Robust Learning Control with Application to HVAC Systems

National Science Foundation

Colorado State University

Electrical & Computer Engineering

Computer Science

Mechanical Engineering

Project Investigators:

Dr. Charles Anderson, CS
Dr. Douglas Hittle, ME
Dr. Peter Young, ECE

Graduate Students

- Michael Anderson
- Christopher Delnero
- David Hodgson
- Matthew Kretchmar
- Jilin Tu

Outline

- Introduction
- Experimental HVAC Platform
- PI Plus Neural Network Controller
- MIMO Robust Controller Design
- Robust Reinforcement Learning Controller
- Results and Discussion
- Concluding Remarks

Motivation

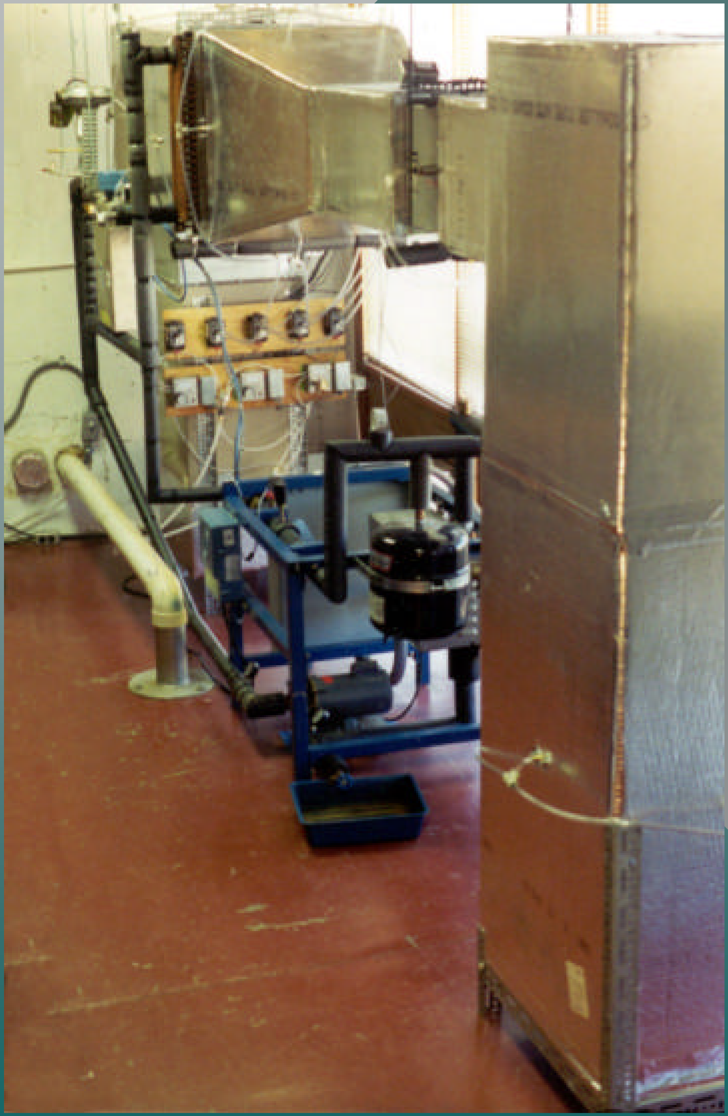
- From the Mechanical Engineering perspective, how can neural networks be applied to highly non-linear, time varying HVAC systems?
- From the Computer Science point of view, how can we train neural networks with reinforcement learning while guaranteeing stability?
- From the Electrical and Computer Engineering view point, how can neural networks be used with robust control systems to improve performance?

Interdisciplinary !

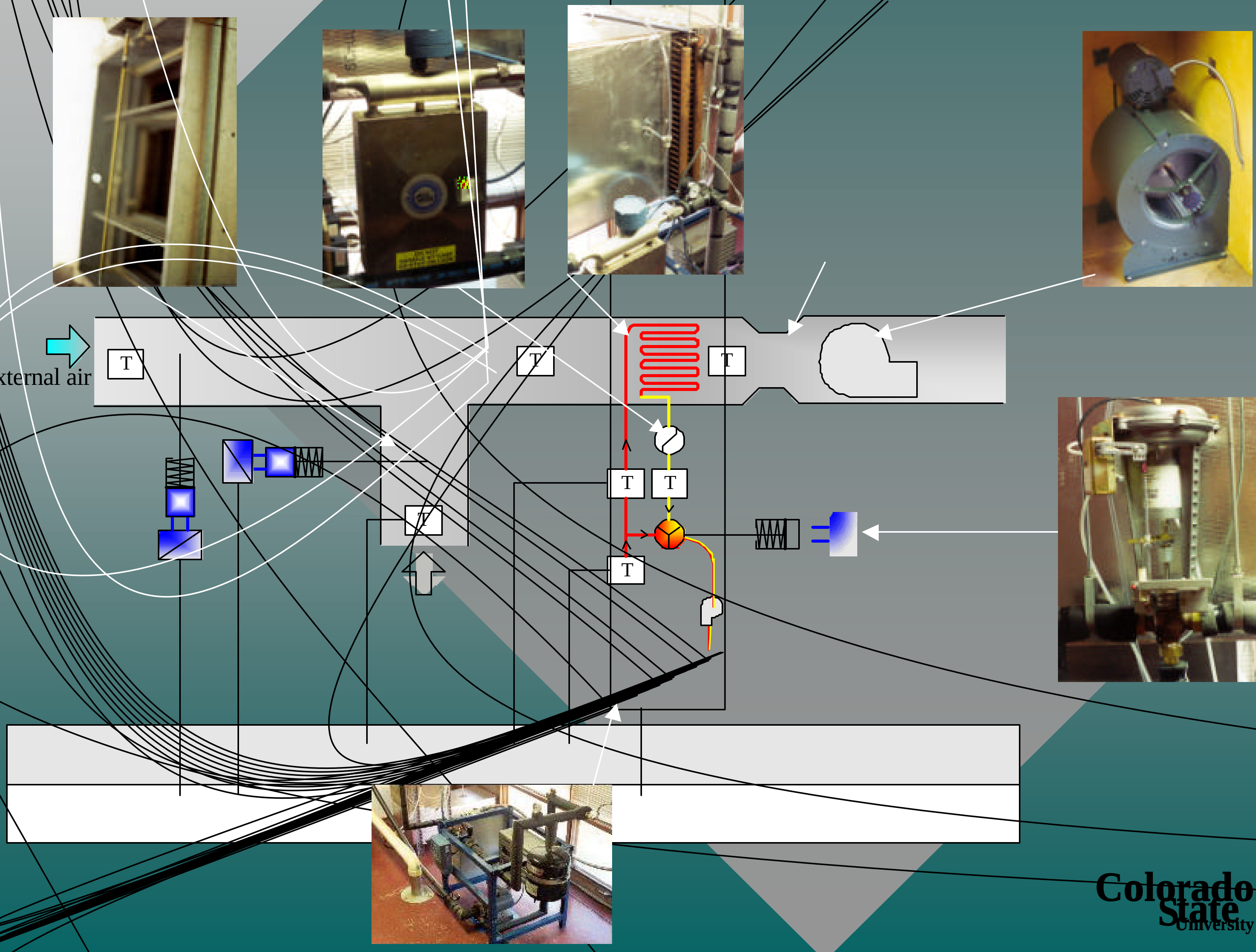
Introduction

- Characteristics of Typical HVAC Systems
  - Energy Transfer via Heating/Cooling Coils
  - Air flow Regulation to Maintain Static Air Pressure
  - Central Water Supply Servicing Multiple Units
- Current HVAC Systems Perform Poorly
  - Complex Nonlinear Time-Varying System
  - Highly Uncertain System Dynamics
  - Interaction of Controlled Variables
  - Controlled via Multiple SISO PID Control Loops

Experimental HVAC System

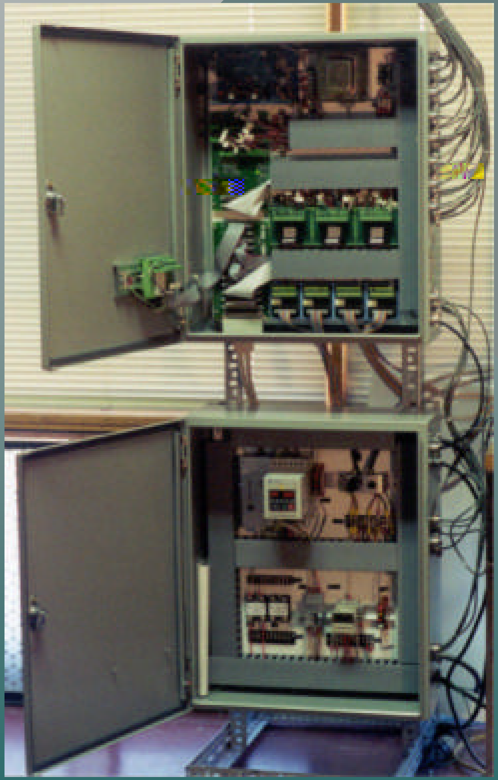


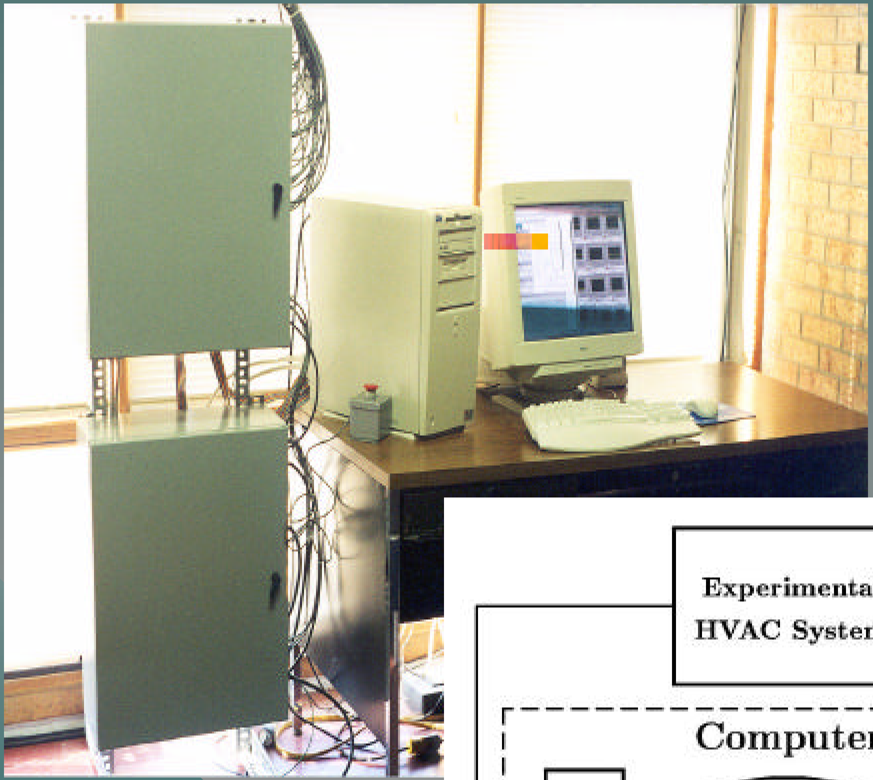
- Simple HVAC System
- Counter Flow Hot Water to Air Heat-Exchanger
- Variable Air Volume
- Mixing Box
- Electric Hot Water Heater
- Controlled Variables:
  - Discharge Air Temperature
  - Mixed Air Temperature
  - Air Flow Rate
  - Hot Water Temperature

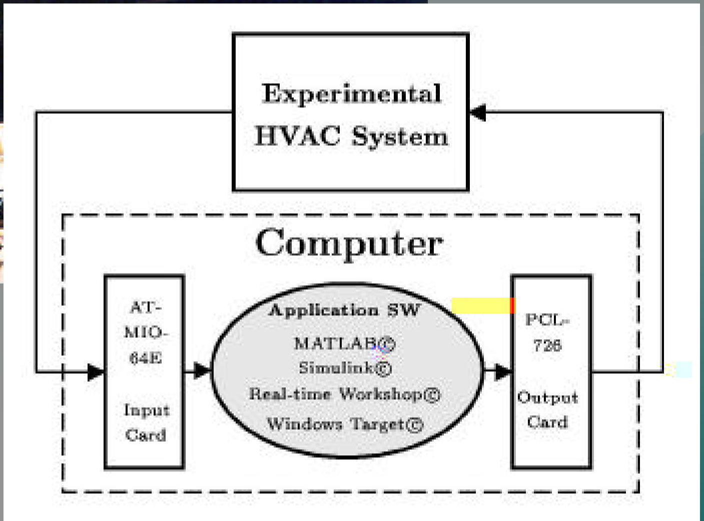


Colorado State University

PC/MATLAB Based Control System







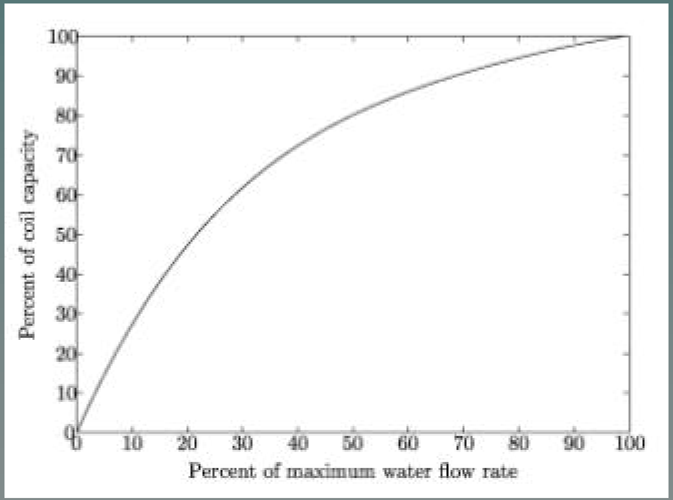
Colorado State University

PI Plus Neural Network Controller

Colorado State University

PI Controller Design

- Nonlinear System
  - Tune at High Gain State
- Parameters Controlled:
  - Water Supply Temperature
  - Air Flow Rate
  - Input Air Temperature
  - Discharge Air Temperature



Heating Coil capacity vrs. water flow rate

Colorado State University

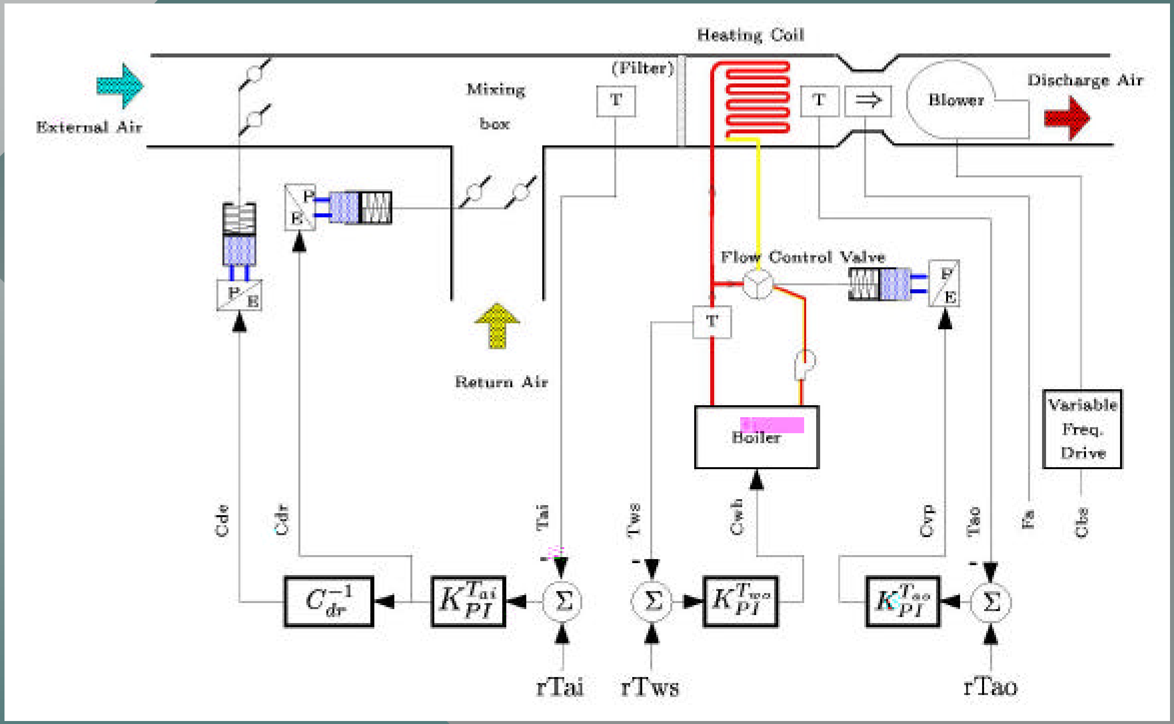
PI Control Algorithm

$$O_\tau = K_p e_\tau + K_i \sum_{j=0}^{\tau} e_j \Delta t$$

$$O_{\tau-1} = K_p e_{\tau-1} + K_i \sum_{j=0}^{\tau-1} e_j \Delta t$$

$$O_\tau = O_{\tau-1} + K_p \left(e_\tau - e_{\tau-1} \right) + K_i e_\tau \Delta t$$

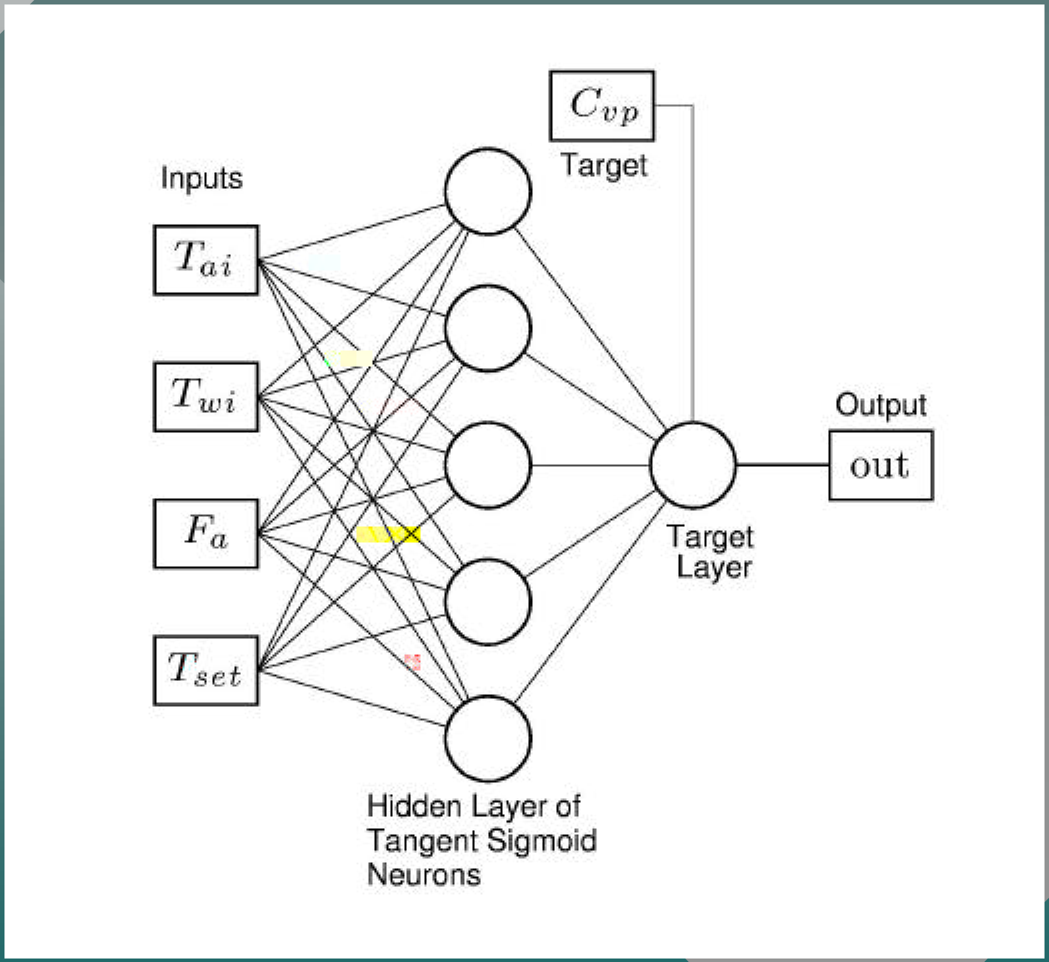
Reference PI Controller



PI Controller	Proportional Gain(K_p)	Integral Gain(K_i)
K_{Tai}	0.20	0.010
K_{Tao}	0.24	0.025
K_{Tao}	1.80	0.015

Colorado State University

Neural Network



Colorado State University

Training

- Back Propagation on:
  - Model Data (Steady State)
    - Curve fit to measured flow vs. control signal
    - Effectiveness heat exchanger model based on physical properties of the coil adjusted based on experiment
  - Experimental Data
    - Wait for the PI controller to achieve steady state

PI Control Plus Neural Net

$$O_\tau = K_p e_\tau + K_i \sum_{j=0}^{\tau} e_j \Delta t$$

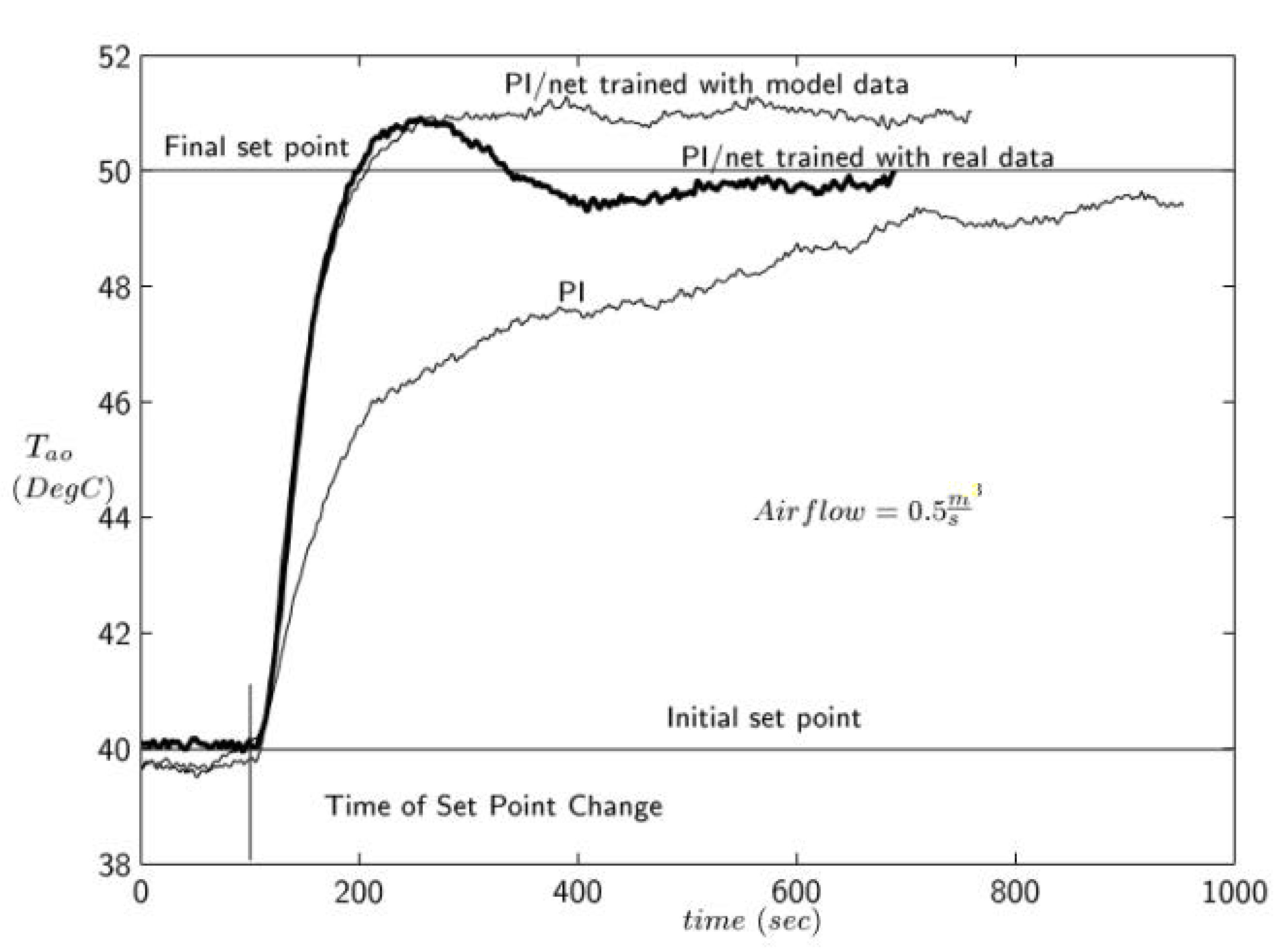
$$O_{\tau-1} = K_p e_{\tau-1} + K_i \sum_{j=0}^{\tau-1} e_j \Delta t$$

$$O_\tau = O_{\tau-1} + K_p \left(e_\tau - e_{\tau-1}\right) + K_i e_\tau \Delta t$$

$$O_\tau = NN + K_p e_\tau + K_i e_\tau \Delta t$$

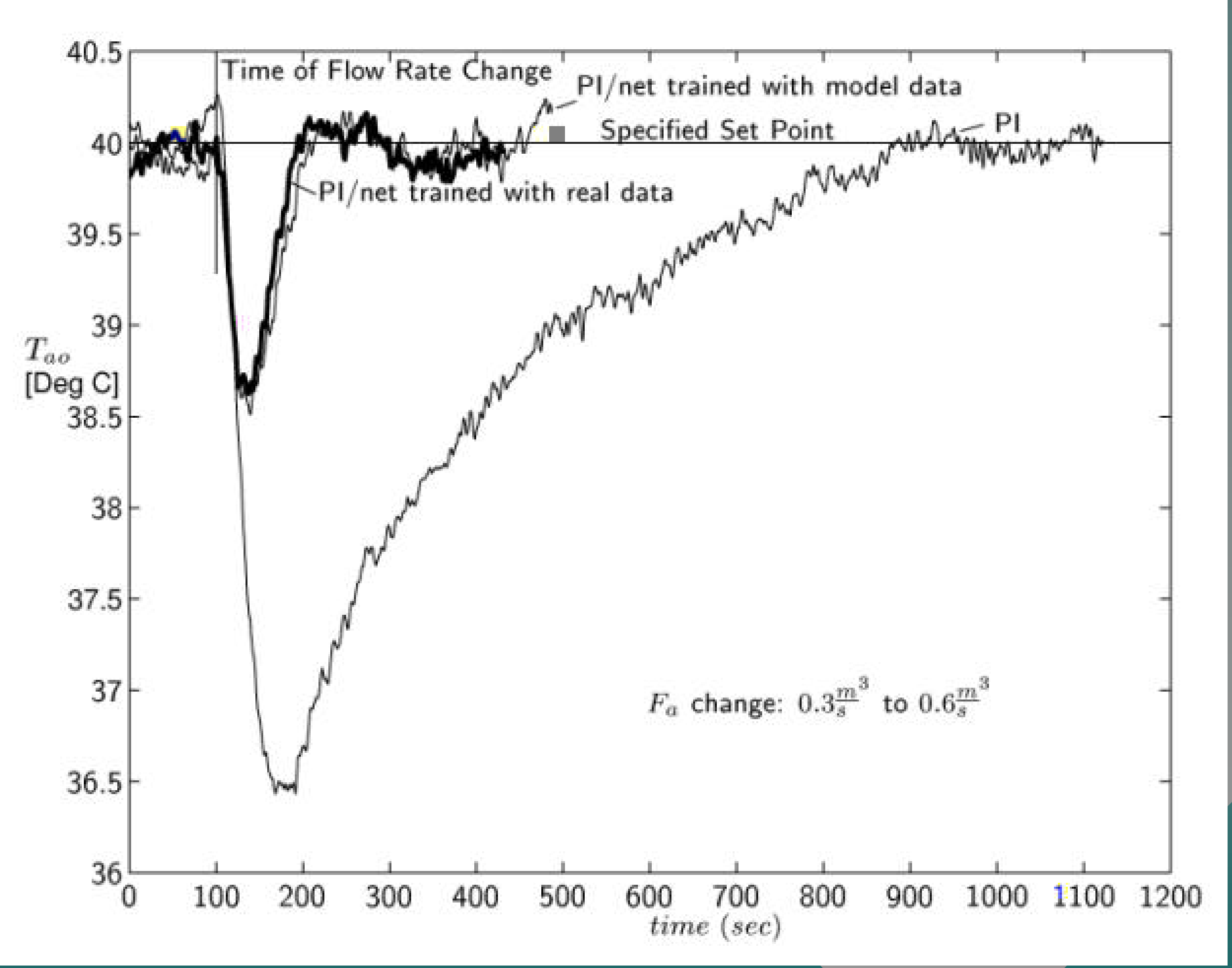
Colorado State University

Response to Set Point Change



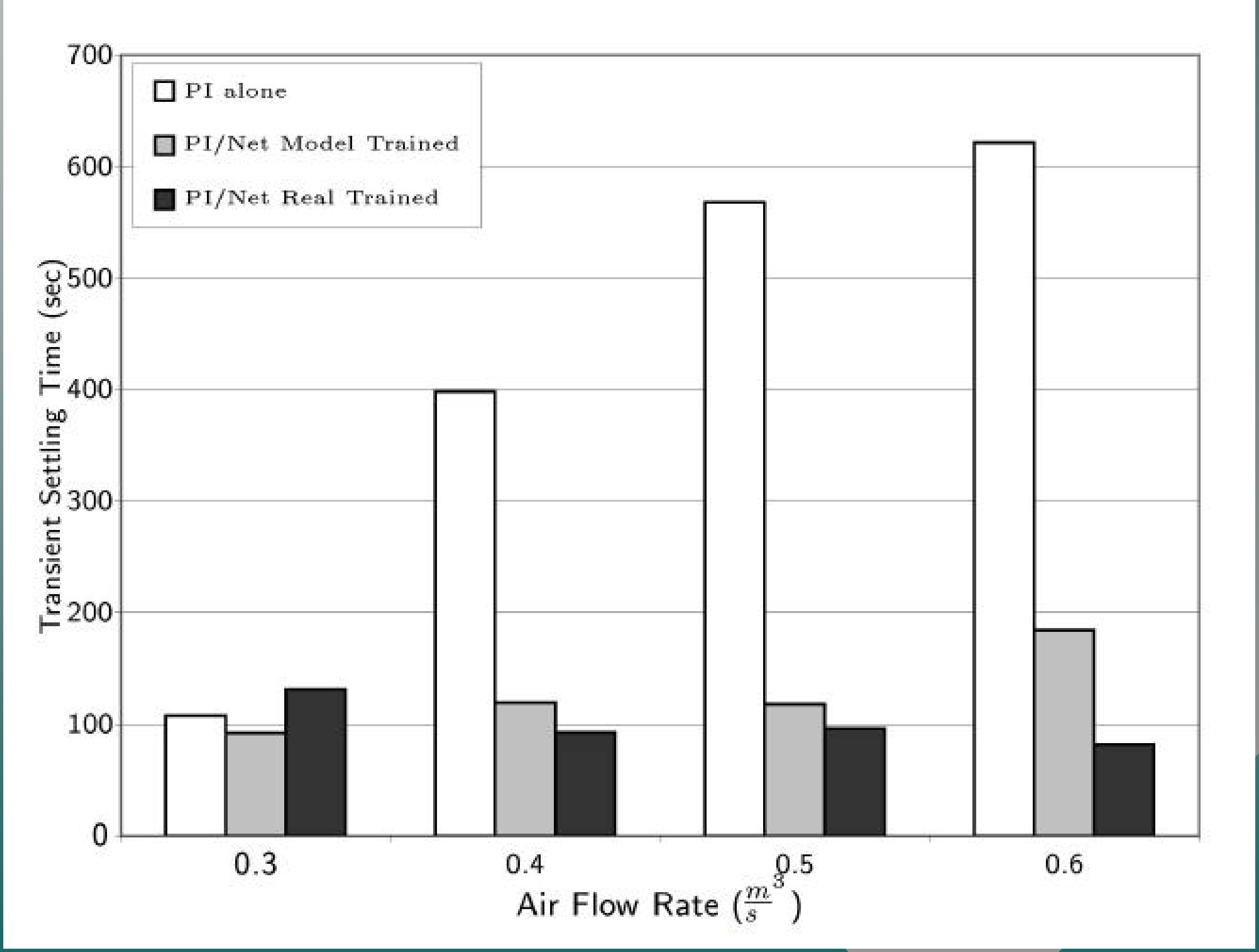
Colorado State University

Disturbance Rejection



Colorado State University

Settling Times



Colorado State University

Advantages

- Improved performance compared to PI alone
- Simple, easy to train neural network
- The combined PI/NN controller is comparatively easy to understand
- Can be implemented in the near term
- MISO Control

Multi-input Multi-output (MIMO) Robust Control of HVAC Systems

Colorado State University

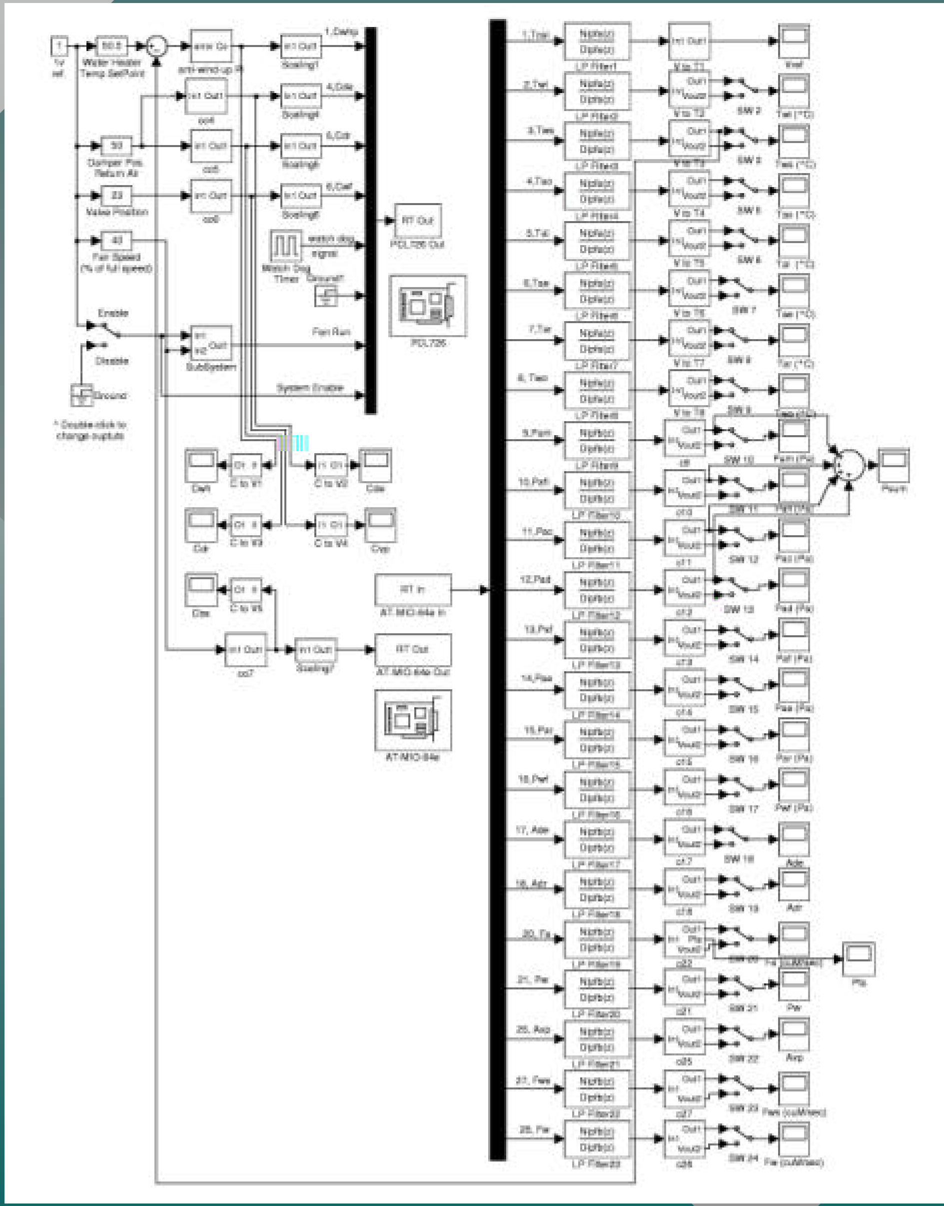
Previous Work - Controllers

- Proportional-Integral-Derivative (PID)
  - Low Gains, Tedious Tuning, Poor Performance
- Optimal
  - Decoupling, Disturbance Rejection
  - Lack of Robustness
- Adaptive
  - Potential to Adapt, Eliminating Need for Tuning
  - Stability Issues
- Robust
  - Potential Loss in Performance
  - Limited to SISO Implementation

Overview

- MIMO Robust Control for HVAC
- Matlab/Simulink Modeling and Design
- True MIMO Analysis and Design
  - Improved System Performance
  - Coordinated Action for Independent Control
- Many Potential System/Controller Architectures
- MIMO Robust Stability and Performance
- Experimental Implementation and Testing

Simulink Interface Model



Colorado State University

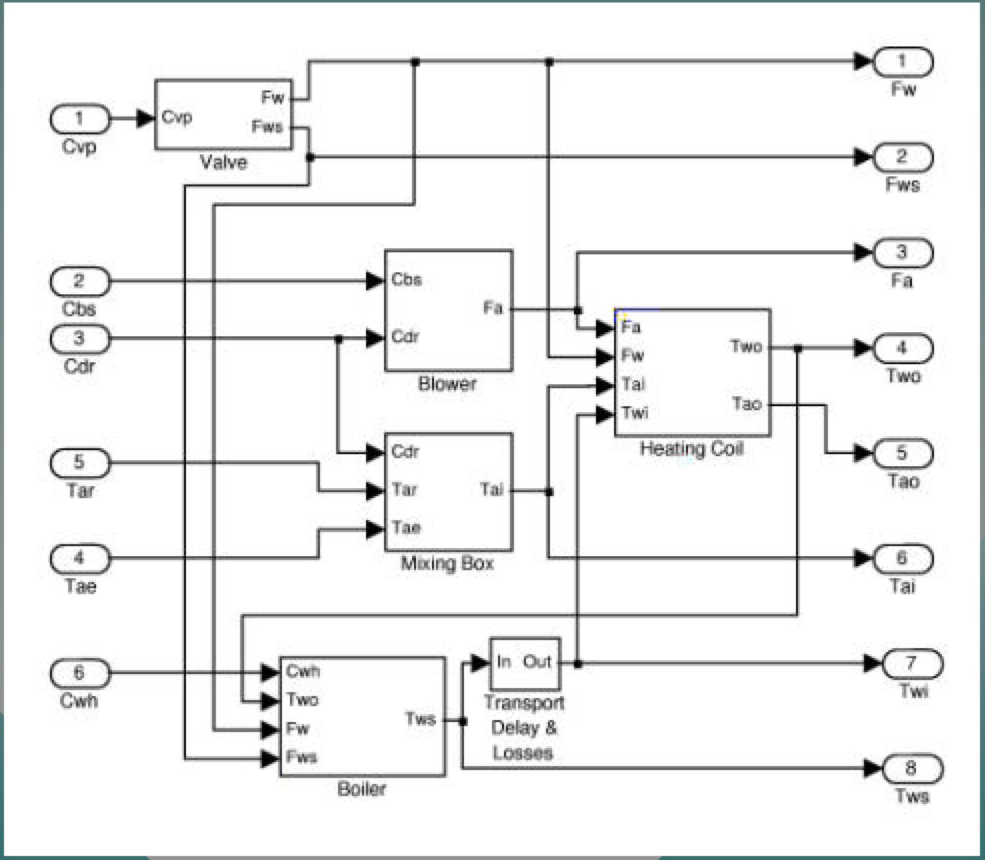
Heating Coil

Mixing box

Valve

HVAC System Model

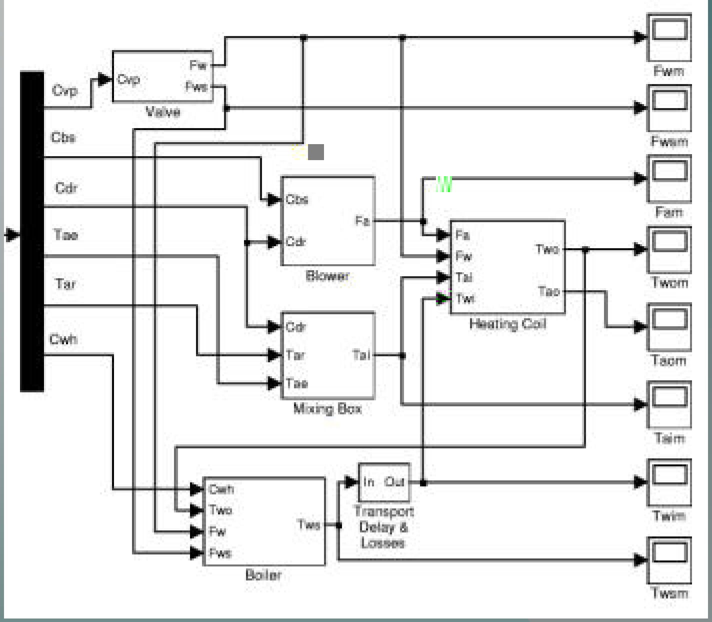
- Dynamic Model for:
  - Controller Design
  - Simulation Testing
- Nonlinear Subsystems
  - Linearization for Design
- Combination of:
  - 1st Principles
  - Data fitting



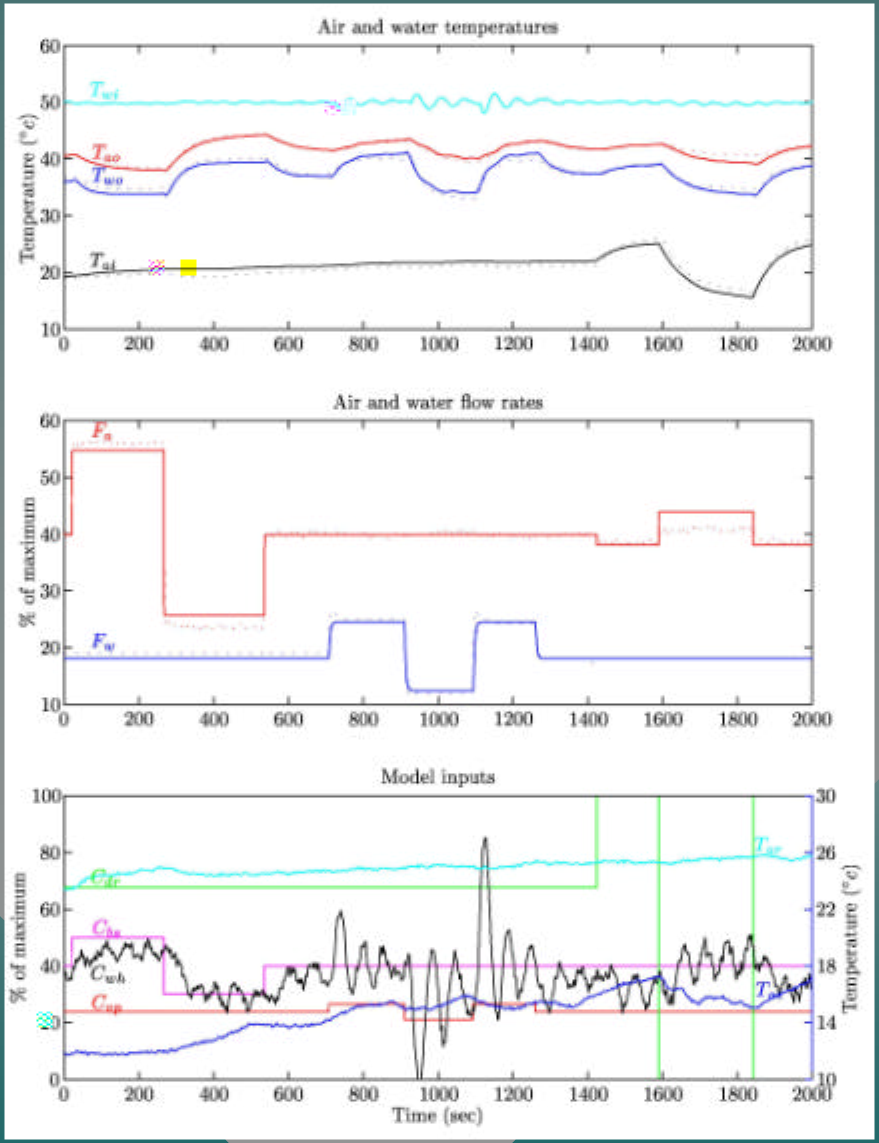
Colorado State University

HVAC System Model

Model of Experimental System



- Dynamic Model for:
  - Controller Design
  - Simulation Testing
- Nonlinear Subsystems
  - Linearization for Design
- 1st Principles and Data fitting



Model and Experimental System Outputs

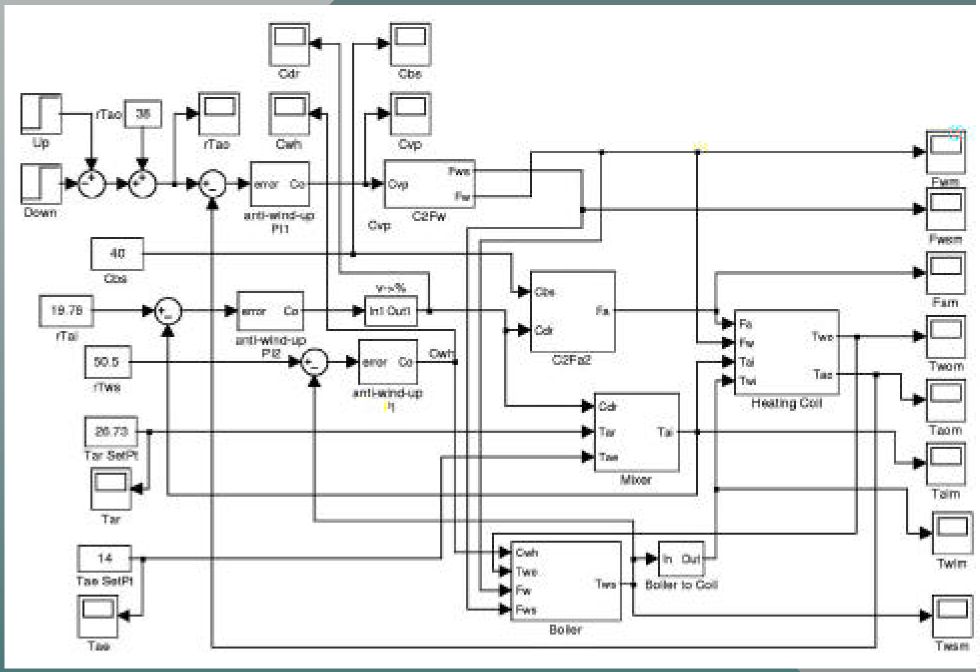
Colorado State University

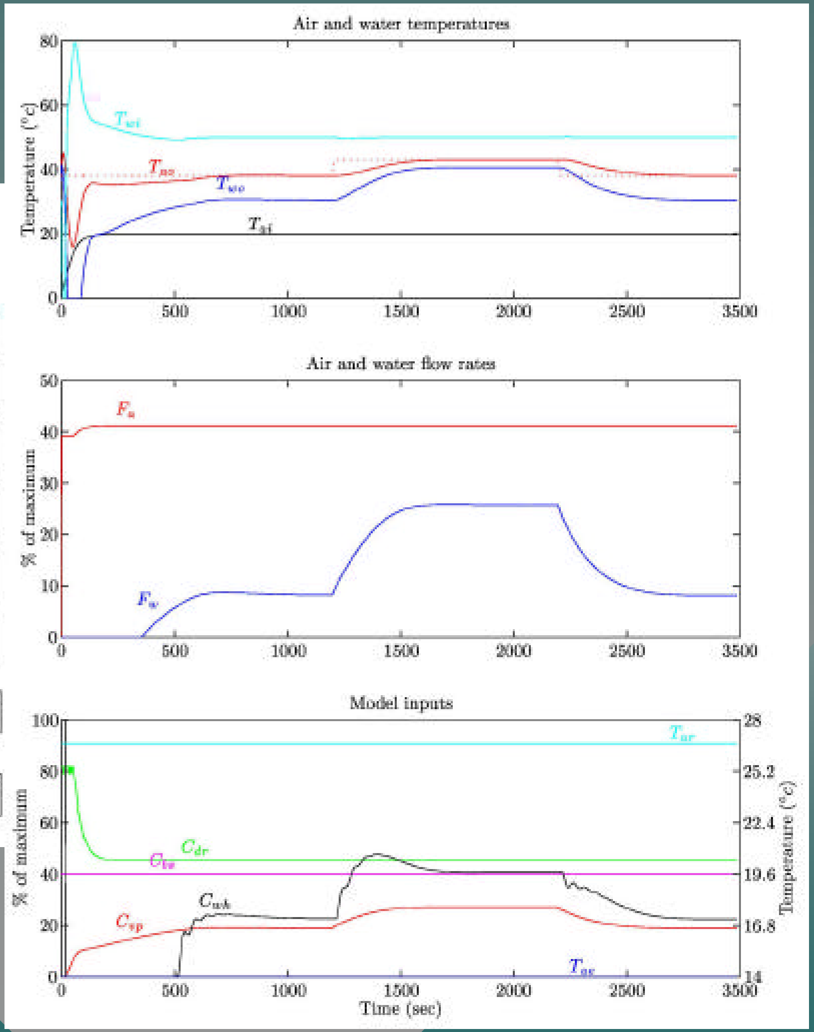
Controller Design

- Basic Design Goals:
  - MIMO Stability and Robustness
  - Independent Control of Key System Variables
  - Discharge Air Temperature and Flow Rate
- Reference Conventional PI Controller
- MIMO Robust Controllers:
  - "Minimal" (3x6) ~ T_{WS} Externally Controlled
  - "Constrained" (4x7) ~ T_{WS} Integrated
  - "Full" (4x7) MIMO

Colorado State University

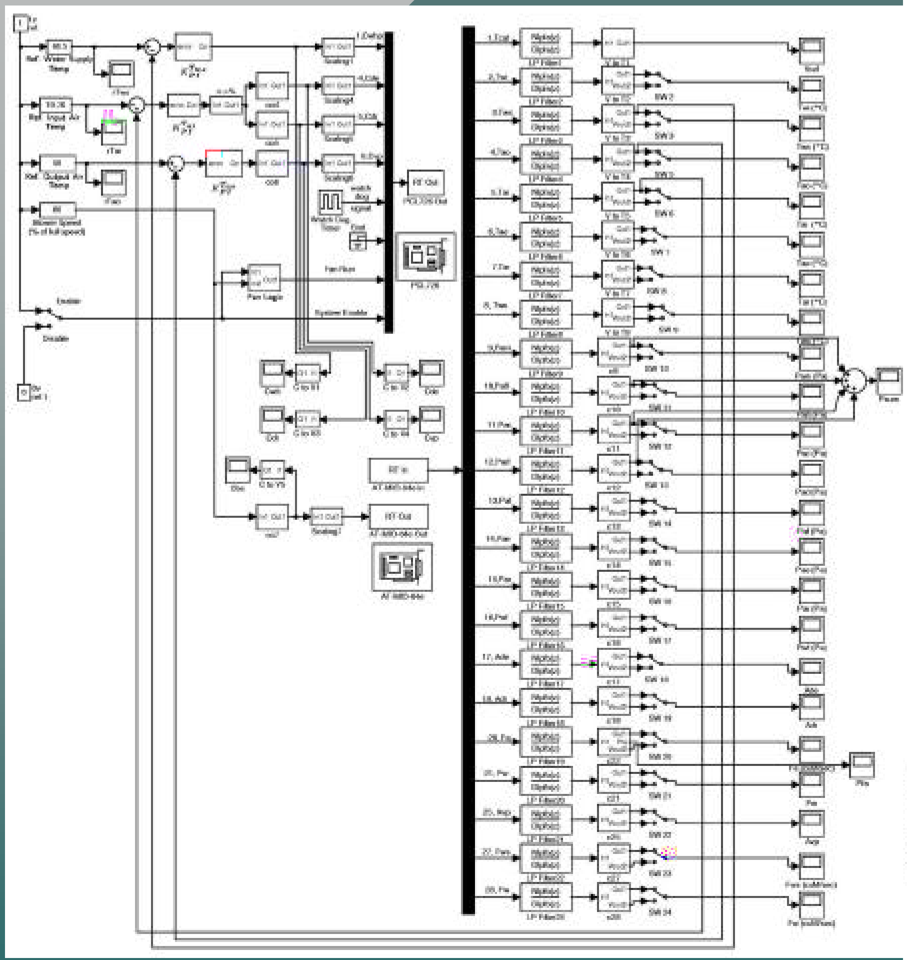
Controller K_{PI} Simulation

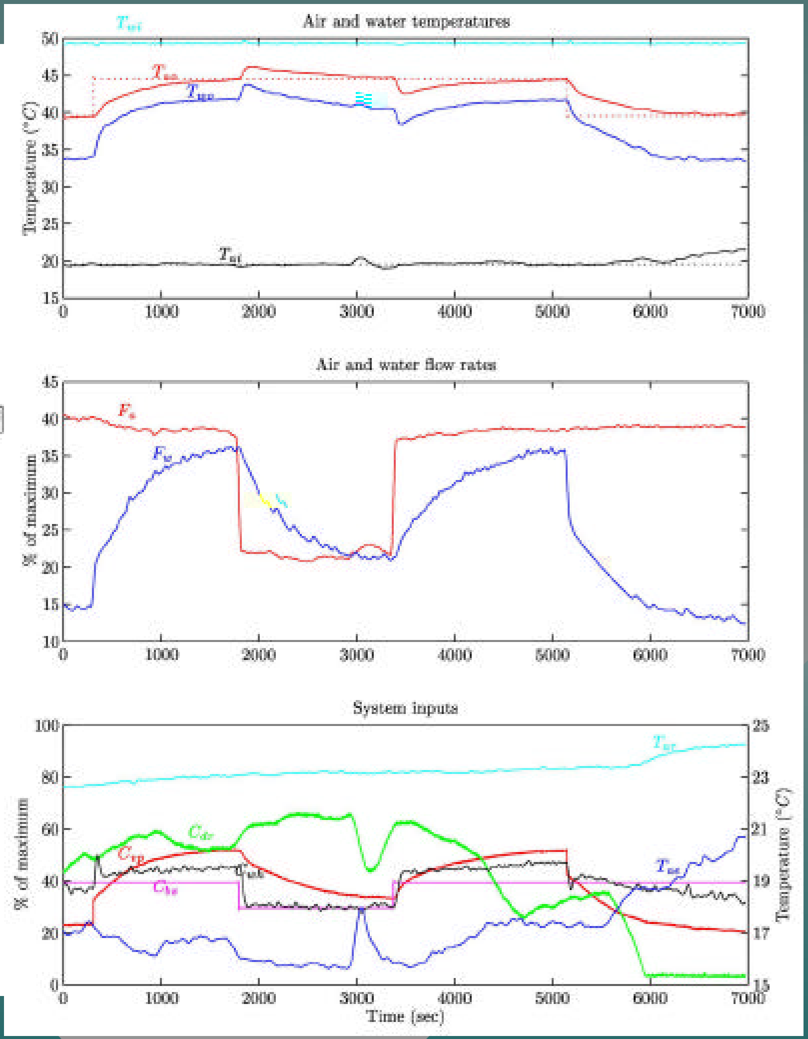




Colorado State University

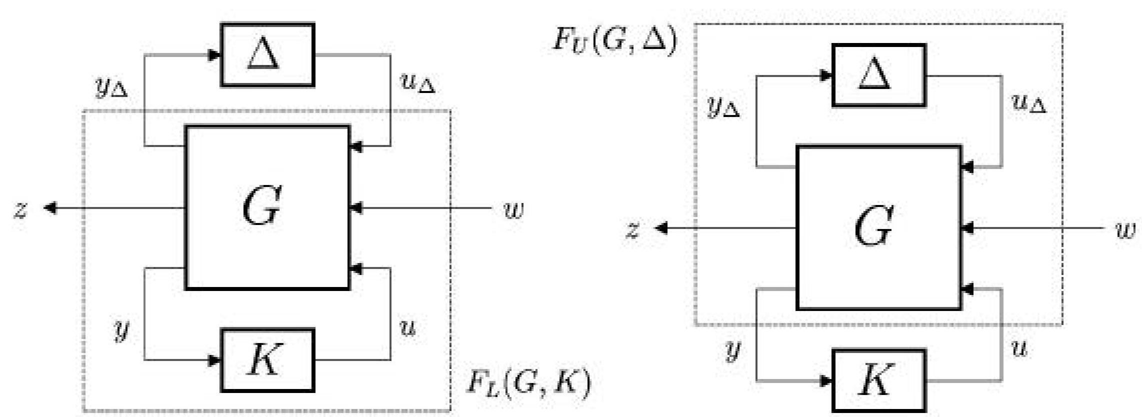
Controller K_{PI} Experimental





Colorado State University

Robust H_∞ Controller Design Using μ-Synthesis



Design Objective:

Find all stabilizing K such that: $\textit{for all } \hat{\Delta} : \|\Delta\|_\infty < 1, \quad \| F_U[F_L(G,K), \Delta] \|_\infty \leq 1$

Use augmented perturbation structure $\hat{\Delta} = \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_p \end{bmatrix}$ to test robust performance

The goal of μ synthesis to minimize over all stabilizing K, the peak value of $\mu_\Delta(\bullet)$

$$\sup_\omega \mu_\Delta(F_L(G,K)(j\omega)) \leq 1$$

Colorado State University

Robust H_∞ Controller Synthesis Using D-K Iteration

$$\sup_{\omega} \mu_\Delta(F_L(G,K)(j\omega)) \le 1$$

No method exists to directly synthesize a robust H_∞ controller, however, *D-K* iteration provides a good approximation:

The upper bound on μ can be expressed as: $\mu_{UB}: \quad \mu_\Delta(M) \le \inf_{D \in \mathcal{D}} \overline{\sigma}(DMD^{-1})$

The μ optimal controller minimizes the peak value of μ_{UB} over frequency (ω), or:

$$\min_{\substack{K \\ stabilizing}} \left(\min_{\substack{D \in \mathcal{D} \\ stable, min-phase}} \|DF_L(G,K)D^{-1}\|_\infty \right)$$

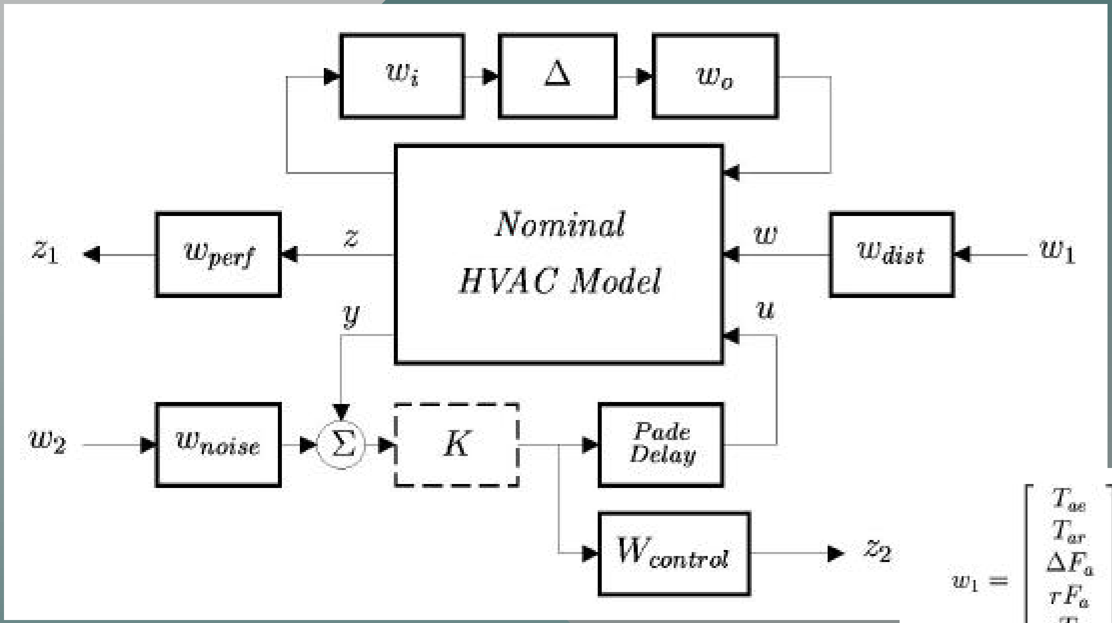
***D-K* iteration process:**

1) Initialize the transfer matrix $D(j\omega)$ (to the identity matrix)
2) Holding D fixed find K stabilizing that solves: $\min_{\substack{K \\ stabilizing}} \|DF_L(G,K)D^{-1}\|_\infty$
3) Holding $F_L(G,K)$ fixed find $D(j\omega)$ to minimize at each frequency $\overline{\sigma}(DF_L(G,K)D^{-1}(j\omega))$
4) Fit $D(j\omega)$ with a stable minimum-phase transfer function and return to step 2

Iteration continues until $\|DF_L(G,K)D^{-1}\|_\infty < 1$ or until H_∞ ceases to decrease

Colorado State University

MIMO Robust Controller Structure

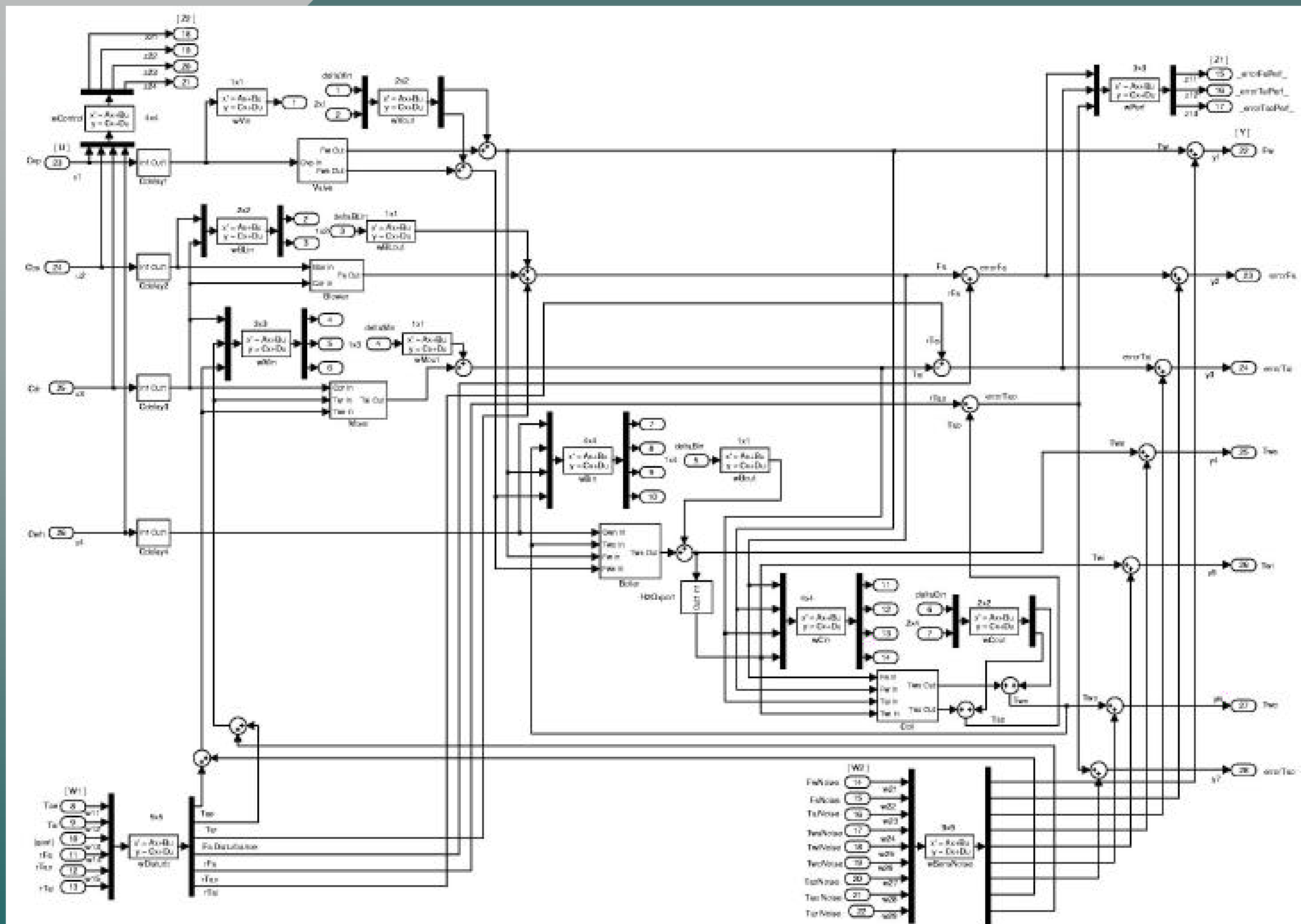


$$w_1 = \begin{bmatrix} T_{ae} \\ T_{ar} \\ \Delta F_a \\ rF_a \\ rT_{ao} \\ rT_{ai} \end{bmatrix} \quad z_1 = \begin{bmatrix} errorF_a \\ errorT_{ai} \\ errorT_{ao} \end{bmatrix} \quad u = \begin{bmatrix} C_{vp} \\ C_{bs} \\ C_{dr} \\ C_{wh} \end{bmatrix}$$

$$w_2 = \begin{bmatrix} noiseF_w \\ noiseF_a \\ noiseT_{ai} \\ noiseT_{ws} \\ noiseT_{wi} \\ noiseT_{wo} \\ noiseT_{ao} \\ noiseT_{ae} \\ noiseT_{ar} \end{bmatrix} \quad z_2 = \begin{bmatrix} C_{vp} \\ C_{bs} \\ C_{dr} \\ C_{wh} \end{bmatrix} \quad y = \begin{bmatrix} F_w \\ errorF_a \\ errorT_{ai} \\ T_{ws} \\ T_{wi} \\ T_{wo} \\ errorT_{ao} \end{bmatrix}$$

Colorado State University

Controller K_{R3} Design Diagram



Colorado State University

Controller K_{R3} Design Interconnection Structure

$$\Delta_{BlkStructure} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 2 & 4 \\ 14 & 5 \end{bmatrix} \begin{array}{l} : \ block\ structure\ for\ \Delta_{Valve}\ (deltaVin) \\ : \ block\ structure\ for\ \Delta_{Blower}\ (deltaBLin) \\ : \ block\ structure\ for\ \Delta_{Mixer}\ (deltaMin) \\ : \ block\ structure\ for\ \Delta_{Boiler}\ (deltaBin) \\ : \ block\ structure\ for\ \Delta_{Coil}\ (deltaCin) \\ : \ block\ structure\ for\ \Delta_{I/O}\ (scale\ W\ \&\ Z) \end{array}$$

Boiler Uncertainty

$$wBin(C_{wh}) = \frac{0.05s+0.003142}{s+3.142}$$

$$wBin(T_{wo}) = 0.005 \qquad wBout = \frac{0.05s+0.0001571}{s+0.1571}$$

$$wBin(F_w) = 1 \times 10^{-6}$$

$$wBin(F_{ws}) = 1 \times 10^{-5}$$

Heating Coil Uncertainty

$$wCin(F_a) = 1 \times 10^{-5}$$

$$wCin(F_w) = 1 \times 10^{-6} \qquad wCoutT_{wo} = \frac{0.1s+0.006283}{s+0.06283}$$

$$wCin(T_{ai}) = 0.001 \qquad wCout(T_{ao}) = \frac{s+0.06283}{s+0.6283}$$

$$wCin(T_{wi}) = 0.001$$

Valve Uncertainty

$$wVin(Cvp) = \frac{0.05s+0.003142}{s+3.142} \qquad wVoutFw = \frac{3\times10^{-6}s+9.425\times10^{-11}}{s+0.0001885}$$

$$wVoutFws = \frac{3\times10^{-6}s+4.712\times10^{-9}}{s+0.04712}$$

Blower Uncertainty

$$wBLin(C_{bs}) = \frac{0.05s+0.03142}{s+31.42} \qquad wBLout = \frac{0.01s+0.01571}{s+3.142}$$

$$wBLin(C_{dr}) = \frac{0.05s+0.03142}{s+31.42}$$

Mixing Box Uncertainty

$$wMin(C_{dr}) = \frac{0.005s+0.0003142}{s+0.3142}$$

$$wMin(T_{ar}) = \frac{0.01s+0.01257}{s+1.257} \qquad wBout = \frac{0.1s+0.0006283}{s+0.06283}$$

$$wBin(Cwh) = \frac{0.1s+0.01257}{s+1.257}$$

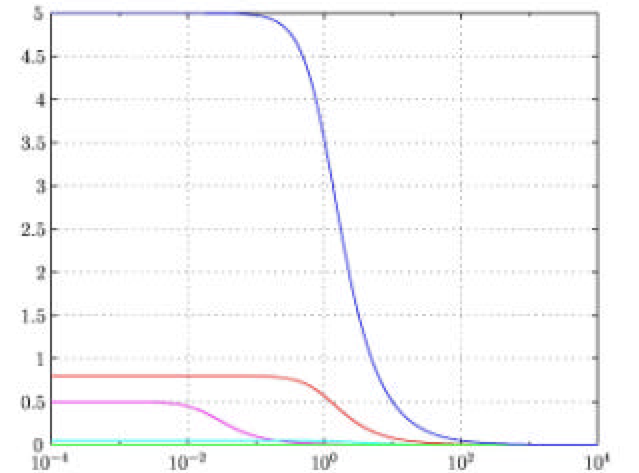
Colorado State University

Controller K_{R3} Design Weights

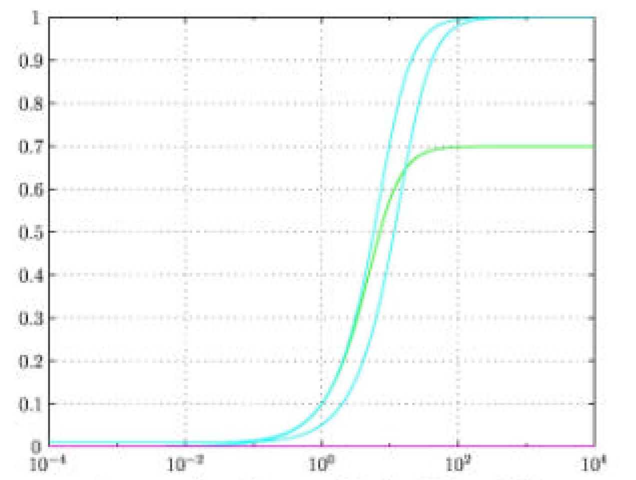
$$wDisturb(T_{ae}) = \frac{0.01}{s+0.02} \qquad wDisturb(T_{ar}) = \frac{0.01}{s+0.02}$$

$$wDisturb(F_a) = \frac{0.01}{s+2} \qquad wDisturb(rF_a) = \frac{0.8}{s+1}$$

$$wDisturb(rT_{ao}) = \frac{5}{s+1} \qquad wDisturb(rT_{ai}) = \frac{5}{s+1}$$



Disturbance & reference weights used in the design of K_{R3}



Sensor noise weights used in the design of K_{R3}

$$wSensNoise(F_w) = \frac{10s+1\times10^{-6}}{1\times10^{6}s+1} \qquad wSensNoise(F_a) = \frac{0.1s+0.001}{0.1429s+1}$$

$$wSensNoise(T_{ai}) = \frac{0.1s+0.001}{0.1s+1} \qquad wSensNoise(T_{ws}) = \frac{0.05s+0.01}{0.05s+1}$$

$$wSensNoise(T_{wi}) = \frac{0.05s+0.01}{0.05s+1} \qquad wSensNoise(T_{wo}) = \frac{0.05s+0.01}{0.05s+1}$$

$$wSensNoise(T_{ao}) = \frac{0.1s+0.01}{0.1s+1} \qquad wSensNoise(T_{ae}) = \frac{0.1s+0.01}{0.1s+1}$$

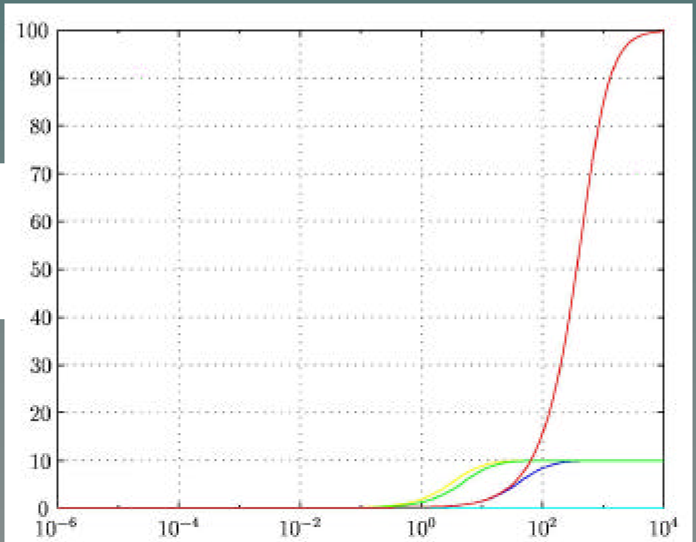
$$wSensNoise(T_{ar}) = \frac{0.1s+0.01}{0.1s+1}$$

Colorado State University

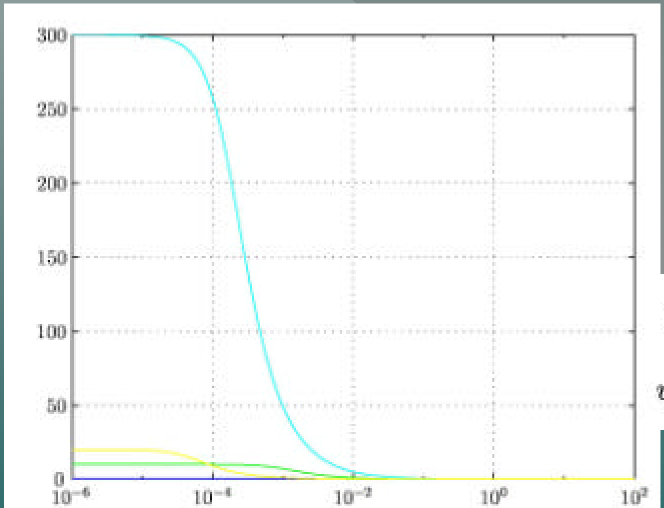
Controller K_{R3} Design Criterion

$$wControl(C_{vp}) = \frac{2s+0.01}{0.2s+1} \qquad wControl(C_{bs}) = \frac{0.1592s+0.01}{0.01592s+1}$$

$$wControl(C_{wh}) = \frac{0.1592s+0.2}{0.001592s+1} \qquad wControl(C_{dr}) = \frac{1.333s+0.02}{0.1333s+1}$$



Control weights used in the design of K_{R3}



Performance weights used in the design of K_{R3}

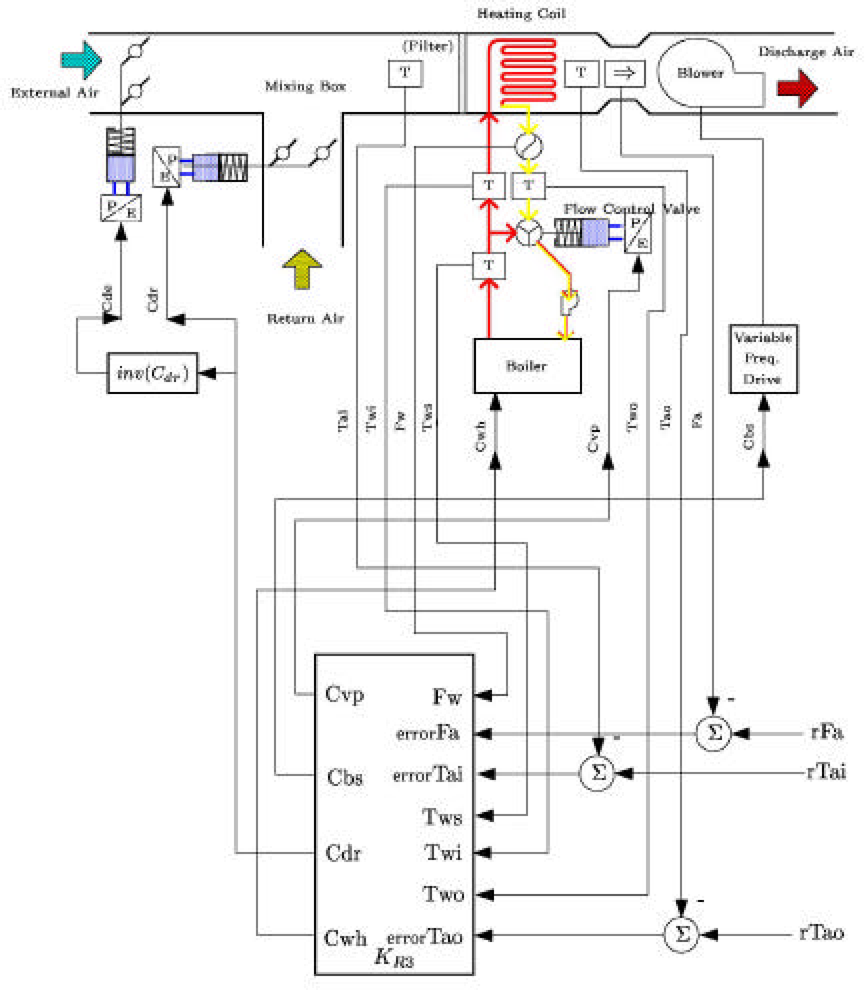
$$wPerf(F_a) = \frac{1\times10^{-6}s+0.05}{s+0.0001667} \qquad wPerf(T_{ai}) = \frac{0.01}{s+0.001}$$

$$wPerf(T_{ao}) = \frac{0.001}{s+5\times10^{-5}}$$

Colorado State University

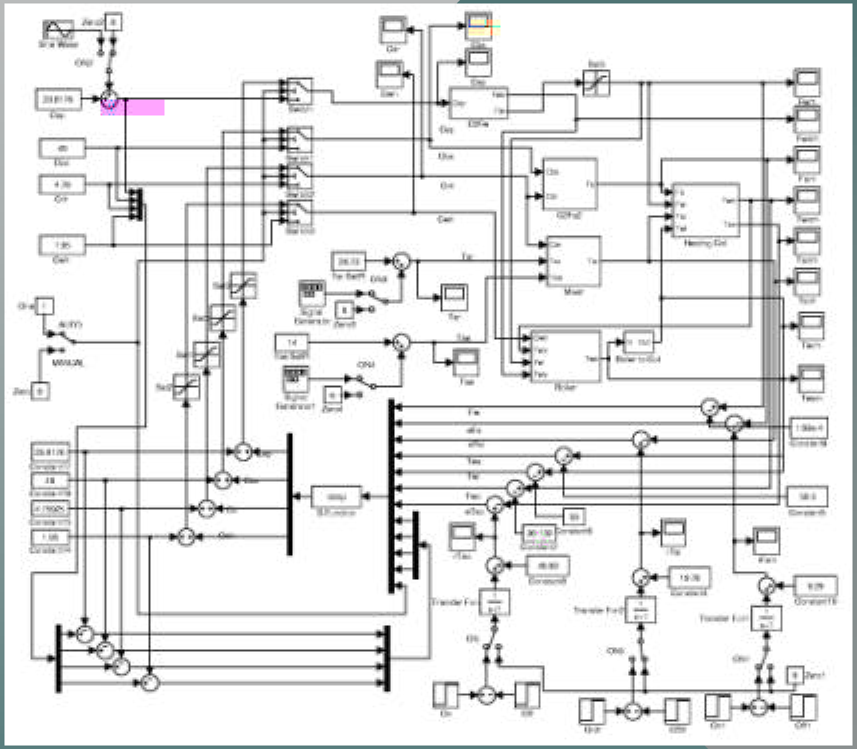
Controller K_{R3} Implementation

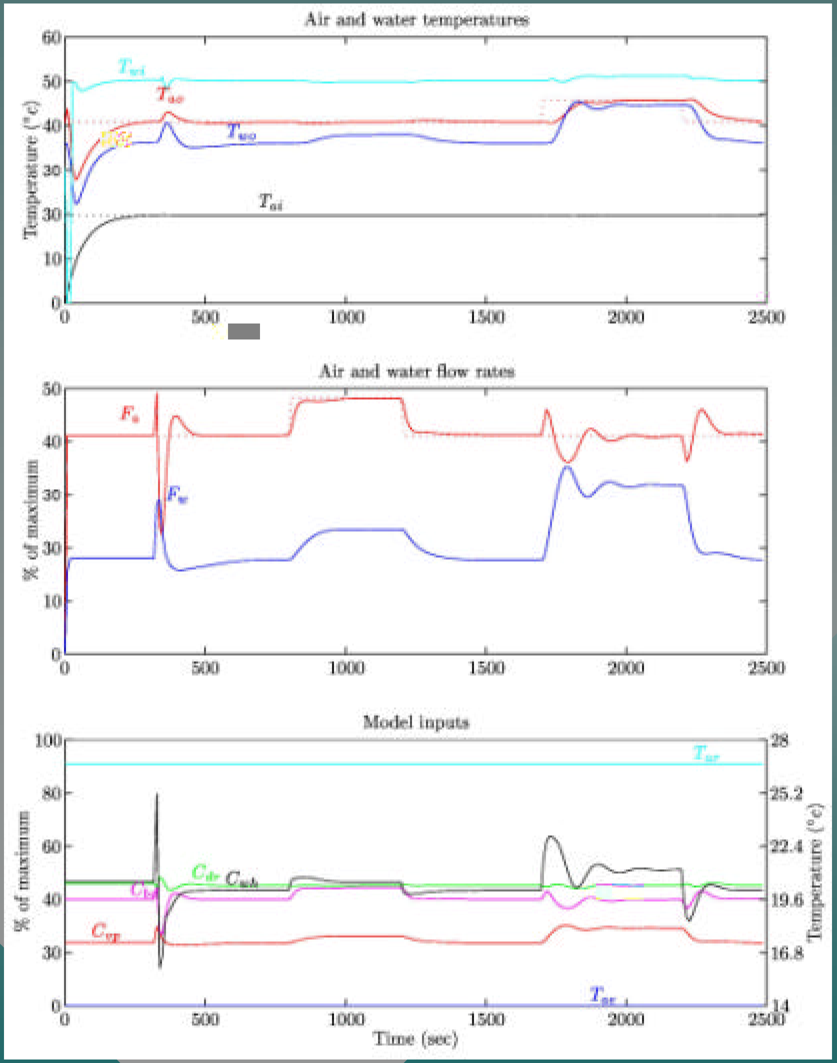
- 4x7 Controller
- Air flow (F_a)
- Input Air Temperature (T_{ai})
- Discharge Air Temperature (T_{ao})



Colorado State University

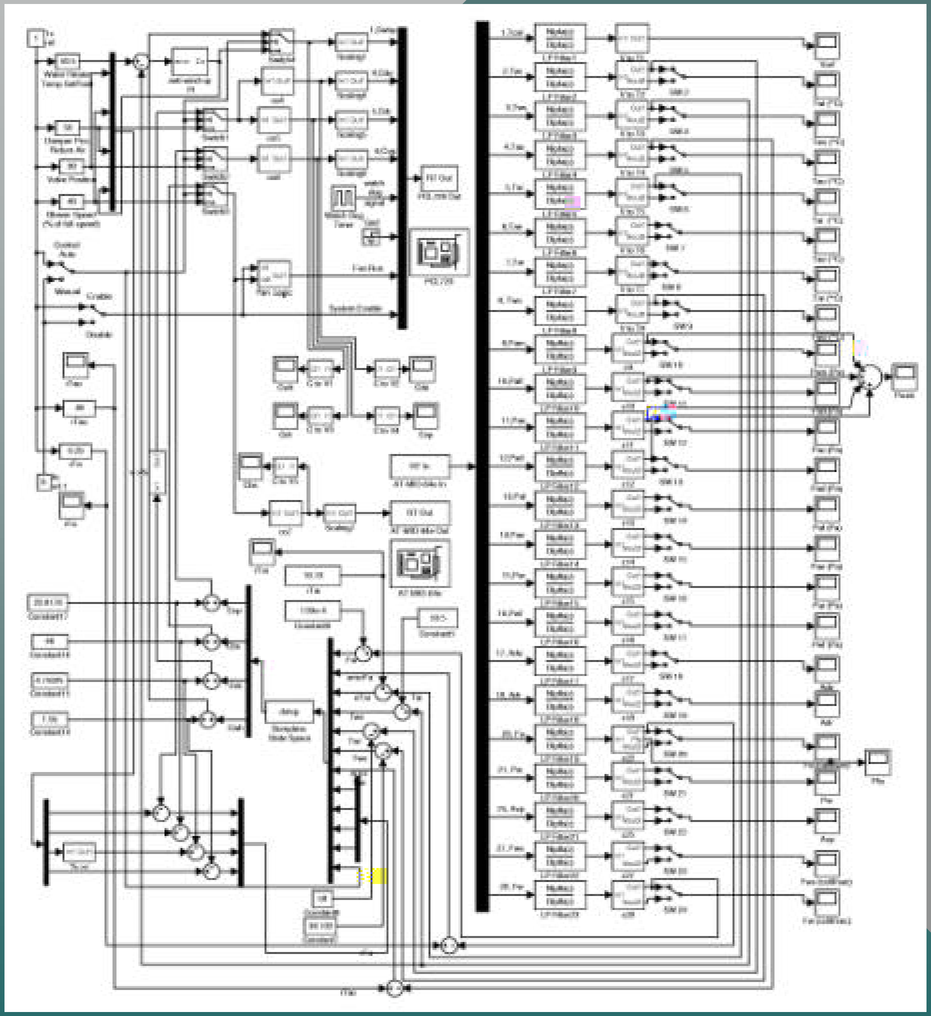
Controller K_{R3} Simulation

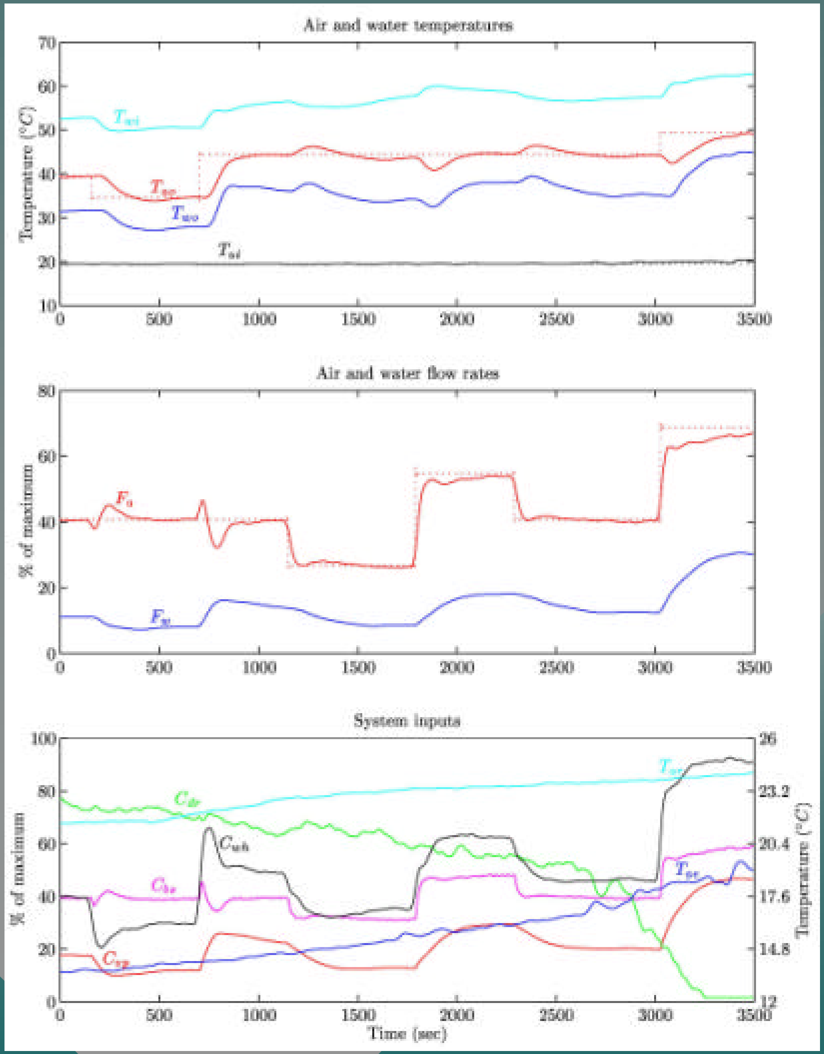




Colorado State University

Controller K_{R3} Experimental

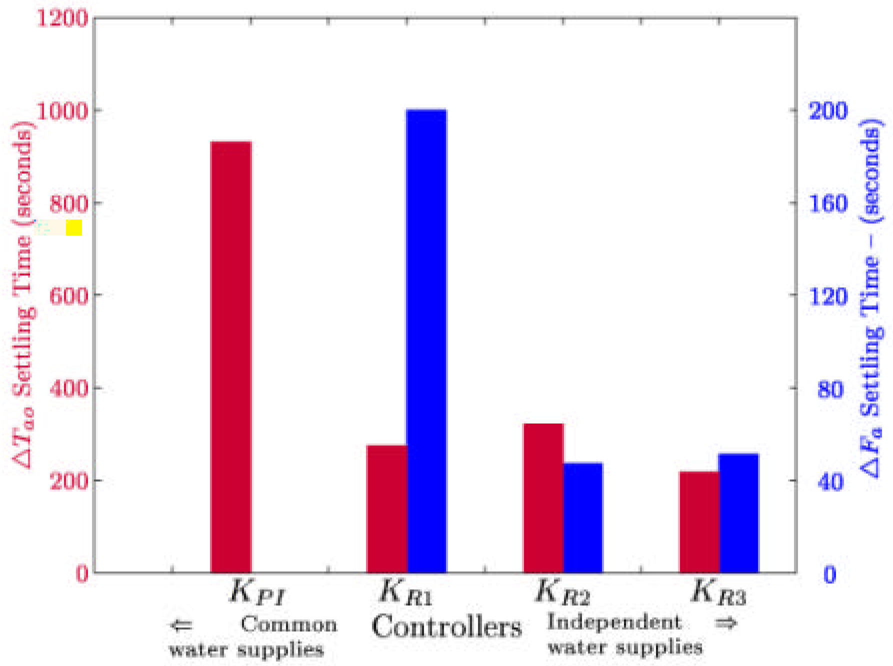




Colorado State University

Comparison of Controller Performance

Performance Measurement		K_{PI}	K_{R1}	K_{R2}	K_{R3}	units
T_{ao}	Rise Time	891	150	178	91	seconds
	Settle Time	931	275	322	218	seconds
	Overshoot	0.0	2.8	2.0	4.0	percent
	Disturbance Rejection	96.0	96.4	98.3	96.3	percent
F_a	Rise Time	< 5	199	42	44	seconds
	Settle Time	< 10	214	46.7	51	seconds
	Overshoot	NA	1.1	1.5	2.0	percent
	Disturbance Rejection	NA	77	83.6	88.7	percent



Colorado State University

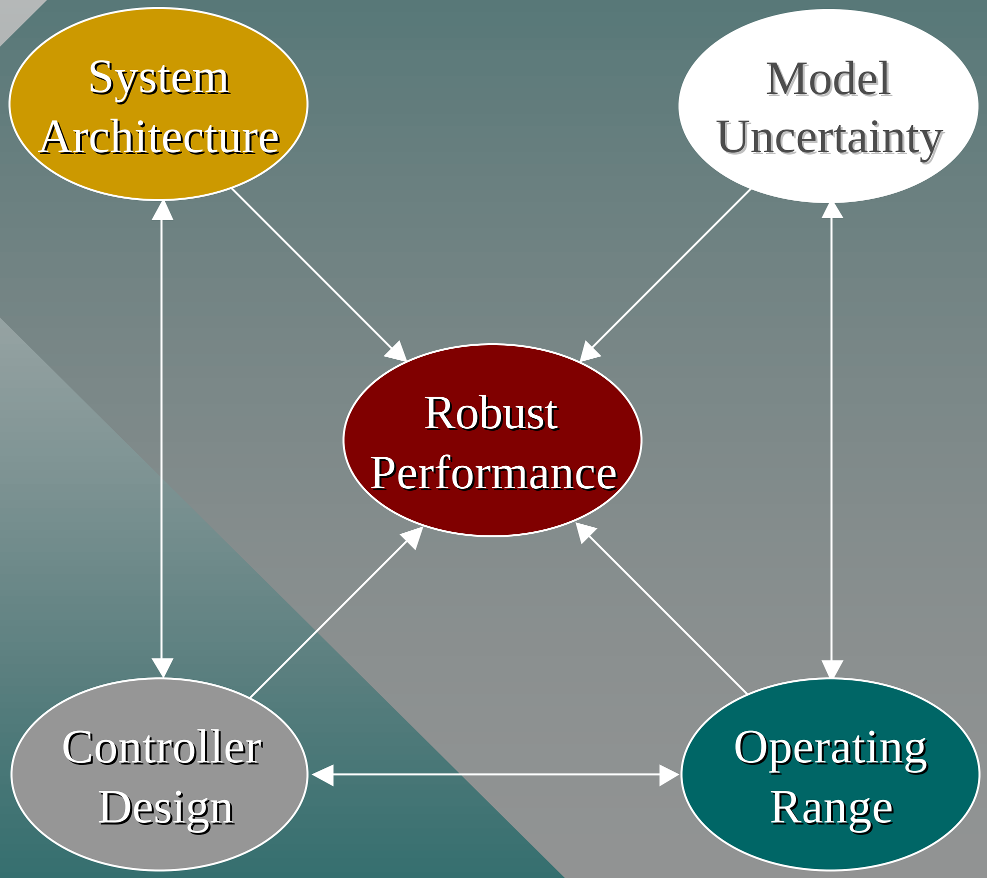
Comparison of Controller Architectures

Controller	Water Supply		Air Supply		Temp. Air Out ($\mathbf{T_{ao}}$)
	$\mathbf{T_{ws}}$ ($\mathbf{T_{wi}}$)	$\mathbf{F_w}$	$\mathbf{T_{ai}}$	$\mathbf{F_a}$	
$\mathbf{K_{PI}}$	SISO (PI)	$\Rightarrow T_{ao}$	SISO (PI)	Open-loop	SISO (PI)
$3 : SISO$	Tracks rT_{ws}	Varied to Control T_{ao}	Tracks rT_{ai}	Manually Controlled	Tracks rT_{ao} $f(F_w)$
$\mathbf{K_{R1}}$	SISO (PI)	MIMO Controlled			
MIMO (3×6) plus one $SISO$	Tracks rT_{ws}	Free Variable	Free Variable	Tracks rF_a (at low freq.)	Tracks rT_{ao} $f(F_w, T_{ai}, F_a^*)$
$\mathbf{K_{R2}}$	MIMO Controlled				
MIMO (4×7)	Tracks rT_{ws}	Free Variable	Tracks rT_{ai} (at low freq.)	Tracks rF_a (at low freq.)	Tracks rT_{ao} $f(Fw, T_{ai}^*, F_a^*)$
$\mathbf{K_{R3}}$	MIMO Controlled				
MIMO (4×7)	Free Variable	Free Variable	Tracks rT_{ai}	Tracks rF_a (at low freq.)	Tracks rT_{ao} $f(Fw, T_{ws}, F_a^*)$

Key: * dynamic participation only (tracks reference signal in steady-state)

Colorado State University

Factors Affecting Performance



Colorado State University

MIMO Control - Conclusions

- Designed and Implemented MIMO Robust Control on HVAC
- Simulation and Experiment
- Coordinated Control Action – Decoupling
- Robustness to Plant Variation – Theory and Experiment
- Large Performance Gains

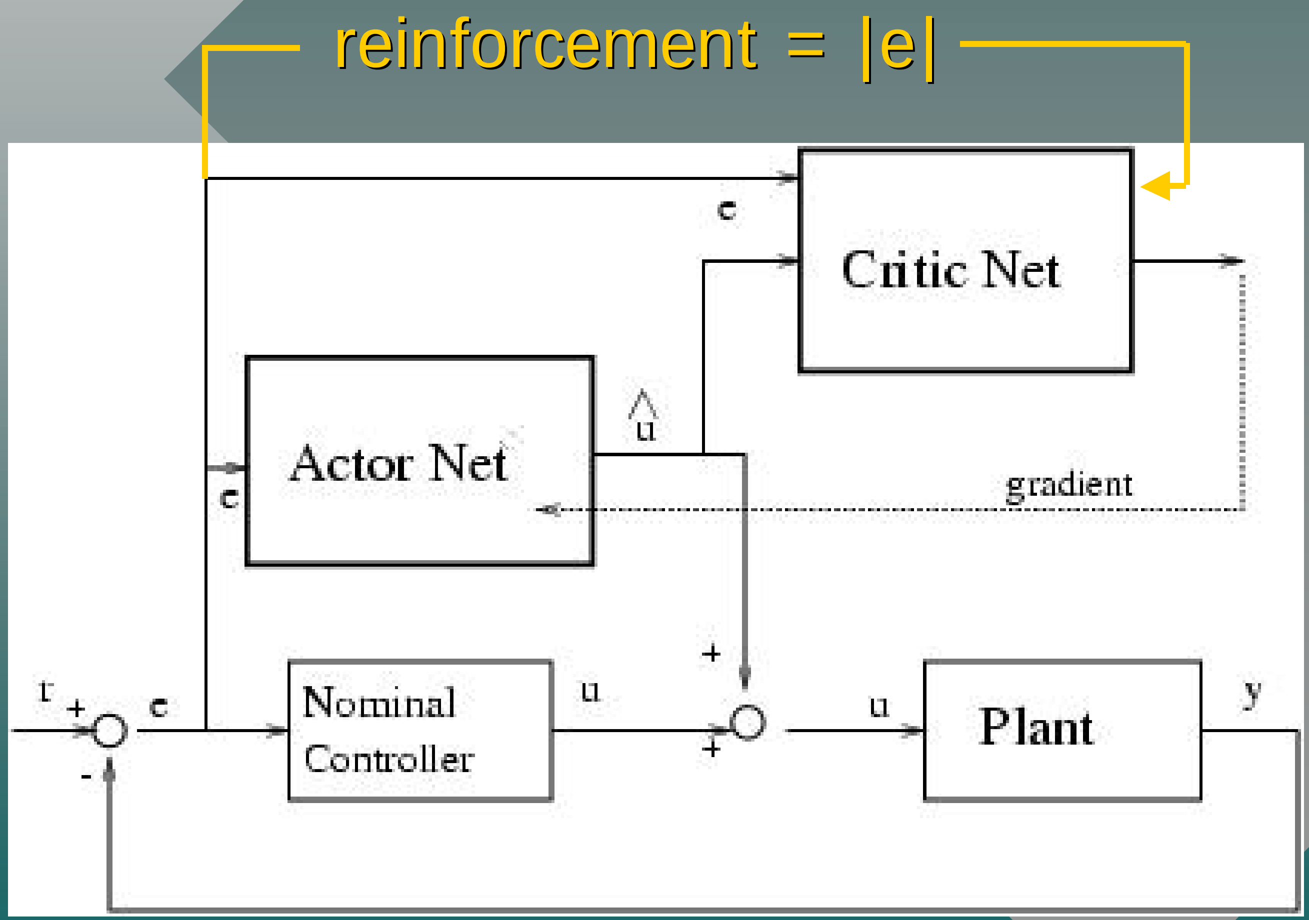
Robust Reinforcement Learning Controller

Colorado State University

Robust Reinforcement Learning

- Motivation
- Review of reinforcement learning
- Our previous results of reinforcement learning for HVAC
- Review of robust control theory
- Incorporating reinforcement learning agent in robust control theory
- Results, Conclusions, Planned Work

Reinforcement Learning Agent in Parallel with Controller



Colorado State University

Reinforcement Learning

Defines a kind of learning problem.

The action you take now may have a delayed effect on system and on performance evaluation.

Must find best sequence of actions, defined as the sequence that optimizes the sum of performance evaluations, or reinforcements.

Commonly formulated as a dynamic programming problem.

Solved by estimating the sum of expected future reinforcements for each state. The multi-step problem becomes a single step decision.

Dynamic programming assumes knowledge of state-transition probabilities.

Reinforcement learning does not. Instead, takes a Monte Carlo approach.

Reinforcement Learning

value function → *state* *action* *policy function* *discount factor* *reinforcement (|error|)*

$$Q_\pi(s_t, a_t) = E_\pi \left\{ \sum_{k=0}^{T} \gamma^k R(s_{t+k}, a_{t+k}) \right\}$$

$$Q_\pi(s_t, a_t) = E_\pi \left\{ R(s_t, a_t) + \sum_{k=1}^{T} \gamma^k R(s_{t+k}, a_{t+k}) \right\}$$

$$= E_\pi \left\{ R(s_t, a_t) + \gamma \sum_{k=0}^{T-1} \gamma^k R(s_{t+k+1}, a_{t+k+1}) \right\}$$

$$= E_\pi \{ R(s_t, a_t) + \gamma Q_\pi(s_{t+1}, a_{t+1}) \}$$

Subtract right side from left to get algorithm for updating Q

$$\Delta Q_\pi(s_t, a_t) = E_\pi \{ R(s_t, a_t) + \gamma Q_\pi(s_{t+1}, a_{t+1}) \} - Q_\pi(s_t, a_t)$$

Replace expectation with sample (Monte Carlo approach)

$$\Delta Q_\pi(s_t, a_t) = \alpha_t [R(s_t, a_t) + \gamma Q_\pi(s_{t+1}, a_{t+1}) - Q_\pi(s_t, a_t)]$$

Temporal-difference error

Reinforcement Learning

$$\Delta Q_{\pi}(s_t, a_t) = \alpha_t [R(s_t, a_t) + \gamma Q_{\pi}(s_{t+1}, a_{t+1}) - Q_{\pi}(s_t, a_t)]$$

This converges on best approximation of value function for policy π.

To also improve the policy:

$$\Delta Q_{\pi}(s_t, a_t) = \alpha_t \left[R(s_t, a_t) + \gamma \min_{a' \in A} Q_{\pi}(s_{t+1}, a') - Q_{\pi}(s_t, a_t)\right]$$

(Q-learning, Watkins, 1989)

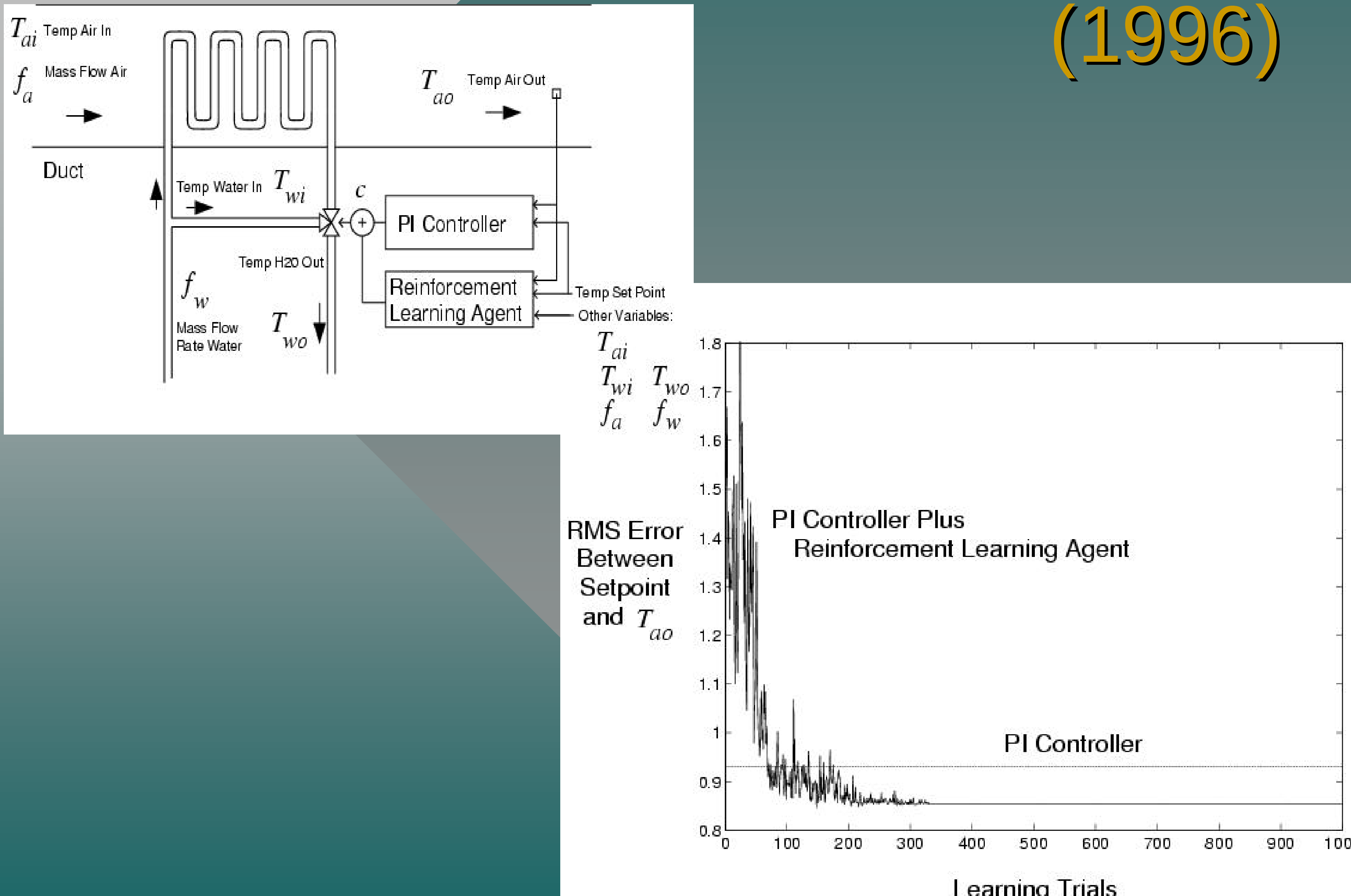
Q implicitly defines the policy:

$$\pi(s_t) = \arg\min_{a \in A} Q(s_t, a)$$

Value function (Q) learned by critic network.

Policy function (π) learned by actor network.

Reinforcement Learning Results (1996)



Colorado State University

Robust Reinforcement Learning?

Learns improved control, but no guarantee of stability.

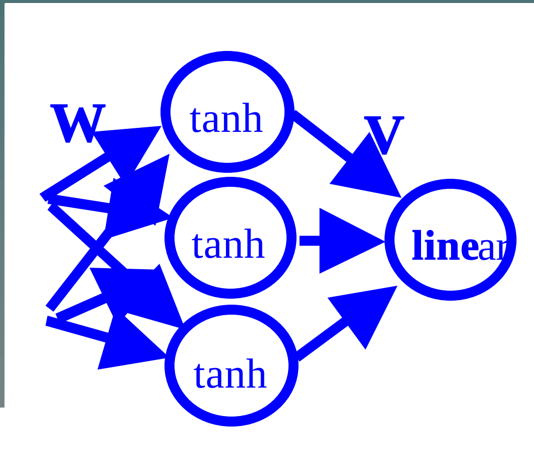
Can we formulate combination of PI control and RL within robust control theory?

Robust control theory is based on linear, time-invariant transfer functions.

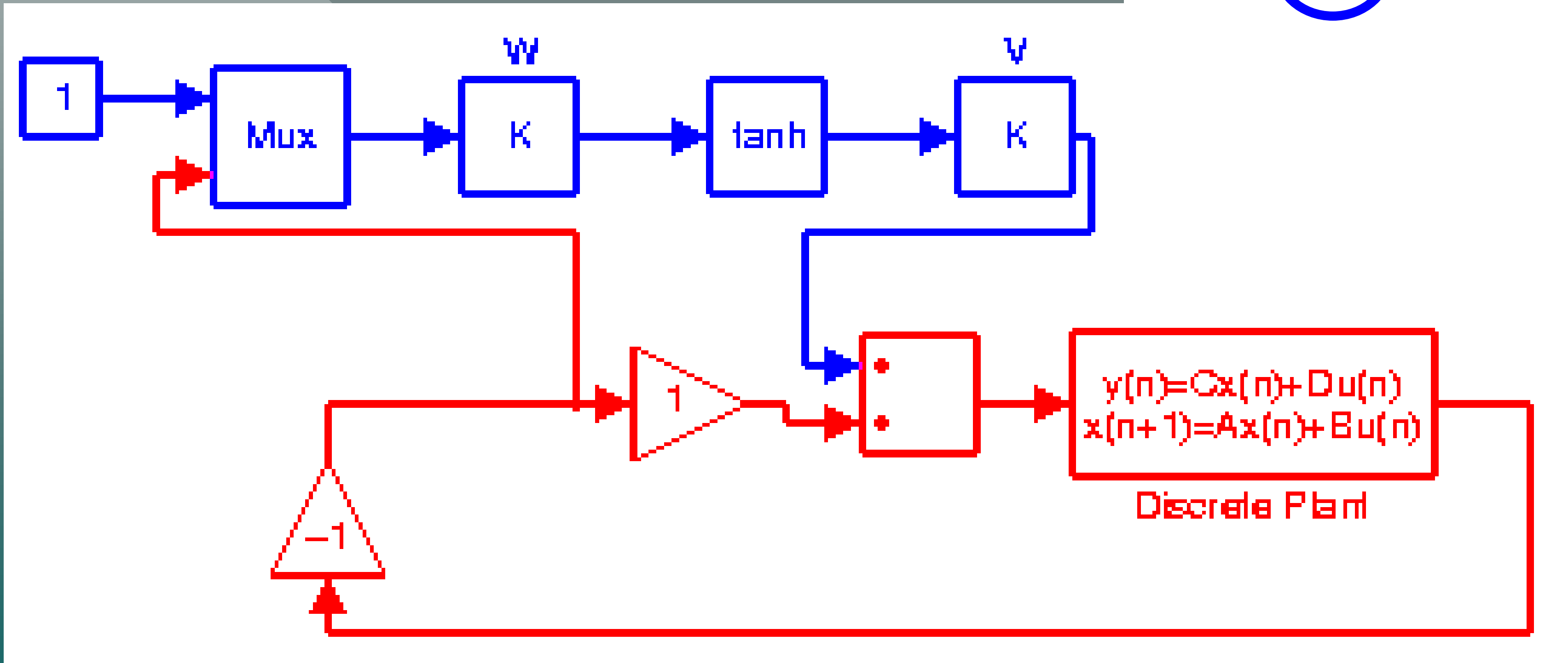
RL agents are nonlinear, because of the units' activation functions.

RL agents are time-varying, because they update their parameters to produce improved behavior.

Neural Net for Learning Agent



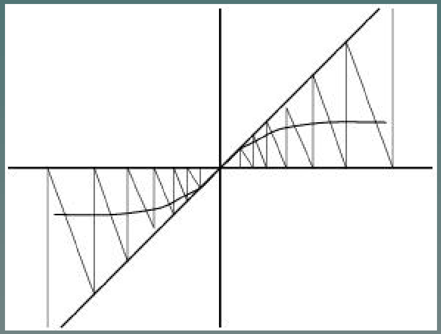
Actor Network (Critic Network not shown)



IQCs for Neural Network as RL Agent

Nonlinear part: tanh

replace with odd, bounded-slope IQC



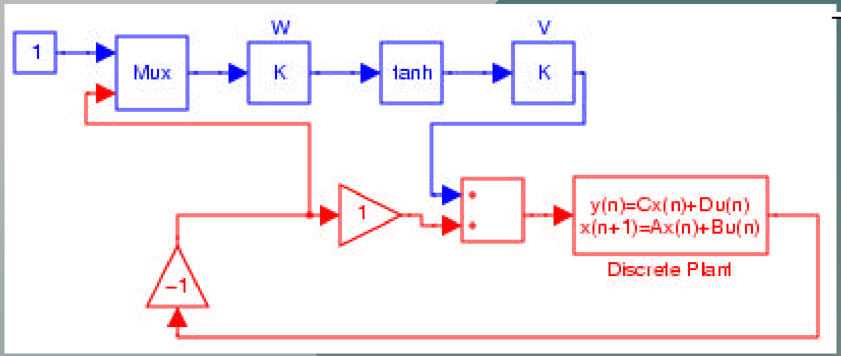
Time-varying part: weight updates

replace with slowly time-varying IQC

Replace with IQCs only for stability analysis, not during operation

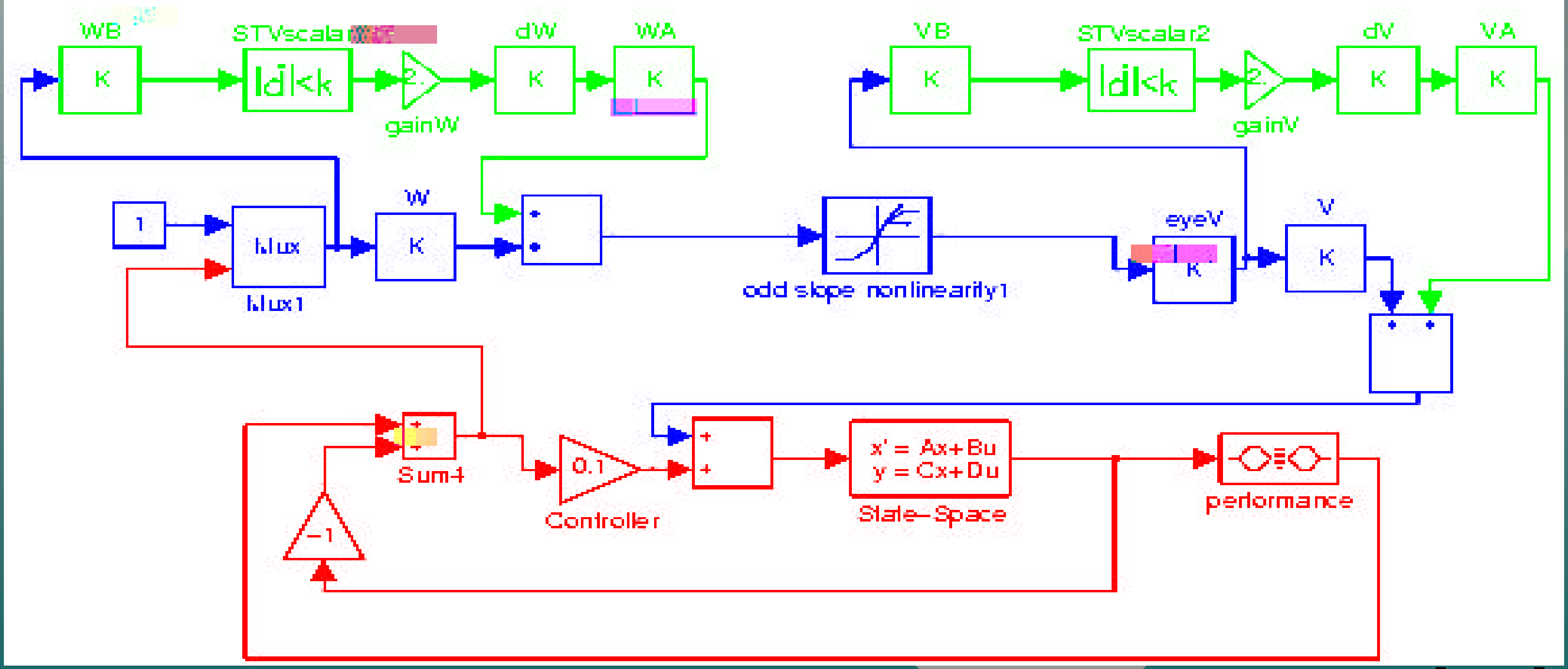
Colorado State University

IQCs for Neural Network as RL Agent



Two-layer neural net as actor, in parallel with controller.

Now with tanh and varying parameters “covered” by IQCs.

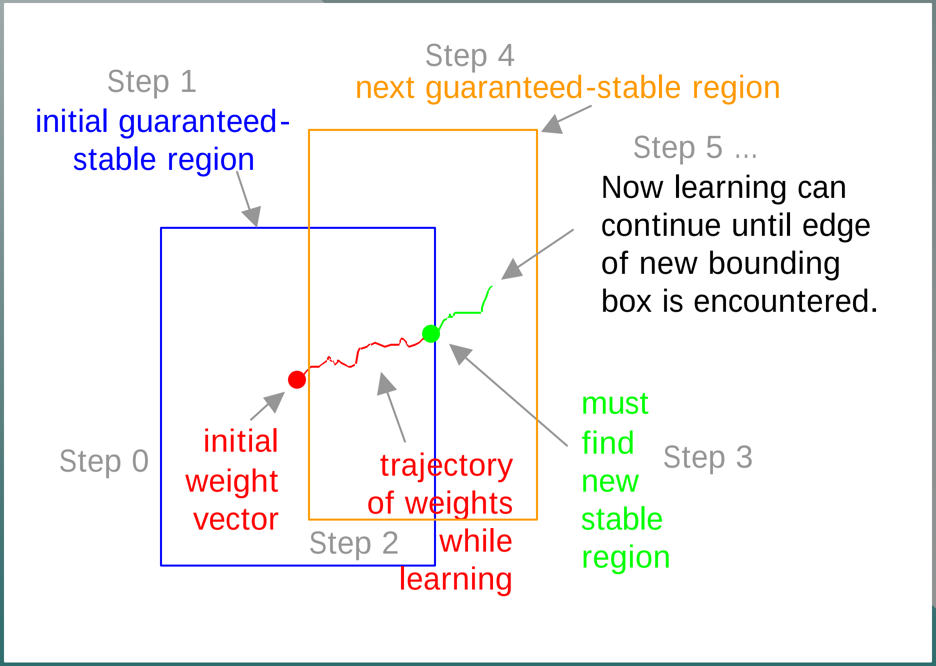


Colorado State University

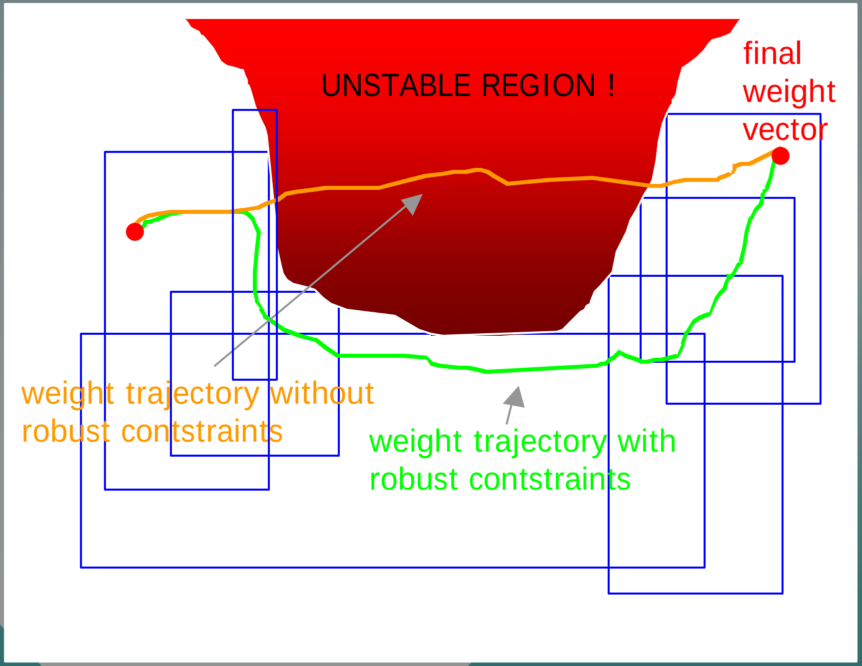
Incorporating Time-Varying IQC in Reinforcement Learning

Reinforcement learning algorithm guides adjustment of actor's weights.

IQC places bounding box in weight space, beyond which stability has not been verified.



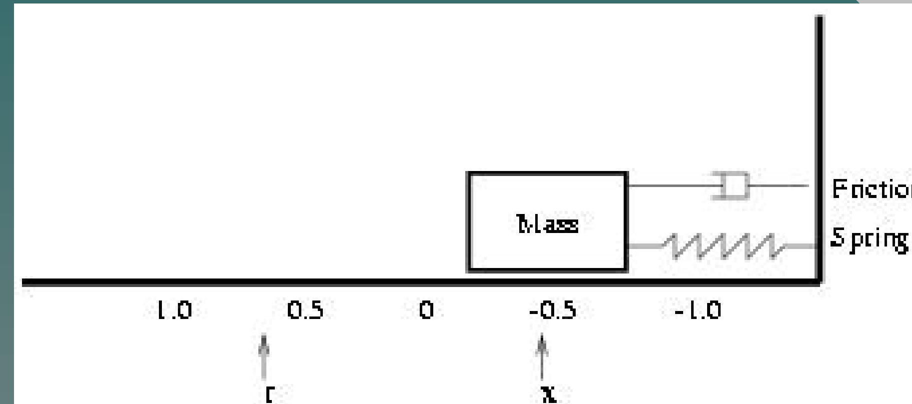
weight space (high-dimensional)

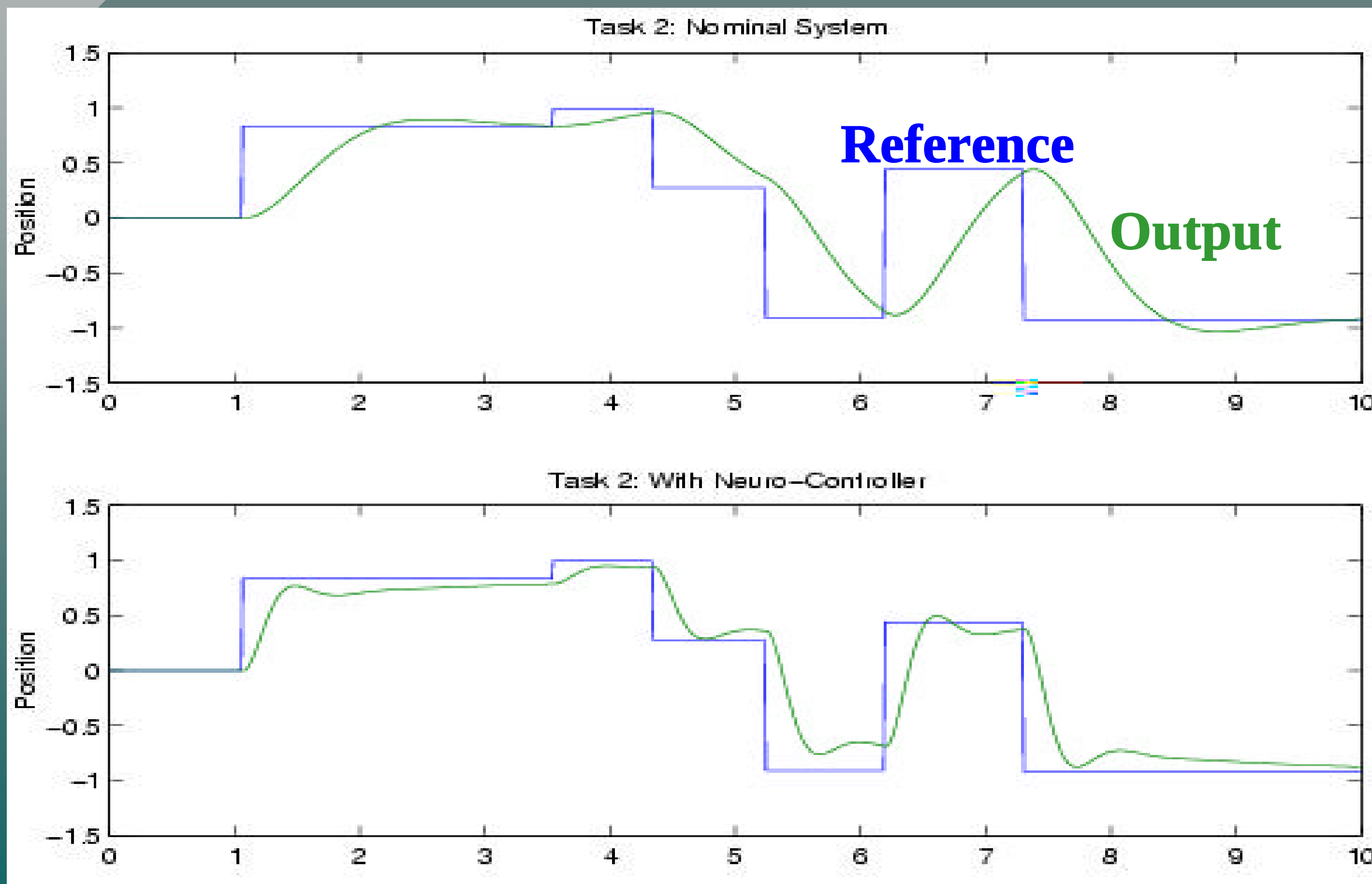


weight space (high-dimensional)

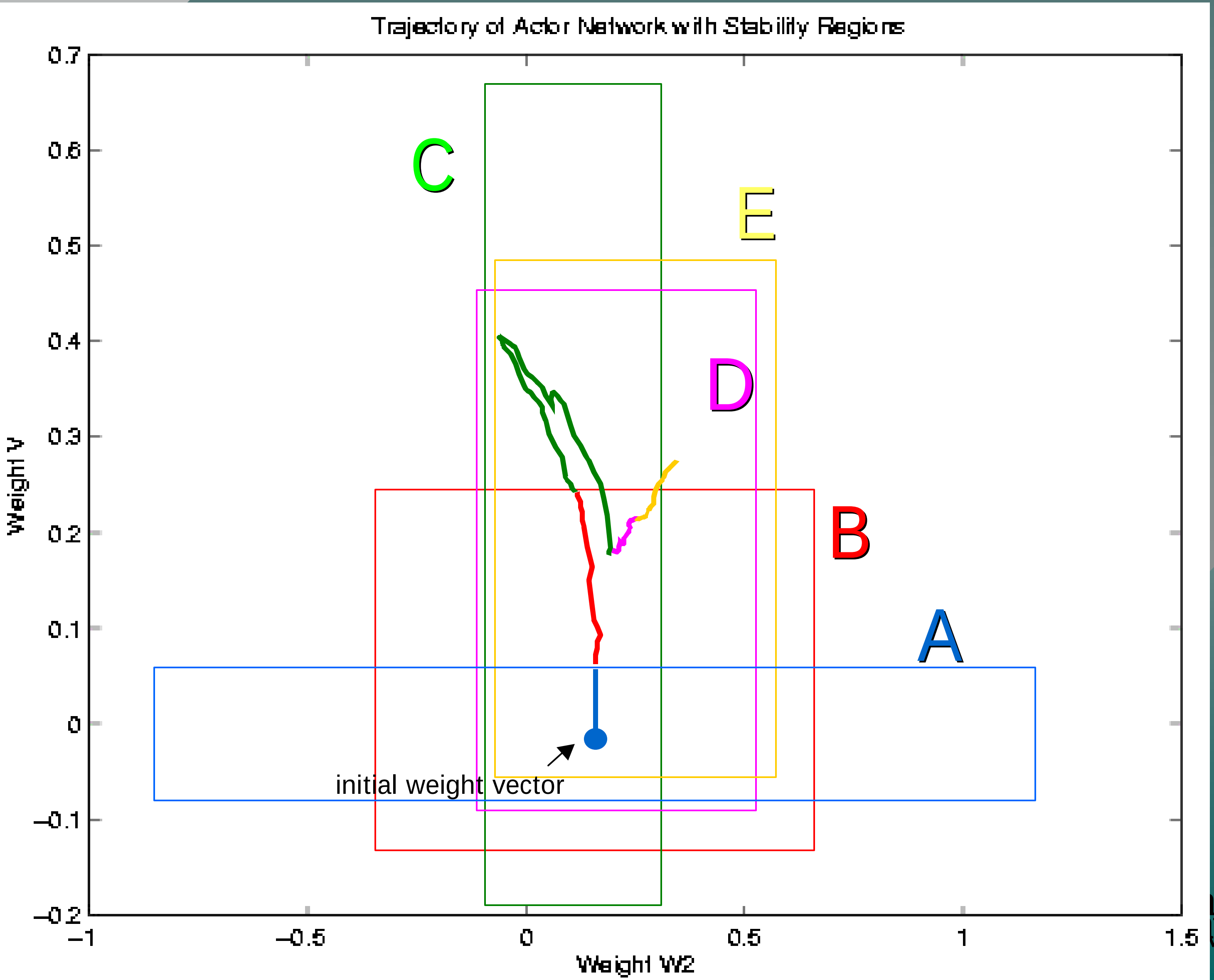
Colorado State University

Test on Simple Simulated Task

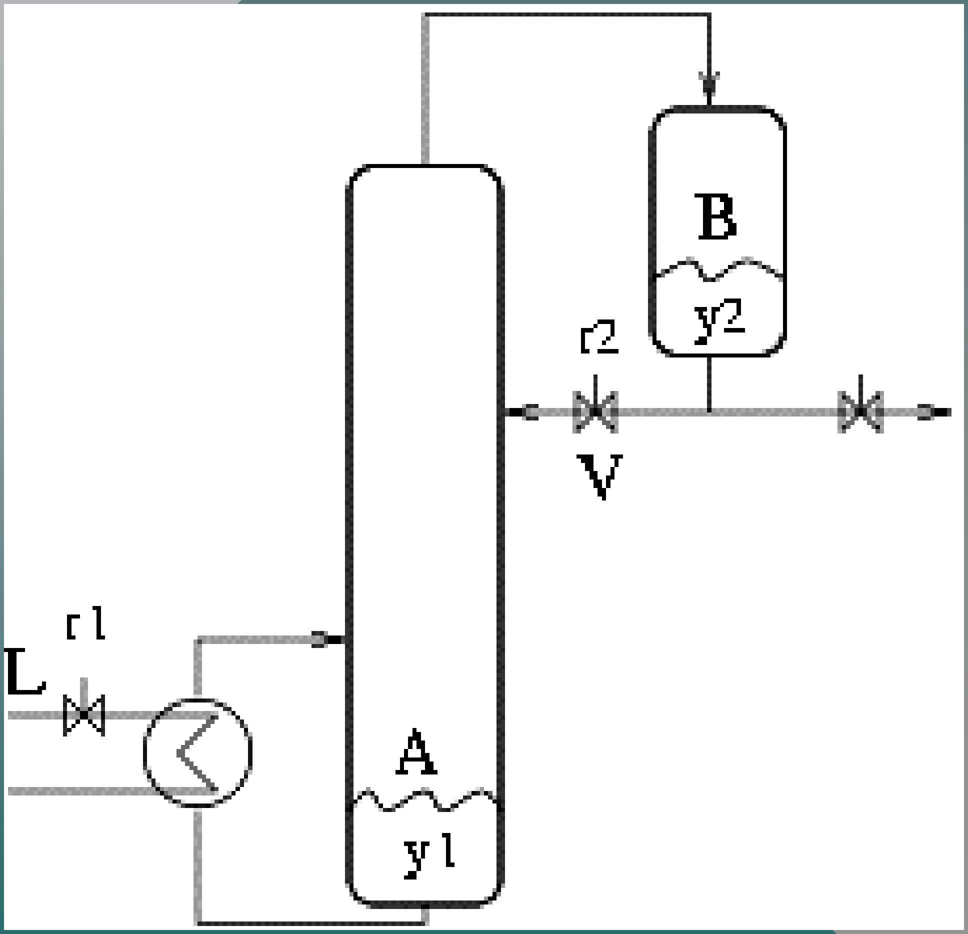




Trajectory of Weights and Bounds on Regions of Stability



Distillation Column

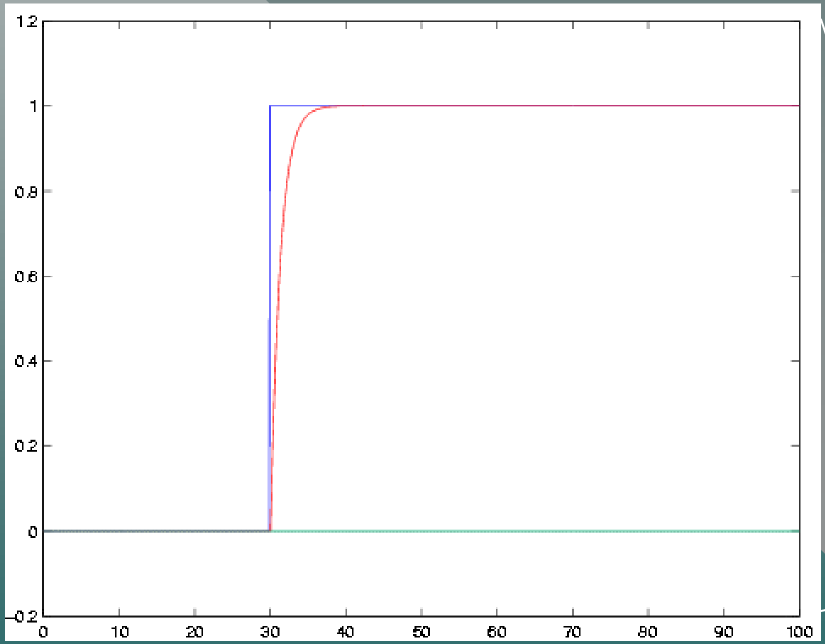


Example of task for which control variables interact in complex way.

Decoupling Controller

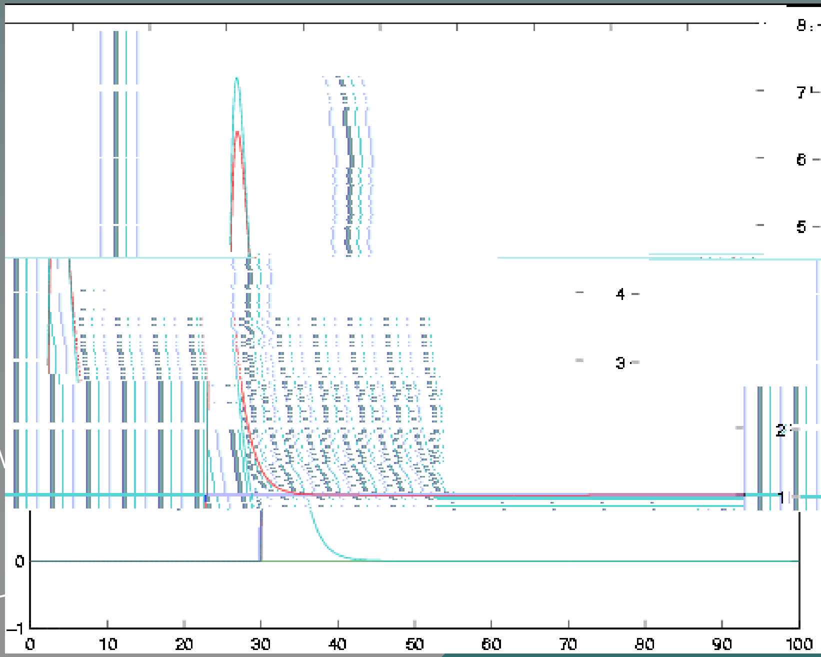
Nominal

Good response



Perturbed

Terrible response

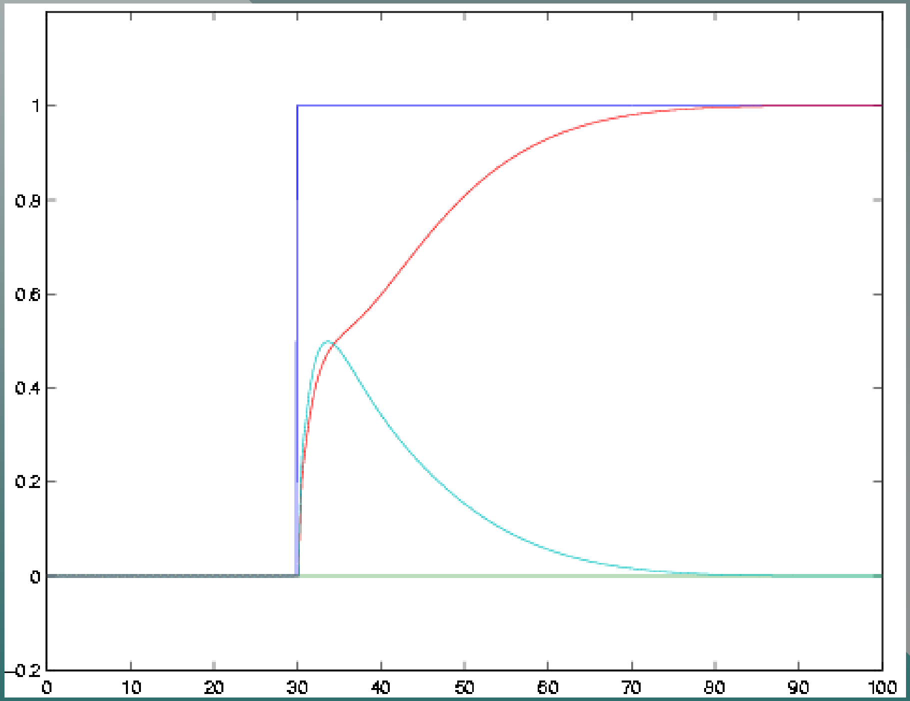


Colorado State University

Robust Controller

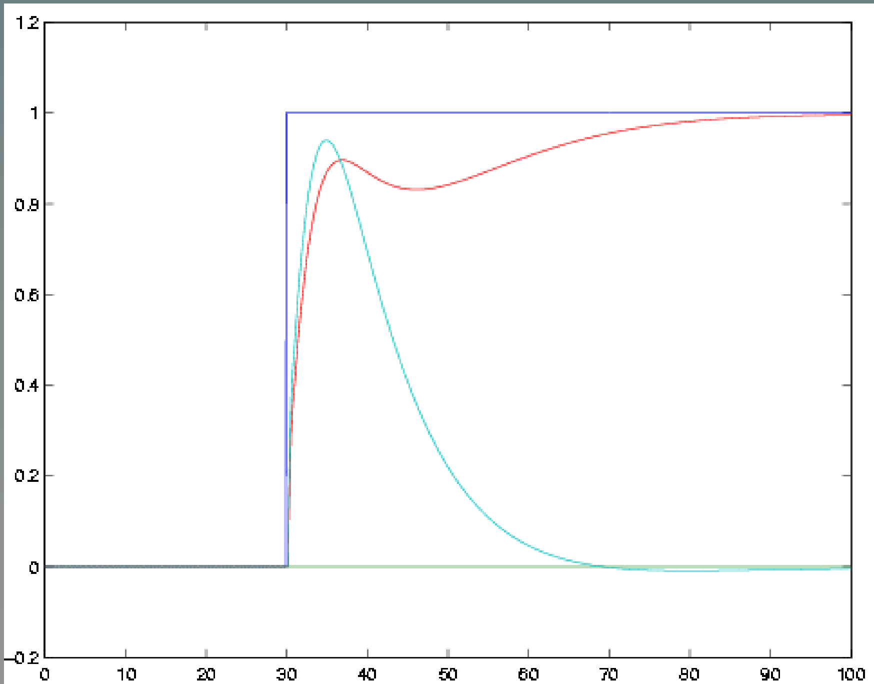
Nominal

Less aggressive response



Perturbed

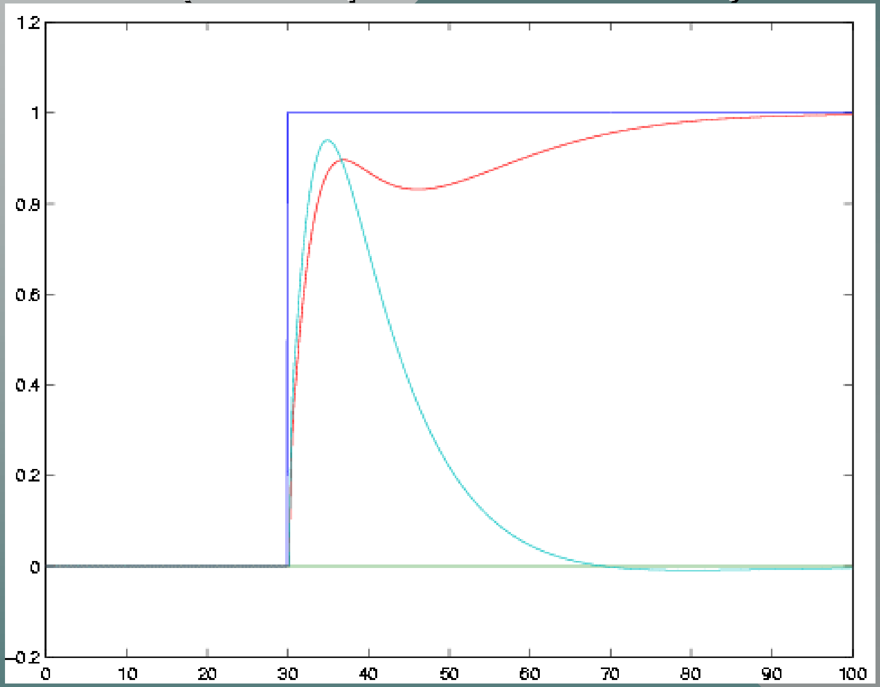
Much improved response



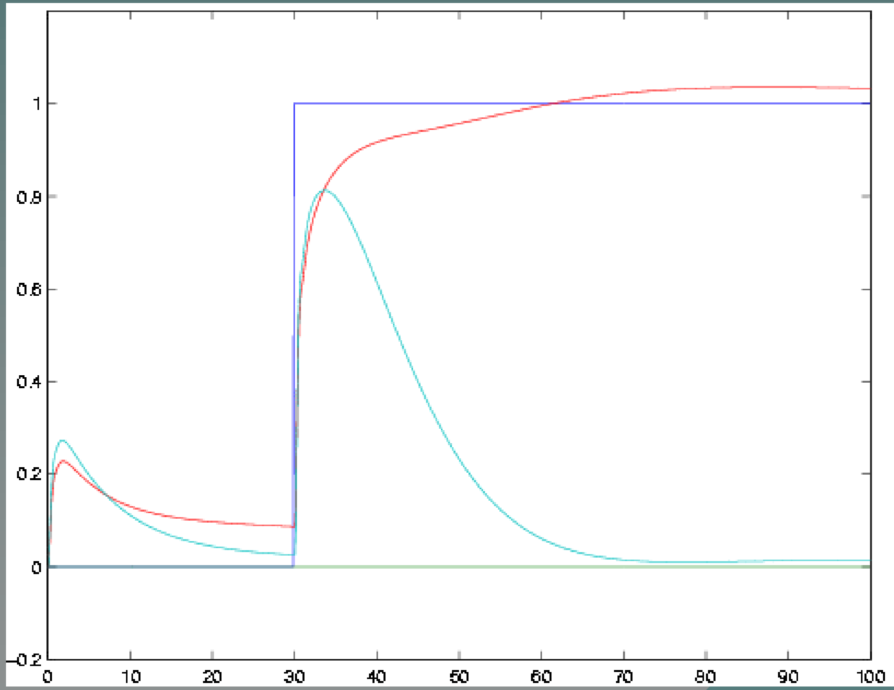
Colorado State University

Robust Reinforcement Learning

Perturbed case, no learning (from previous slide)



Perturbed case, with learning



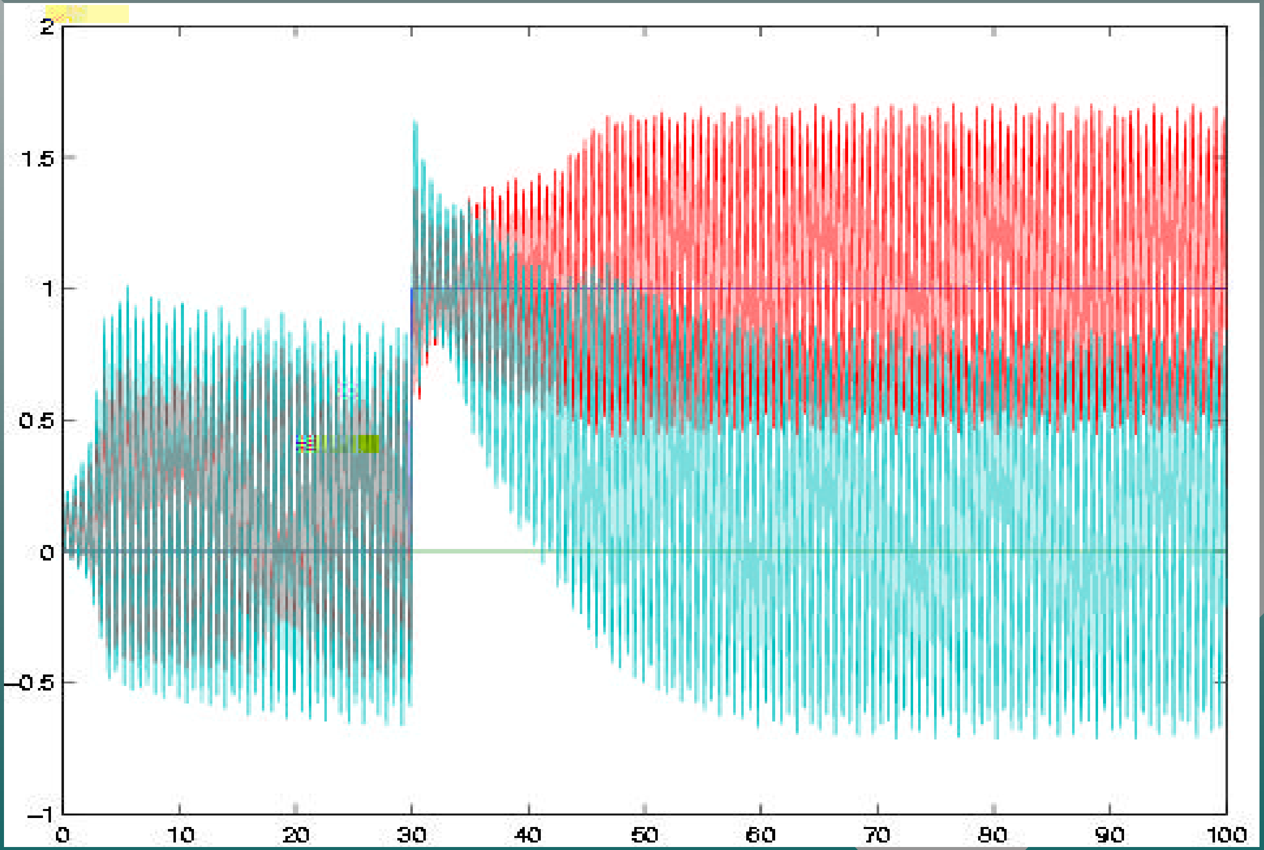
	Sum Squared Error
Nominal Controller	0.646
Robust Controller	0.286
Robust RL Controller	0.243

Through learning, controller has been fine-tuned to actual dynamics of real plant without losing guarantee of stability !

Colorado State University

Reinforcement Learning without IQCs

Ultimately achieves same good performance, but during learning periods of instability occur.



Colorado State University

Conclusions

- IQC bounds on parameters of tanh and sigmoid networks exist for which the combination of a reinforcement learning agent and feedback control system satisfy the requirements of robust stability theorems. (static and dynamic stability)
- Resulting robust reinforcement learning algorithm improves control performance while avoiding instability on several simulated problems.

Colorado State University

Current Work

- Applying robust reinforcement learning to HVAC model and real HVAC system.
- Developing continuous versions of reinforcement learning.
  - Continuous state, action needed for high-dimensional control problems
- Investigating value-gradient method (based on Werbos' heuristic dynamic programming, 1987).
  - Uses known or learned model of system dynamics.
  - Can result in much faster learning.

Colorado State University

Significant Project Outcomes

- Built Experimental HVAC System
- Developed models of HVAC System
- Developed and Implemented PI Plus Neural Network Control
  - Improved performance
  - Simple to implement and train
  - MISO
  - Applicable to many processes

Significant Project Outcomes (cont.)

- Designed MIMO Robust Controllers
- Implemented MIMO Control
  - Reduced T_{ao} Settle Time by over 300%
  - Decoupling of Controlled Variables
  - Simultaneous, Coordinated Control Action
  - Controllers are Insensitive to Model Uncertainty

Colorado State University

Significant Project Outcomes (cont.)

- Designed Robust Reinforcement Learning Controller
- Tested Controller on Standard Problems
  - For the first time, a control neural network can be trained while guaranteeing robust stability
  - A potential breakthrough in the application of neural networks to control
  - Training is by reinforcement learning, obviating the need for training data sets.

Colorado State University

Impact of Project

- Dramatic Improvement versus Current HVAC Control
  - Improved Efficiency
  - Stability and Robustness
  - Coordinated MIMO Action
- MIMO Robust Control
- First Guarantee of Stability During Reinforcement Learning.
- Potential Cost Savings
- Installation and Maintenance
- 6 Publications
- Currently Pursuing 2 Patents
- 4 Masters and 2 PhD Students

Colorado State University

Future Directions

- Dissemination into Industry
- Implementation of Robust Learning Control on MIMO HVAC System
- Large Scale Experimental Platform
- Gain-Scheduled Controllers
- Nonlinear Modeling – PDE Approach
- Robust Reinforcement Learning Control Theoretical Advances
- Advanced Robust Learning Algorithms