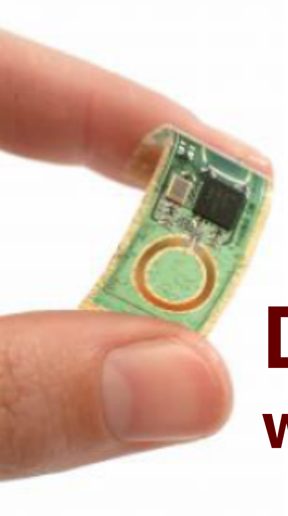


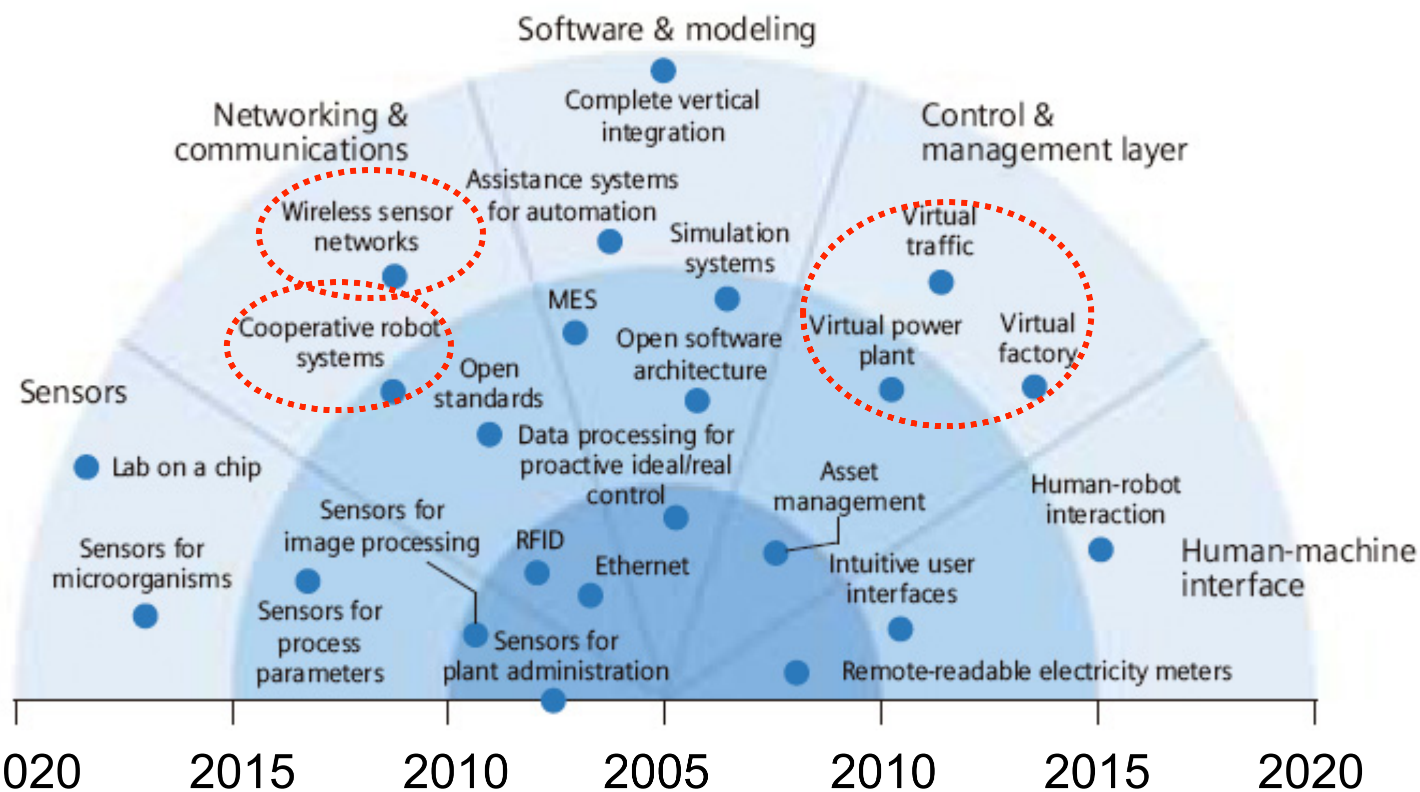
Distributed Control-as-a-Service
with Wireless Swarm Systems

Prof. Rahul Mangharam
Director, Real-Time & Embedded Systems Lab
Dept. Electrical & Systems Engineering
Dept. Computer & Information Science
University of Pennsylvania
rahulm@seas.upenn.edu

Penn Engineering

mLAB Real-Time & Embedded

Discrete and Process Wireless Control



Wireless Control/Actuation ← Wireless Monitoring **→ Wireless Control/Actuation**

Roadmap from Siemens

Our Focus: Industrial Control Systems

Natural gas processing plants

Oil-refineries

Paper pulp manufacturing

$120 Billion/Year market

PLC Architectures and Software are from the mid-1980s to early-1990

Software Issues with Industrial Automation

- **Automotive assembly lines lose over $22,000/minute downtime**
- **Systems are rigid, difficult to maintain, operate and diagnose**

Goal: Plug-n-Play Wireless Automation Control Systems

Penn Engineering

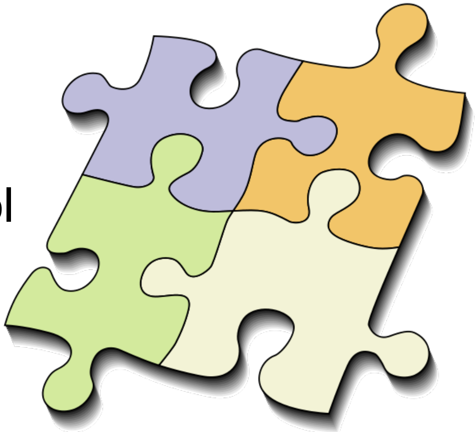
mLAB Real-Time & Embedded

Advantages of wireless control system architectures

1) Plug-n-Play capabilities:

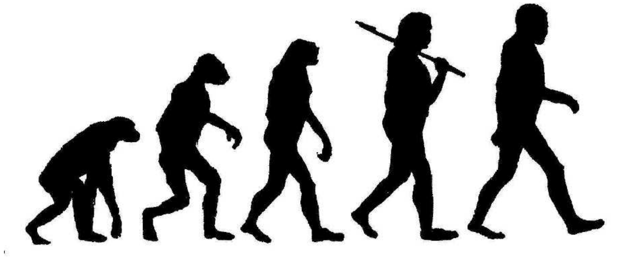
Minimizes downtime with efficient recovery from controller faults as re-connecting the logical I/O lines of a wireless backup controllers is seamless.

2) Compositionality:



Enables system evolution through logical expansion/ contraction of plants and controllers with composable control systems. Suitable for emerging markets.

3) Runtime adaptation:



Control stability and performance are maintained in the presence of node, link and topological changes.

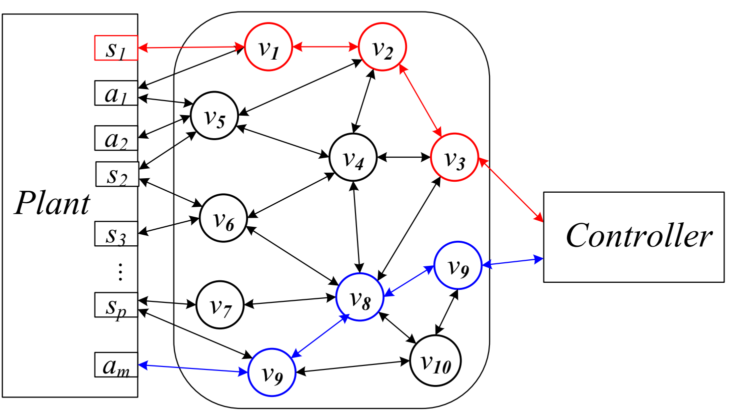
Two Disruptive Approaches for Wireless Control

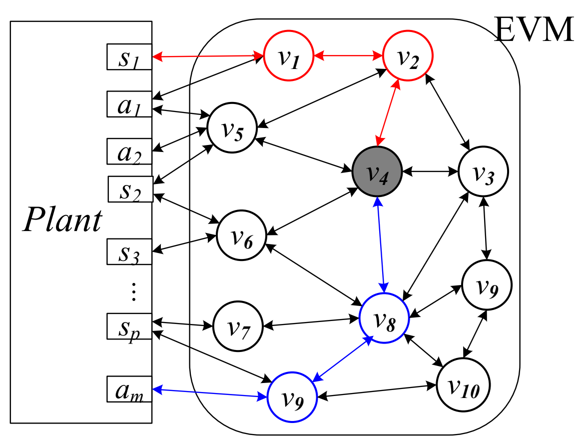
A Embedded Virtual Machines

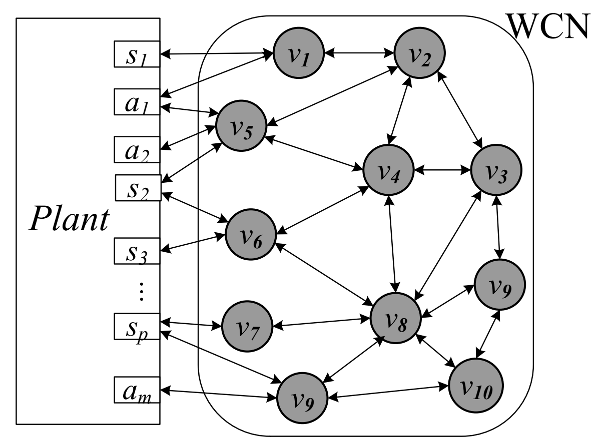
- Runtime abstraction of wireless control
- Controller tasks migrate across physical nodes
- Robust to topological changes

B Wireless Control Network

- Fully distributed in-network approach
- Low computational overhead and simple, static scheduling
- Enables plant & controller composition







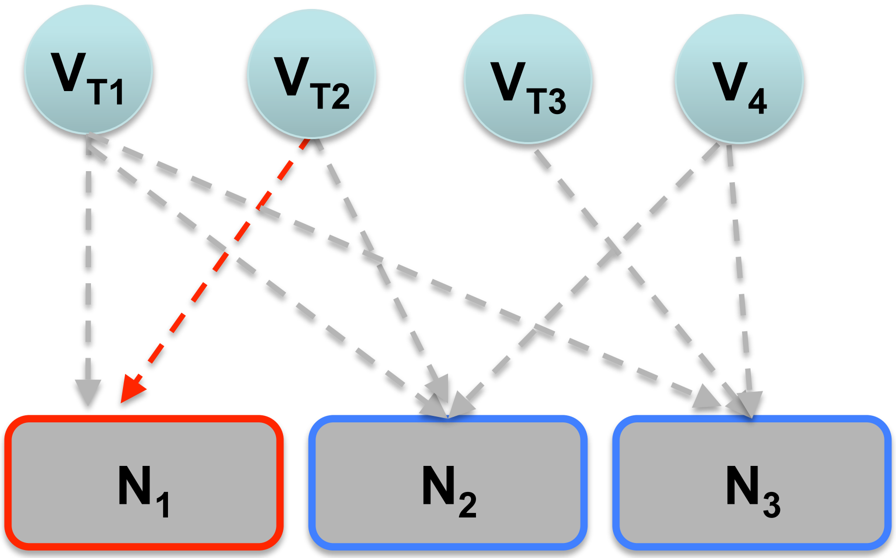
Published in: IEEE WCPS 2009, RTAS'10, CDC_1'10, CDC_2'10, ACM TECS'11, IEEE TAC'11, CDC'11, ACC'11, CDC'12, IPSN'12, JSAC'12, IISc'13…

Embedded Virtual Machines

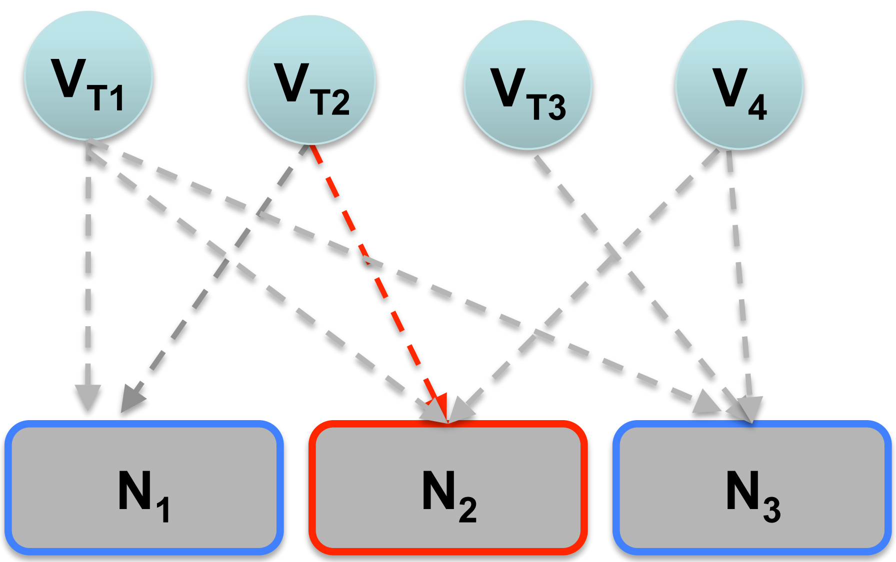
For Robust Wireless Control

The Core Idea

Virtual Task to Physical Resource Mapping



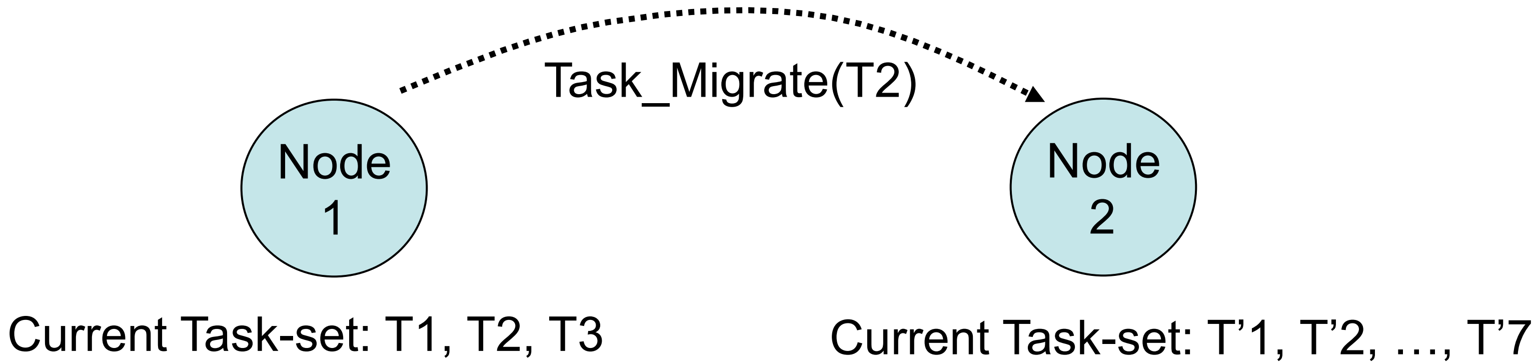
V1	1	N1	active	1s	100ms
V1	1	N3	backup	1s	50ms
V1	1	N2	dormant	1s	100ms
V2	0	N2	active	0.5s	50ms
V2	1	N1	backup	0.5s	50ms
V3	1	N3	active	0.5s	200ms
V4	1	N2	active	0.25s	200ms
V4	1	N3	backup	0.25s	100ms



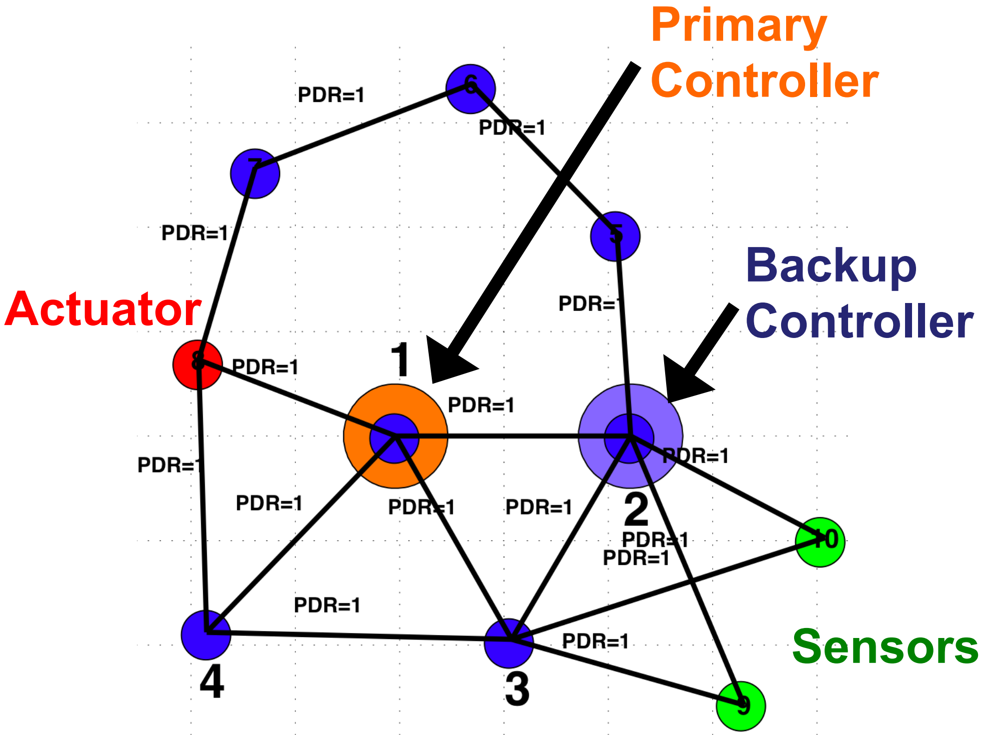
V1	1	N1	active	1s	100ms
V1	1	N3	backup	1s	50ms
V1	1	N2	dormant	1s	100ms
V2	1	N1	active	0.5s	50ms
V3	1	N3	active	0.5s	200ms
V4	1	N2	active	0.25s	200ms
V4	1	N3	backup	0.25s	100ms

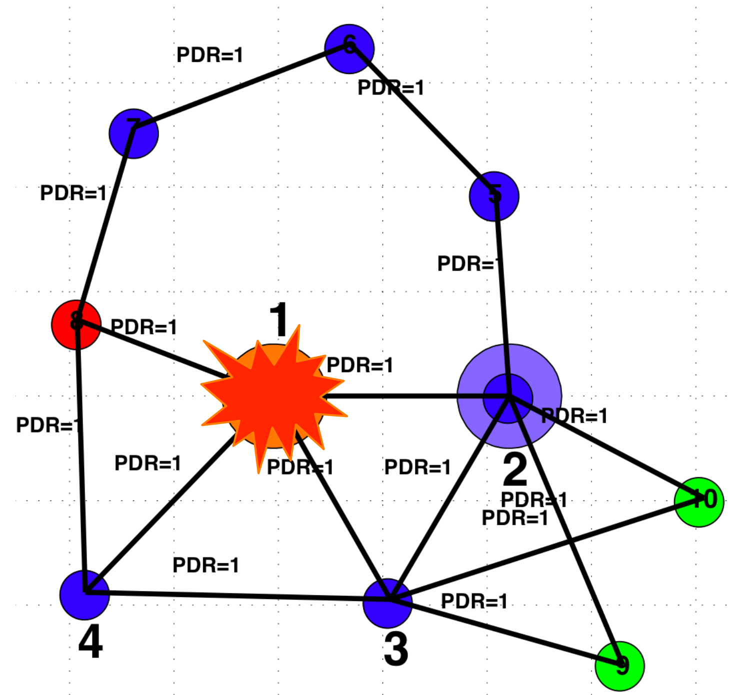
Runtime Task Management

- Task Migrate
  - Task instructions
  - Stack
  - Data
  - Associate libraries
  - Control/Schedule/Resource meta data



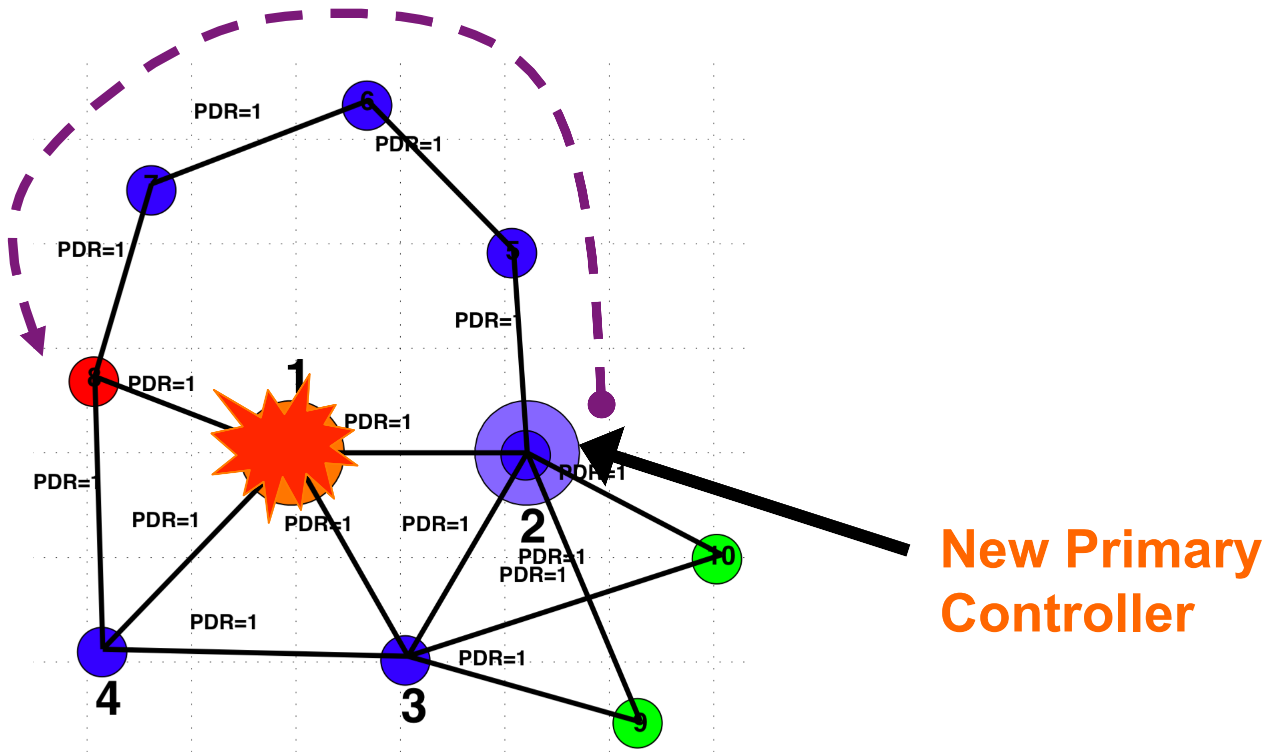
Example: When Routing Fails → Migrate





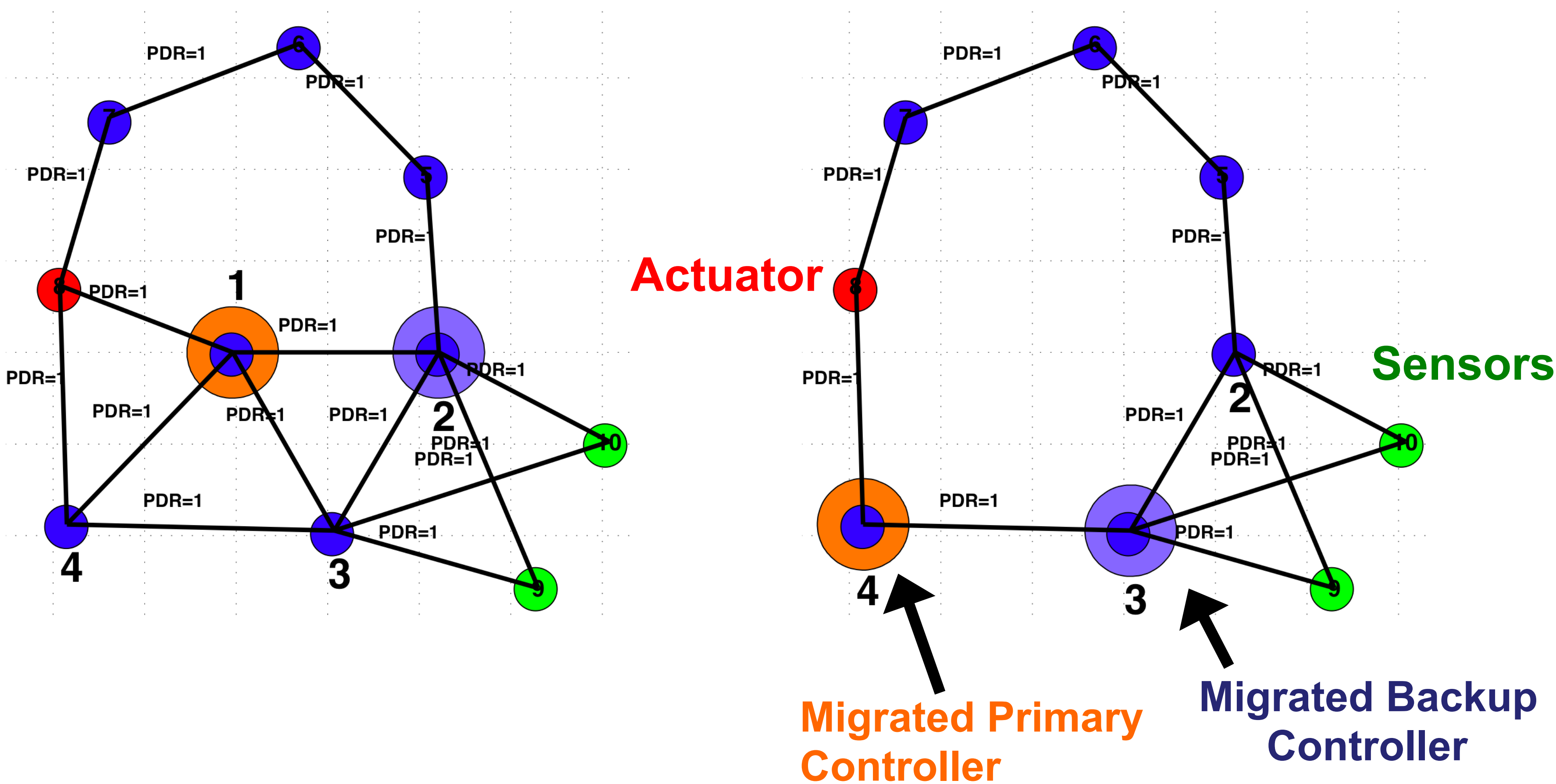
Example: When Routing Fails → Migrate

What if we just used the backup and re-routed the control path?



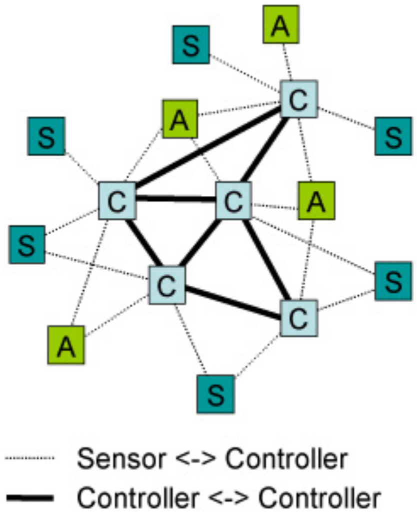
Routing from Backup Controller to Actuator fails to meet Stability Constraint

Example: When Routing Fails → Migrate



Embedded Virtual Machines

- **Distributed runtime system** that dynamically selects primary-backup sets of controllers to guarantee QoS given spatial and temporal constraints of the underlying wireless network.



(a)

- **Algorithms** to program virtual components and ***maintain functional and para-functional invariants*** across the system.

Focus on controller reliability and fault tolerance

The Goal

- **Maintain Functional Invariant**
  - Control Law

Control Stability

- **Maintain Para-Functional Invariants**
  - Timeliness, reliability, fault-tolerance
- **Predictable outcomes** in presence of controller / link failures
  - For planned changes
- **Graceful degradation without violating safety**
  - For unplanned changes

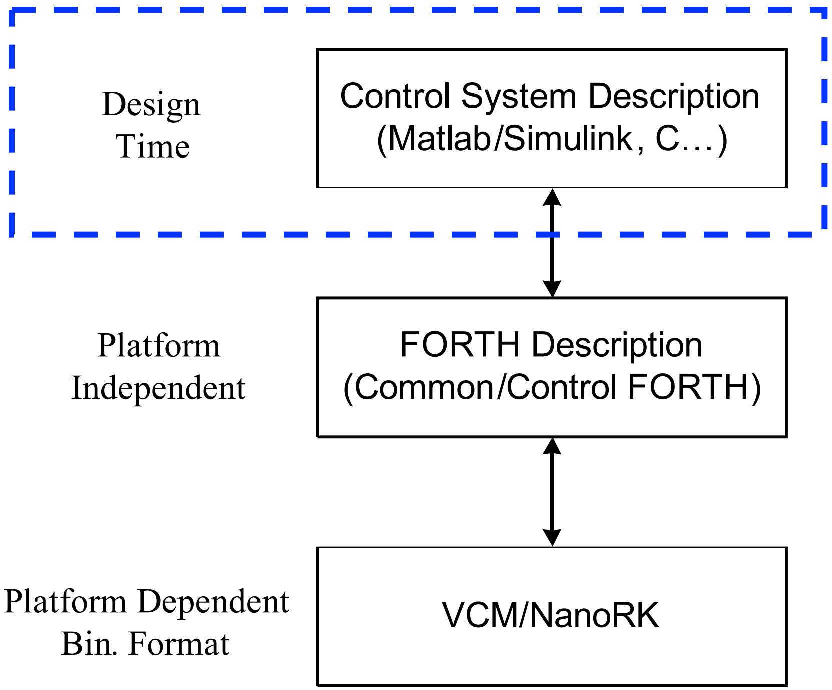
Reliability

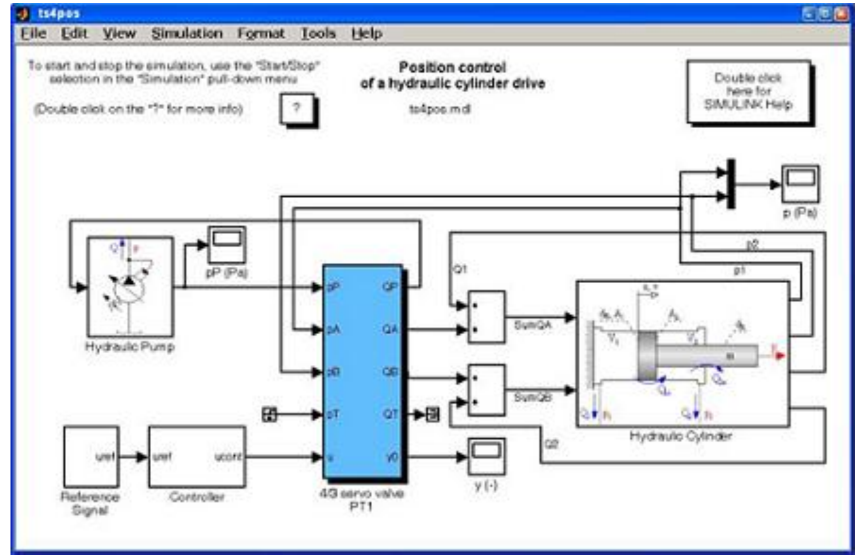
- **Composability: for multiple plants at runtime**
  - Increase functionality or respond failure

Composition

- **Adaptive Runtime Resource Re-appropriation**
  - Optimization for Dynamic changes in service/throughput

EVM Design Flow
Design Time with Simulink

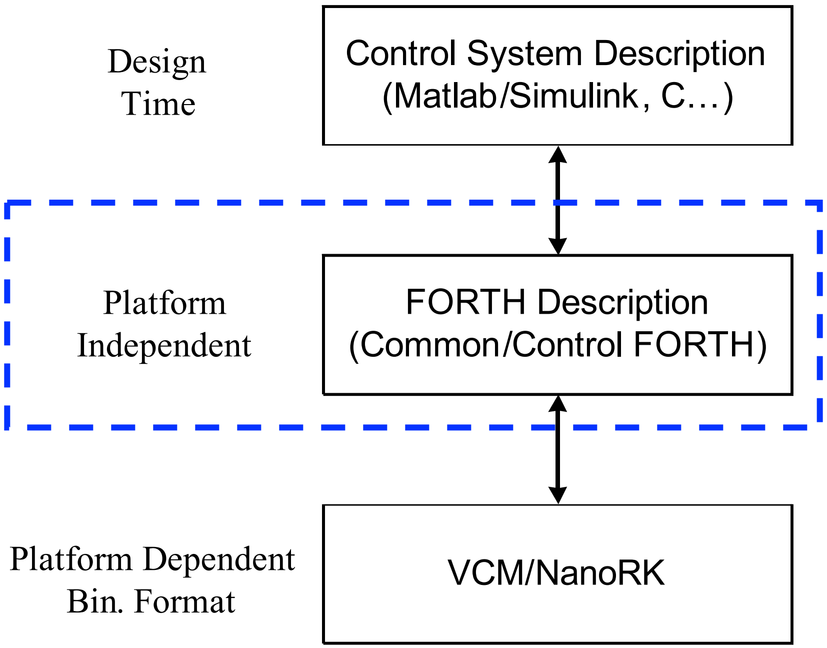




Penn Engineering

LAB
Real-Time & Embedded

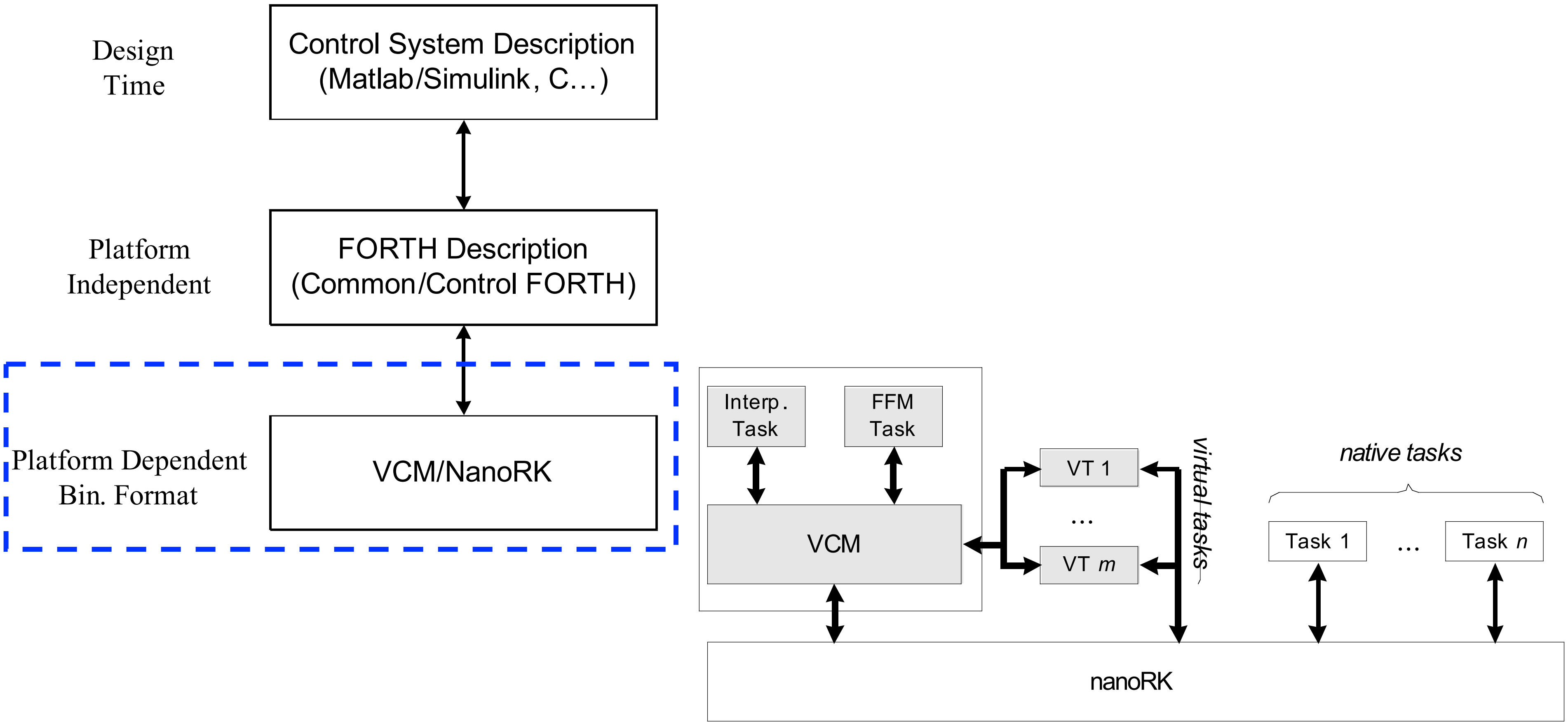
EVM Design Flow
Domain Specific Language Interpreter



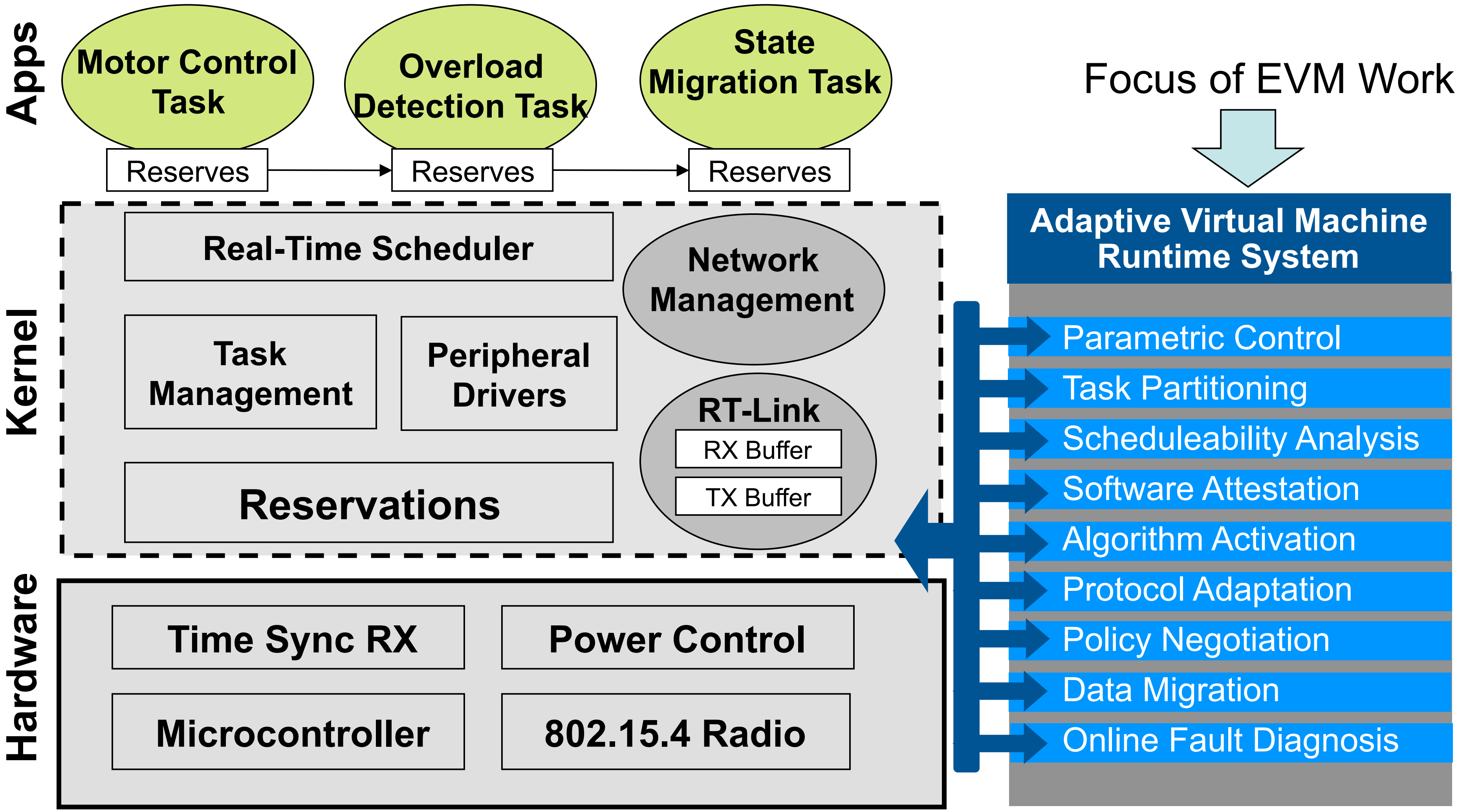

```
 01 CONSTANT  PORT                                        \ Port assignments
/bit-mask      name    bit-mask            name
1 CONSTANT MOTOR            8 CONSTANT    FAUCET
2 CONSTANT CLUTCH           16 CONSTANT   DETERGENT
4 CONSTANT PUMP    32 CONSTANT   LEVEL
                                                          \ Device control
: ON ( mask -- ) PORT C@ OR PORT C! ;
: OFF ( mask -- ) INVERT PORT C@ AND PORT C! ;
                                                          \ Timing functions
: SECONDS ( n -- ) 0 ?DO 1000 MS LOOP ;
: MINUTES ( n -- ) 60 * SECONDS ;
: TILL-FULL ( -- ) BEGIN PORT C@ LEVEL AND UNTIL ; \ Wait till level switch is on
                                                          \ Washing machine
: ADD ( mask -- ) DUP ON 10 SECONDS OFF ;
: DRAIN ( -- ) PUMP ON 3 MINUTES ;
: AGITATE ( -- ) MOTOR ON 10 MINUTES MOTOR OFF ;
: SPIN ( -- ) CLUTCH ON MOTOR ON 5 MINUTES MOTOR OFF CLUTCH OFF
        PUMP OFF ;
: FILL-TUB ( -- ) FAUCET ON TILL-FULL FAUCET OFF ;
                                                          \ Wash cycles
: WASH ( -- ) FILL-TUB DETERGENT ADD AGITATE DRAIN ;
: RINSE ( -- ) FILL-TUB AGITATE DRAIN ;

: WASHER ( -- ) WASH SPIN RINSE SPIN ;                    \ Top-level control
```

EVM Design Flow
Platform Dependent Binary



Embedded Virtual Machine
Architecture



Runtime Parametric and Programmatic Control

Wireless Control Network

A Simple Distributed Method for Control over Wireless Networks

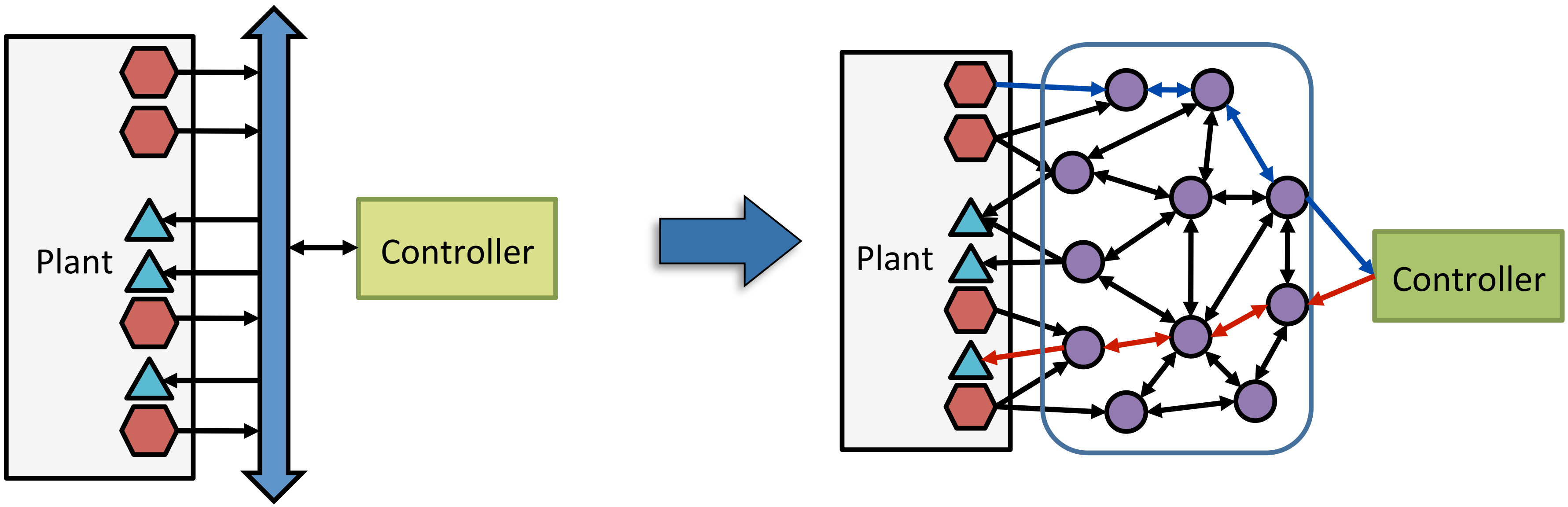
Penn Engineering

LAB Real-Time & Embedded

Goal: Use multi-hop wireless networks for closed-loop control
Move from open-loop monitoring to distributed control!

For the past 40 years control architectures have been based on wired networks

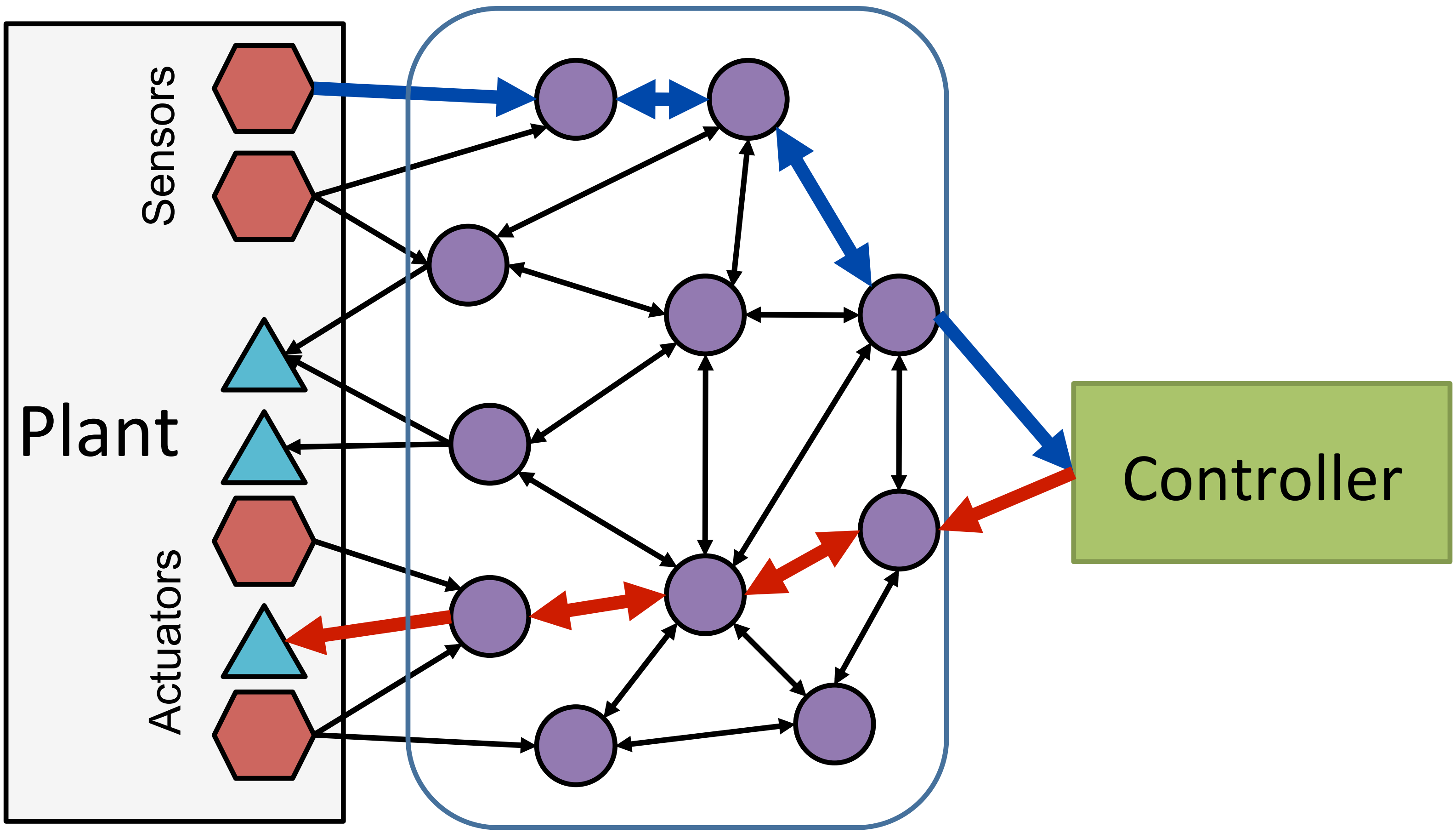
- Sensors ( ) and Actuators ( ) are installed on a plant
- Communicate with controller ( ) over a wired network



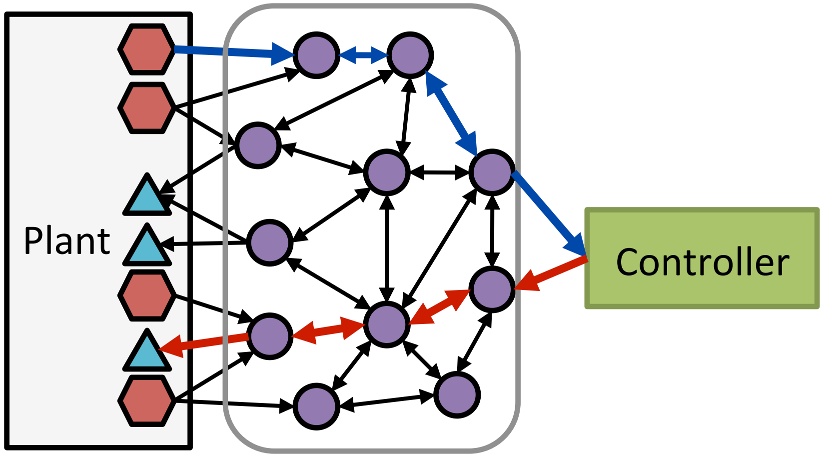
Wired Control System

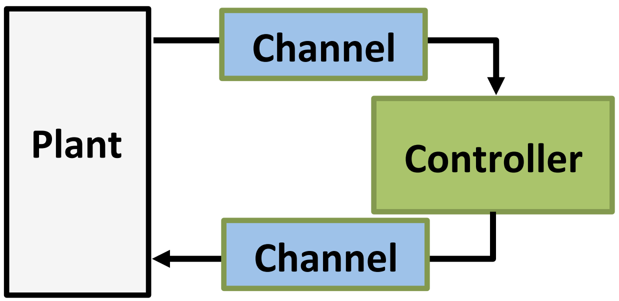
Wireless Control System

Route assignment is static



Control System's view of the Network



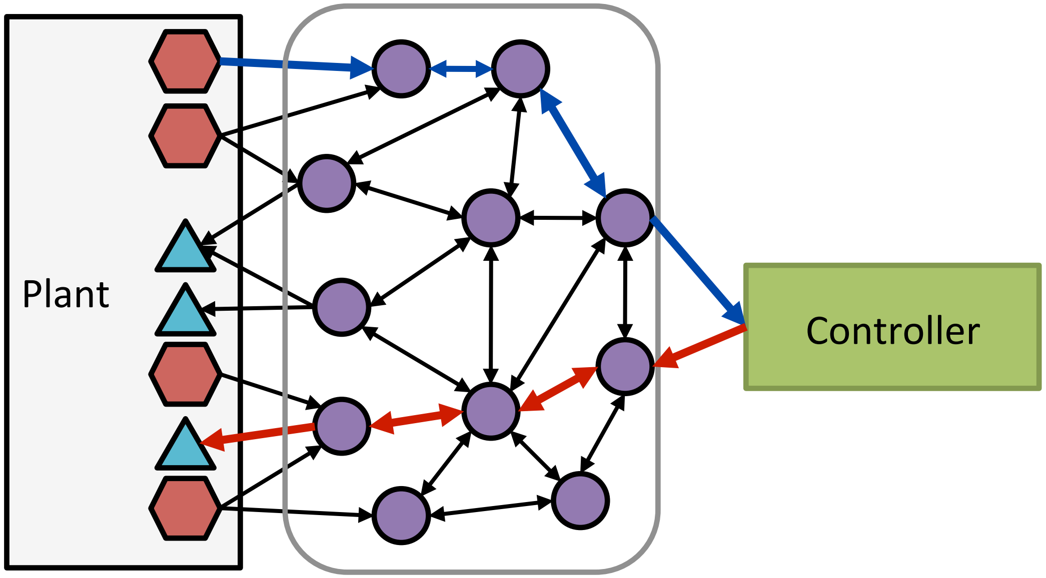


Abstracts away system design to an ideal network

Route assignment is static

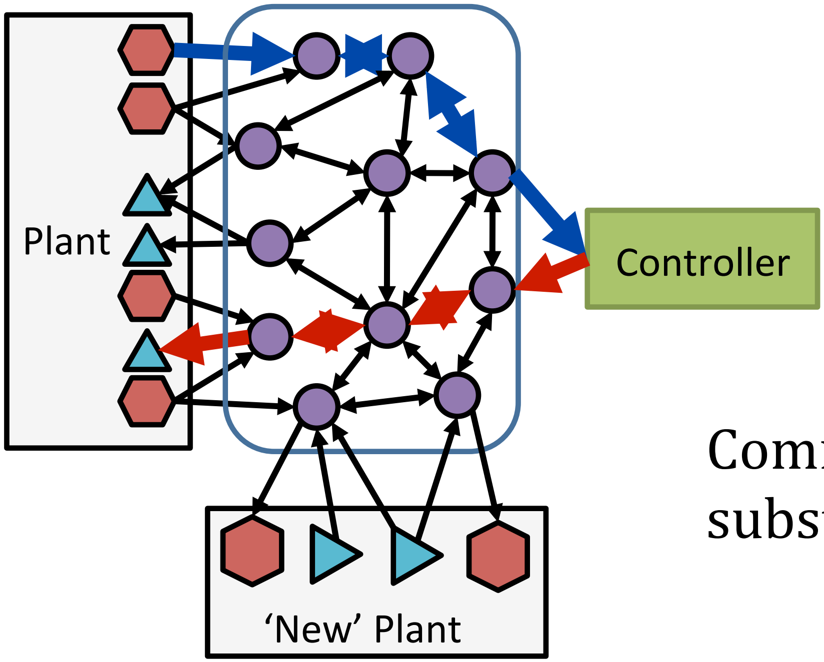
- Control problems impose strict delay requirements

Why we need Distributed Control ?



- Problem I: Changes involve global reorganization
  - Requires significant software support (e.g., EVM, Etherware)

Networks for industrial automation systems are usually shared among several control loops!



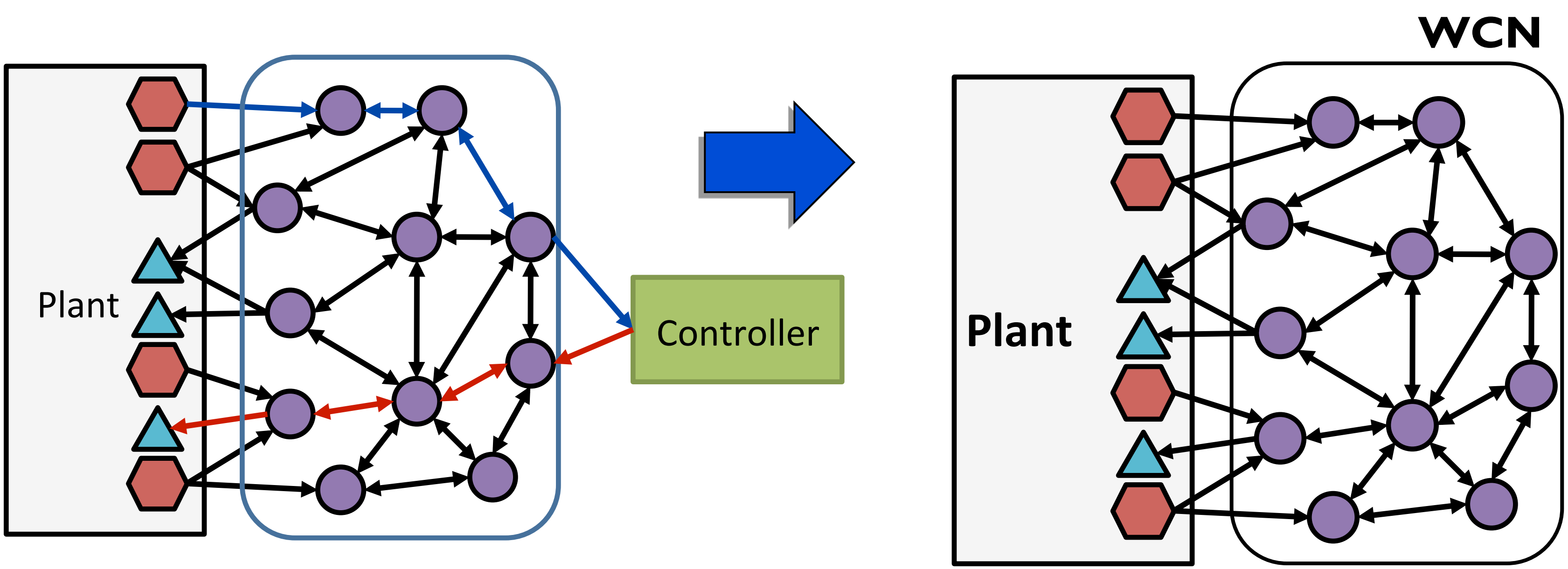
Communication substrate is shared!

- **Problem II**: New feedback loops might affect the existing loops
  - requires full schedule recalculation (problem at runtime)

It is necessary to provide a composable control scheme!

Multi-hop Control Networks: Architectures

- Sensors ( ) and Actuators ( ) are installed on a *plant*
- Communicate with controllers ( ) over a wireless network



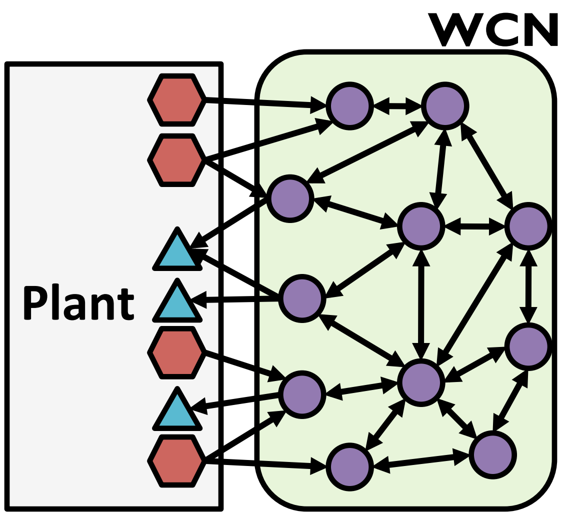
Out-of-network computation

Distributed In-network computation

Outline: Control-as-a-Service

A simple distributed method for control over wireless networks

1. Wireless Control Networks (WCNs)
2. Modeling
3. Synthesis of Optimal WCNs
4. Robustness
5. Case Study

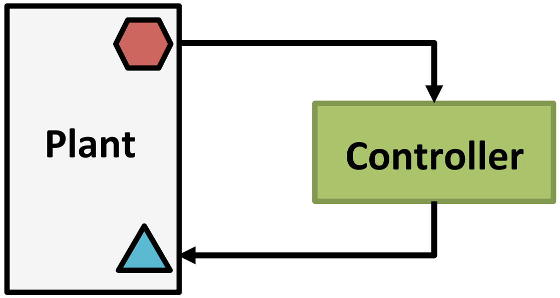


Penn Engineering

Standard Feedback Control Schemes

Linear-Time Invariant model of the plant:

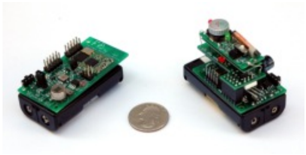
$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k] + \mathbf{B_w}\mathbf{u}_w[k]$$
$$\mathbf{y}[k] = \mathbf{C}\mathbf{x}[k],$$



Linear dynamic feedback controllers:

$$\mathbf{x_c}[k+1] = \mathbf{A_c}\mathbf{x}_c[k] + \mathbf{B_c}\mathbf{y}[k]$$
$$\mathbf{u}[k] = \mathbf{C_c}\mathbf{x}_c[k] + \mathbf{D_c}\mathbf{y}[k]$$

- **Goal**: Leverage the computational capability *in the network*
- Each node acts as a **local** linear dynamical controller
  - Resource constrained nodes → Small states
  - States of the nodes' neighbors are considered as inputs!

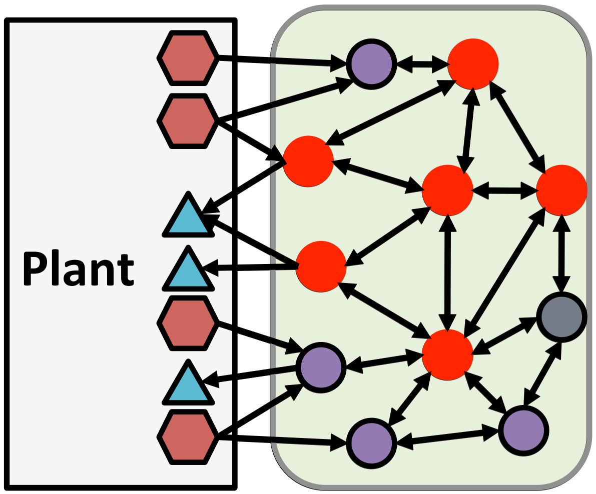


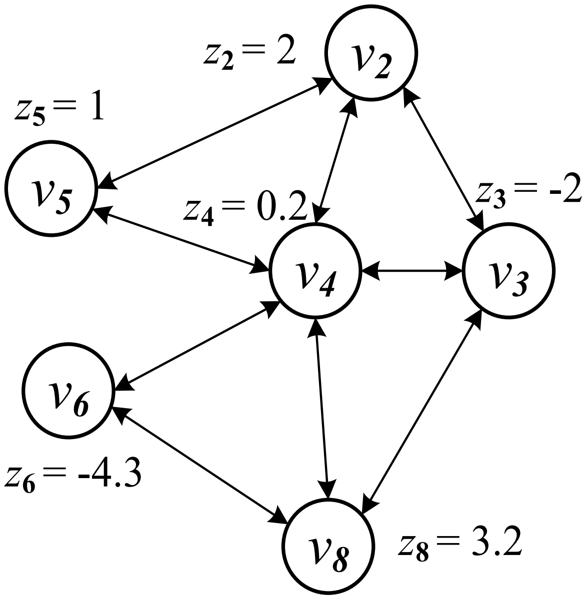
Penn Engineering

Linear Iterative Scheme

- Each node maintains its (*possibly vector*) state
  - Transmits state exactly once in each step (per frame)
  - Updates state using linear iterative strategy

- Example:



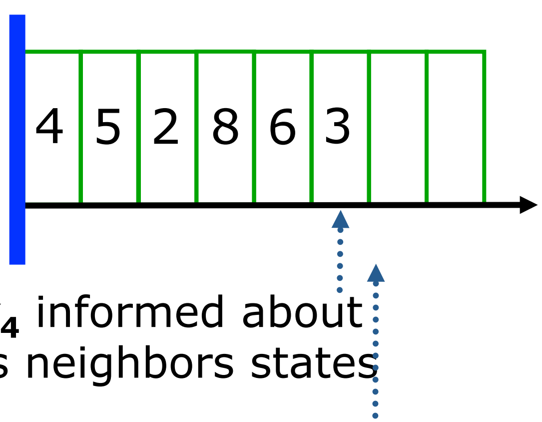


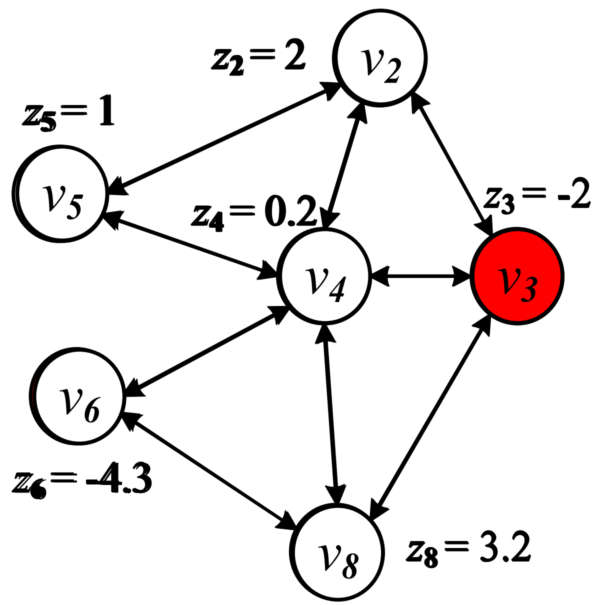
Initial state

Linear Iterative Scheme

- Each node maintains its (*possibly vector*) state
  - Transmits state exactly once in each step (per frame)
  - Updates state using linear iterative strategy

- Example:





WCN Modeling

- Each node maintains state $z_i[k]$

- Node state update procedure:

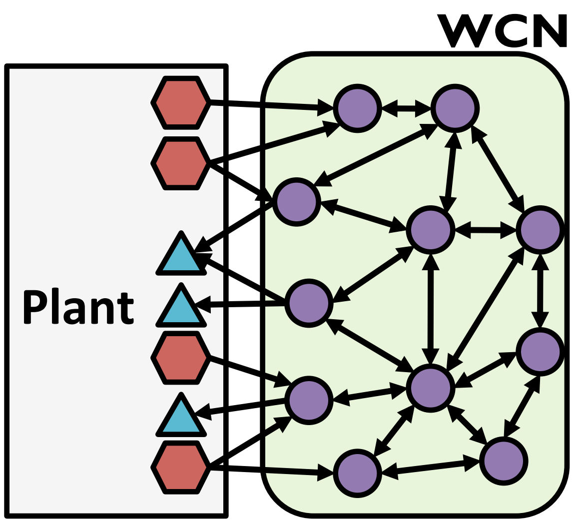
$$z_i[k+1] = w_{ii} z_i[k] + \sum_{v_j \in \mathcal{N}_{v_i}} w_{ij} z_j[k] + \sum_{s_j \in \mathcal{N}_{v_i}} h_{ij} y_j[k]$$

From neighbors **From sensors**

- Actuator update procedure:

$$u_i[k] = \sum_{j \in \mathcal{N}_{a_i}} g_{ij} z_j[k]$$

From actuator's neighbors



Each node acts as part of a dynamical compensator

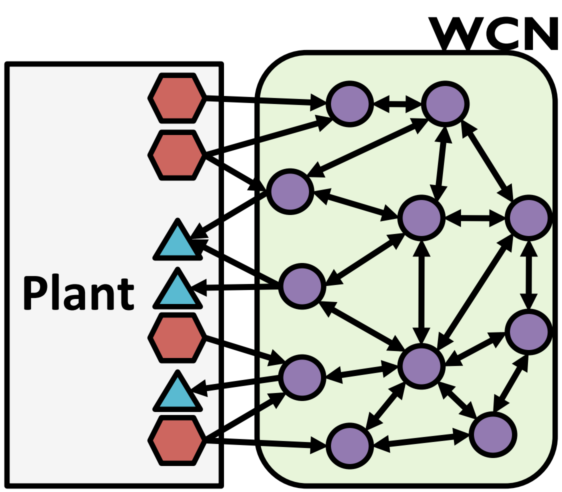
WCN Modeling

Network acts as a linear dynamical compensator

$$\mathbf{z}[k+1] = \underbrace{\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1} & w_{N2} & \cdots & w_{NN} \end{bmatrix}}_{\mathbf{W}} \mathbf{z}[k] + \underbrace{\begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1p} \\ h_{21} & h_{22} & \cdots & h_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N1} & h_{N2} & \cdots & h_{Np} \end{bmatrix}}_{\mathbf{H}} \mathbf{y}[k]$$

$$\mathbf{u}[k] = \underbrace{\begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1N} \\ g_{21} & g_{22} & \cdots & g_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ g_{m1} & g_{m2} & \cdots & g_{mN} \end{bmatrix}}_{\mathbf{G}} \mathbf{z}[k]$$

Structural constraints: Only elements corresponding to existing links (link weights) are allowed to be non-zero



WCN modeling: Closing the loop

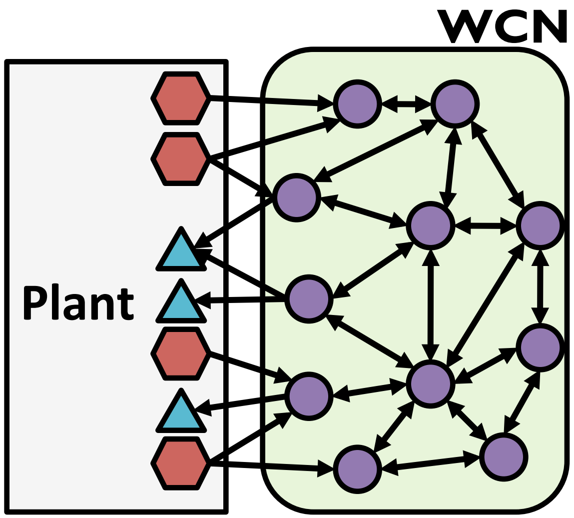
- Overall system state:

$$\hat{\mathbf{x}}[k] = \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{z}[k] \end{bmatrix} \begin{matrix} \leftarrow \text{plant} \\ \leftarrow \text{network} \end{matrix}$$

- Closed-loop system:

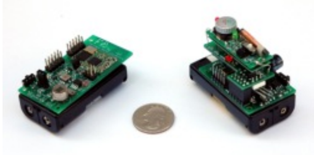
$$\hat{\mathbf{x}}[k+1] = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{BG} \\ \mathbf{HC} & \mathbf{W} \end{bmatrix}}_{\hat{\mathbf{A}}} \underbrace{\begin{bmatrix} \mathbf{x}[k] \\ \mathbf{z}[k] \end{bmatrix}}_{\hat{\mathbf{x}}[k]} + \underbrace{\begin{bmatrix} \mathbf{B}_w \\ \mathbf{0} \end{bmatrix}}_{\hat{\mathbf{B}}} \mathbf{u}_w$$

- Matrices W, G, H are structured
- Sparsity constraints imposed by topology!



Penn Engineering

Advantages: **Simple & Powerful**



- **Low overhead**
  - Easily incorporated into existing wireless networks
    (e.g., ISA100.11a or wirelessHART)
- **Simple scheduling**
  - Each node needs to transmit only once per frame
  - Static (conflict-free) schedule
- **No routing!**
  - Easily deals with multiple sensing/actuation points

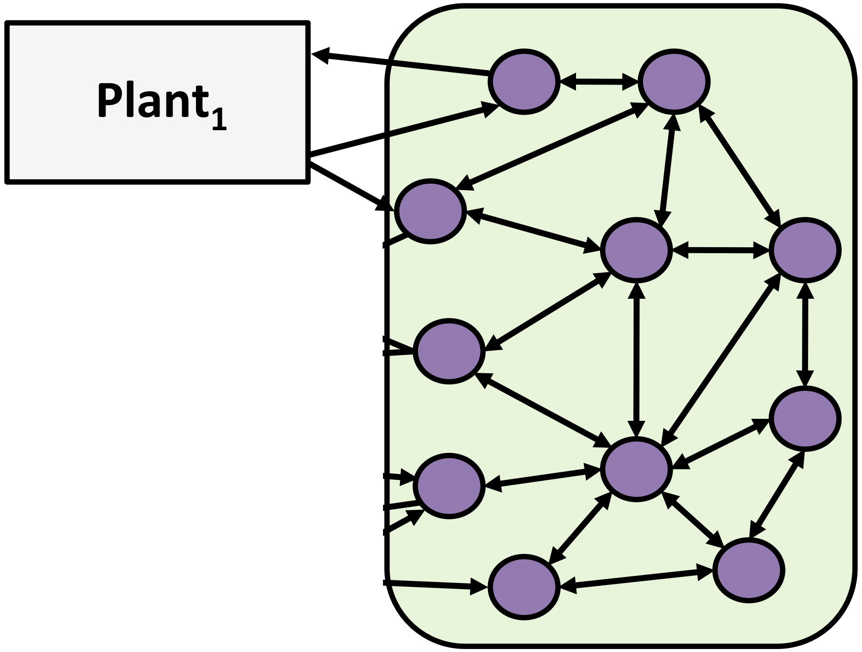
Advantages: **Compositionality**

- Adding new control loops is easy!
  - Does not require any communication schedule recalculation

- WCN configurations can be combined

Stable configuration

$$(\mathbf{W}_1, \mathbf{H}_1, \mathbf{G}_1) \in \Psi$$

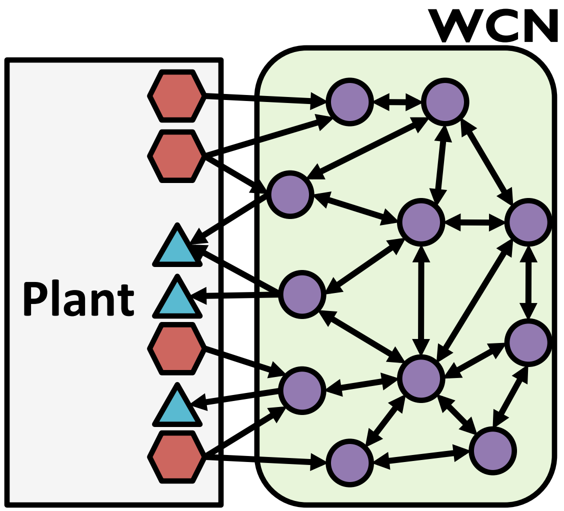


Outline

A simple distributed method for control over wireless networks

1. Wireless Control Networks (WCNs)
2. Modeling
3. **Synthesis of Optimal WCNs**
4. **Robustness**
5. Case Study

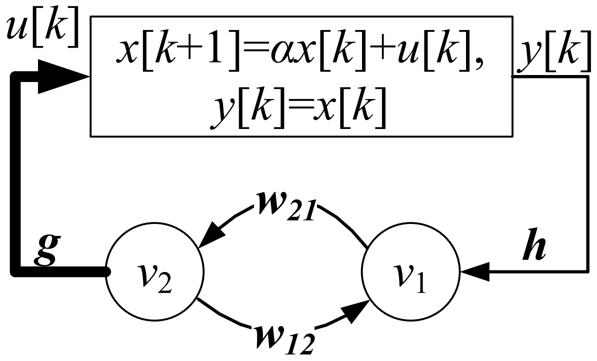
$$\hat{\mathbf{x}}[k+1] = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{BG} \\ \mathbf{HC} & \mathbf{W} \end{bmatrix}}_{\hat{\mathbf{A}}} \underbrace{\begin{bmatrix} \mathbf{x}[k] \\ \mathbf{z}[k] \end{bmatrix}}_{\hat{\mathbf{x}}[k]} + \underbrace{\begin{bmatrix} \mathbf{B}_w \\ \mathbf{0} \end{bmatrix}}_{\hat{\mathbf{B}}} \mathbf{u}_w$$



Penn Engineering

Robustness to Link Failures

- Example

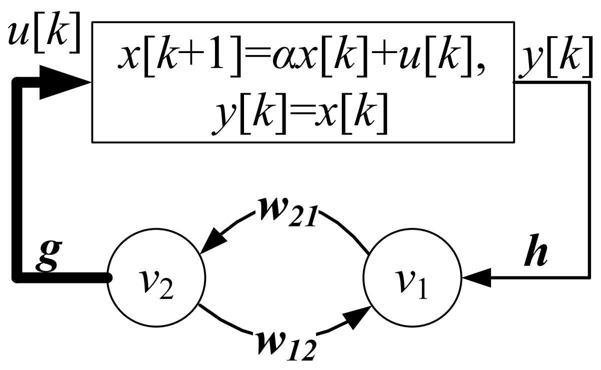


- Maximal message drop probability which guarantees MSS, α=2

	WCN (scalar state)	WCN ($\mathbb{R}^2$ state)	$oWCN$ (scalar state)	$oWCN$ ($\mathbb{R}^2$ state)
$N=2$	$p_m = 1.18\%$	$p_m = 1.30\%$	$p_m = 10.46\%$	$p_m = 17.82\%$
$N=3$	$p_m = 1.32\%$	$p_m = 1.46\%$	$p_m = 11.24\%$	$p_m = 17.88\%$
$N=4$	$p_m = 1.41\%$	$p_m = 1.54\%$	$p_m = 11.46\%$	$p_m = 17.88\%$
		$oWCN$ ($\mathbb{R}^3$ state)	$oWCN$ ($\mathbb{R}^4$ state)	$oWCN$ ($\mathbb{R}^5$ state)
$N=2$		$p_m = 20.40\%$	$p_m = 20.48\%$	$p_m = 20.64\%$

Robustness to Link Failures

- Example – WCN with observer style updates



For α=2, maximal message drop probability which guarantees MSS

$$p_{max} \approx 21\% < 25\%$$

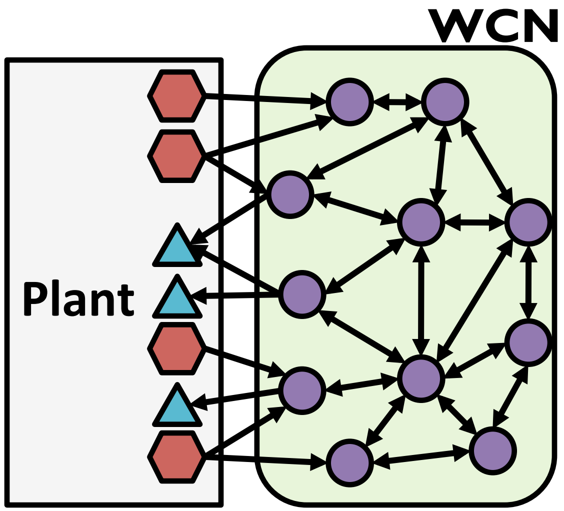
Approaching theoretical limit for robustness with centralized controllers!

Penn Engineering

Outline

A simple distributed method for control over wireless networks

1. Wireless Control Networks (WCNs)
2. Modeling
3. Synthesis of Optimal WCNs
4. Robustness
5. **Case Study**

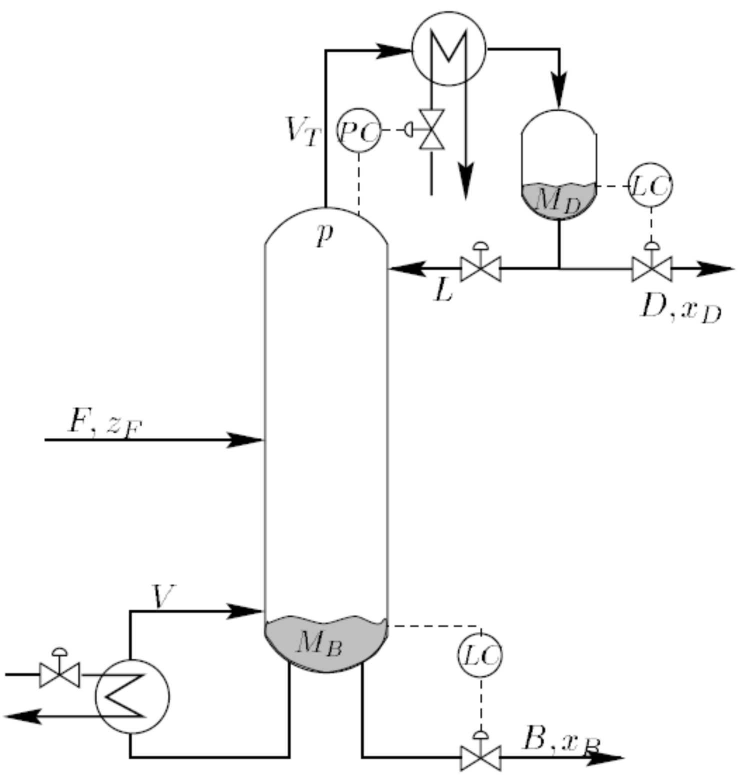


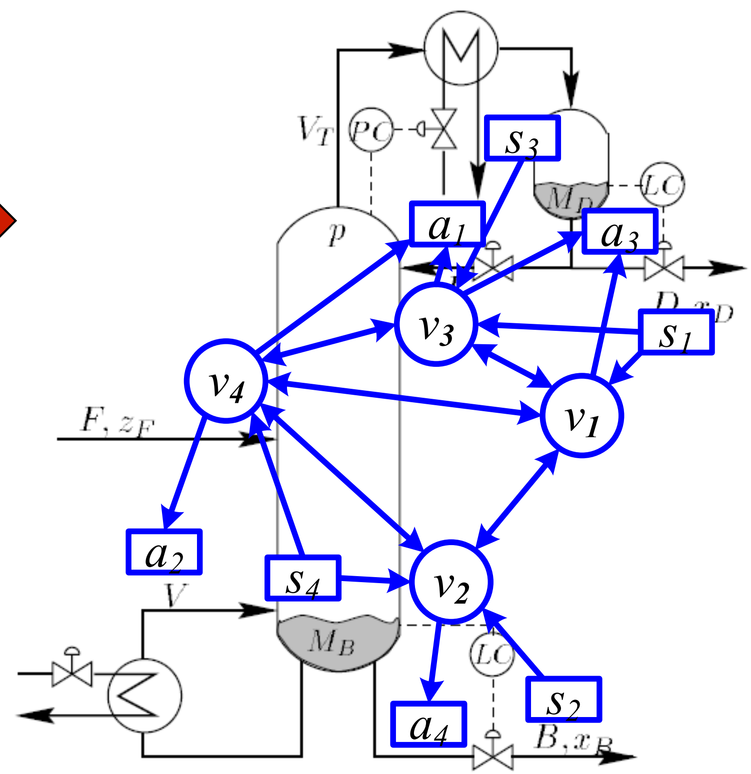
Penn Engineering

WCN demo: Distillation column process control

- Distillation column control
  - Plant **continuous-time** model contains 8 states, 4 inputs, 4 outputs

- Distillation column structure





System configuration

WCN demo: Distillation column process control

- Distillation column control
  - Plant model contains 8 states, 4 inputs, 4 outputs
- WCN contains 4 nodes

Stable configuration (obtained after plant discretization):

node→node

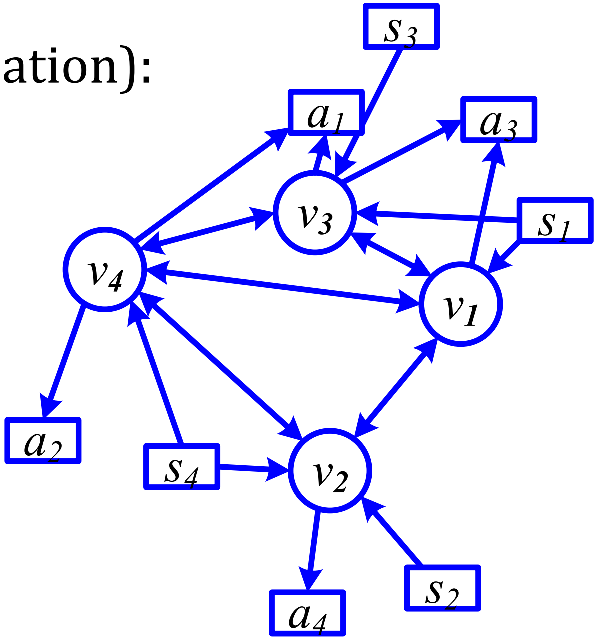
$$\mathbf{W} = \begin{bmatrix} -0.470 & 0.339 & -0.260 & -0.390 \\ -1.117 & -0.145 & 0 & -0.269 \\ 0.0514 & 0 & -0.703 & 0.600 \\ 0.854 & 0.277 & -0.086 & -0.112 \end{bmatrix}$$

sensor→node

$$\mathbf{H} = \begin{bmatrix} 1.260 & 0 & 0 & 0 \\ 0 & 0.104 & 0 & 0.075 \\ 0 & 0 & 0.421 & 0 \\ 0 & 0 & 0 & -0.034 \end{bmatrix}$$

node→actuator

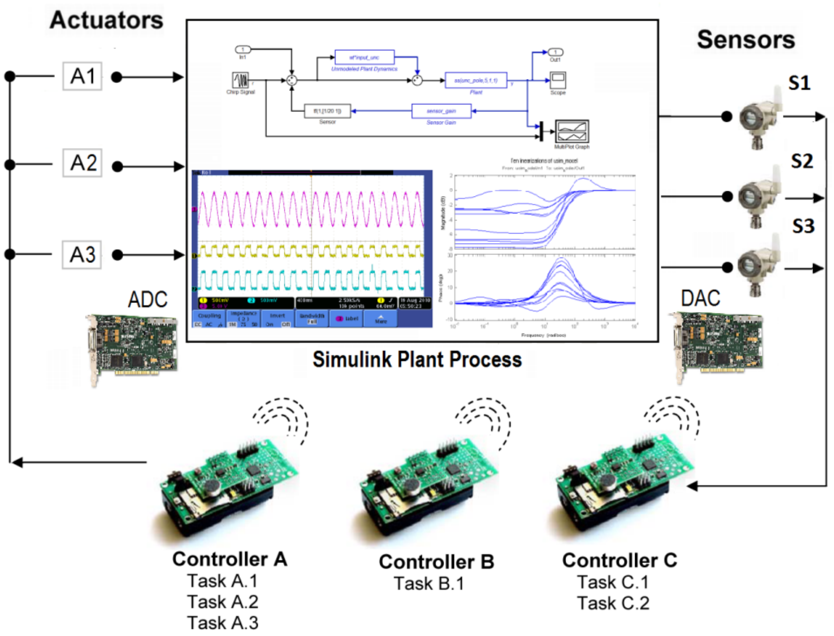
$$\mathbf{G} = \begin{bmatrix} 0 & 0 & -0.226 & -0.459 \\ 0 & 0 & 0 & 0.102 \\ 0.120 & 0 & 1.072 & 0 \\ 0 & 2.549 & 0 & 0 \end{bmatrix}$$



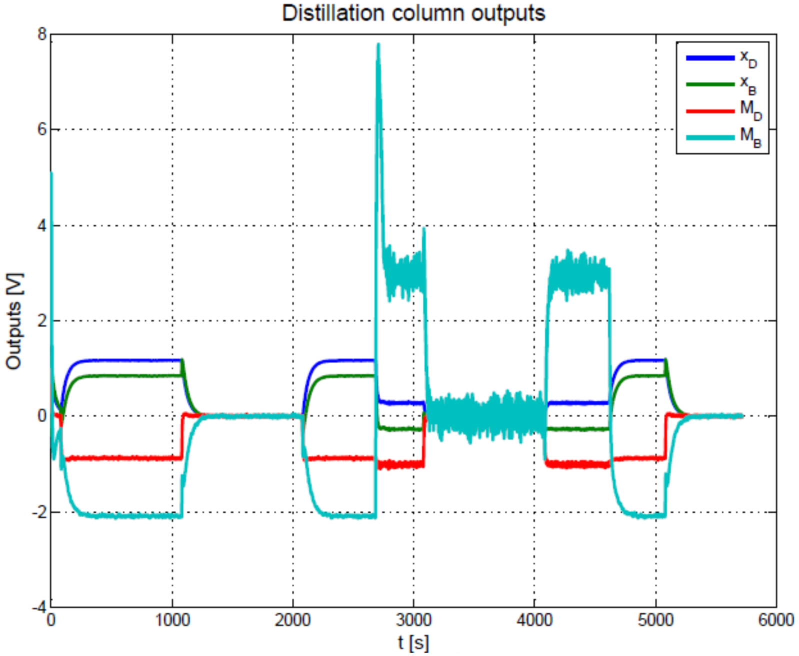
Network topology

WCN demo: Distillation column process control

Process-in-the-loop test-bed



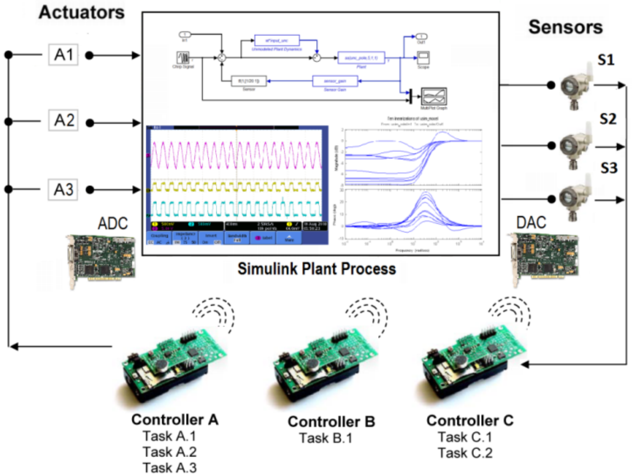
Scenario I: v_1 turned OFF/ON



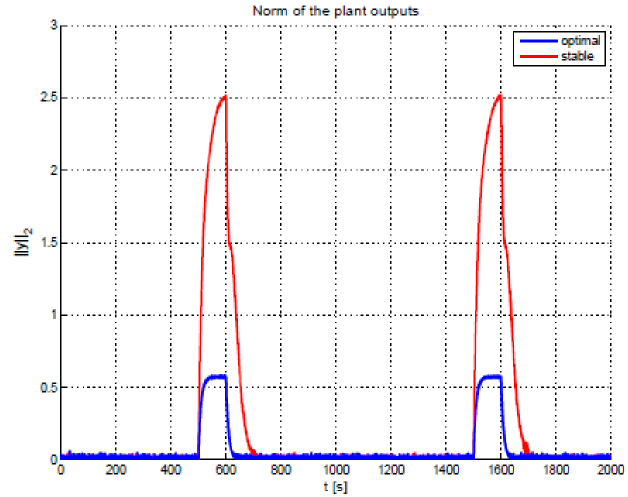
Penn Engineering

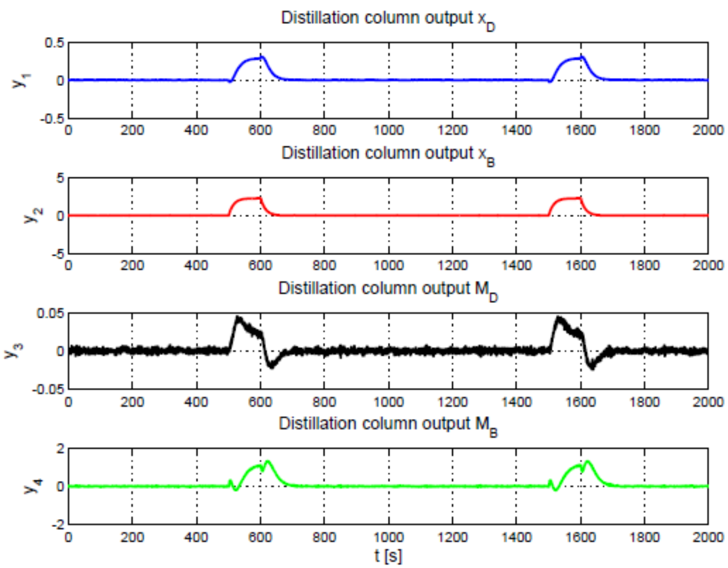
WCN demo: Distillation column process control

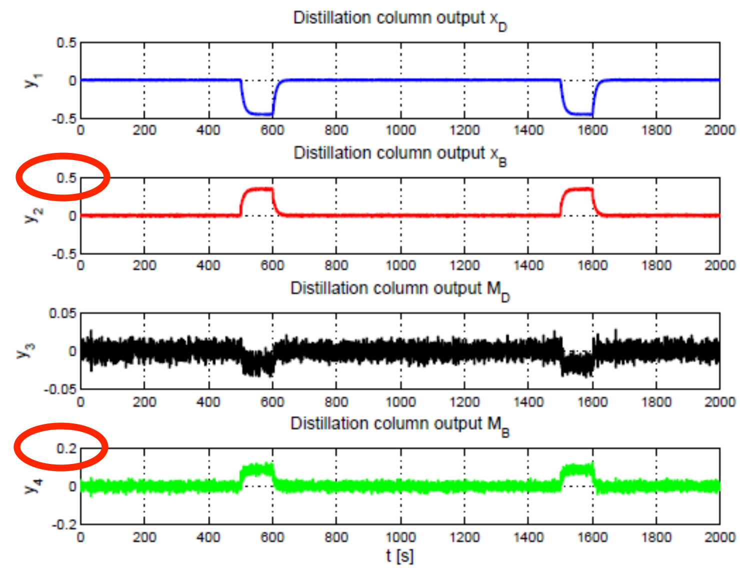
Process-in-the-loop test-bed



Scenario II: Optimal control





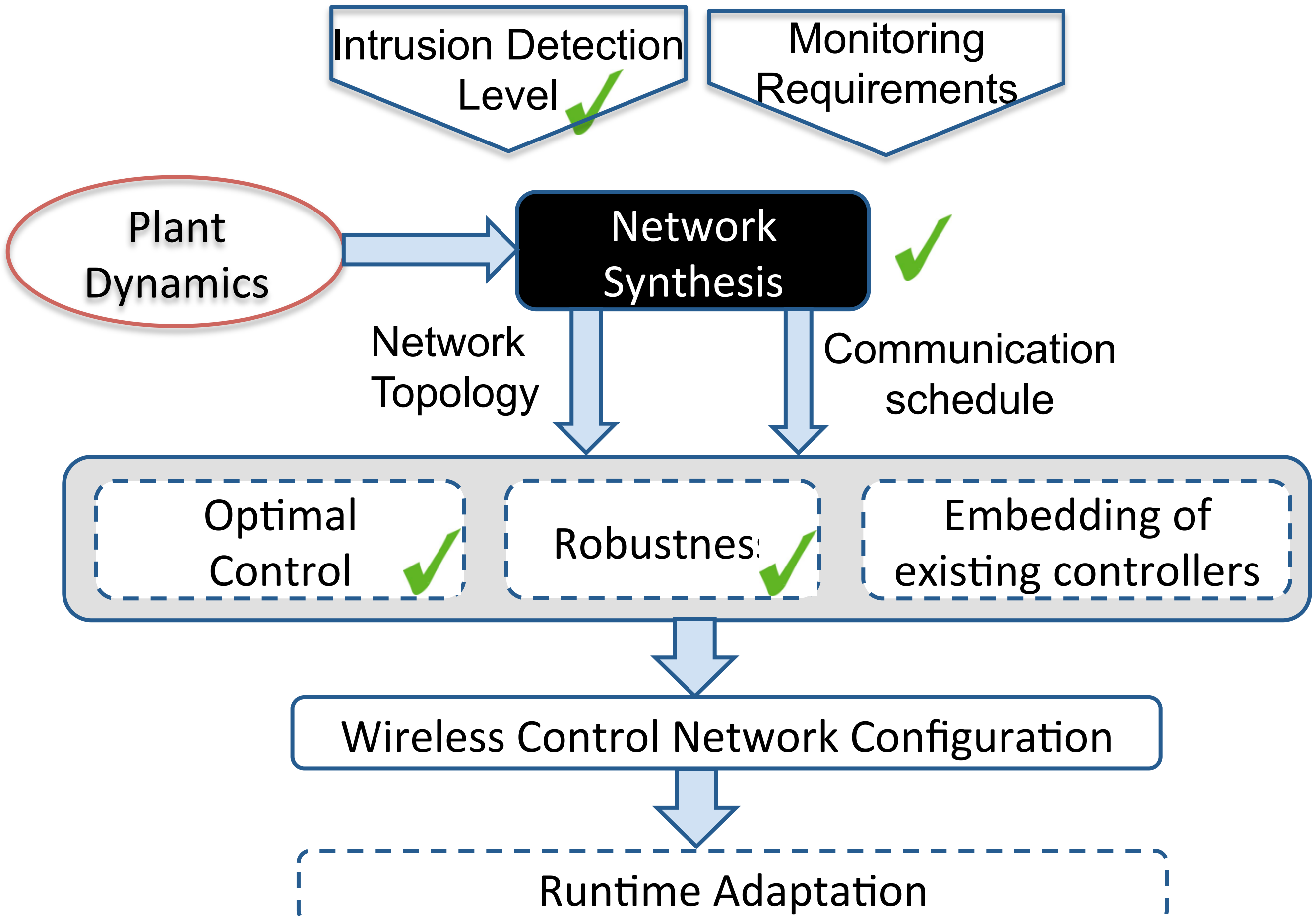


Penn Engineeri

Winners of the Honeywell Users Group 2011 Industrial Wireless Competition

PRECISE Ph.D. Student: Miroslav Pajic. **Embedded Sys MS Student:** Harsh Jain

Controller-as-a-Service with Wireless Control Networks



Thank You

PRECISE

PENN RESEARCH IN EMBEDDED COMPUTING AND INTEGRATED SYSTEMS ENGINEERING

Penn
UNIVERSITY of PENNSYLVANIA

UNIVERSITY OF
WATERLOO

Demo