

Dependability Evaluation of SISO Control Theoretic Power Managers for Processor Architectures

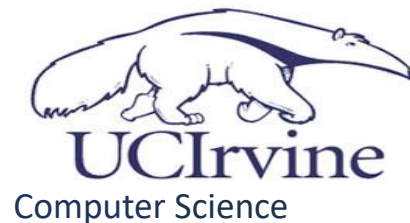
Authors:

Sina Shahhosseini, Kasra Moazzemi, Amir Rahmani, Nikil Dutt

Affiliation:

University of California, Irvine

TU Wien, Austria



Introduction



- The power dissipation has become important constraint in resource management in emerging processors
- Controlling the peak and average power of cores can improve reliability and reduce cost
- Essential to control the power based on application behavior
- Some approaches:

- Ad-hoc approaches

No knowledge about input variation

- Control theoretic

Provides formal guarantees

Requires precise controller design

We study dependability of
control theoretic power
management methods

Control Theoretic Approach



Single-Input Single-Output (SISO) controllers

- Widely used in industry
- Deployed as PI, PID or lead-lag
- Guarantees for stability and robustness

Considerations in design and implementation of SISO:

- Generic average based controllers does not work all the time
- Phasic behavior of applications
- Implementation in various layers

Software controllers

- Ease of implementation and flexibility
- Limited on response time to changes in the system

Hardware controllers

- Fixed implementation
- Higher responsiveness to sensor measurements



Controller design steps



Three main steps in design of SISO controllers

1. System Identification

- Abnormal behaviors in model -> imprecise controller design
- Highlight importance of proper modeling

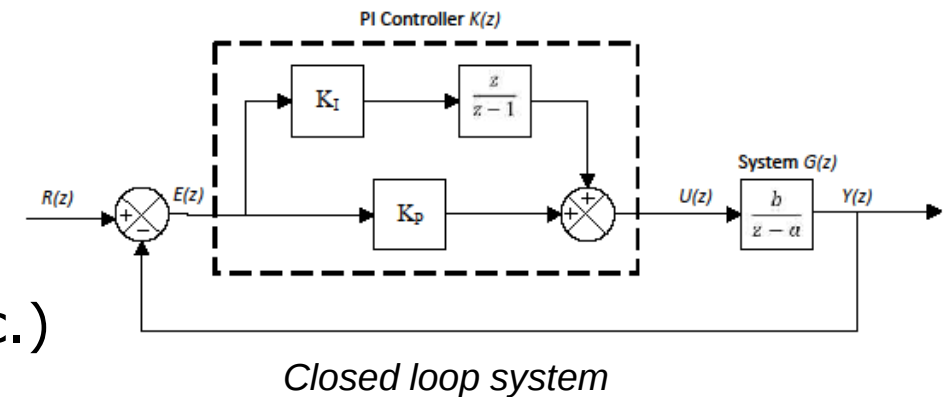
2. Controller design

- Well established field for controller design
- Main focus of the designers
- Multiple methods (Customized, Average, Etc.)

3. Performance Analysis

- Analyze the capabilities of the deployed controller to the test workload

Two steps often-neglected important aspects in the design of a controller. In this work we analyze different cases in each step of controller design



Experimental Setup



In order to study the effects of proper system identification and performance analysis we modified an architectural simulator to a closed loop system

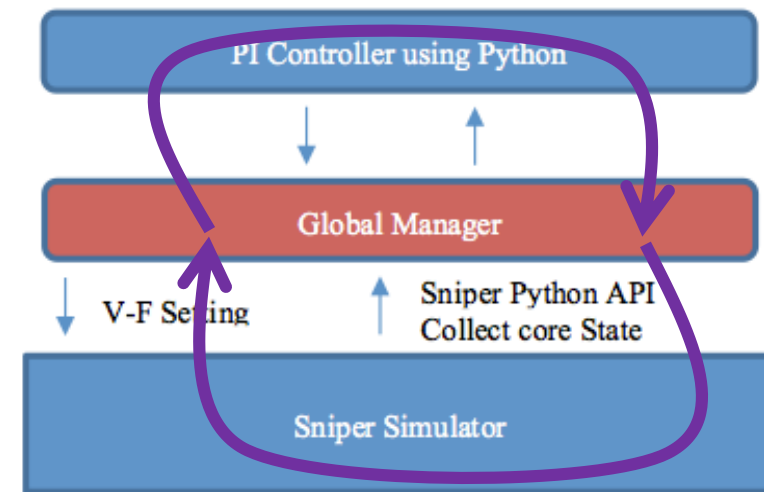
Simulation Framework

- Sniper architectural simulator (Nehalem 45nm)
- McPAT for power estimation
- “Global manager” added to manage DVFS
- Both hardware and software emulation

Benchmark Categorization

- SPLASH2 comprehensive benchmark suite
- Two custom micro benchmarks exercising
 1. Computational Bound
 2. Memory Bound

Lets take a look at some of selected observations from our study...

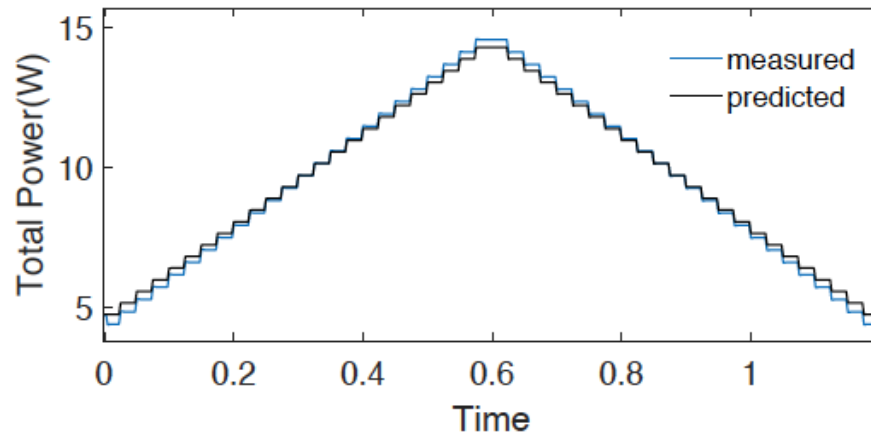


Improved simulator for closed loop runtime DVFS

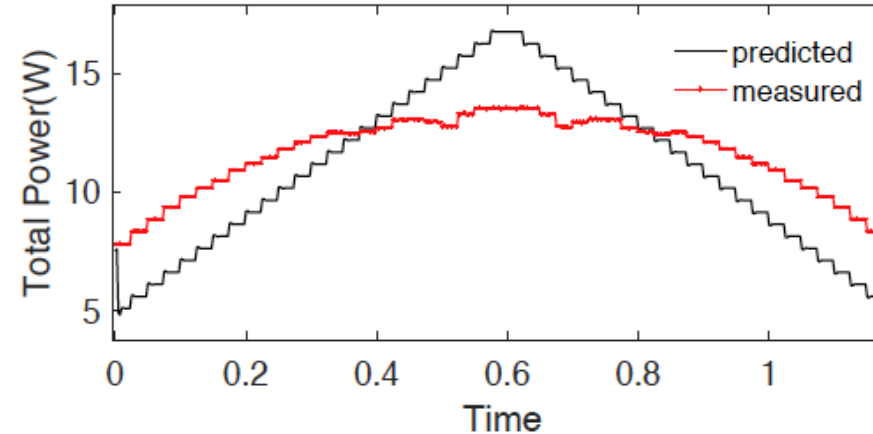
First step: System Identification

(1) Workload types variation

- Change CPU frequency from 1 GHz to 3.3 GHz with steps of 100 MHz
- CPU bound benchmark closely fits the measured model
- Memory bound benchmark lacks the ability to fit into the expected model



CPU Bound



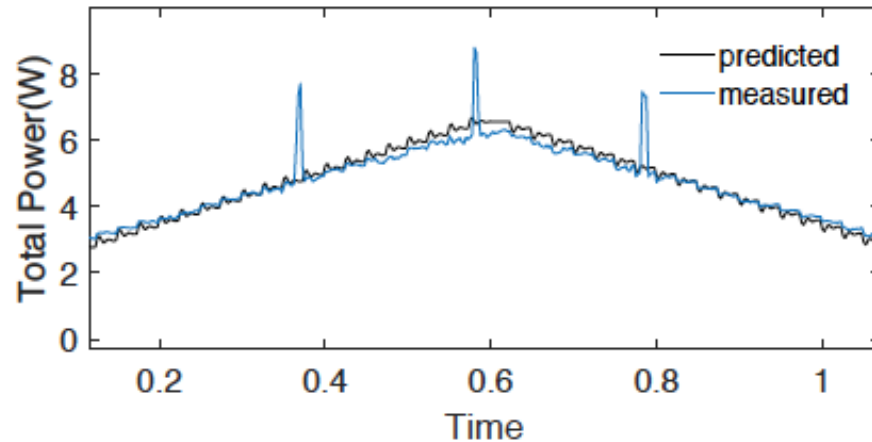
Memory Bound

Diverse workloads exhibit different degrees of reaction to power management control. Selecting correct controller knobs is crucial

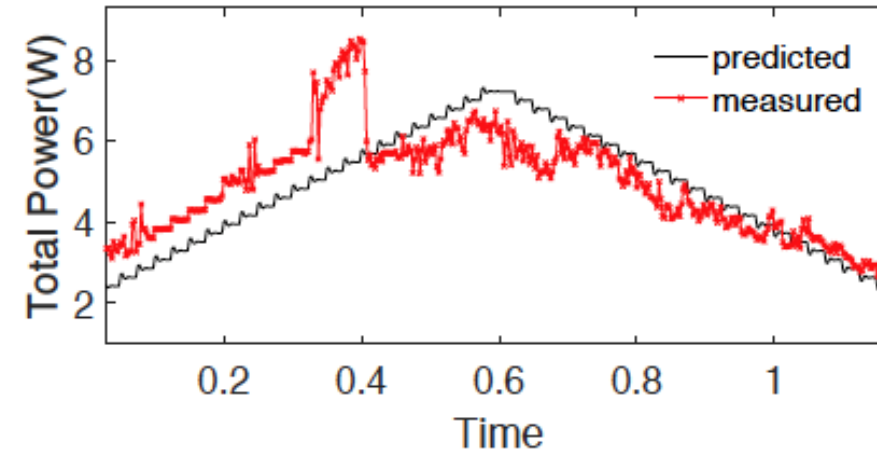
First step: System Identification

(2) Application phasic behavior

- Rapid changes in behavior during system identification
- (left) Barnes workload demonstrating stochastic behavior in shape of spikes
- (right) Identified model for the Raytrace workload exhibits a long period of underestimation



Barnes Workload



Raytrace Workload

Phase changes in system identification can help designers in grasping required controller response

Second step: Controller design



- Widely used practice in control theory
- Comprehensive controller design can be performed by the Matlab
- Typically there are three ways that designers choose to design a controller for a computer system:
 1. Customized case
System identification and controller design stages are performed individually on each class of applications
 2. Average and worst case
Worst case scenario that can respond with slower speed but which provide larger margins of guarantees
 3. Corner Cases
For the workloads that exhibit abnormal behavior in system identification



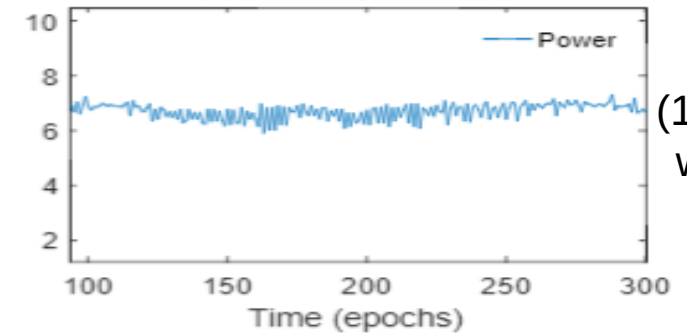
Third step: Performance Analysis



Goal: Follow power references set by designer

(1) Customized controllers

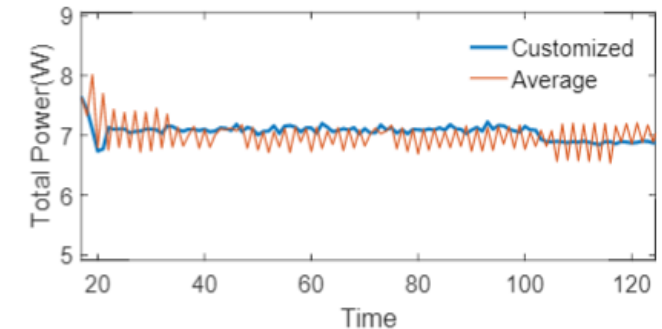
- Example of well-tuned controllers with proper behavior for Volrend benchmark, the power reference is set to 7W



(1) Volrend workload

(2) Average case controllers

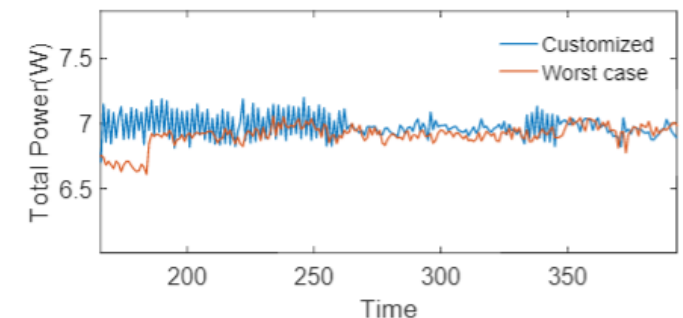
- Fmm workload with average and customized case
- Customized with less ripples vs average with quick changes to follow the reference (rapid)



(2) Fmm workload

(3) Worst case controllers

- Raytraced workload with average and customized case
- Sluggishness to reach the reference but close tracking afterwards (conservative)



(3) Ray workload

Third step: Performance Analysis

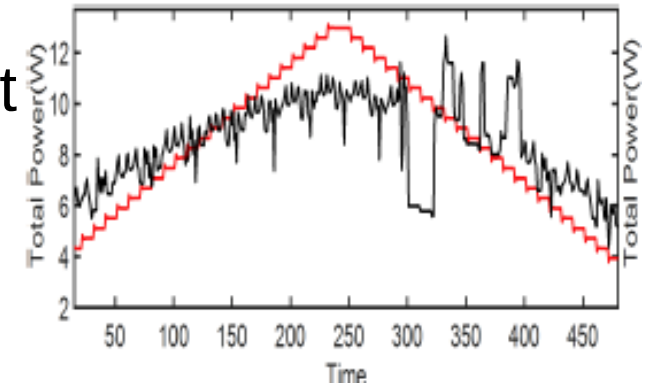


Corner Cases

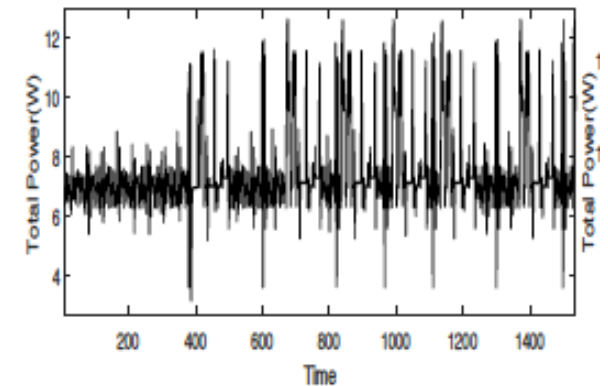
- Ocean Workloads
- In system identification predicted power model does not follow measured power (Top)
- PI- Controller is not able to converge the power to reference (Bottom)

We asked ourselves **WHY?**

- Average power usage higher than other benchmarks and average IPS lower
- Moving global manager to hardware with 10x faster response didn't show improvement
- Further investigation show memory bound features and changes in frequency isn't effective



Ocean benchmark system identification



Ocean benchmark tracking behavior

Discussion & Contributions



For the corner cases

- Not responsive to control knobs
- Corner cases either require more advanced controllers (non-linear, adaptive)
- Or different/more configuration knobs such as memory bandwidth

Each type of control design has their benefits when meeting the requirements

- Hand tuned controllers performs well in all well identified cases
- Average case controller -> responsive and rippled
- Worst case controller -> not fast but reliable

Additional contributions of this paper

- Outline a robustness classification of workloads based on their characteristics
- Identify early symptoms that can cause instability in a controller and lay down guidelines for stable and robust SISO controller design for power management



Conclusion



1. Generic controllers are not always the answer
2. Many applications require proper system identification before controller design
3. Selecting the correct layer for controller deployment is crucial. Some applications require fast response that can only be provided by hardware controllers
4. Responsiveness of workloads to control knobs is essential for a proper power management
5. Sometimes classic SISO controller are not the answer and there is a need for adaptive control. In future works, we try to address new set of controllers for resource management



Questions



Contact Information



Authors:

Sina Shahhosseini, Kasra Moazzami, Amir Rahmani, Nikil Dutt

E-mail:

{sshahhos, moazzemi, amirr1, duttg}@uci.edu

Affiliation:

University of California, Irvine



Background

- Controller design for feedback loops consists of three main stages:
- Modeling
- Controller Design
- Performance evaluation
 - The merit of a controller is measured in terms of
 - Accuracy, Overhead, Robustness and Flexibility
 - Evaluate how well a controller satisfies these properties while executing different types of workloads

$$u(k) = u(k-1) + (K_P + K_I)e(k) - K_P e(k-1)$$

