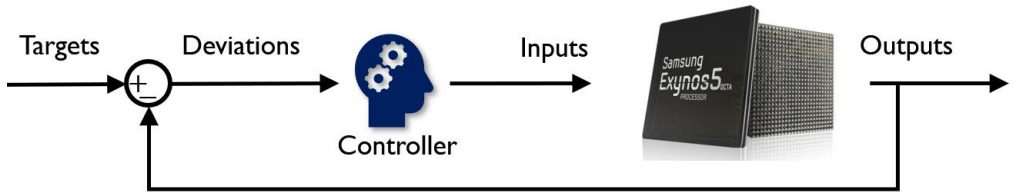
COMBINING CONTROL AND LEARNING

So Far...

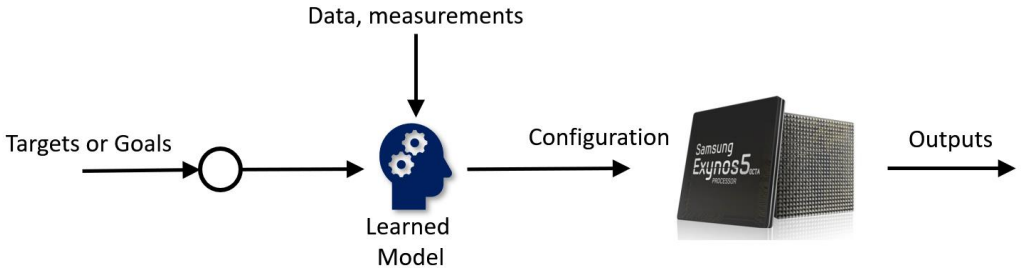
Control Theory

- Feedback to evolve state
- Policy is fixed
- Designed with a dynamic model



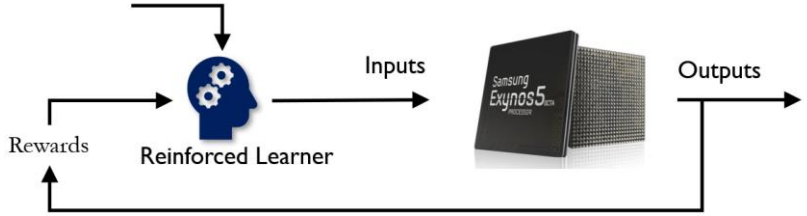
Supervised learning

- No feedback
- Policy is fixed
- No dynamics

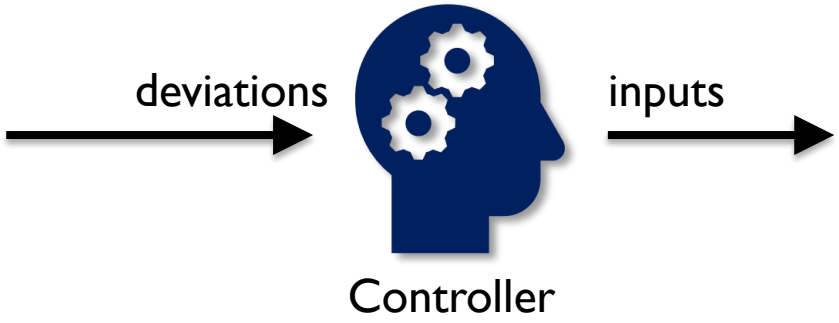


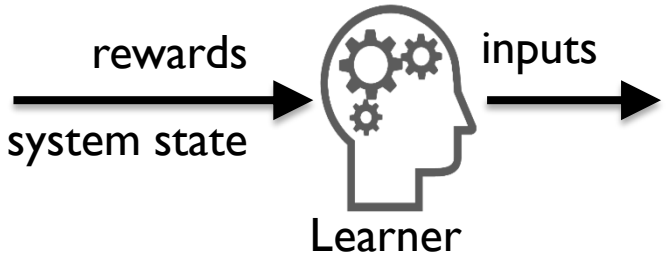
Reinforcement learning

- Feedback to improve actions
- Feedback to change policy
- Model-free



Control and Reinforcement Learning





[**A, B, C, D**] and *state*

$$\text{inputs} = \mathbf{C} \times \text{state} + \mathbf{D} \times \text{deviations}$$

$$\text{state}_{\text{new}} = \mathbf{A} \times \text{state} + \mathbf{B} \times \text{deviations}$$

- State: controller's property
  - Vector of real values
- State evolution implicitly uses model

Policy and *action*

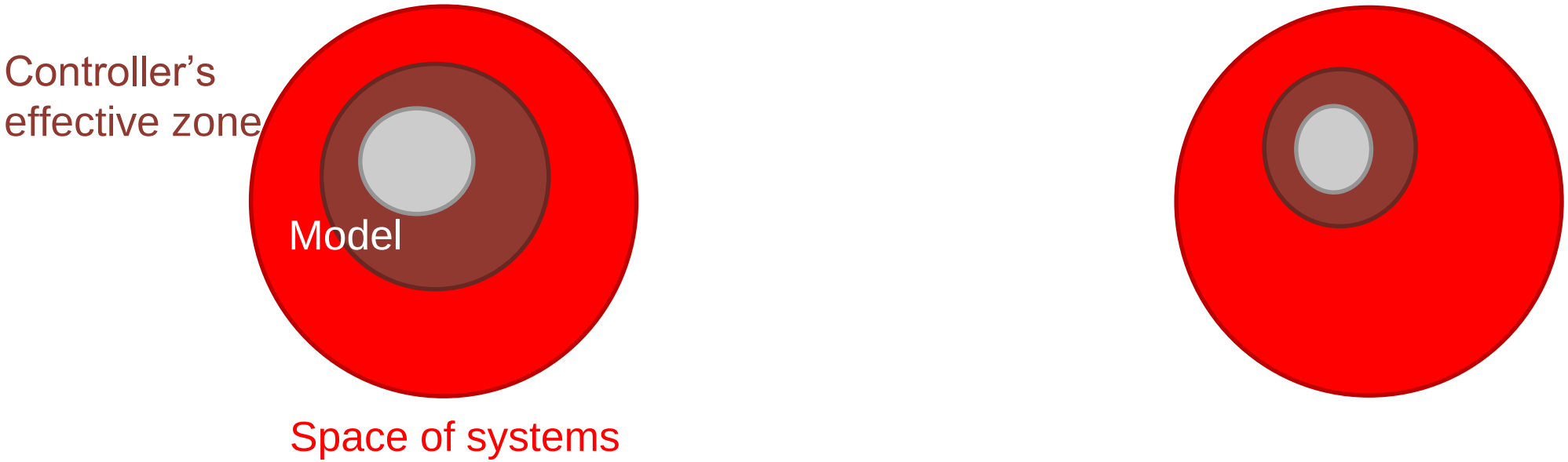
Policy: (system state, value of the reward) → action

- State: system's property,
  - enumerated by designers
  - Possible actions are also enumerated
- Needs no information about system dynamics

Comparison of Approaches

Technique	Model Requirements	Guarantees	Modeling Error Tolerance
Control Theory	Dynamic model	Strong	Formally analyzable
Open-loop Learning	Implicit	Statistical	Confidence
Reinforcement Learning	Model-free	Minimal	Unknown

Generalizability of Implementation



Generalizability: Does control for system A work when the model is very wrong, or for system B which has different inputs, outputs?

Robustness: Does control for system A work for system A', or if you used an inaccurate model for system A?

Why is Generalizability of Control Important?

- Computing systems are built to be generalizable
  - Computers run many types of applications
  - Each application can also have its own set of inputs, outputs
- Computing landscape is changing rapidly
  - New designs emerge quickly
- A universal control system avoids re-designing for each new system

Comparison of Approaches

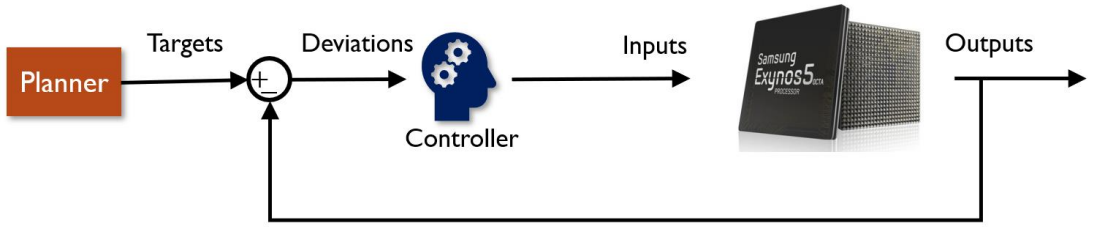
Technique	Model Requirements	Guarantees	Modeling Error Tolerance	Generalizability of Implementation
Control Theory	Dynamic model	Strong	Formally analyzable	Low
Open-loop Learning	Implicit	Statistical	Confidence	High for some types
Reinforcement Learning	Model-free	Minimal	Unknown	Medium
	Low	Strong	Formal	High

Insights to Combine Control and Learning Approaches

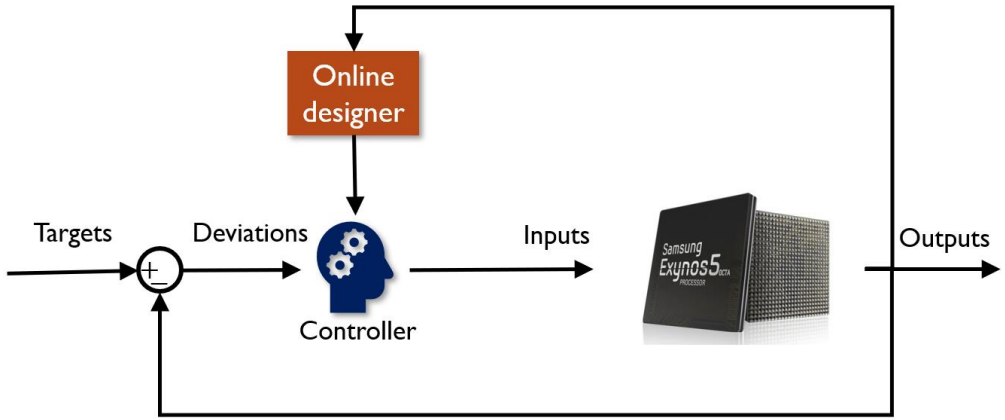
- Control has stronger formal guarantees
  - We like this!
- Control has more demanding requirements on modeling
  - We don't like this, but…
- …models capture structure: system's dynamic evolution, tradeoffs
  - E.g., humans capture the most relevant structure, and this enables correct operation under large modeling/training errors
  - We like this!
  - Machine Learning is a powerful tool to learn structure

Some Combinations of Learning and Control

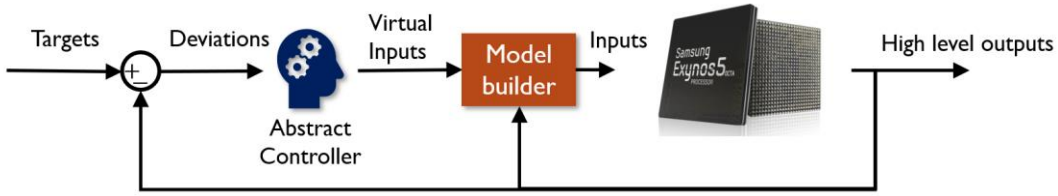
- Learning to set targets for controllers
  - targets met using standard control designs



- Learning to set controller hyper-parameters
  - guardbands, predictions horizon, etc.



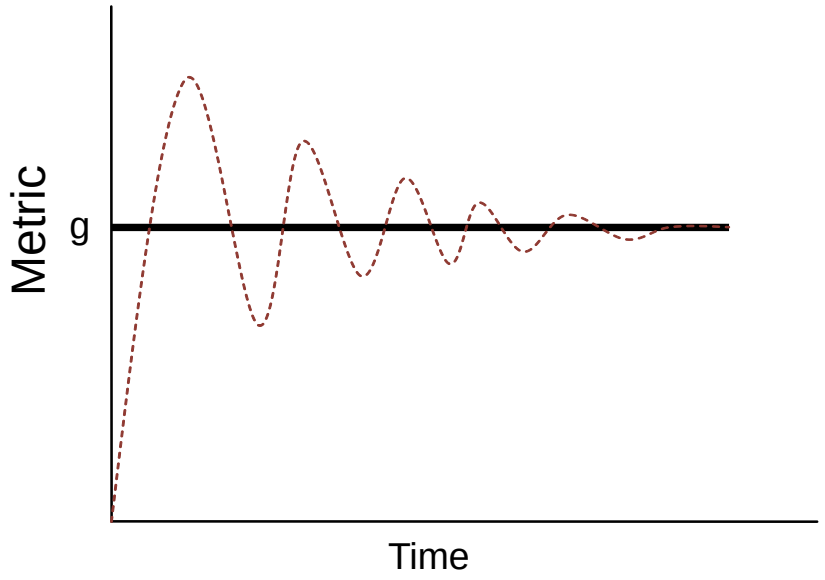
- Abstract control
  - Virtualize structure: inputs, outputs and dynamics



Intuition for Abstract Control

Control System

- Formalize Dynamic Response



- But: Model is system specific

Machine Learning

- Generalizable Models

- But: No Dynamic Formalism

Combination Should Allow Generalization of Control

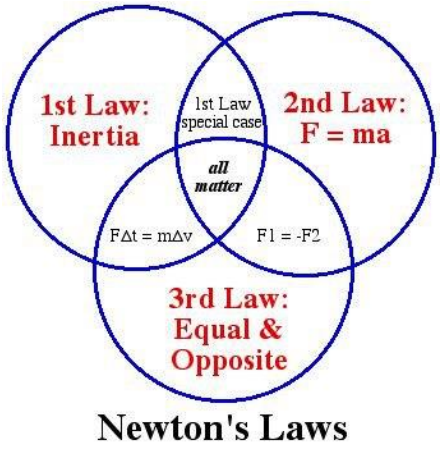
Case Study with ML as Model Builder

- Goal:
  - Create a generalized PID implementation
  - Handle multiple inputs
  - Handle categorical parameters
  - Add optimization
  - One implementation that can control many things
- Encode dynamic models in a library
- Users specify what can be controlled and what the goals are
  - Advanced users could have some input into dynamic response
- Allow learning (machine or human) to customize control to specific problem

The World vs. Computer Engineering

All Other Engineering

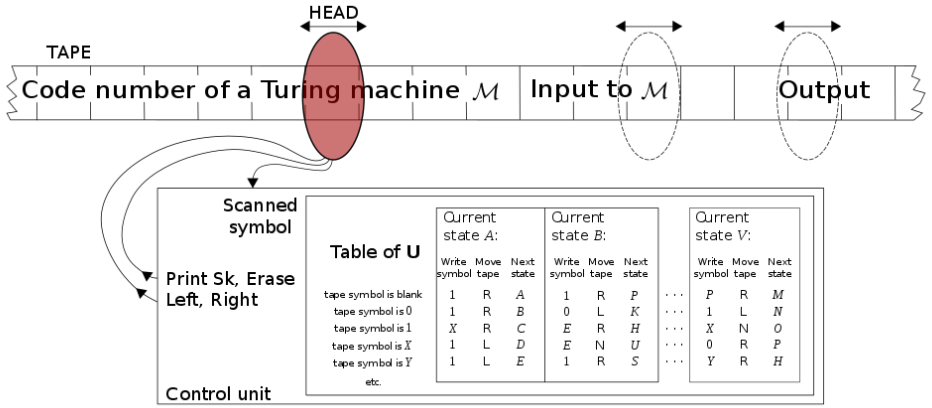
- General Theory



- Specific Implementation

Computer and Software Engineering

- General Theory



- General Implementation

Controlling Computer Systems

Traditional Control Engineering

- General Theory

$$\begin{cases} \dot{X} = AX + Bu \\ y = CX + Du \end{cases}$$

- Specific Implementations

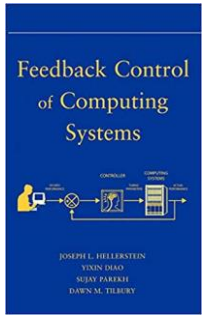
$$speed_k = \alpha * torque_{k-1}$$

$$error_k = \textbf{desired speed} - speed_k$$

$$torque_k = torque_{k-1} + \frac{1 - \text{pole}}{\alpha} * error_k$$

Computer and Software Control Engineering

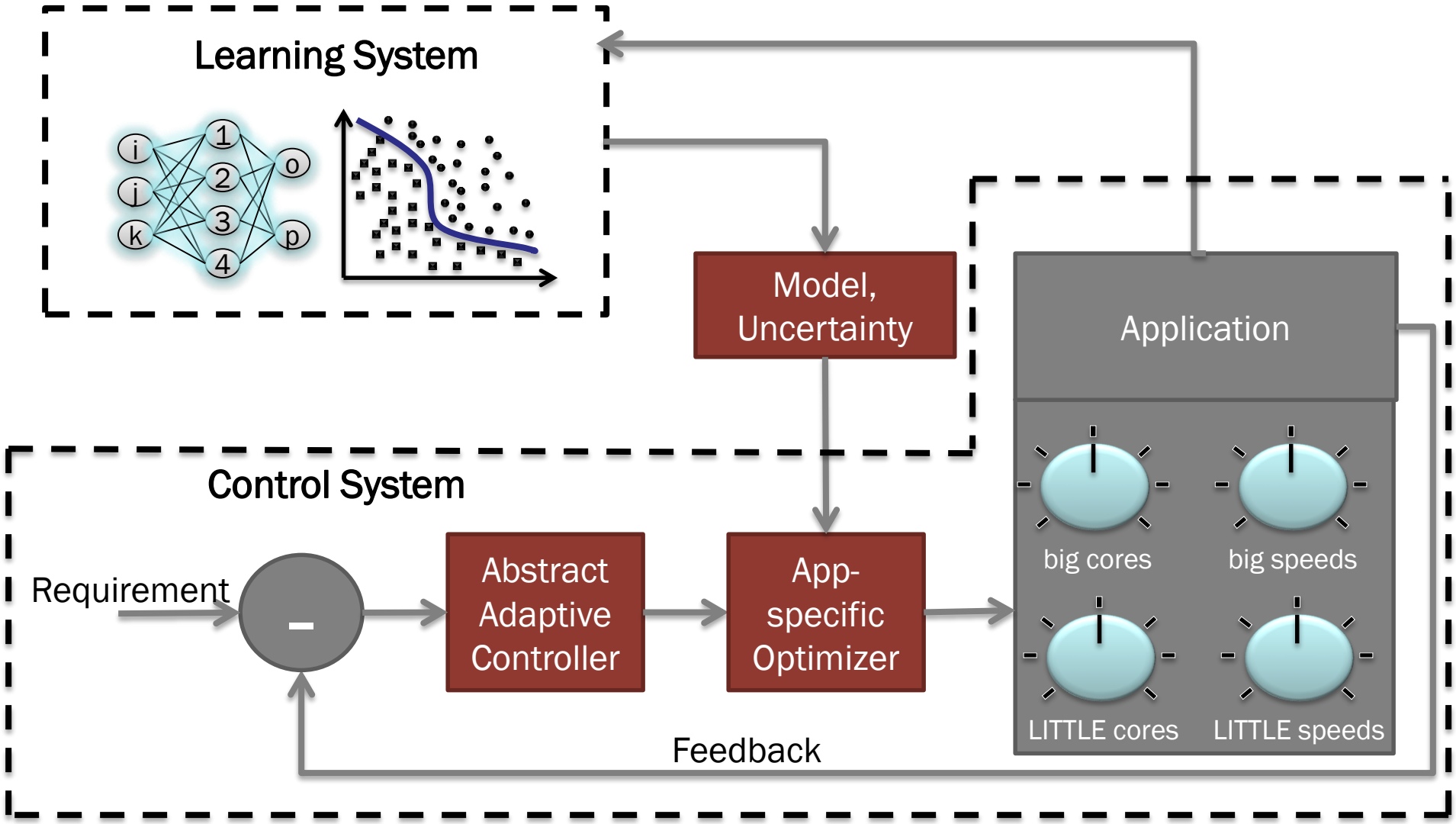
- General Theory



- General Implementations

CALOREE: Combining Learning and Control

[ASPLOS 2018, SysML 2018, IEEE Micro Top Picks HM]

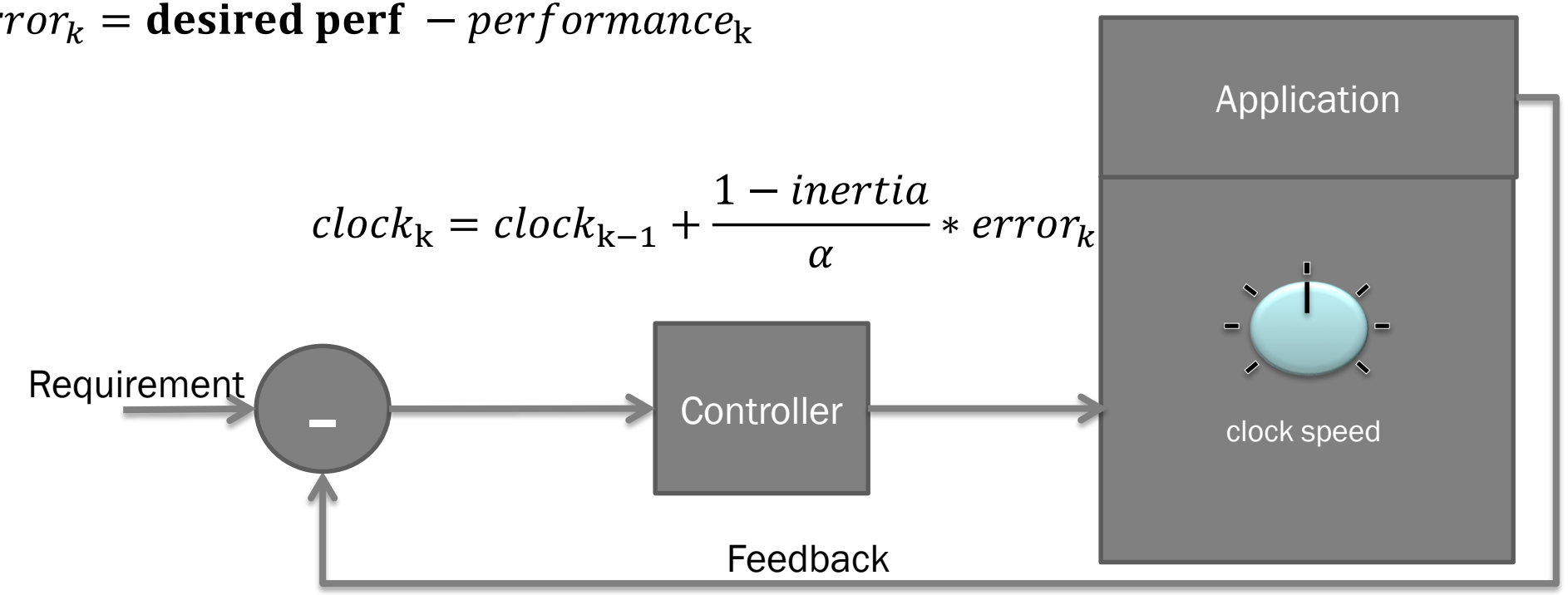


Reminder: Typical Control Systems

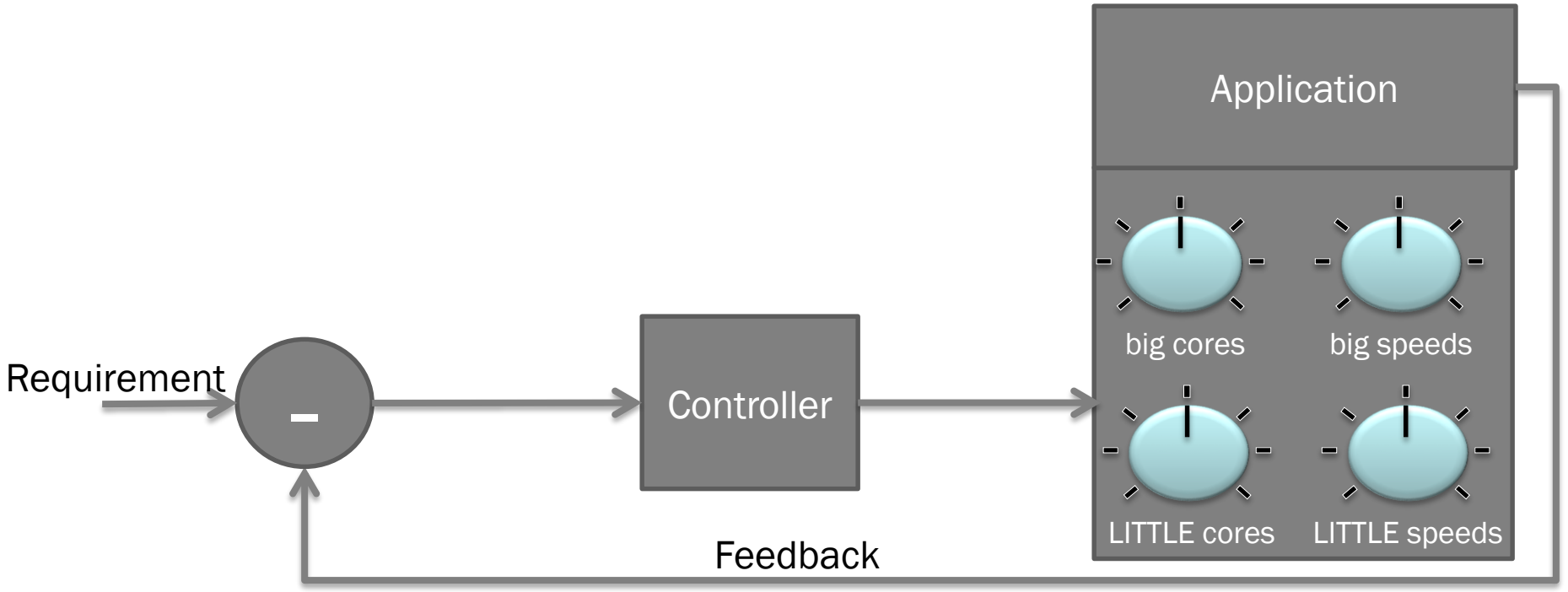
$$performance_k = \alpha * clock_{k-1}$$

$$error_k = \textbf{desired perf} - performance_k$$

$$clock_k = clock_{k-1} + \frac{1 - inertia}{\alpha} * error_k$$

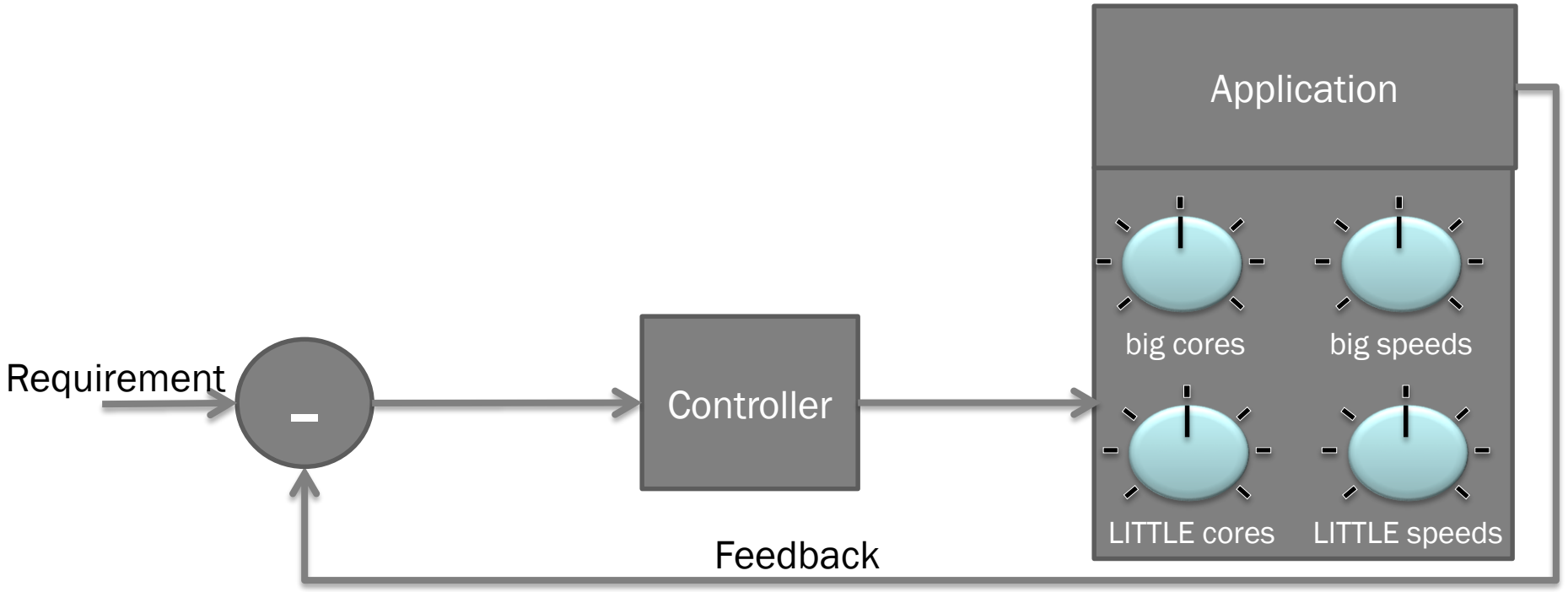


Reminder: Typical Control Systems



Reminder: Typical Control Systems

Specific Implementation Useless on New System

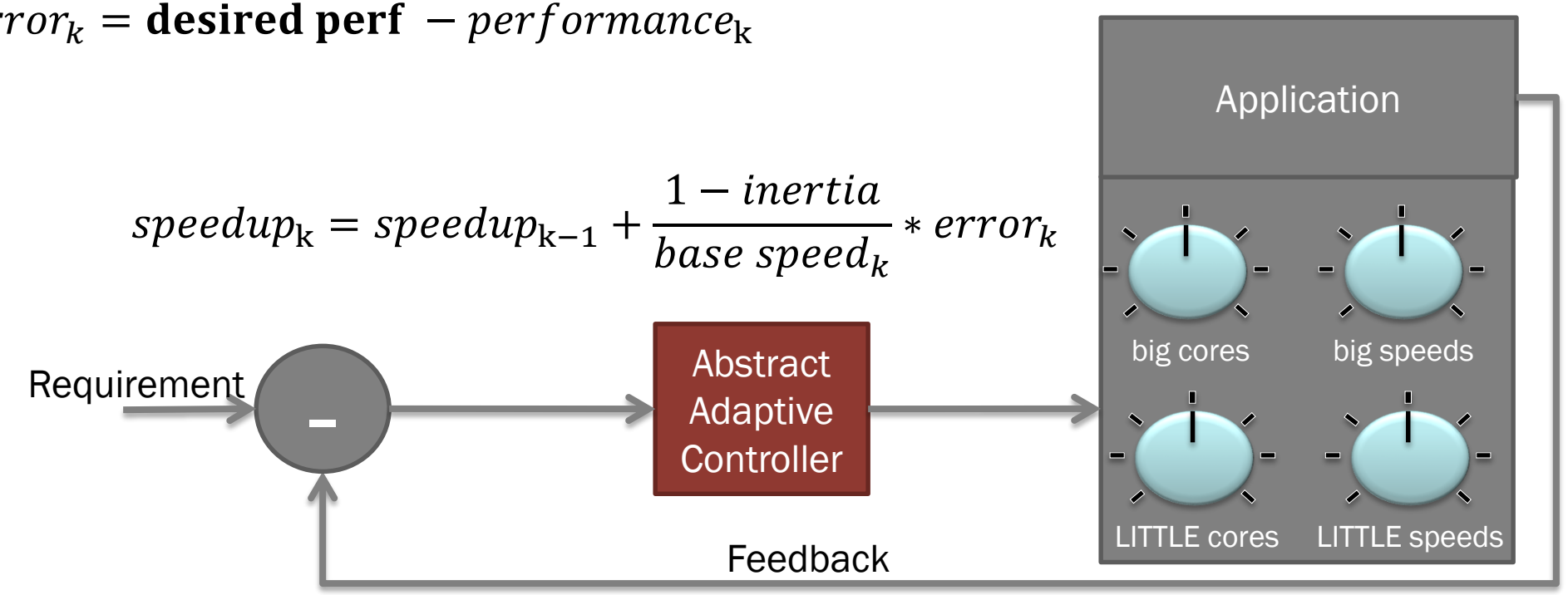


Computer Engineering: Generalized Control Implementations?

$$performance_k = base\ speed_k\ * speedup_{k-1}$$

$$error_k = \textbf{desired perf}\ - performance_k$$

$$speedup_k = speedup_{k-1} + \frac{1 - inertia}{base\ speed_k} * error_k$$



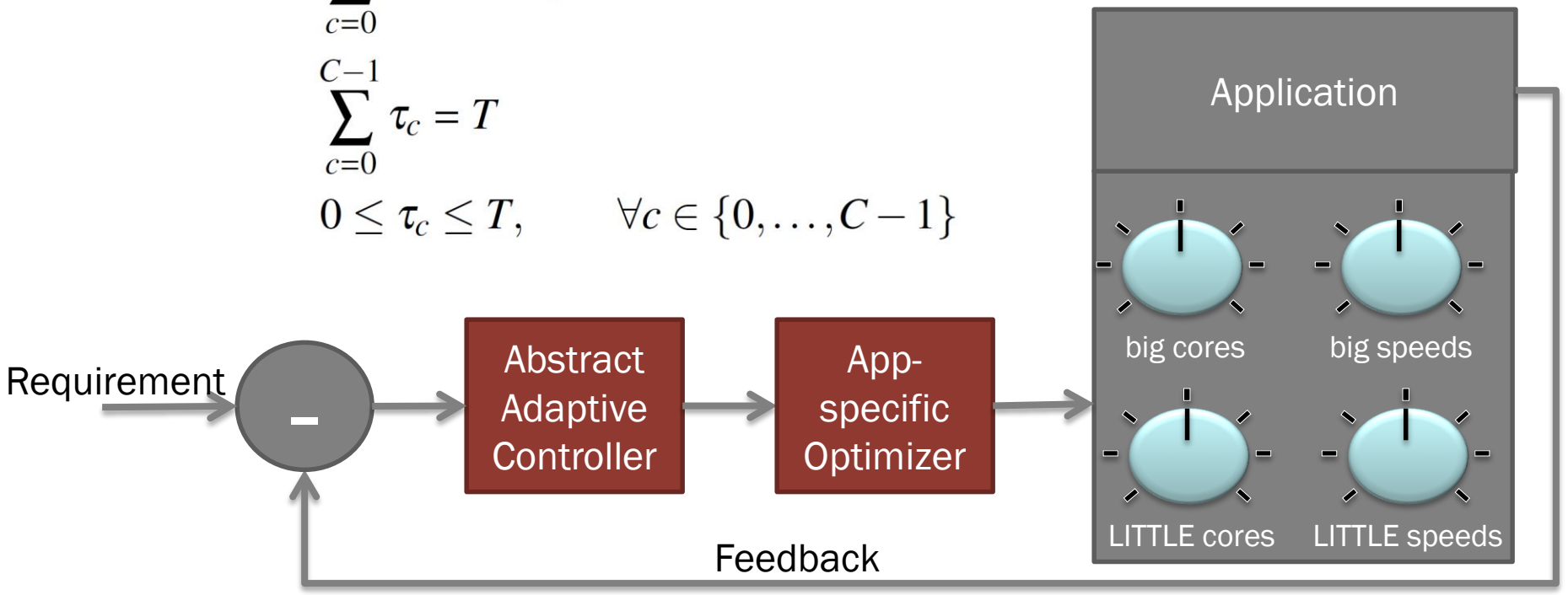
Translating Speedup Into Configurations

$$\underset{\tau \in \mathbb{R}^C}{\text{minimize}} \quad \sum_{c=0}^{C-1} \tau_c \cdot p_c$$

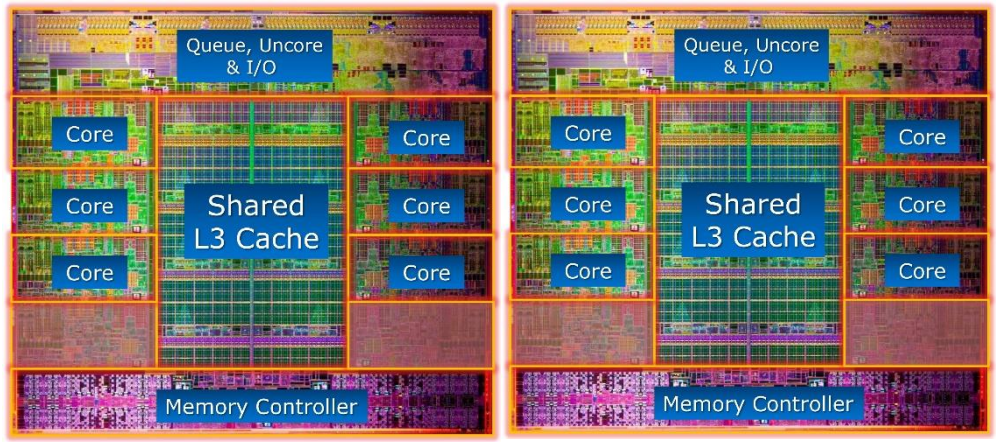
$$s.t. \quad \sum_{c=0}^{C-1} \tau_c \cdot s_c = speedup(t)T$$

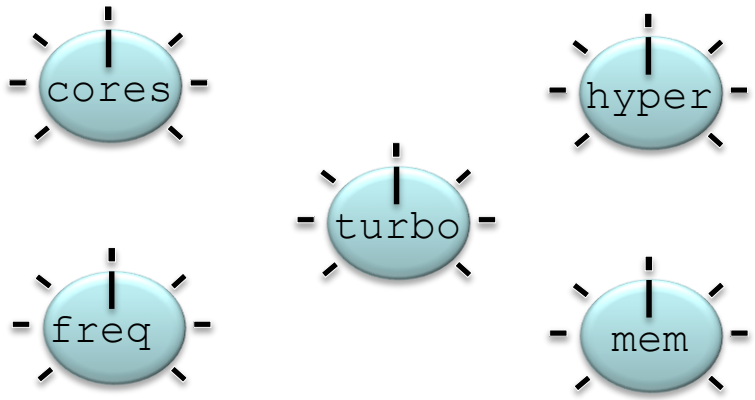
$$\sum_{c=0}^{C-1} \tau_c = T$$

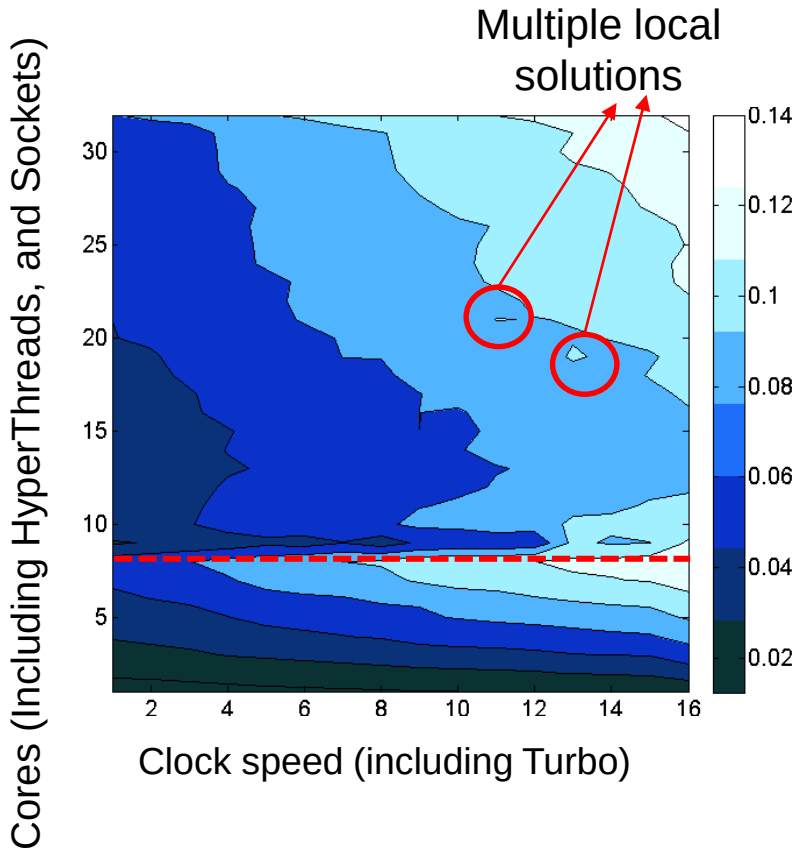
$$0 \leq \tau_c \leq T, \qquad \forall c \in \{0, \ldots, C-1\}$$



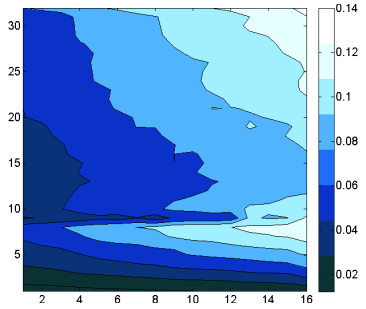
Computing System Complexity

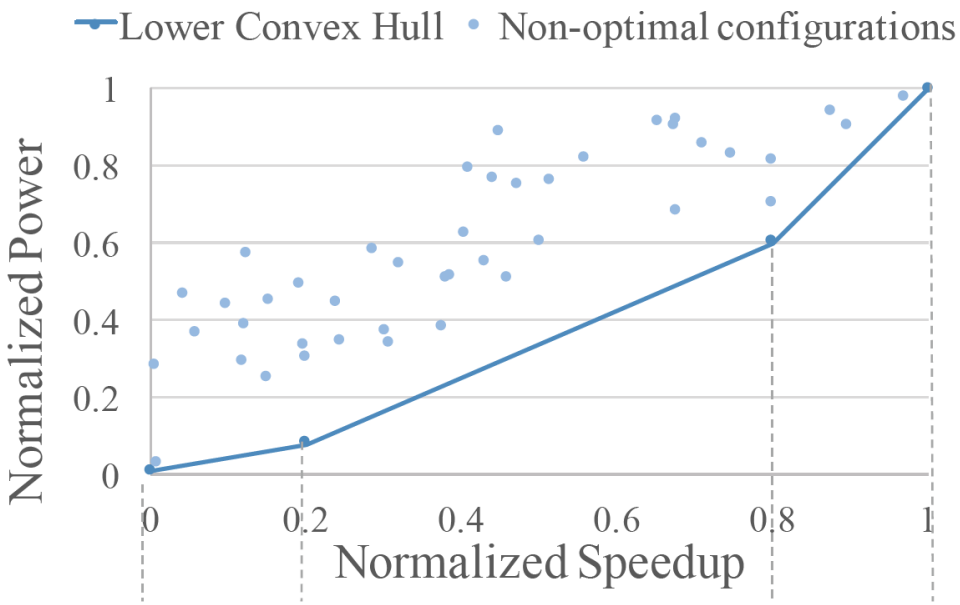






Interface Between Learning and Control

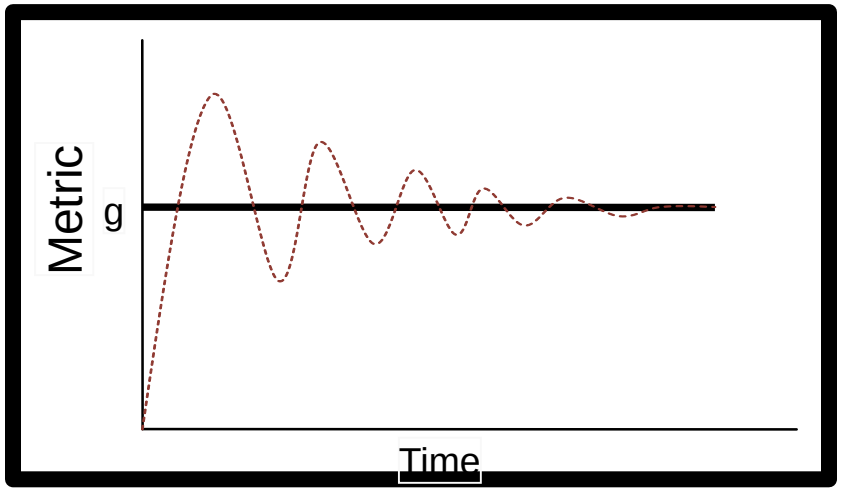




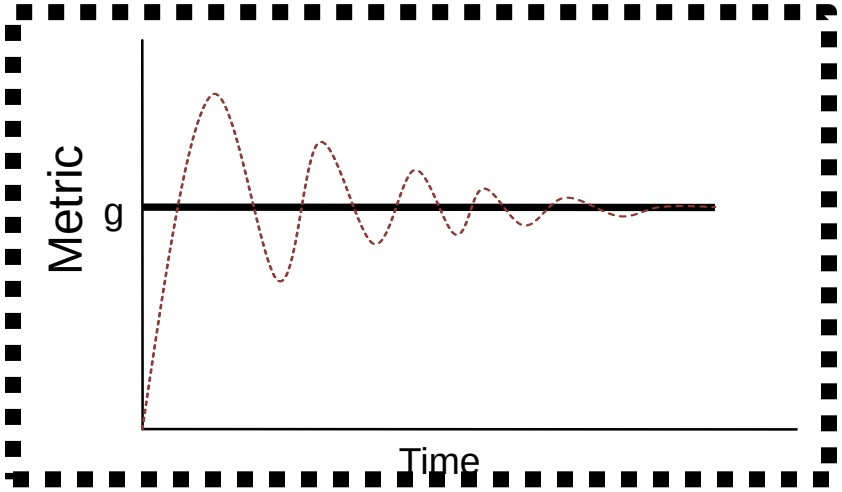
Use This Yourselves!

- poet.cs.uchicago.edu
- Library implementation of abstract, adaptive control system
- https://github.com/libpoet/bard/blob/master/inc/poet.h

Need One More Piece for Stable Control



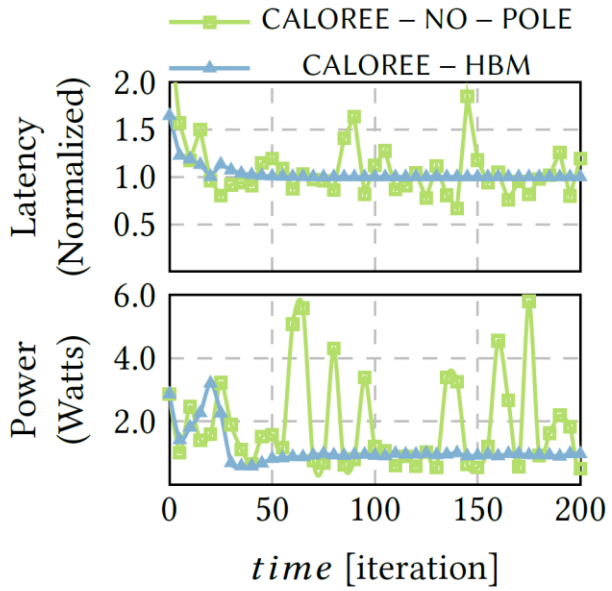
Need One More Piece for Stable Control



$$speedup_{\mathrm{k}} = speedup_{\mathrm{k-1}} + \frac{1 - inertia}{base\ speed} * error_k$$

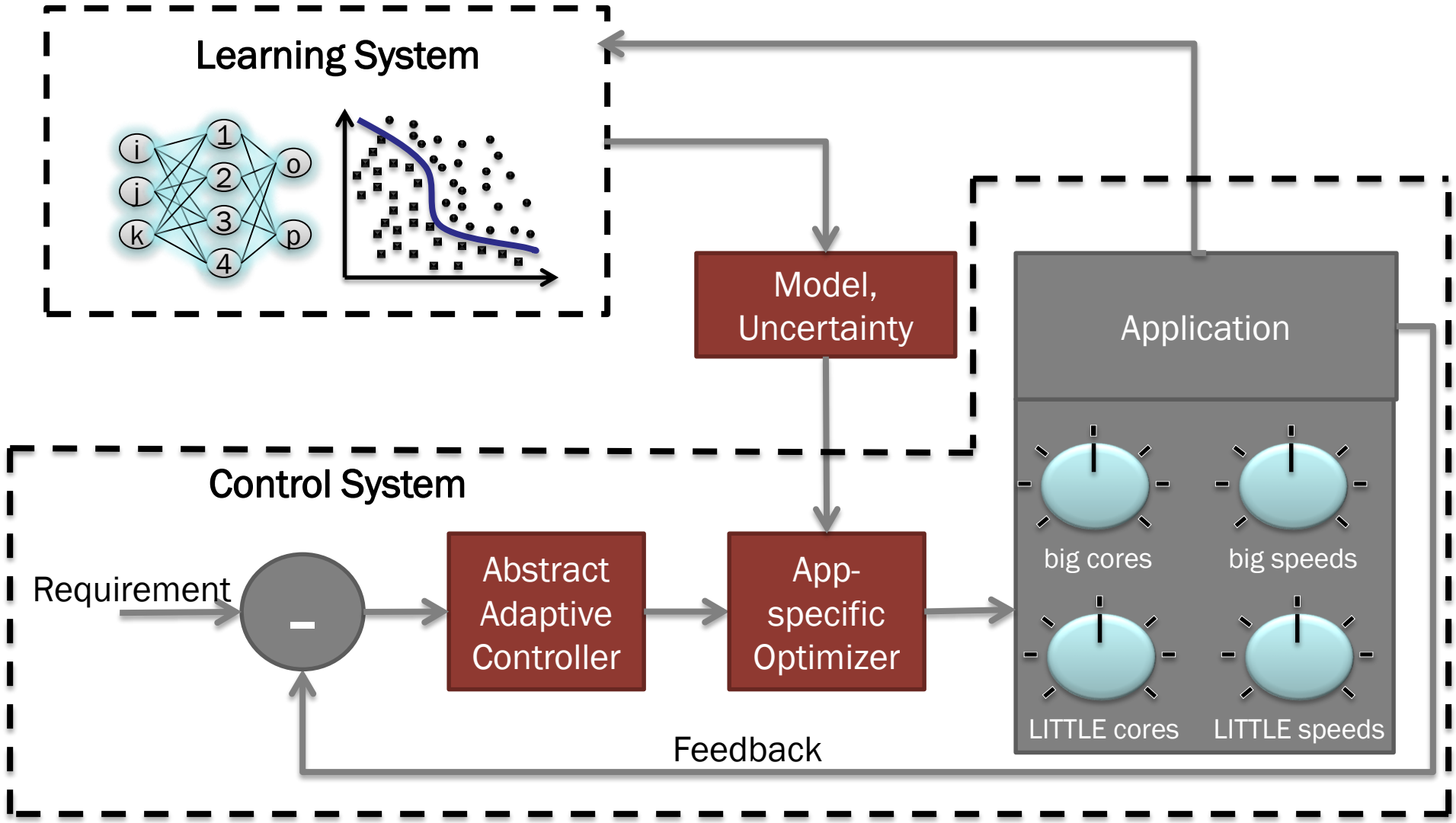
~~$0 \leq \text{inertia} < 1$~~

$$\left\lfloor 1 - \left\lfloor \frac{\max \widehat{speedup}}{\min \widehat{speedup}} - 3\sigma \right\rfloor \right\rfloor \leq \text{inertia} < 1$$



CALOREE: Combining Learning and Control

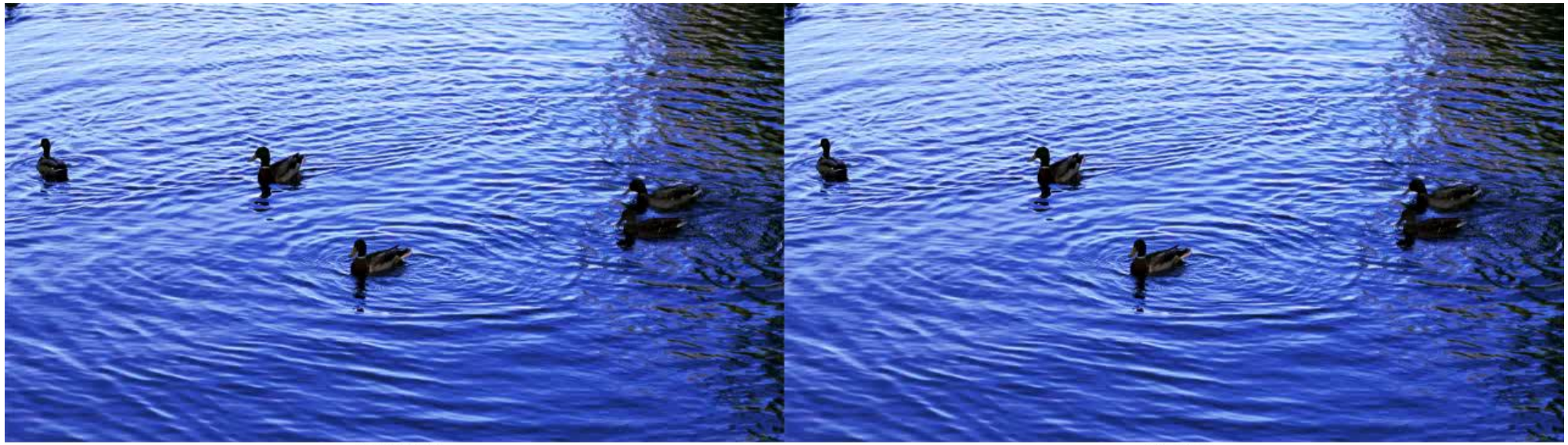
[ASPLOS 2018b, SysML 2018, IEEE Micro Top Picks HM]



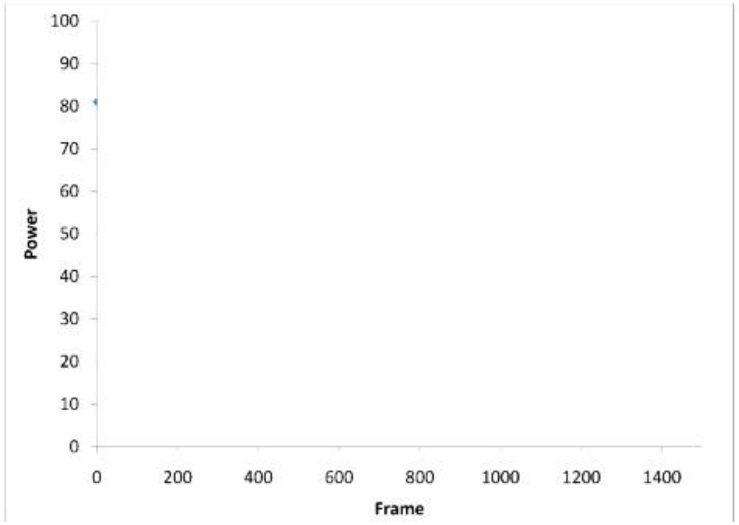
Testing Generality of Self-aware Computing

Video Encoder Example

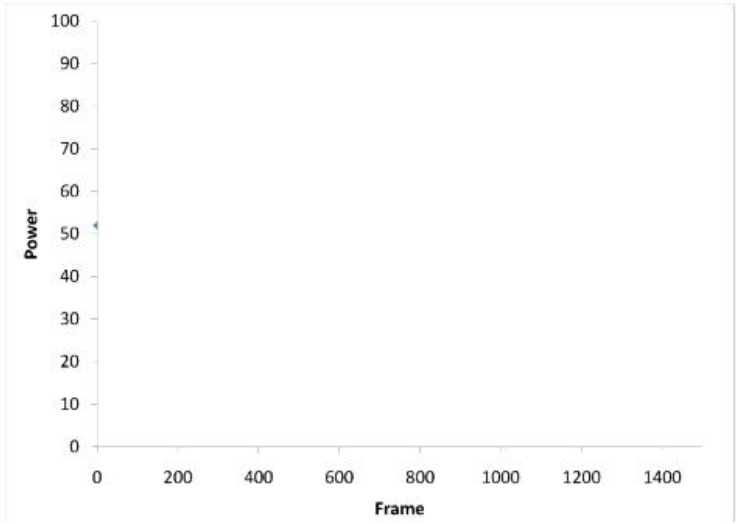
Running on Linux x86



WCET

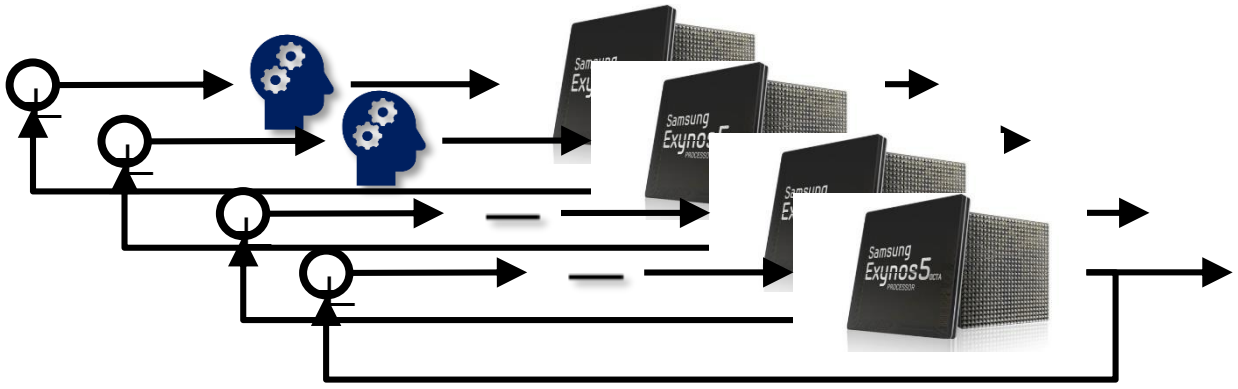


SEEC



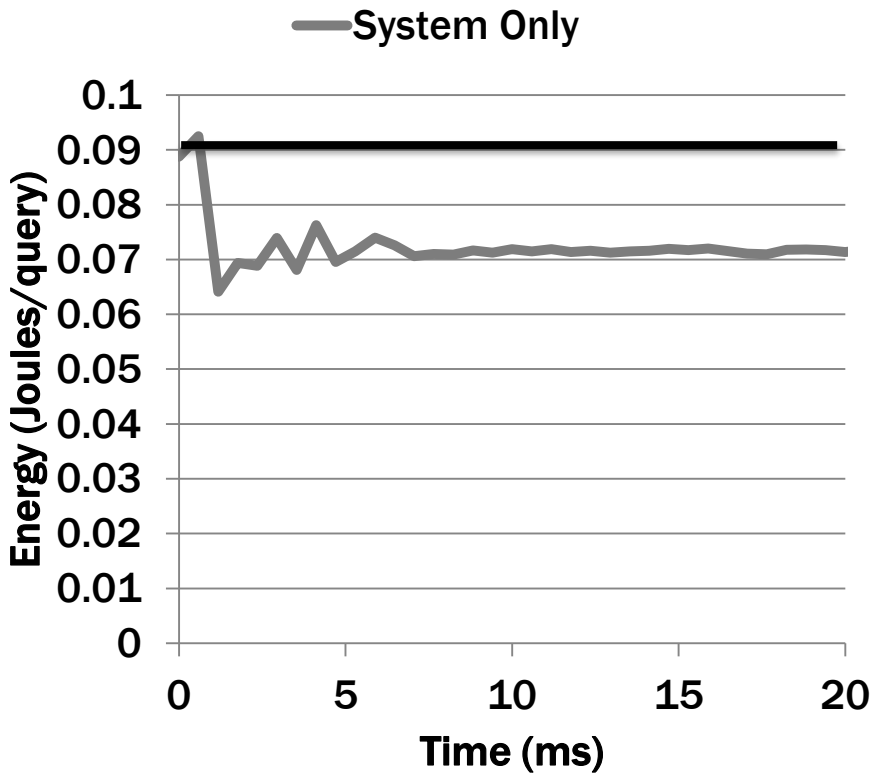
Further Extensions and Research Challenges

- If one controller is good….

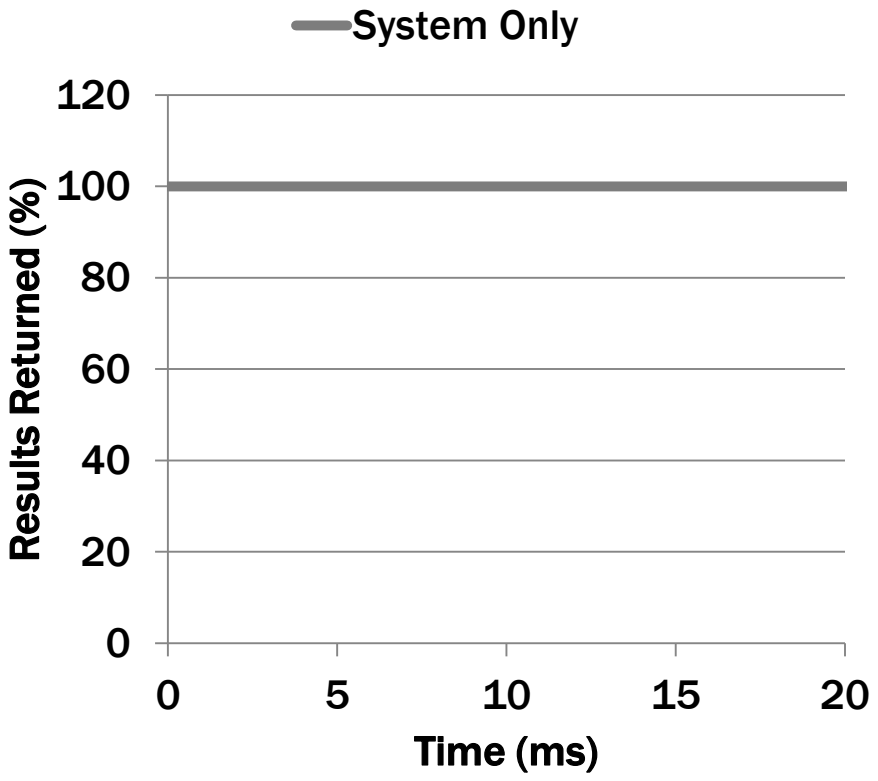


Uncoordinated Control

Energy

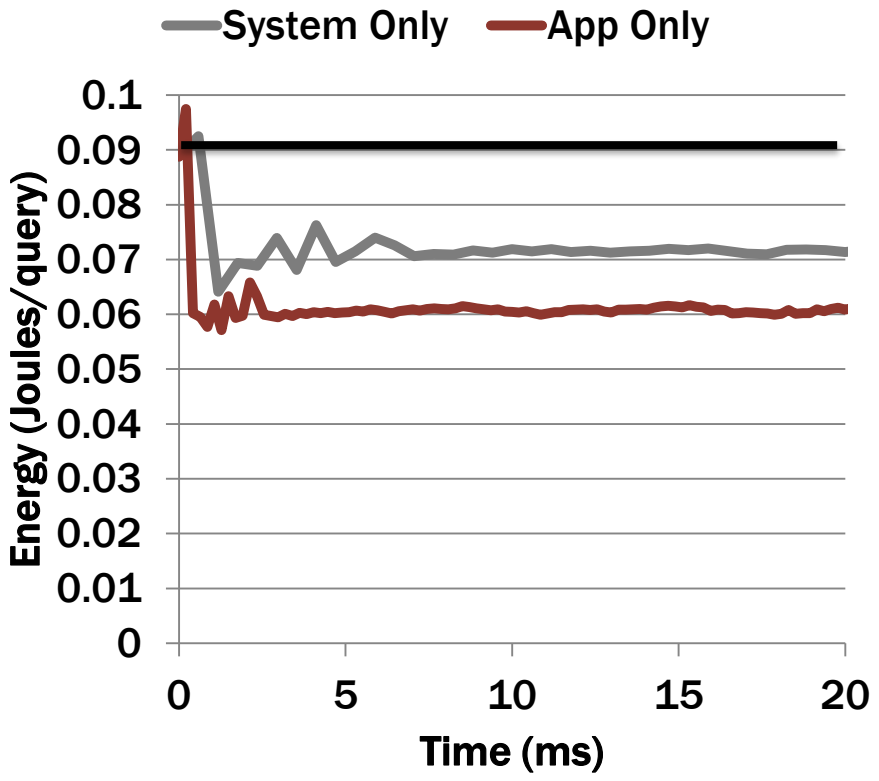


Accuracy

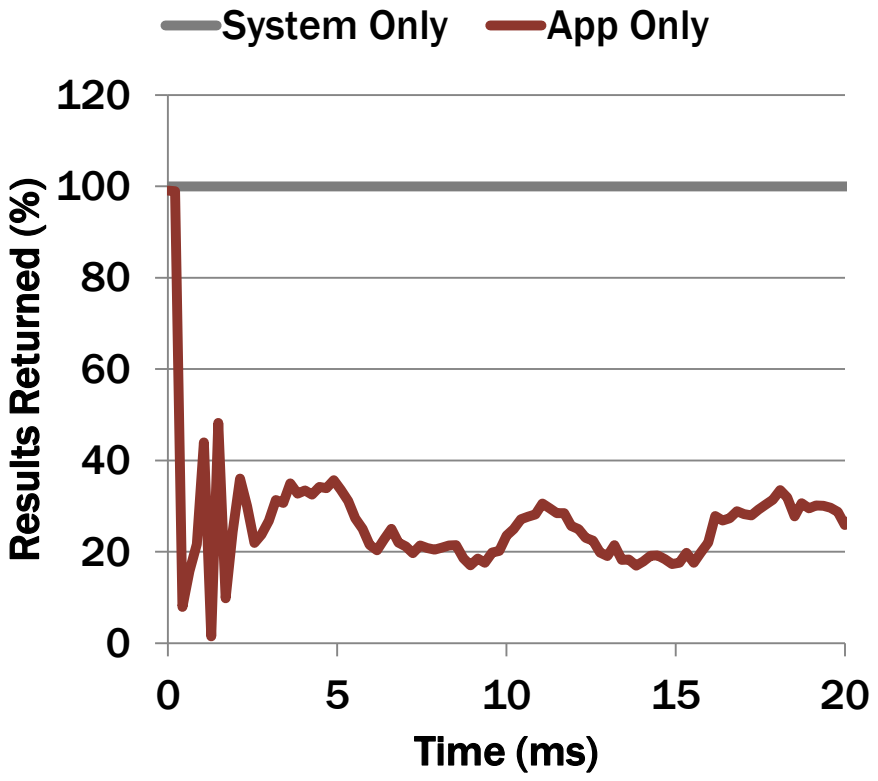


Uncoordinated Control

Energy

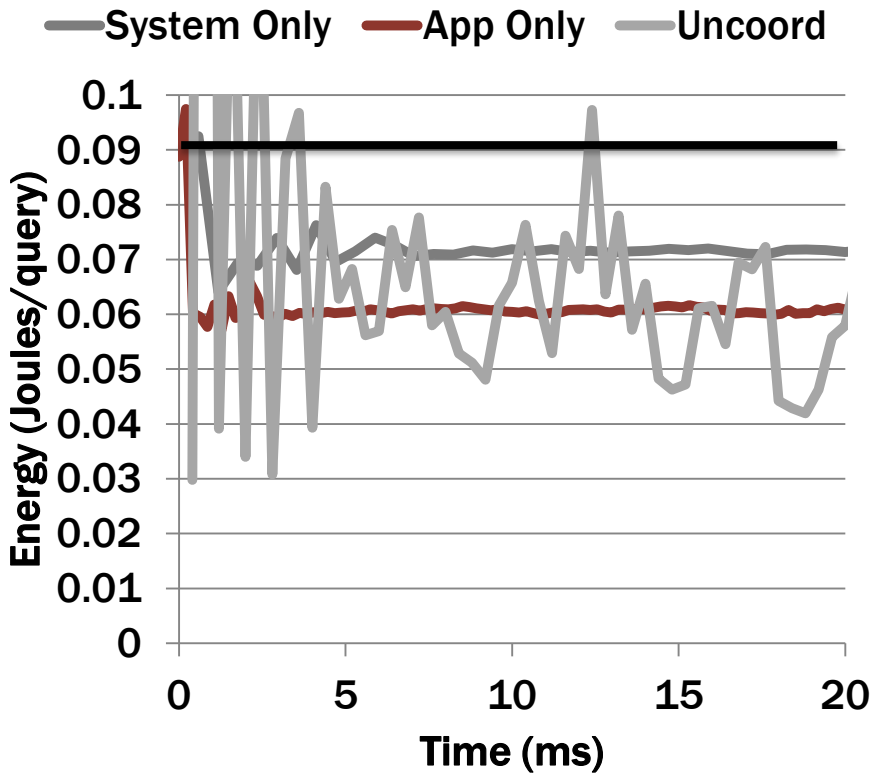


Accuracy

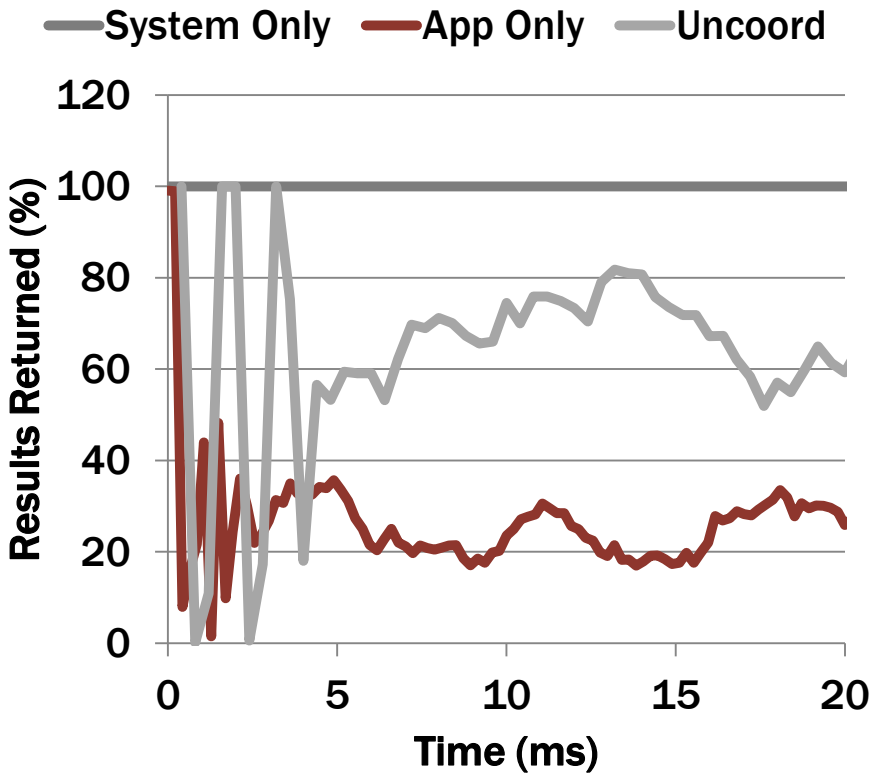


Uncoordinated Control

Energy

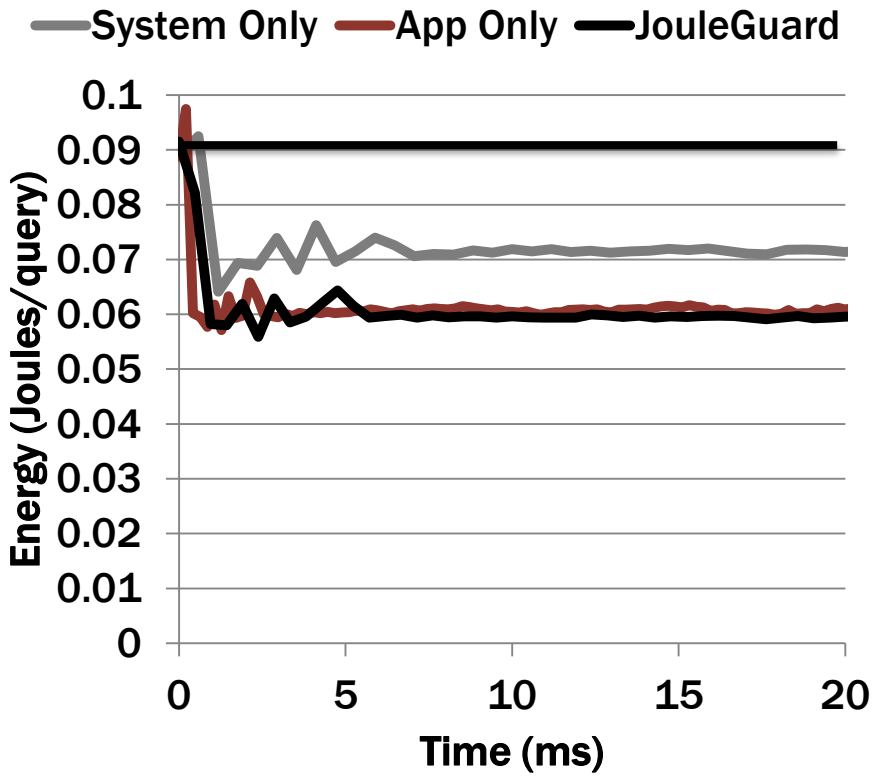


Accuracy

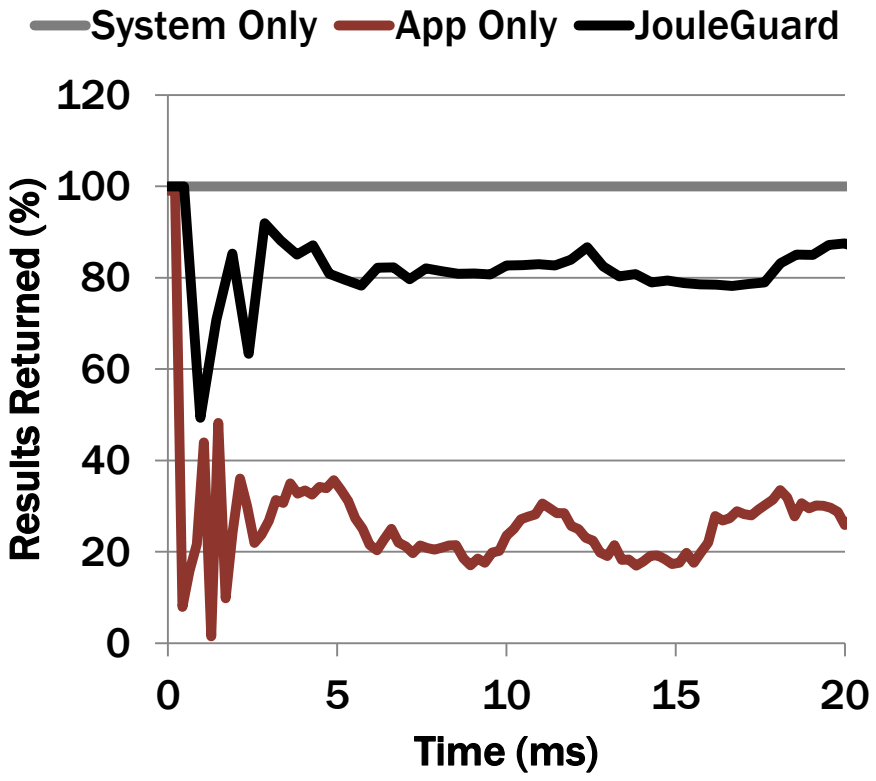


Uncoordinated Control

Energy

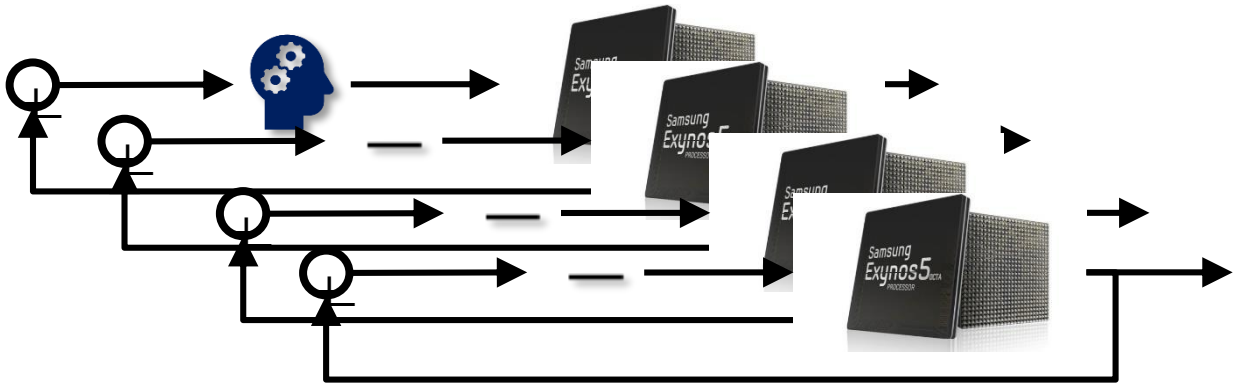


Accuracy



Further Extensions and Research Challenges

- If one controller is good….

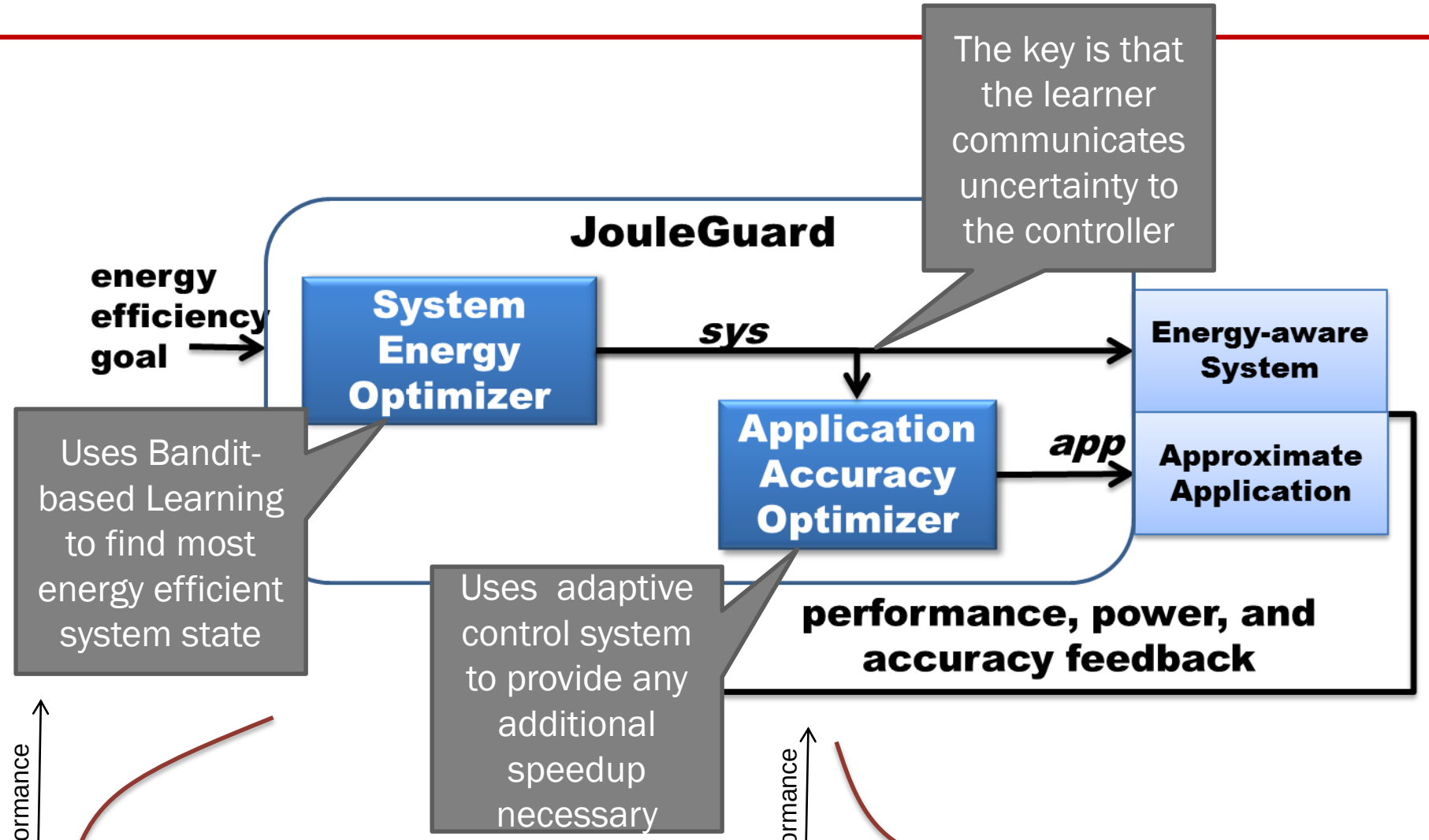


- Coordination of multiple adaptive systems (learning or control)
  will be a fundamental barrier to acceptance

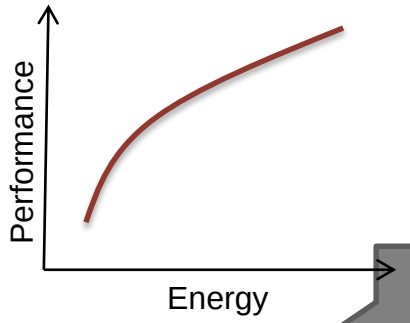
How can we maintain modularity and independence while adapting?

Backup Slides

JouleGuard


JouleGuard
energy efficiency goal
System Energy Optimizer
sys
Application Accuracy Optimizer
app
Energy-aware System
Approximate Application
performance, power, and accuracy feedback
The key is that the learner communicates uncertainty to the controller
Uses Bandit-based Learning to find most energy efficient system state
Uses adaptive control system to provide any additional speedup necessary
Performance
Energy
Performance
Accuracy

Bandit Based Learner



Current Energy Efficiency Estimate

$$\hat{p}_{sys}(t) = (1-\alpha)\cdot\hat{p}_{sys}(t-1) + \alpha\cdot\bar{p}_{sys}(t)$$
$$\hat{r}_{sys}(t) = (1-\alpha)\cdot\hat{r}_{sys}(t-1) + \alpha\cdot\bar{r}_{sys}(t)$$

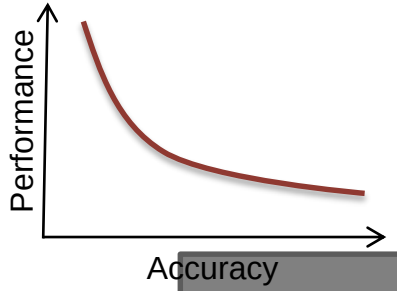
Model Uncertainty

$$\delta(t) = e^{\frac{-\left|\alpha\left(\frac{\bar{r}_{sys}(t)}{\bar{p}_{sys}(t)} - \frac{\hat{r}_{sys}(t)}{\hat{p}_{sys}(t)}\right)\right|}{5}}$$

Exploration Versus Exploitation [Tokic 2010]

$$x(t) = \frac{1-x(t)}{1+x(t)}$$
$$\rho(t) = \frac{1}{|Sys|}\cdot\rho(t) + \left(1-\frac{1}{|Sys|}\right)\cdot\epsilon(t-1)$$

Control Theoretic Optimizer



$$error(t) = r(t) - \bar{r}(t)$$

Difference Between Current Speed and Needed Speed

$$s(t) = s(t-1) + \frac{(1 - pole(t)) \cdot error(t)}{\hat{r}_{bestsys}(t)}$$

Additional Speedup

Key Insight

$\delta(t)$

$$pole(t) = \begin{cases} \delta(t) > 2: & 1 - 2/\delta(t) \\ \delta(t) \leq 2: & 0 \end{cases}$$

Formal guarantees that energy converges to desired target based on adaptive pole placement and gain scheduling – details in paper