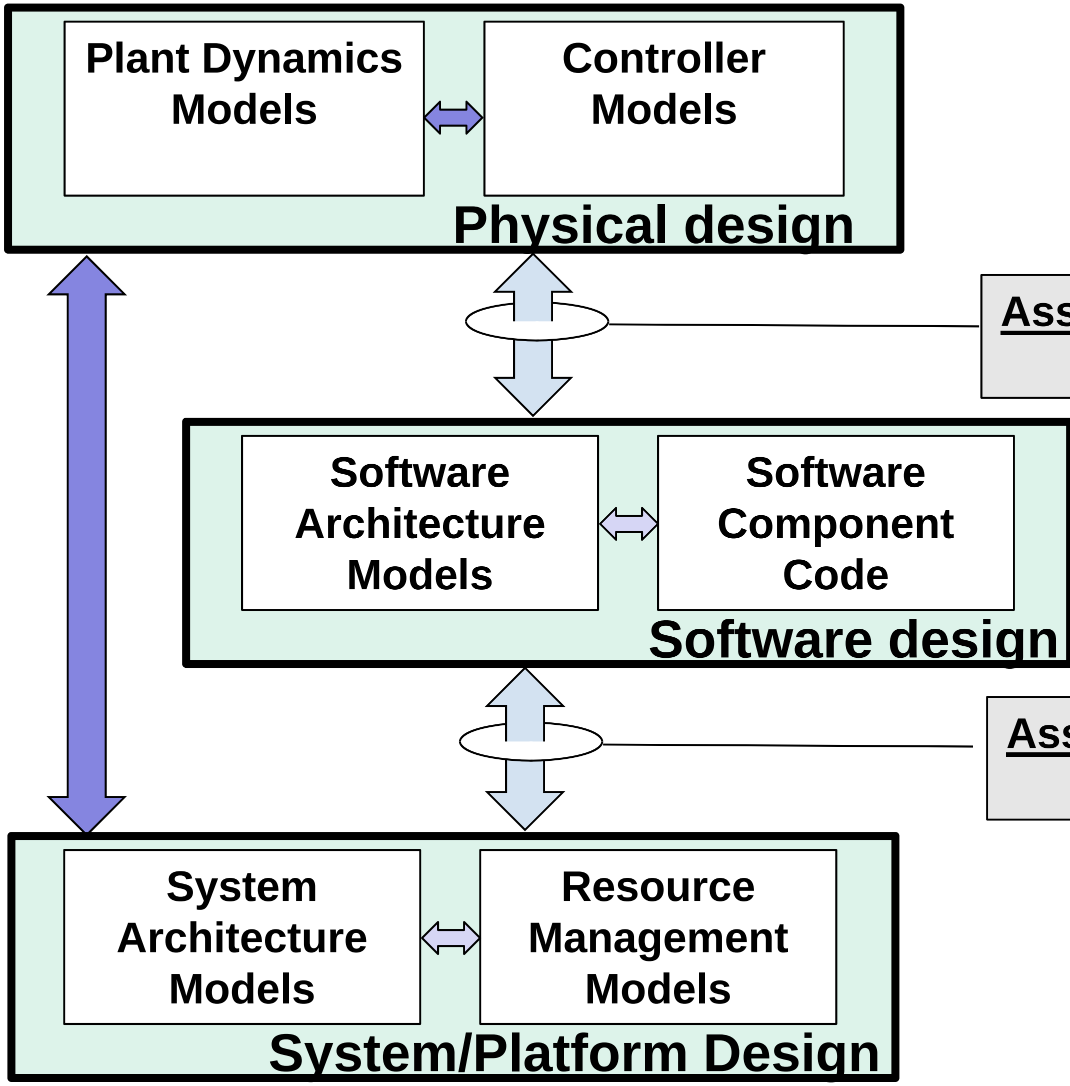
- *Properties*: deadlock, invariants, security,…
- *Abstractions*: logical-time, concurrency, atomicity, ideal communication,..

Assumption: Effects of platform properties can be neglected

Systems : $B(t_j) = \kappa_p(B_1(t_i),\ldots,B_k(t_i))$

- *Properties*: timing, power, security, fault tolerance
- *Abstractions*: discrete-time, delays, resources, scheduling,

Composition Inside Abstraction Layers



Controller dynamics is developed without considering implementation uncertainties (e.g. word length, clock accuracy) optimizing performance.

Assumption: Effects of digital implementation can be neglected

Software architecture models are developed without explicitly considering systems platform characteristics, even though key behavioral properties depend on it.

Assumption: Effects of platform properties can be neglected

System-level architecture defines implementation platform configuration. Scheduling, network uncertainties, etc. are introduce time variant delays that may require re-verification of key properties on all levels.

Composition and Heterogeneity

- Consequence of the lack of composability across system layers
  - intractable interactions
  - unpredictable system level behavior
  - full-system verification does not scale
- Approach: simplification strategies
  - ***Orthogonalization:*** **Use passivity for decoupling stability and implementation induced time variant delays**
  - **…**

Outline

- Effects of heterogeneity
- **Passivity-based design**
- Toward a high-confidence model-based design tool chain

Physical layer: Passivity-based design

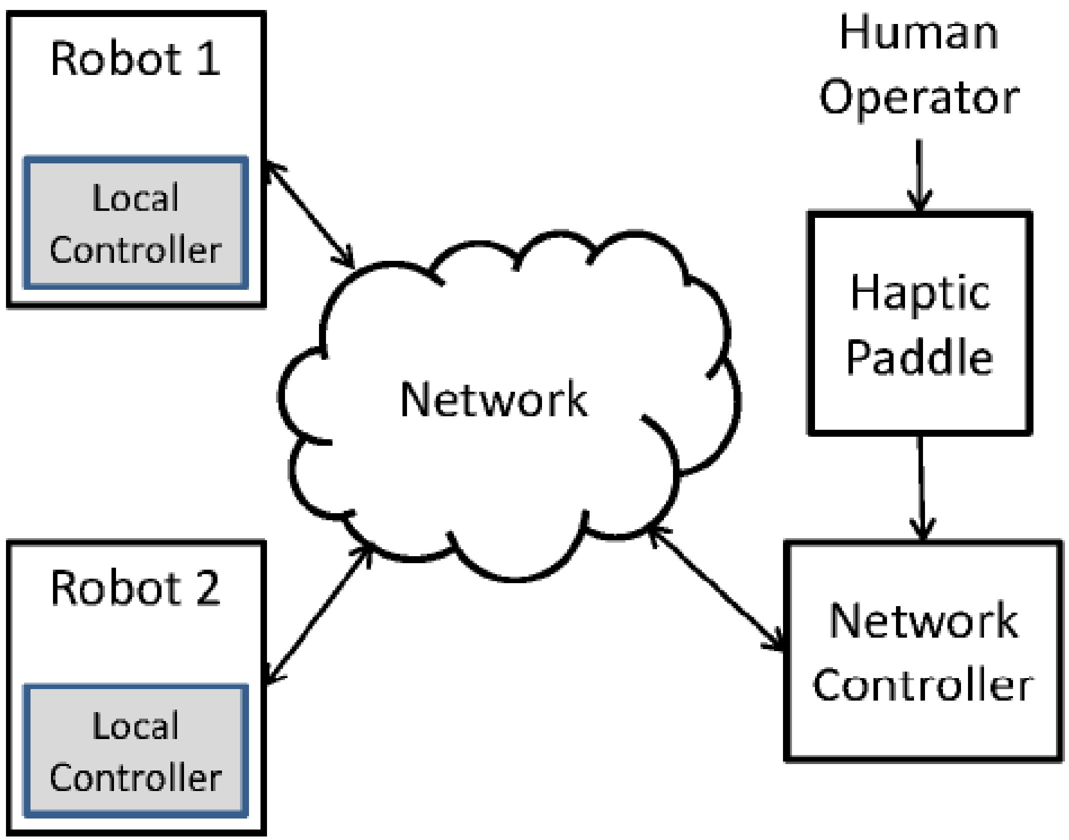
Key idea: Passivity-based design of networked control systems provides robustness to time-varying delays

- Various mathematical definitions
  - A passive system only stores and dissipates energy but cannot generate energy of its own
- Passive systems interact in a stable manner
  - When connected in either a parallel or negative feedback manner the overall system remains passive
- Passive control theory applies to
  - Linear and nonlinear systems
  - Continuous and discrete-time systems
- Easier and safer to control
  - Independent joint PD controller for robotic manipulator
  - Asymptotic stability for set-point tracking

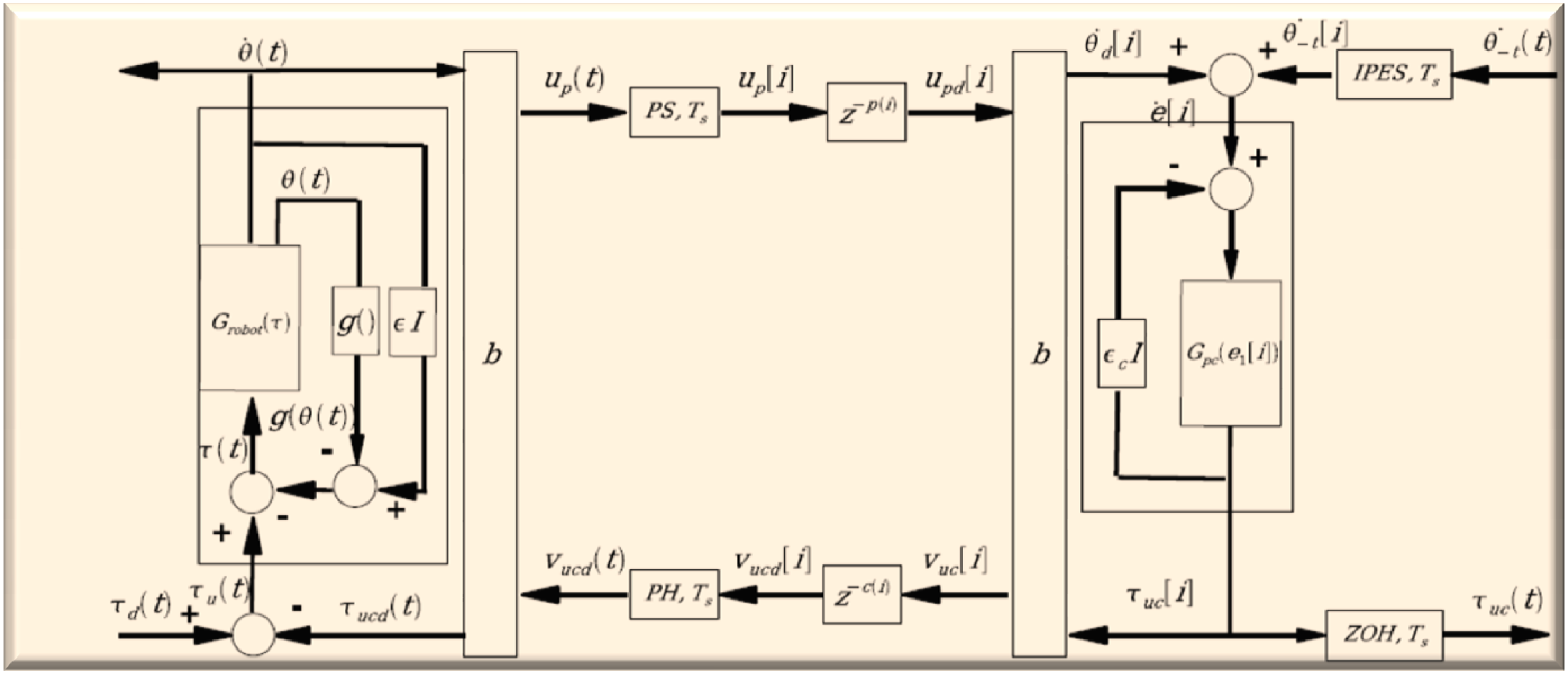
Background on Passivity

- Milestones:
  - Wave digital filters (Fettweis, 70's)
  - Passive structures (Peceli, 80's)
  - Teleoperation over the Internet (Niemmeyer, 04)
  - Power junctions (Kottenstette, Antsaklis, 08)
- Work at ISIS:
  - Design tool suite and extension through applications (Eyisi, Hall, Porter, Kottenstette, Koutsoukos, Sztipanovits)

Networked Control System



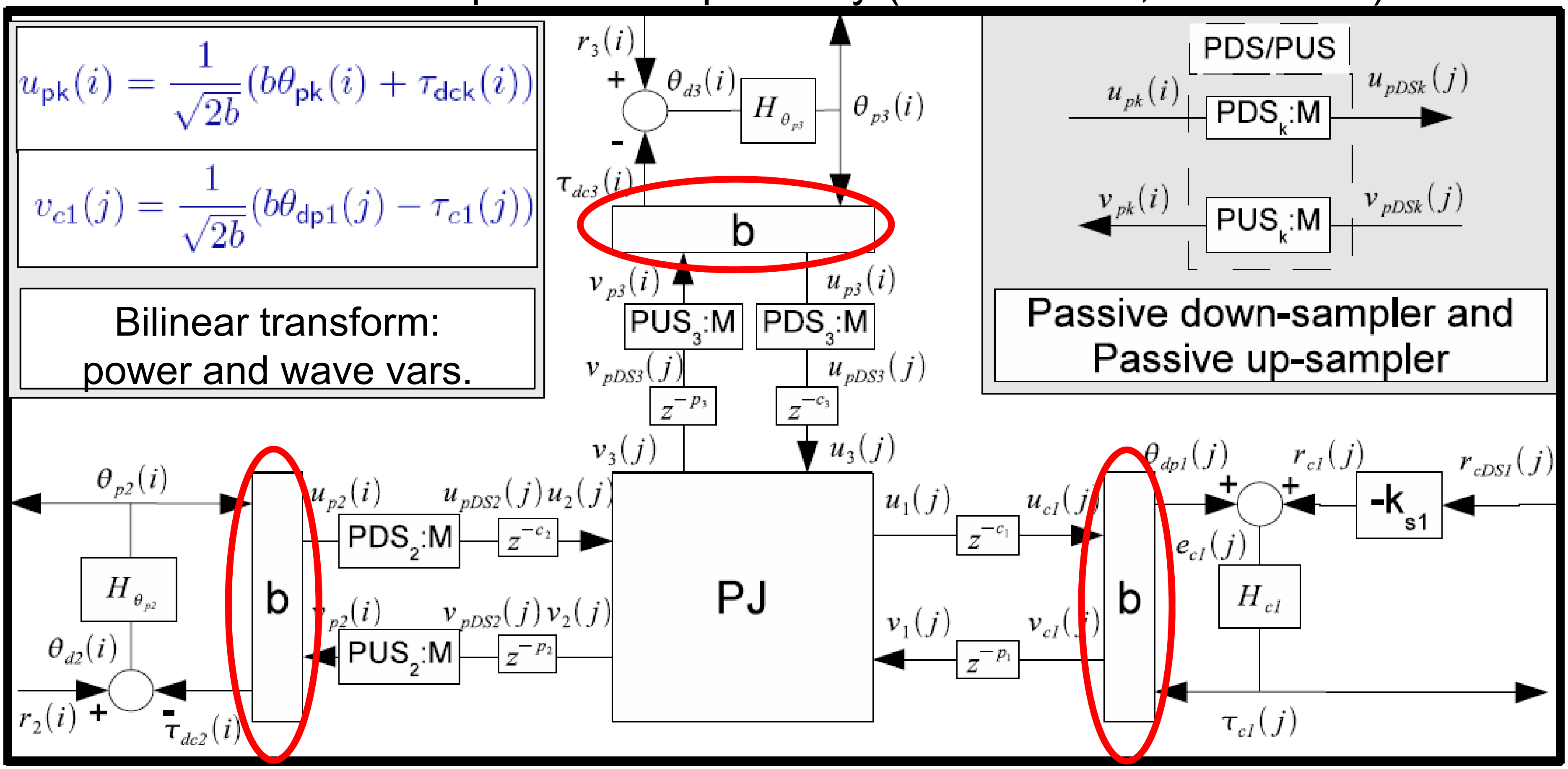
Basic concepts



- Wave variables were introduced by Fettweis in order to circumvent the problem of delay-free loops and guarantee that the implementation of wave digital filters is realizable
- Wave variables defined by a bilinear transformation under which a stable minimum phase continuous-time system is mapped to a stable minimum phase discrete-time system. The transformation preserves passivity.

Passivity-Based NCS Architecture

Modularization – composition for passivity (Kottenstette, Kotsoukos)

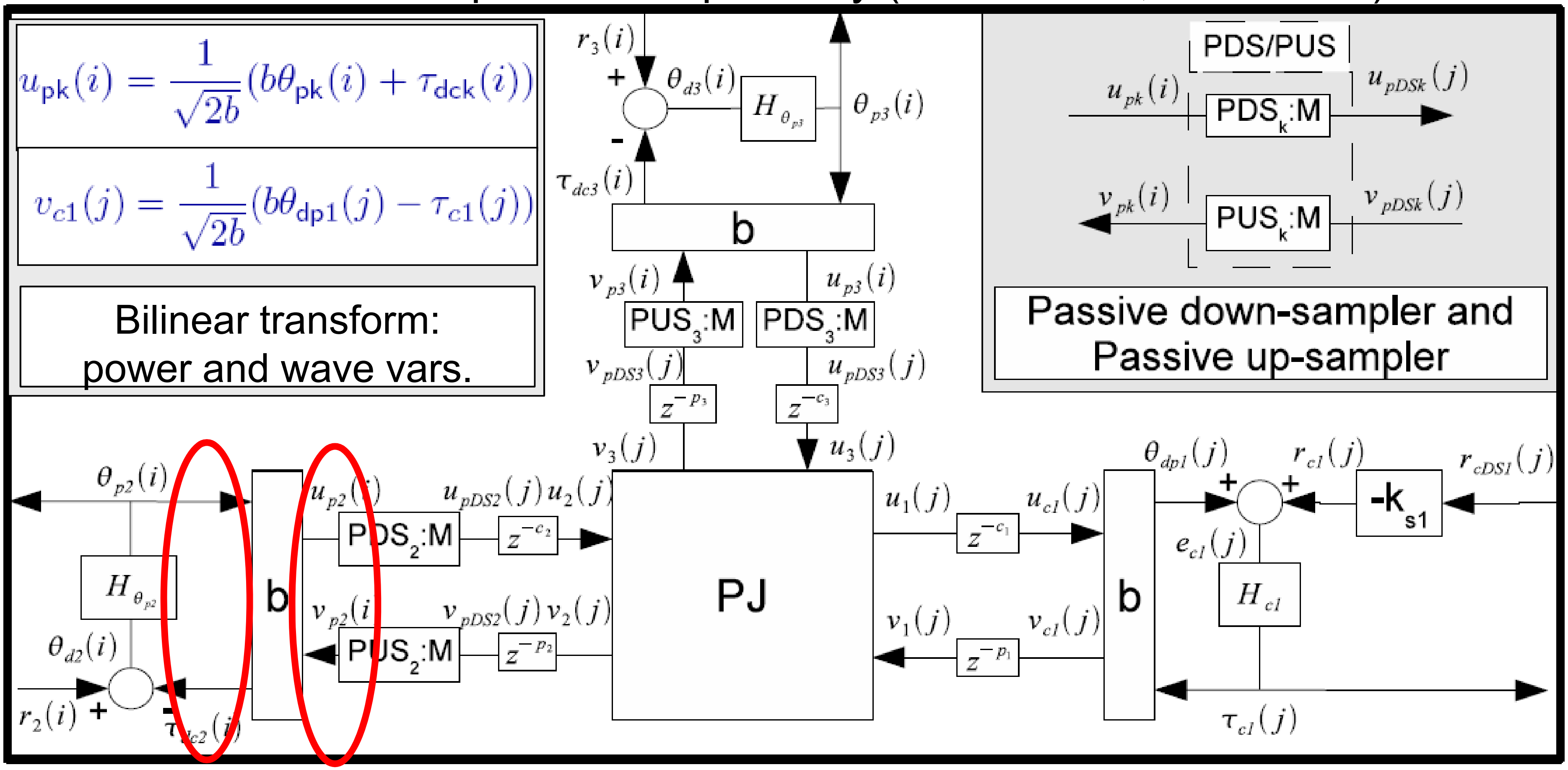


- **Bilinear transform (b)**
- Power and Wave variables
- Passive down- and up-sampler (PUS, PDS)

- Delays
- Power junction
- Passive dynamical system

Passivity-Based NCS Architecture

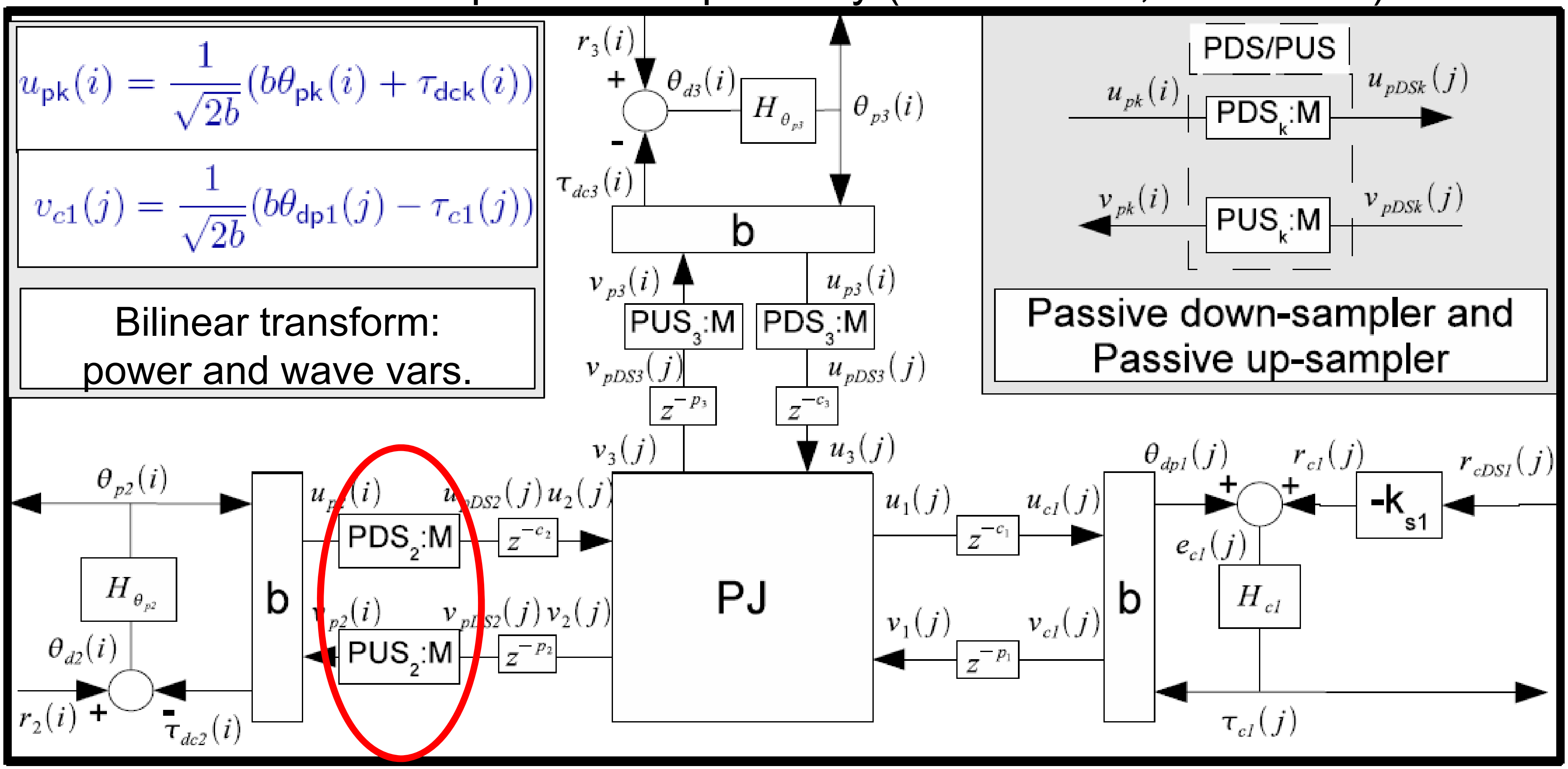
Modularization – composition for passivity (Kottenstette, Kotsoukos)



- Bilinear transform (b)
- **Power and Wave variables**
- Passive down- and up-sampler (PUS, PDS)
- Delays
- Power junction
- Passive dynamical system

Passivity-Based NCS Architecture

Modularization – composition for passivity (Kottenstette, Kotsoukos)

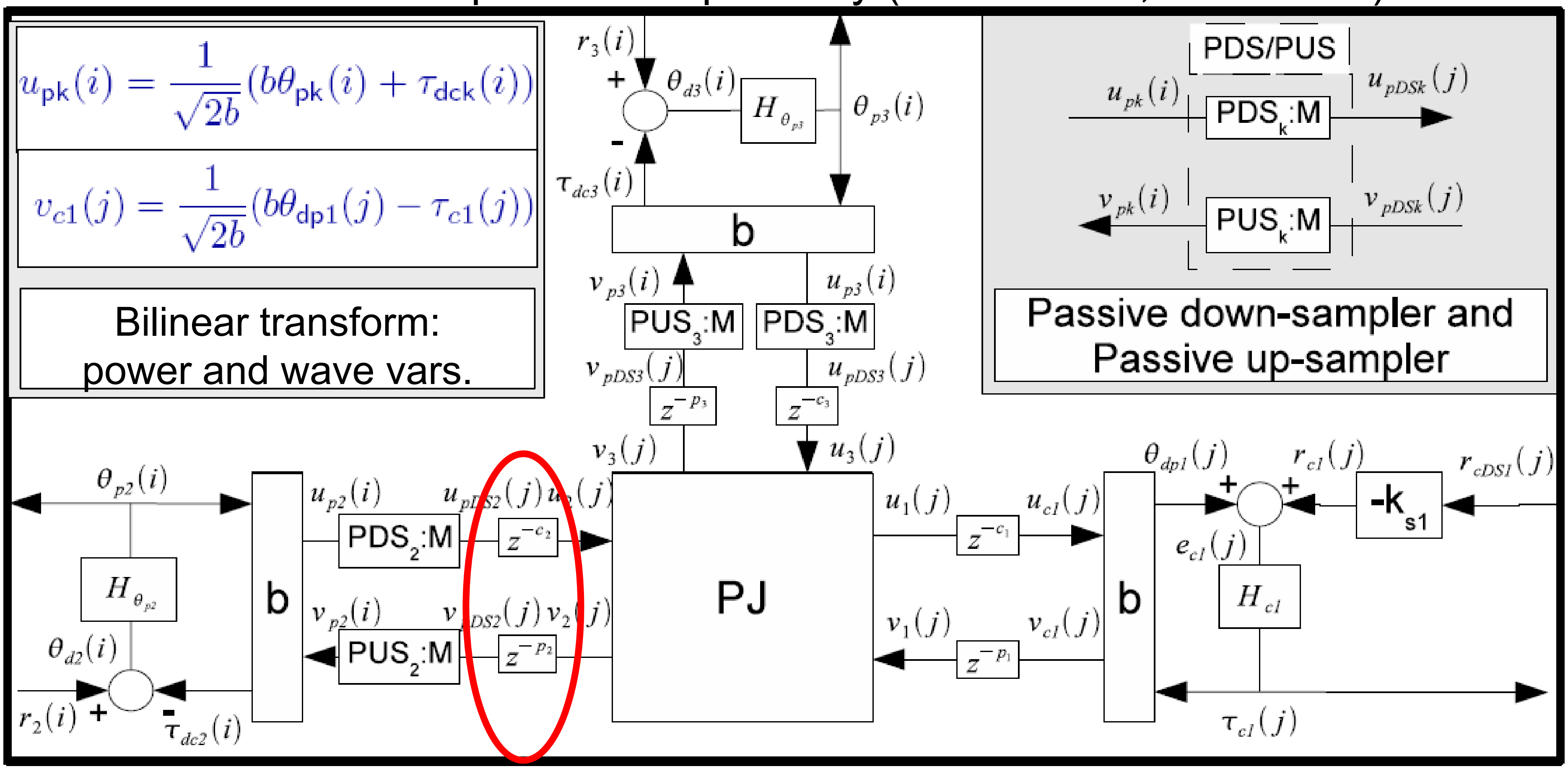


- Bilinear transform (b)
- Power and Wave variables
- **Passive down- and up-sampler (PUS, PDS)**

- Delays
- Power junction
- Passive dynamical system

Passivity-Based NCS Architecture

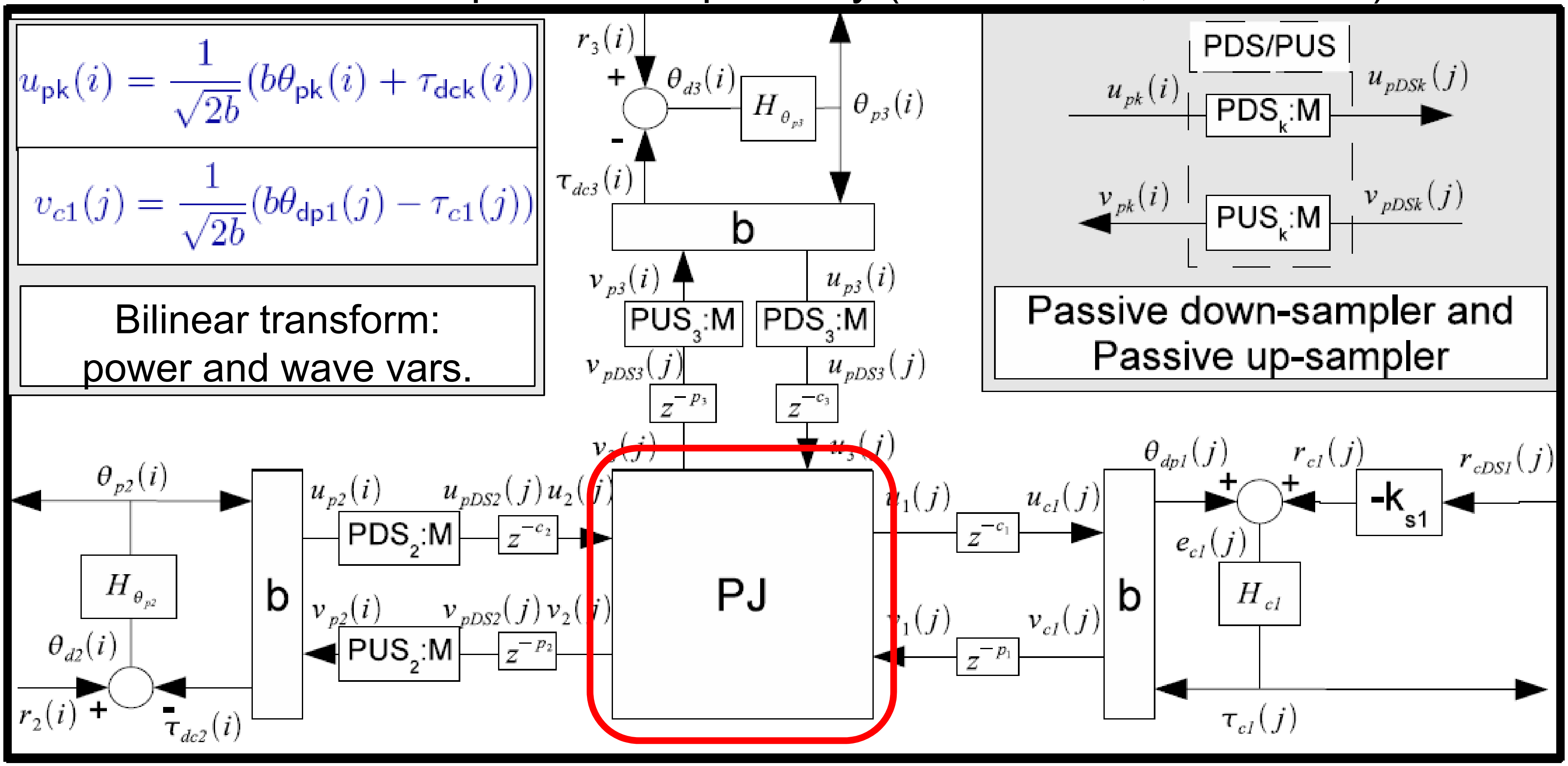
Modularization – composition for passivity (Kottenstette, Kotsoukos)



- Bilinear transform (b)
- Power and Wave variables
- Passive down- and up-sampler (PUS, PDS)
- **Delays**
- Power junction
- Passive dynamical system

Passivity-Based NCS Architecture

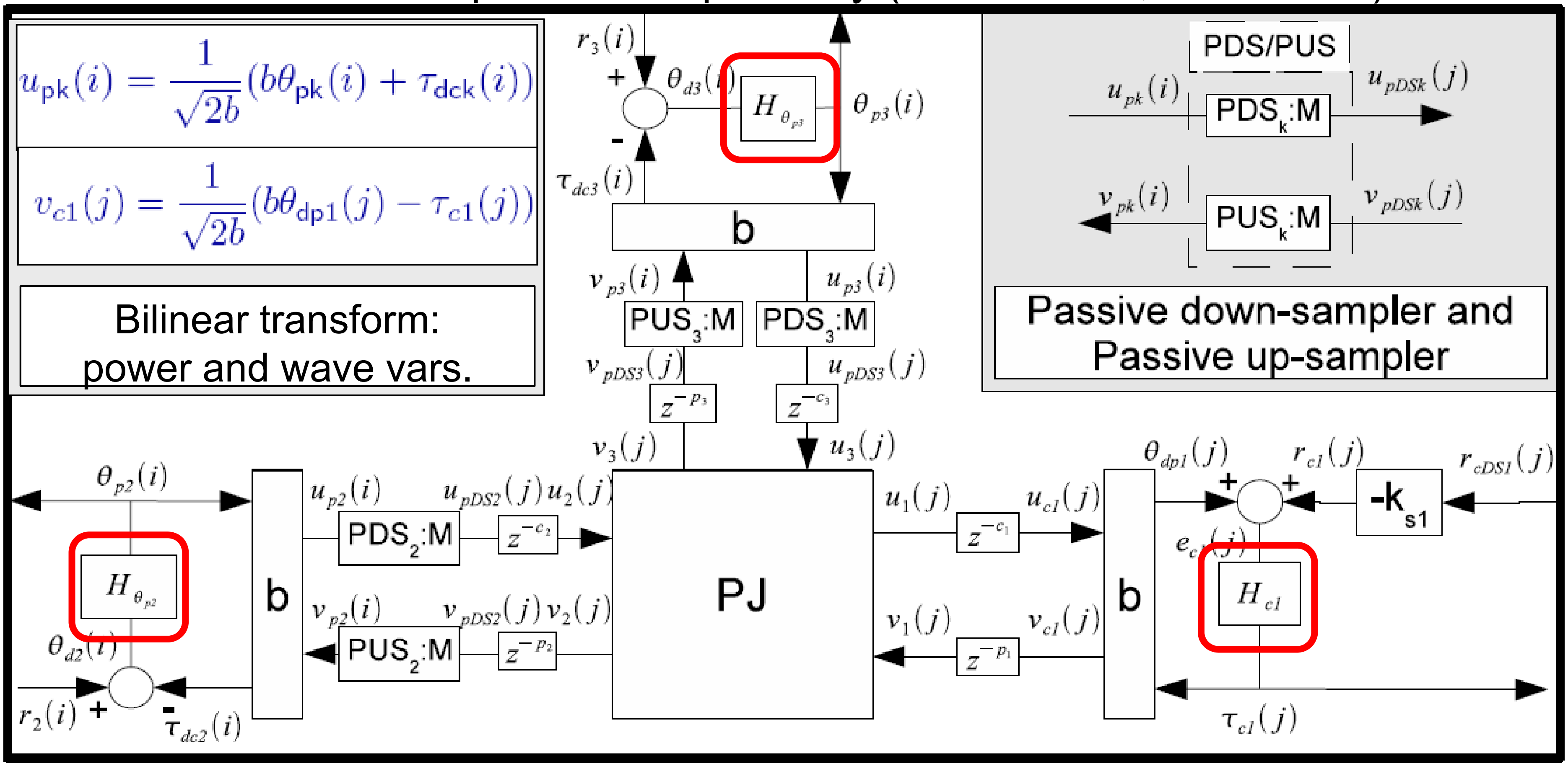
Modularization – composition for passivity (Kottenstette, Kotsoukos)



- Bilinear transform (b)
- Power and Wave variables
- Passive down- and up-sampler (PUS, PDS)
- Delays
- **Power junction**
- Passive dynamical system

Passivity-Based NCS Architecture

Modularization – composition for passivity (Kottenstette, Kotsoukos)



- Bilinear transform (b)
- Power and Wave variables
- Passive down- and up-sampler (PUS, PDS)
- Delays
- Power junction
- **Passive dynamical system**

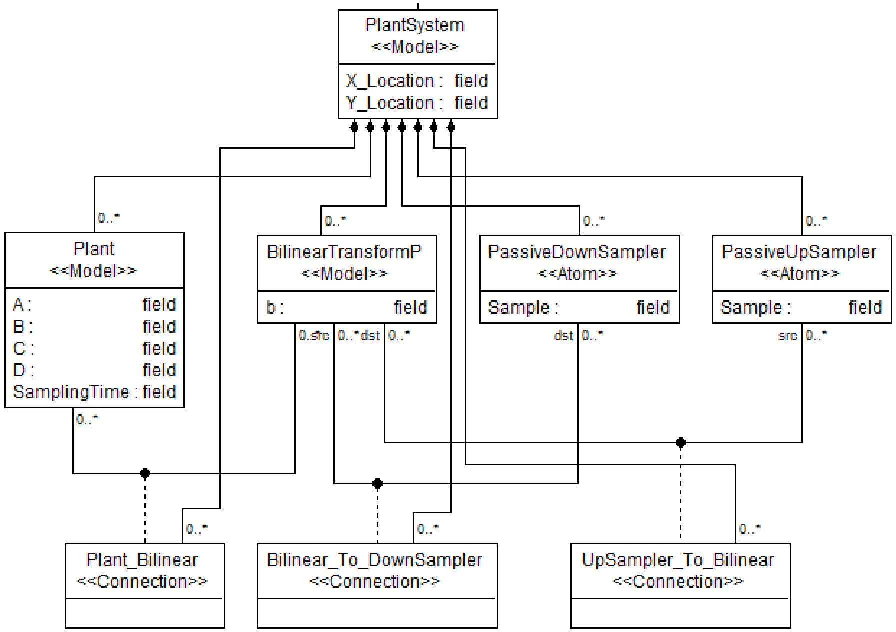
DSML for passivity-based design: PaNeCS

- Developed by Emeka Eyisi using the Model Integrated Computing (MIC) tools, (GME, UDM).
- PaNeCS Meta-model
  - **Main Components**
    - Plant Subsystem
    - Controller Subsystem
    - PowerJunction Subsystem
    - Network Subsystem

Plant Subsystem

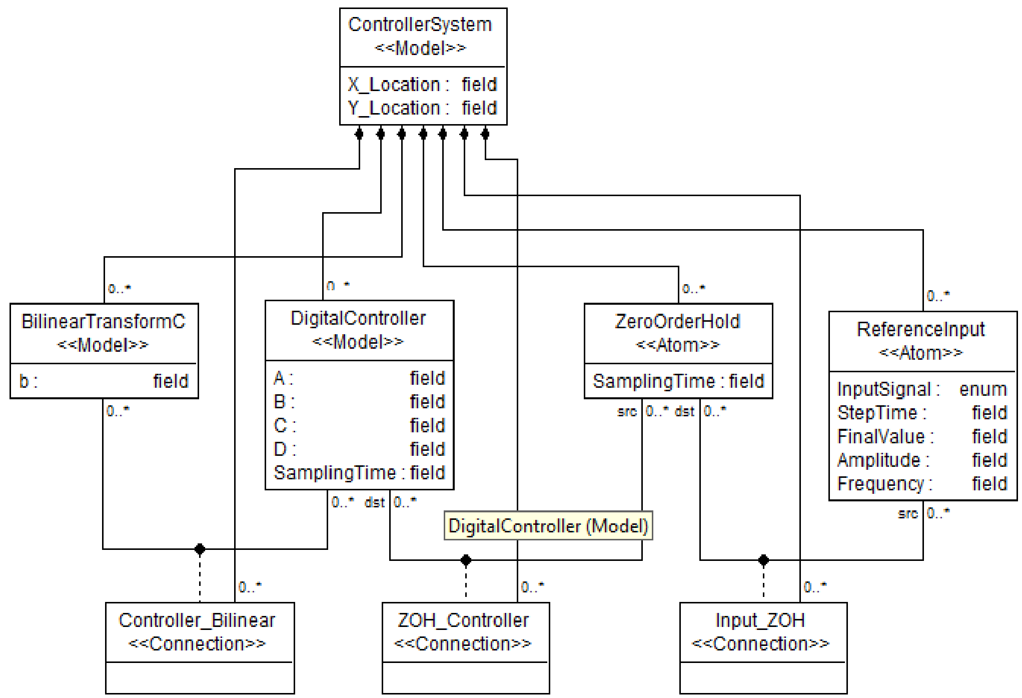
- **Components**
  - Plant
    - Discrete-Time LTI
  - BilinearTransformP
  - PassiveDownSampler
  - PassiveUpSampler
- **Interconnections**
  - Plant_Bilinear
  - Bilinear_To_DownSampler
  - UpSampler_To_Bilinear

$$x(k+1) = Ax(k) + Bu(k)$$
$$y(k) = Cx(k) + Du(k)$$



Controller Subsystem

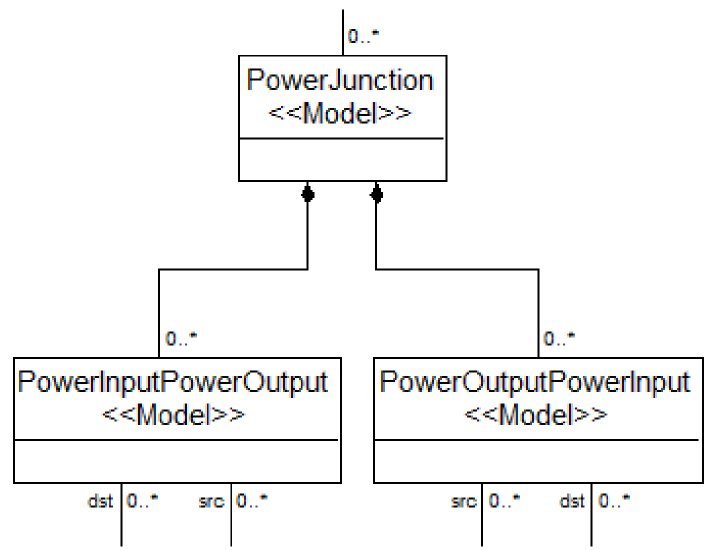
- **Components**
  - DigitalController
  - BilinearTransformC
  - Reference Input
  - ZeroOrderHold
- **Interconnections**
  - Controller_Bilinear
  - ZOH_Controller
  - Input_ZOH



PowerJunction Subsystem

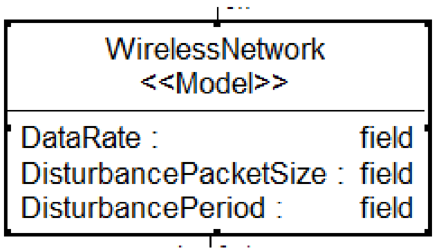
- **Components**
  - PowerInputPowerOutput (Plant connection to PowerJunction)
  - PowerOutputPowerInput (Controller connection to PowerJunction)

$$\sum_{k=m+1}^{n}(u_k^T u_k - v_k^T v_k) \ge \sum_{j=1}^{m}(u_j^T u_j - v_j^T v_j)$$



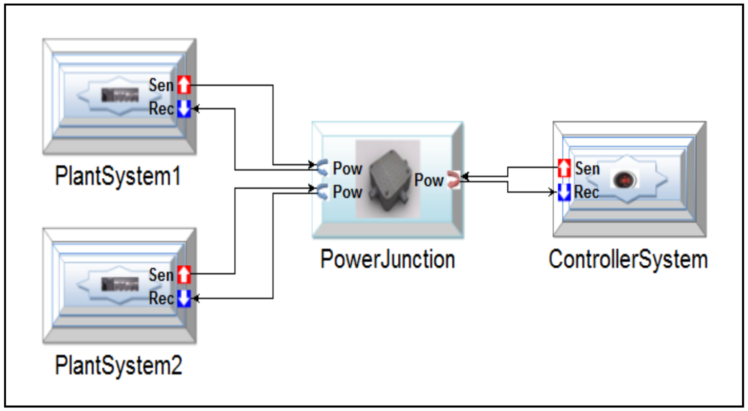
Network Subsystem

- **Network representation**
  - Defines parameters for the network
  - Ability to introduce network disturbance for simulation purposes



Control Design Aspect

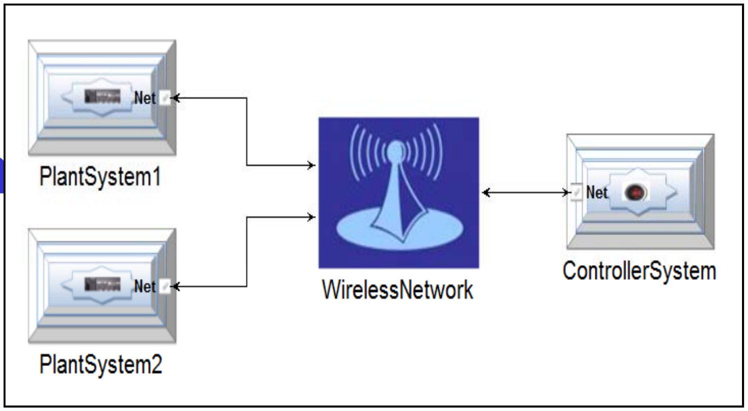
- Provides visualization of the control modeling layer indicating flow of control and sensor signals.
- Components represent control design concepts.
- Visible Components
  - Plant Subsystem
  - Controller Subsystem
  - Powerjunction



Control Design Aspect

Platform Aspect

- Provides visualization of the physical platform layer indicating the flow of data packets over the network.
- Components represent physical entities
- Visible components
  - Plant Subsystem
  - Controller subsystem
  - Wireless network



Platform Design Aspect

Sample OCL Implementation

- OCL Implementation
  - Connection between BilinearTransformC and DigitalController

Description: There must be one connection between the DigitalController block and the BilinearTransformC block

Equation:

```
self.connectionParts("Controller_Bilinear").size
()=1
```

Passivity Analysis

- In order to achieve the desirable properties of passive systems
  - Analyze the networked control system
- Analysis of the NCS
  - Component Analysis
  - System-level Analysis

Component Analysis

- Analyze individual components of the NCS
  - Only Plant and Controller Components
- Designed Model Interpreter Tool integrated in GME visits each tool and invokes the analysis function.

$$\begin{bmatrix} A^T P A - P - \tilde{Q} & A^T P B - \tilde{S} \\ (A^T P B - \tilde{S})^T & -\tilde{R} + B^T P B \end{bmatrix} \leq 0$$

$$\tilde{Q} = C^T Q C, \ \tilde{S} = C^T S + C^T Q D$$

$$\tilde{R} = D^T Q D + (D^T S + S^T D) + R$$

$$\exists \varepsilon > 0, \ Q = -\varepsilon I, \ R = 0, \ S = \tfrac{1}{2} I$$

Kottenstette, Antsaklis 2008

- CVX semi-definite programming tool (SDP) used in a Matlab script to solve LMI.

System-level Analysis

- Due to the “correct-by-construction” approach
  - Network as a whole ensure global robustness by a combination of
    - Individual components satisfaction of passivity constraints.
    - Passive Composition constraints encoded in the modeling language.
- Reduction in the analysis burden of verifying passivity.

PaNeCS Design Flow

NCS Modeling → SL/tt model generator

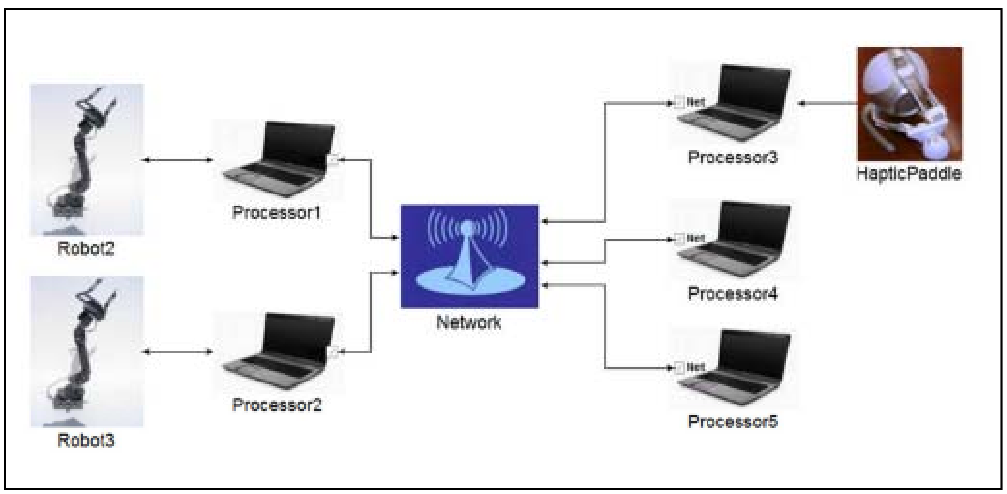
PaNeCS GME → **PaNeCS 2tt** → **Simulink TrueTime** →

- structural constraints
- component passivity analysis

- model transformation

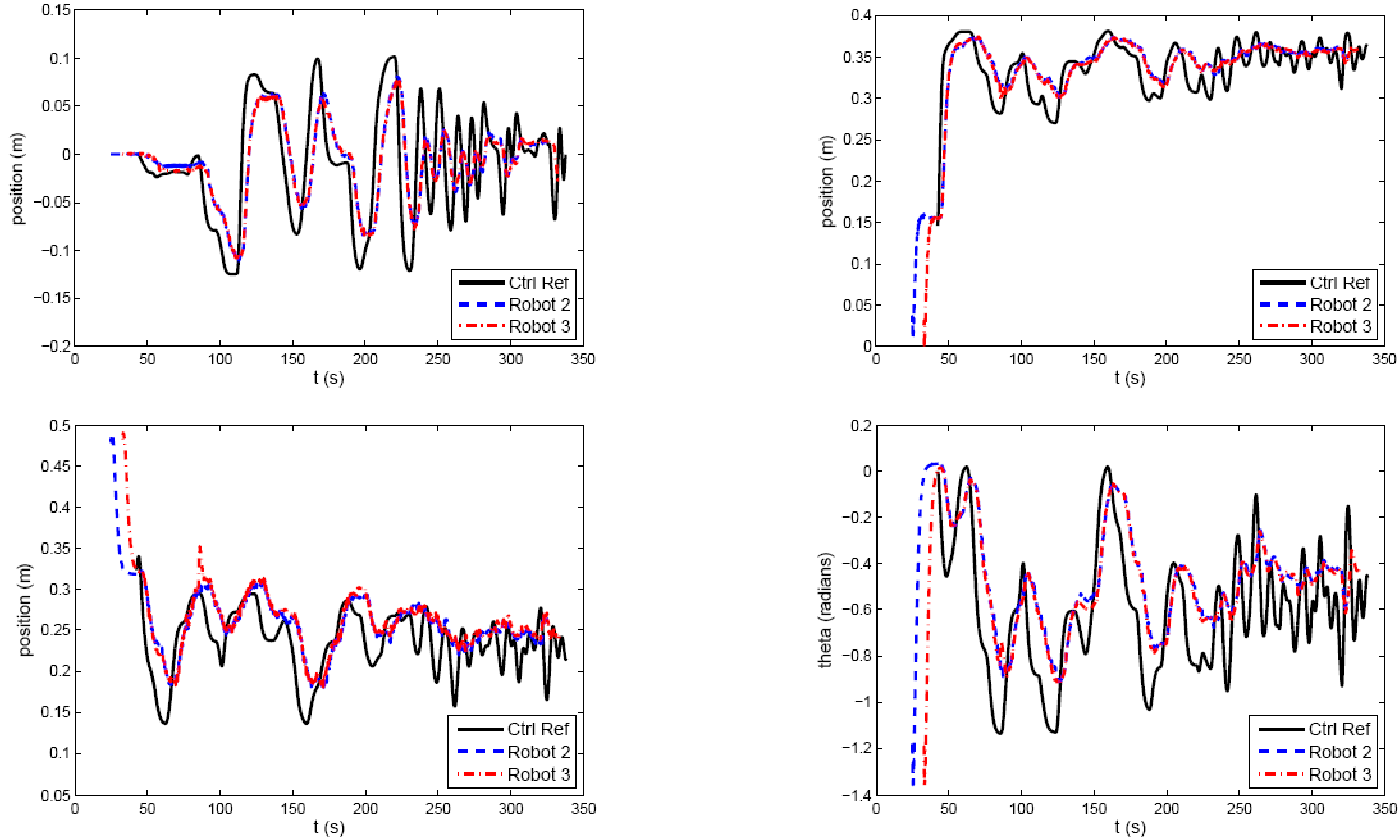
- behavior simulation

Experimental Setup



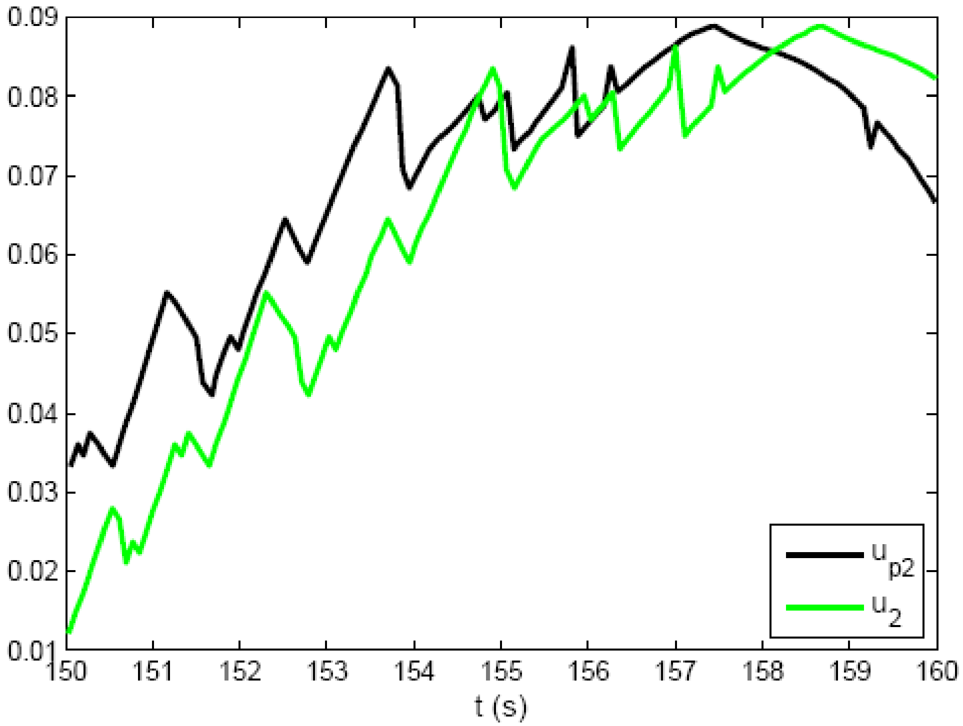
- Two CrustCrawler robotic arms
  - 4 DOF with AX-12 smart servos at each joint
- Novint haptic paddle
- Five networked Windows PCs with Matlab/Simulink

Experiment 1: Nominal Case



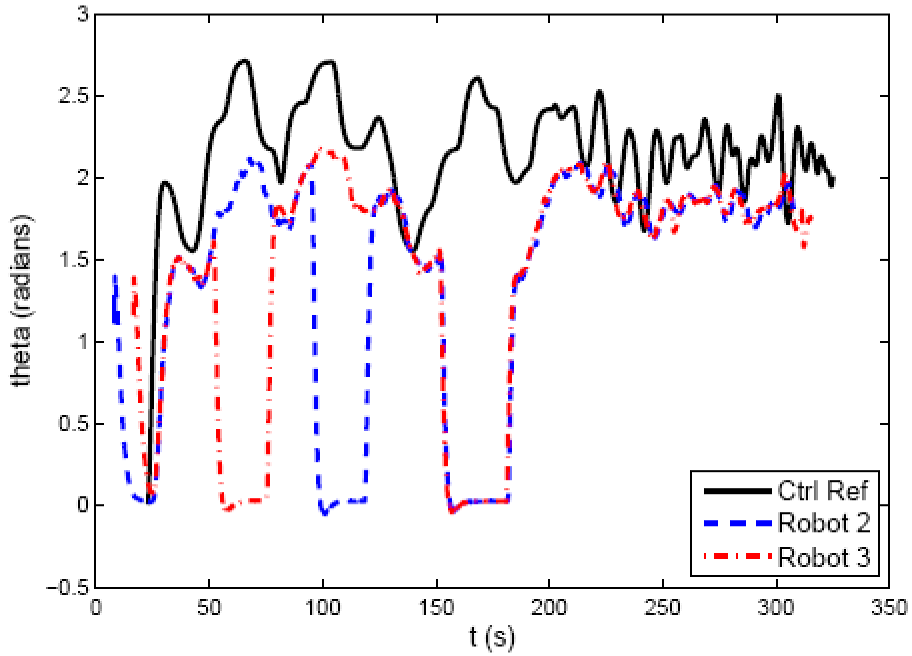
x-y-z coordinates and angle of joint 2 of reference, robot 2, and robot 3

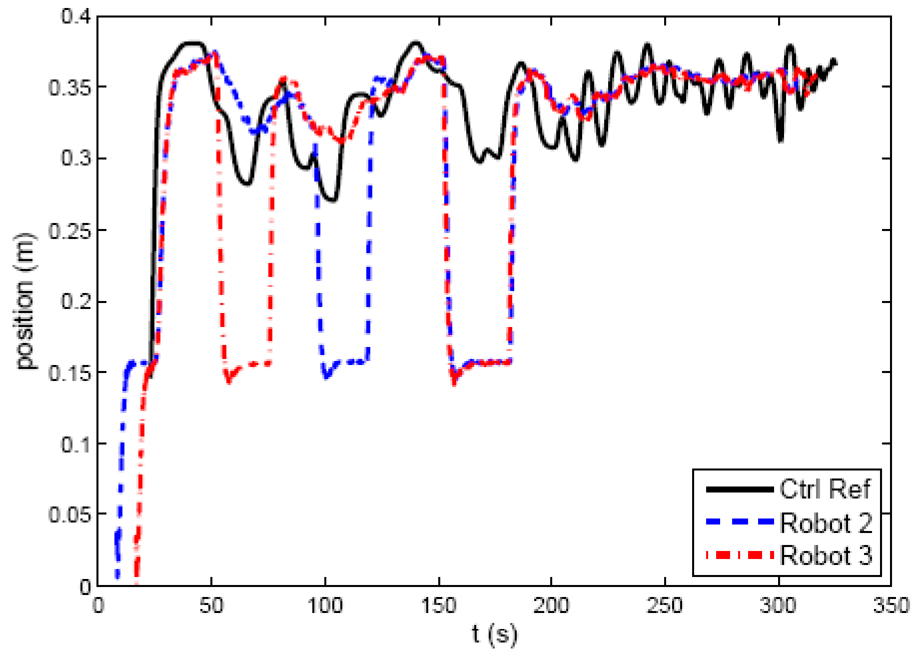
Experiment 1: Time Delay



Time Delay Between Robot 2 and Power Junction

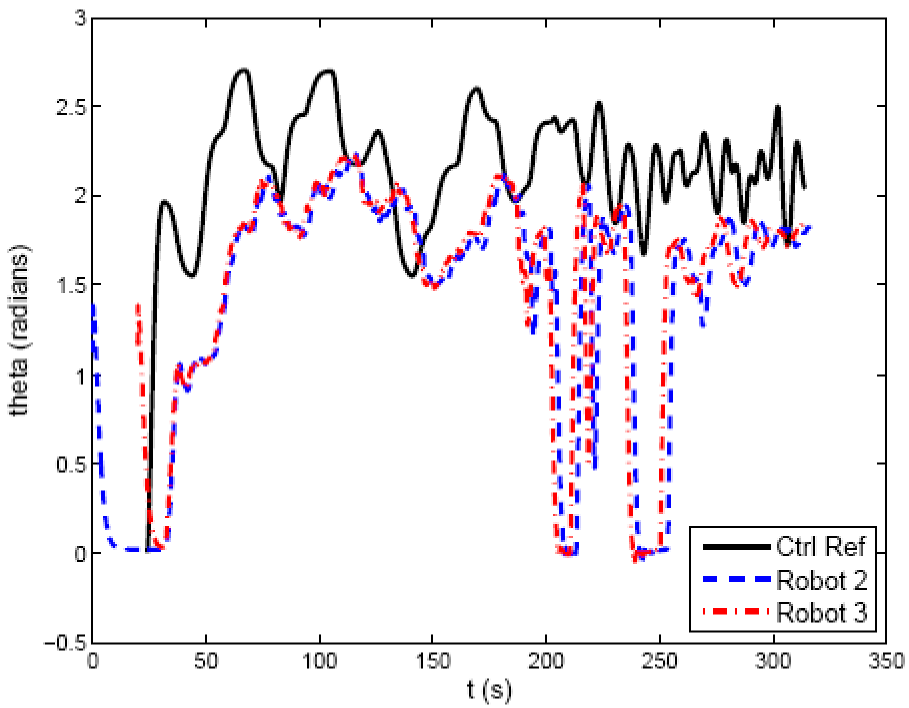
Experiment 2: Persistent Link Interruptions

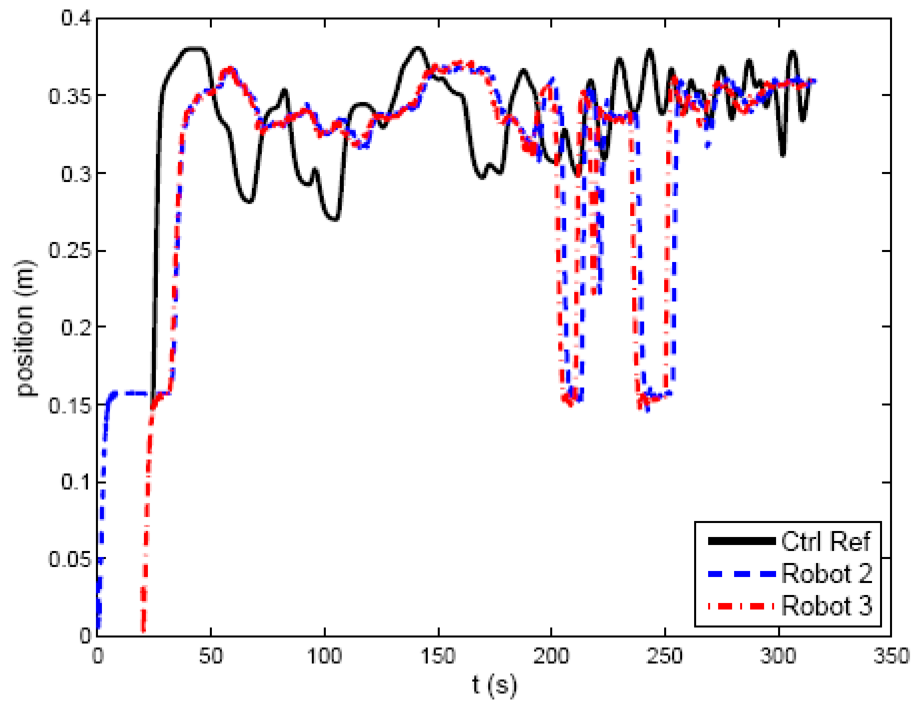




Angle of joint 3 and y coordinate of reference, robot 2, and robot 3

Experiment 2: Intermittent Wireless Connection



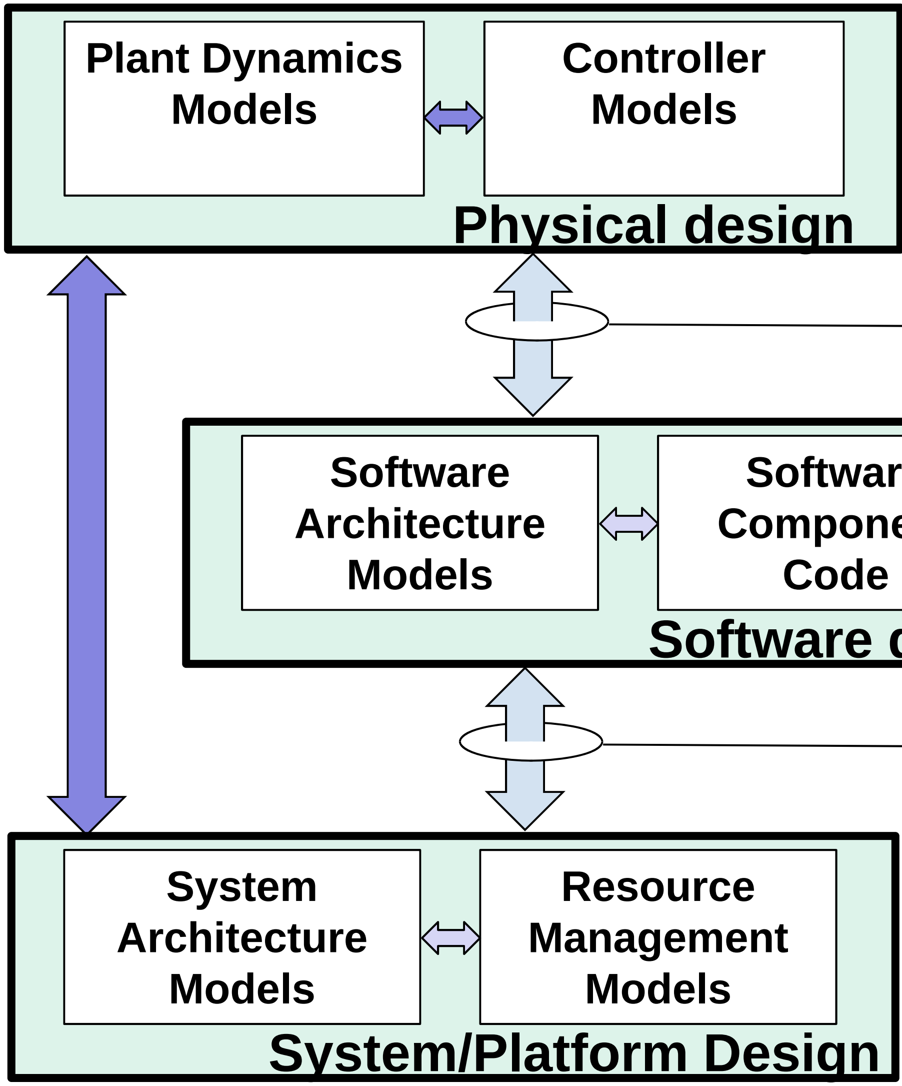


Angle of joint 3 and y coordinate of reference, robot 2, and robot 3

Outline

- Effects of heterogeneity
- Passivity-based design
- **Toward a high-confidence model-based design tool chain**

Model-based Design Flow



Controller dynamics is developed without considering implementation uncertainties (e.g. word length, clock accuracy) optimizing performance.

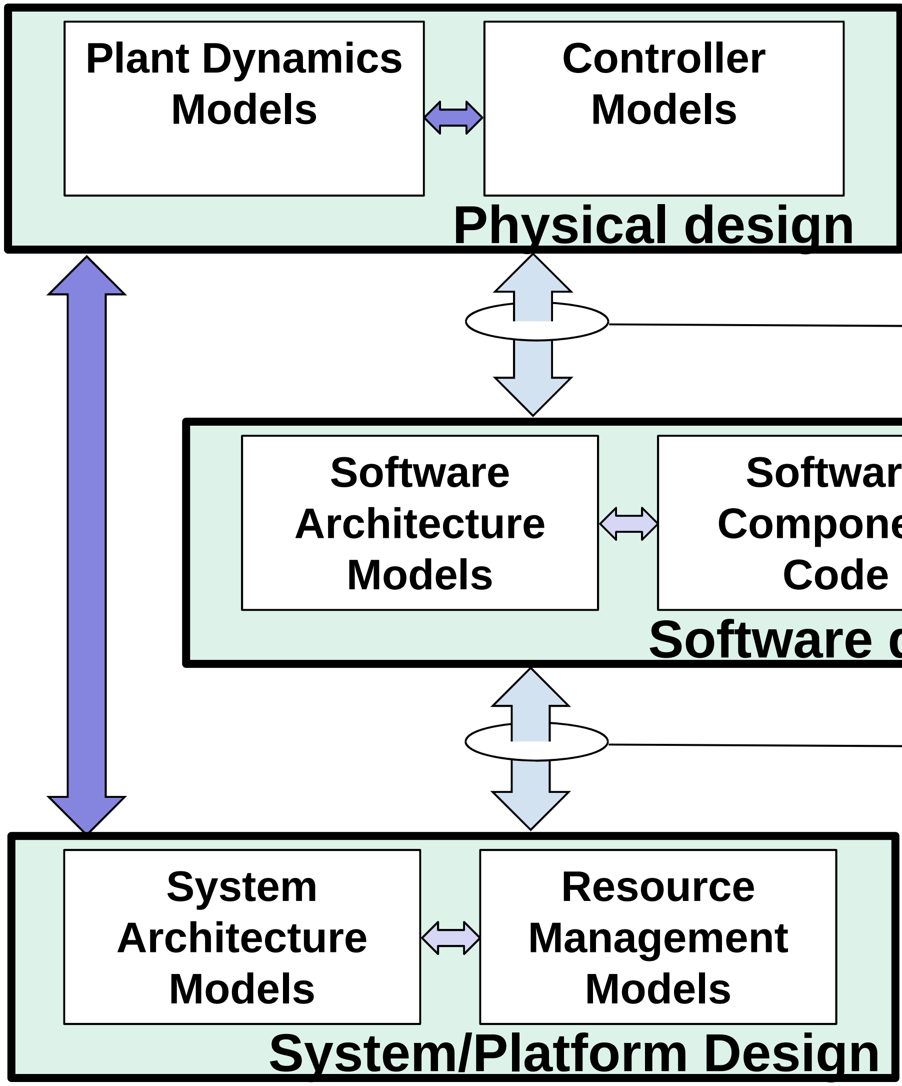
Assumption: Effects of digital implementation can be neglected

Software architecture models are developed without explicitly considering systems platform characteristics, even though key behavioral properties depend on it.

Assumption: Effects of platform properties can be neglected

System-level architecture defines implementation platform configuration. Scheduling, network uncertainties, etc. are introduce time variant delays that may require re-verification of key properties on all levels.

Model-based Design Flow



Controller dynamics is developed using passivity-based design. Current work focuses on performance optimization and safety.

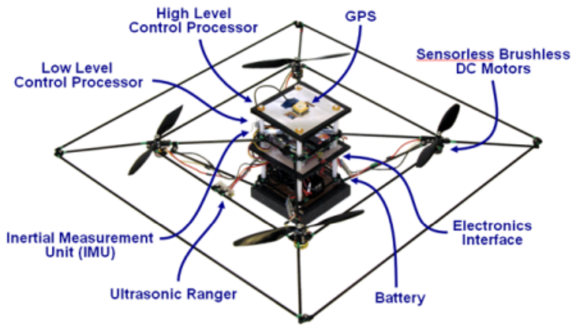
Assumption: Time varying delays do not impact stability.

Software models need to be verified for essential properties e.g. deadlock freeness. -> **BIP**

Assumption: Effects of platform on properties can be analyzed

System-level architecture defines implementation platform configuration. Scheduling, network uncertainties, etc. are introduce time variant delays that may require re-verification of key properties on all levels.

Experimental Platform

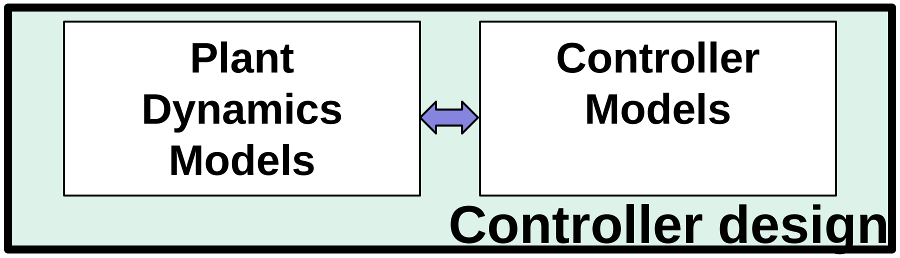


Real-time Simulation Platform 'Virtual plant'

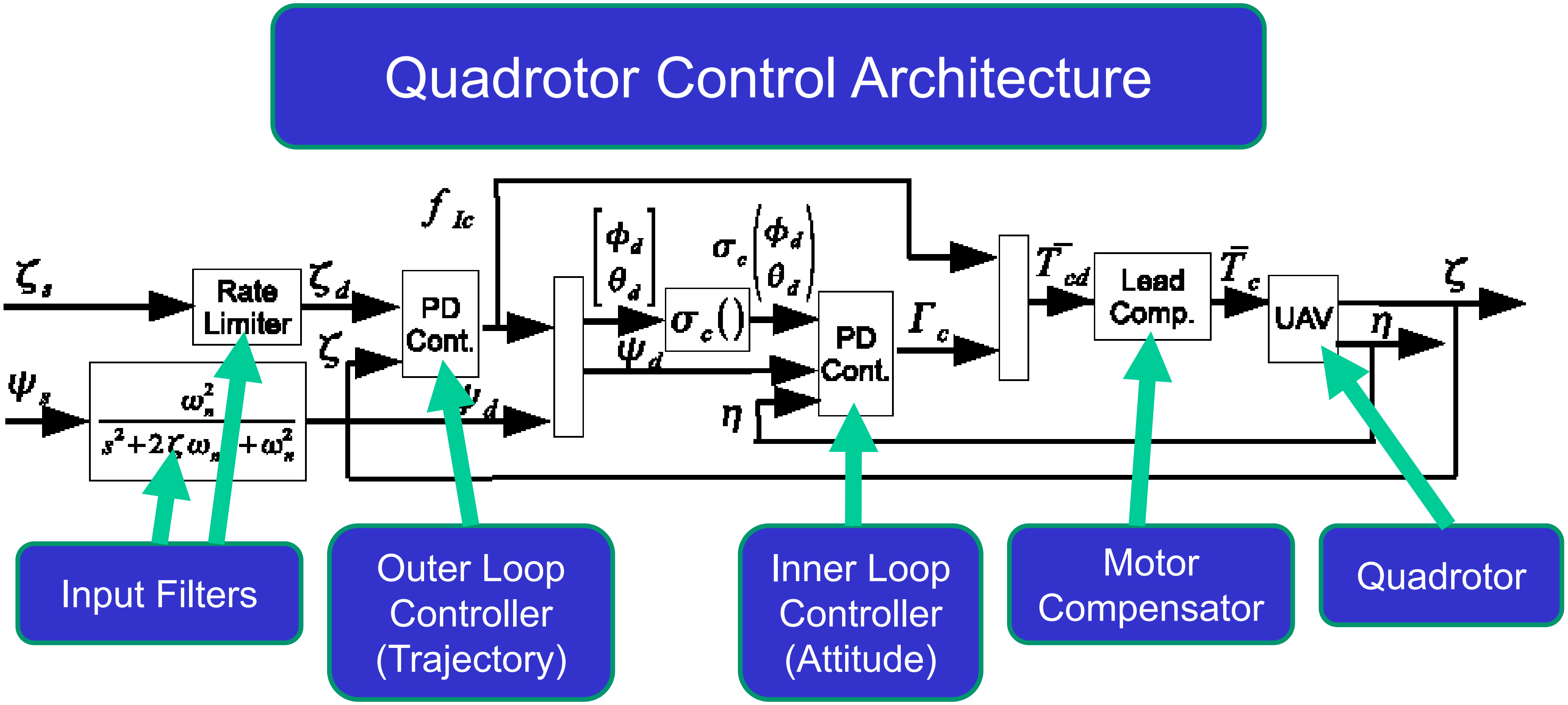
- Gumstix/Robostix

- Linux + AVR micro
- TT Virtual Machine on Linux/UDP + FreeRTOS
- No fault tolerance (yet)

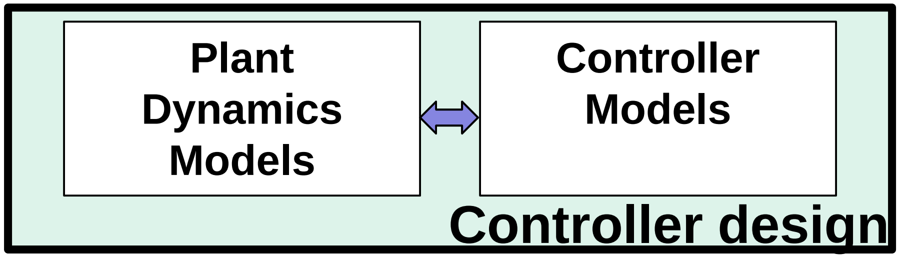
Design Flow: Passive Controllers



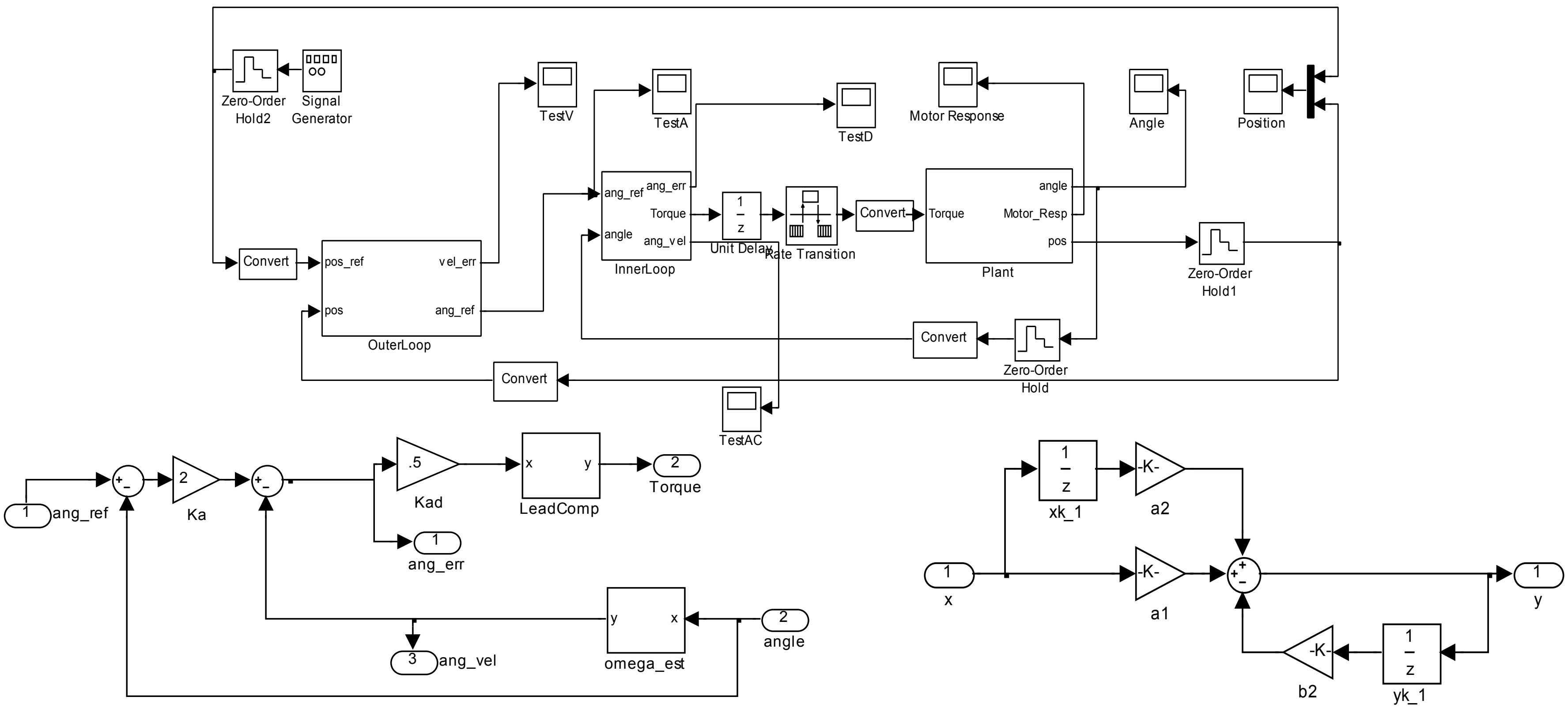
Passive systems are robust to network delays and quantization errors. We can design controllers to “passify” many systems that are not quite passive.



Design Flow: Passive Controllers

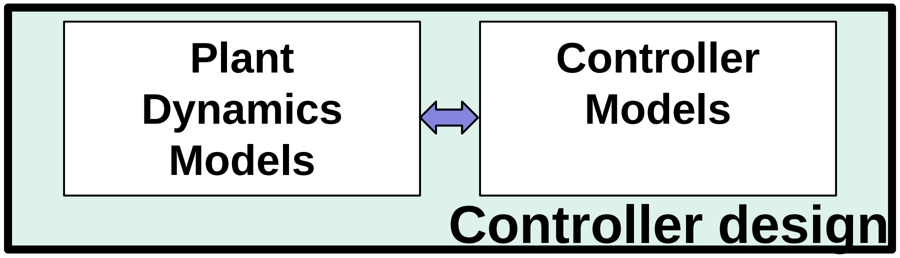


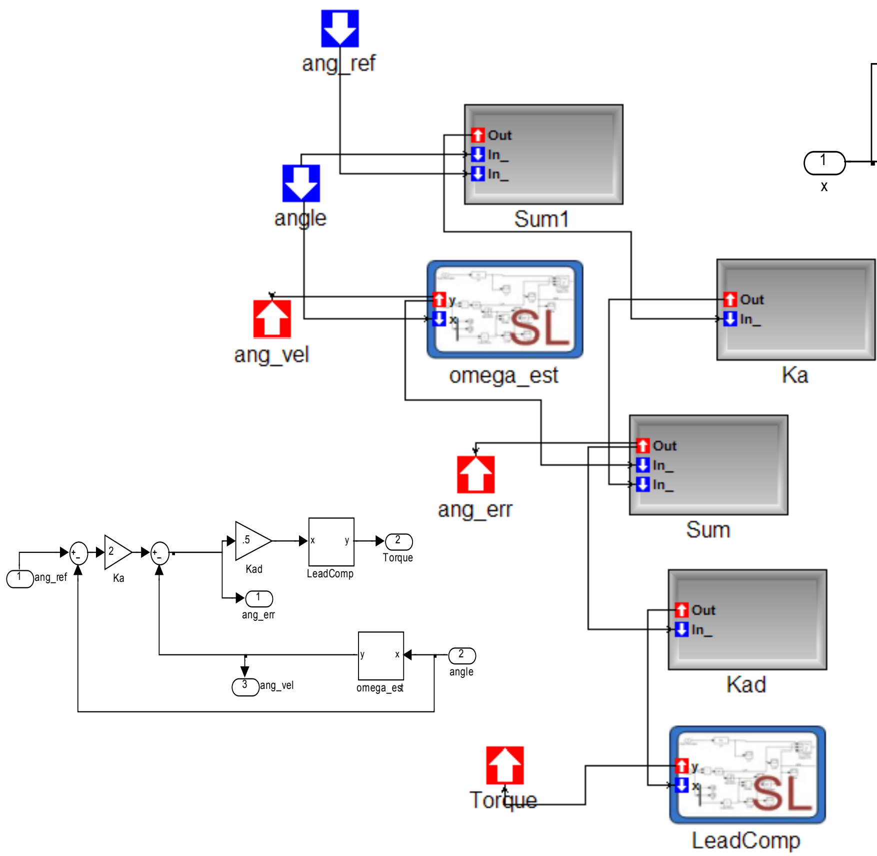
Simulink for control design

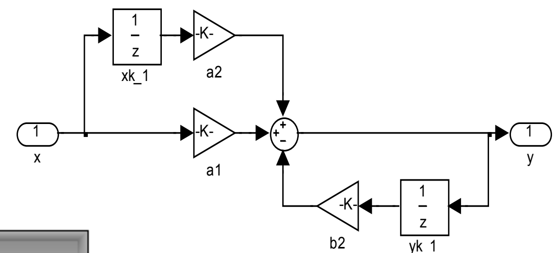


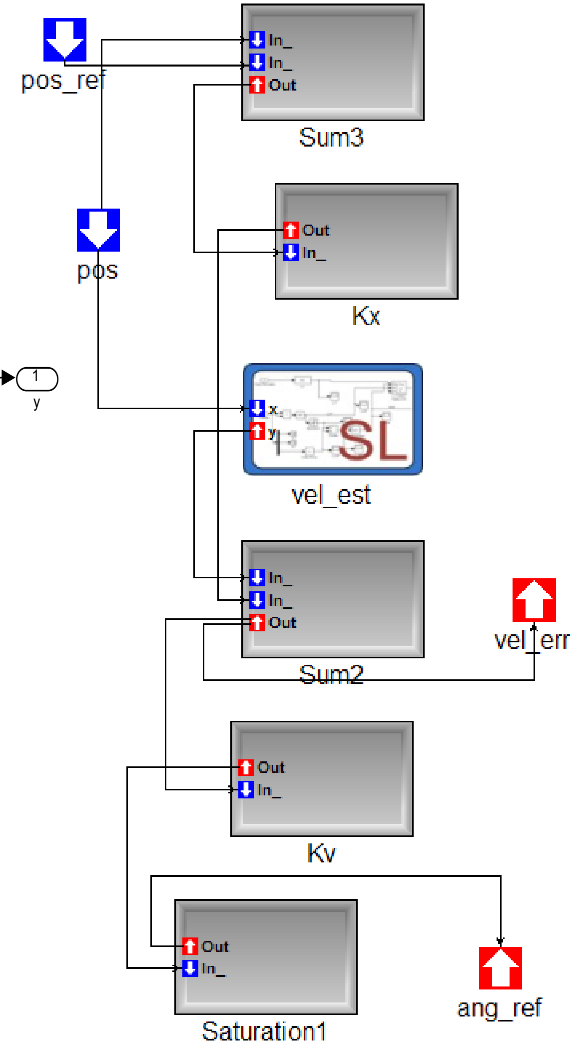
This example is a subset of the full quadrotor model we're using for testing.

Design Flow: Moving toward implementation



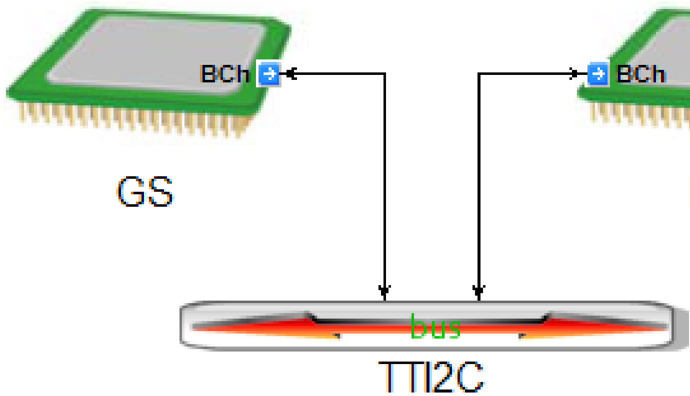






Simulink models imported into GME

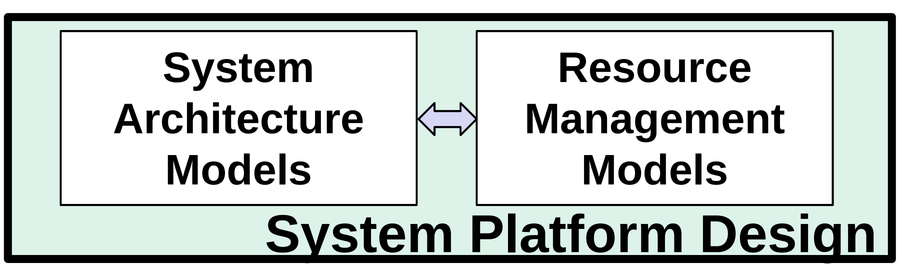
Design Flow: Platform



Two processors connected by a time-triggered bus

System timing parameters are captured here:

- *fundamental tick time of processing units*
- *data transfer setup times*
- *bus rates*

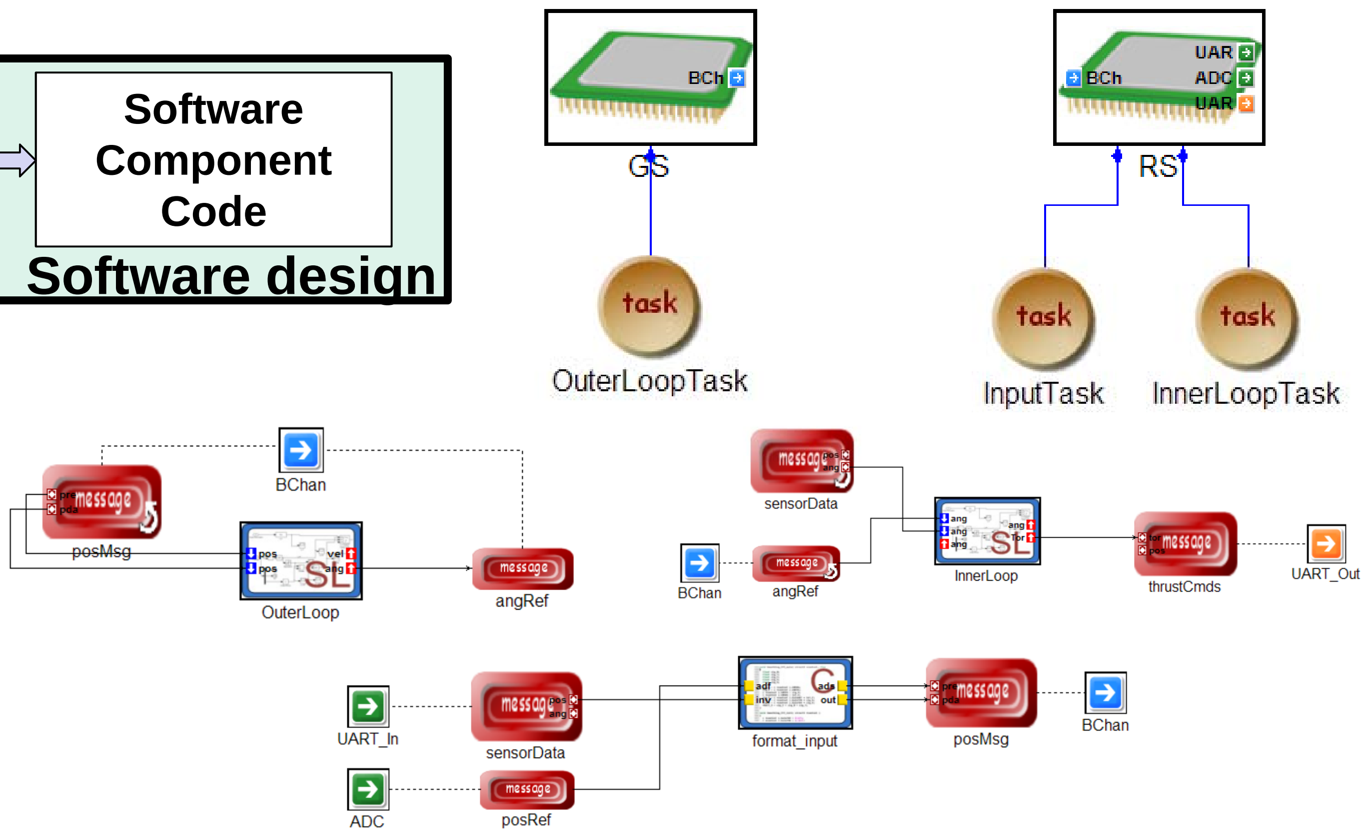


Design Flow: Software

Software architecture (types and dependencies)

Software Architecture Models ⟺ **Software Component Code**

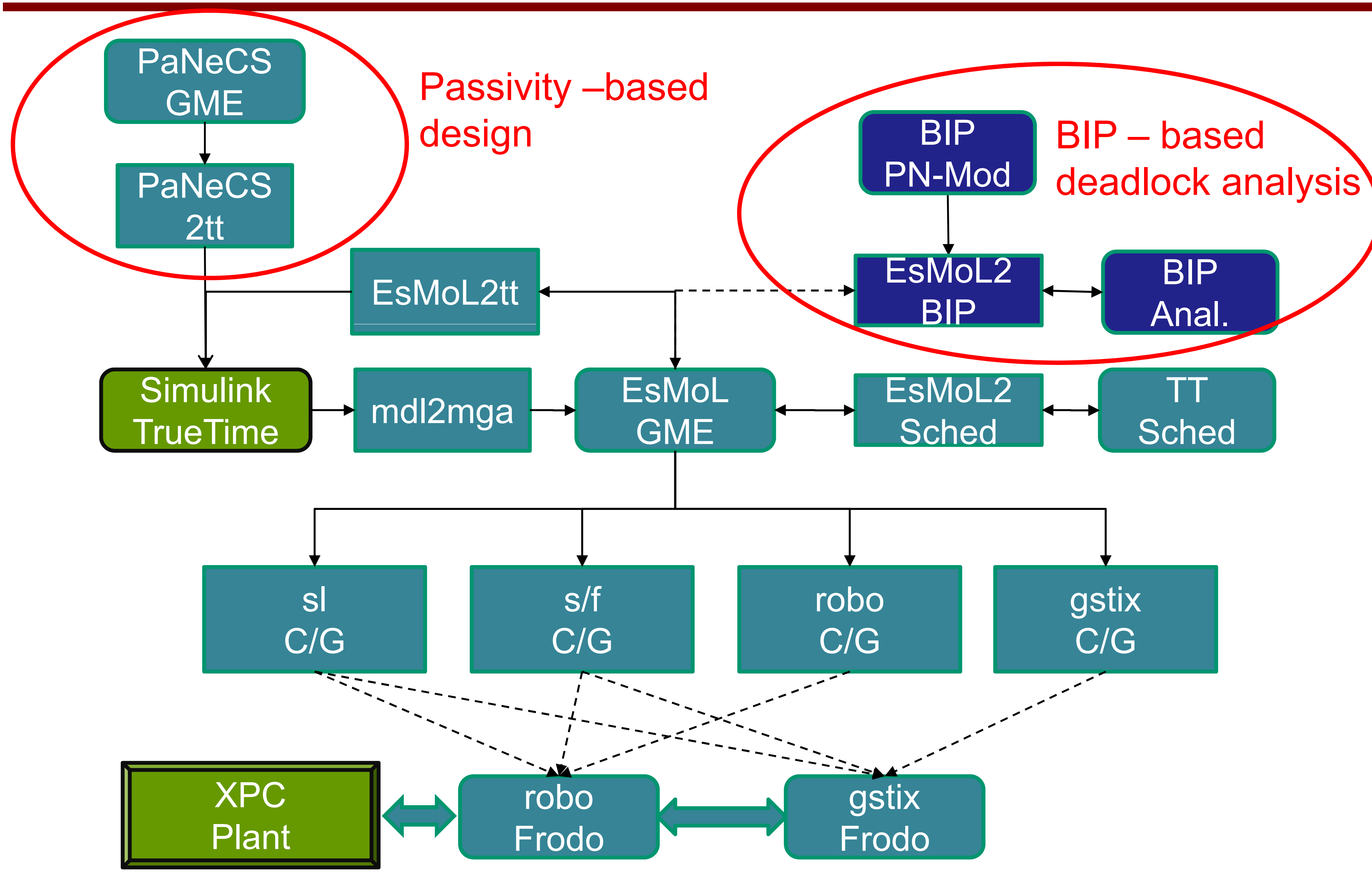
Software design



Deployment details:

- *tasks*
- *messages*
- *processor assignments*

HCDDES Tool Chain



Summary

- Composition in heterogeneous systems requires decoupling among design concerns
- Decoupling requires significant effort, but the benefits are also significant: this is the primary tool for decreasing complexity
- There is a performance tradeoff; in safety critical systems it still may be the right choice.

Some References

- Kottenstette, N., J. Hall, X. Koutsoukos, P. J. Antsaklis, and J. Sztipanovits, "Digital Control of Multiple Discrete Passive Plants Over Networks", *Int. J. of Systems, Control and Communications* Special Issue on "Progress in Networked Control Systems" (in press) available as ISIS Technical Report# ISIS-09-102
- Emeka Eyisi, Joseph Porter, Joe Hall, Nicholas Kottenstette, Xenofon, Koutsoukos and Janos Sztipanovits: "PaNeCS: A Modeling Language for Passivity-based Design of Networked Control Systems", Second International Workshop on Model Based Architecting and Construction of Embedded Systems ACES09, Denver CO October 6, 2009
- Nicholas Kottenstette, Xenofon Koutsoukos, Joe Hall, Panos Antsaklis, and Janos Sztipanovits, ``Passivity-Based Design of Wireless Networked Control Systems for Robustness to Time-Varying Delays'', *29th IEEE Real-Time Systems Symposium (RTSS 2008)*, Barcelona, Spain, November 30 - December 3, 2008
- Joseph Porter, Peter Volgyesi, Nicholas Kottenstette, Harmon Nine, Gabor Karsai, Janos Sztipanovits: "An Experimental Model-Based Rapid Prototyping Environment for High-Confidence Embedded Software," *20th International Symposium on Rapid System Prototyping (RSP09)*, Paris, France, June 23-26, 2009

Passive Up-sampling and Down-sampling

- Because of bandwidth constraints, the local digital controllers for each robot run at a faster rate than the network controller
- Ensure that no energy is generated, and thus passivity is preserved
- Passive down-sampling

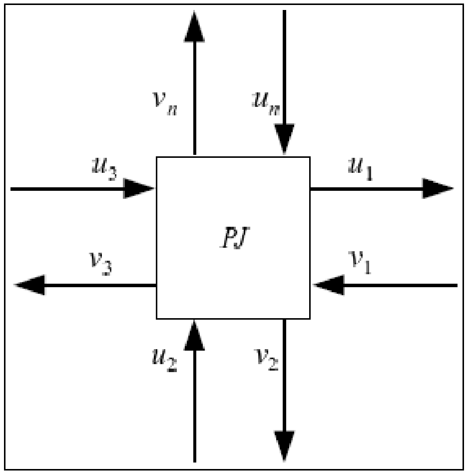
$$u_{\mathrm{pDSk}_k}(j) = \sqrt{\sum_{i=M(j-1)}^{Mj-1} u_{\mathrm{pk}_k}^2(i)}\,\mathrm{sgn}\Big(\sum_{i=M(j-1)}^{Mj-1} u_{\mathrm{pk}_k}(i)\Big)$$

- Passive up-sampling

$$v_{\mathrm{pk}_k}(i) = \sqrt{\frac{1}{M}}\,v_{\mathrm{pDSk}_k}(j-1),\ i = Mj, \ldots, M(j+1)-1$$

where $i = \left\lfloor \frac{t}{T_s} \right\rfloor$ and $j = \left\lfloor \frac{t}{MT_s} \right\rfloor$

Power Junction



- Compose a network in which multiple passive plants can be interconnected to multiple passive controllers
- Interconnect wave variables from multiple controllers and plants such that the total power input is always greater than or equal to the total power output

$$\sum_{k=m+1}^{n} (u_k^{\mathsf{T}} u_k - v_k^{\mathsf{T}} v_k) \geq \sum_{j=1}^{m} (u_j^{\mathsf{T}} u_j - v_j^{\mathsf{T}} v_j)$$