Energy Efficient Soft Real-Time Computing through Cross-Layer Predictive Control

Guangyi Cao and Arun Ravindran

Department of Electrical and Computer Engineering

University of North Carolina at Charlotte

UNC CHARLOTTE

Organization of Talk

- Motivation
- Related Work
- Cross-Layer Control Framework
- Evaluation Methodology
- Experimental Results
- Future Directions

UNC CHARLOTTE

- Motivation
- Related Work
- Cross-Layer Control Framework
- Evaluation Methodology
- Experimental Results
- Future Directions

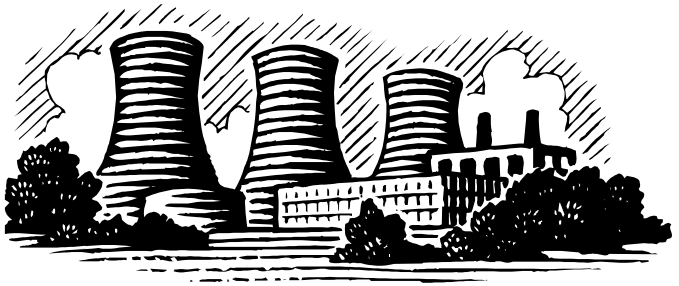
UNC CHARLOTTE

Data Center Energy Consumption

- In 2012, data centers consumed equivalent of 30GW of power



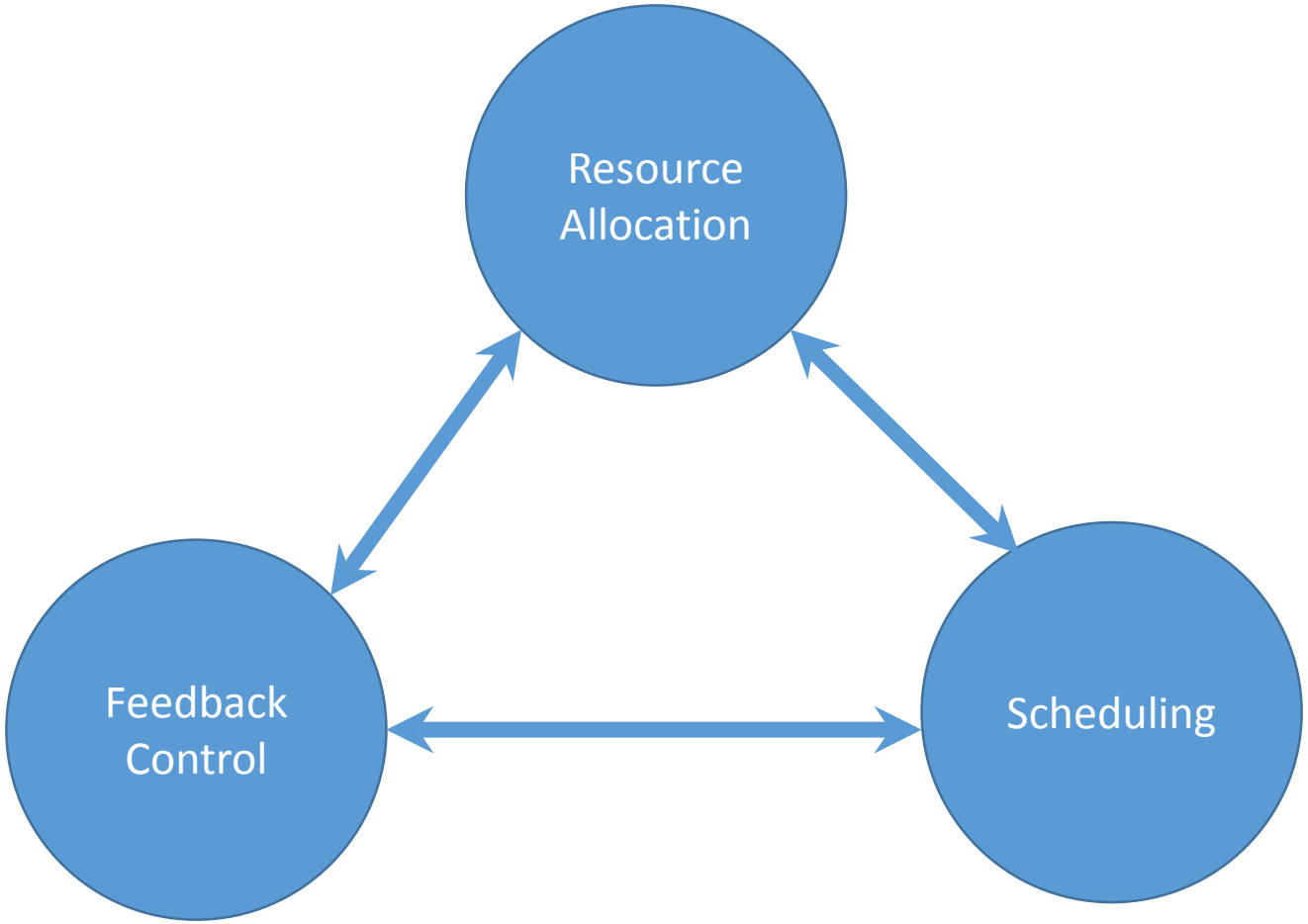
Source: BalticServers, Wikimedia



- Servers typically operate between 10% to 50% of their maximum utilization level
- Server idle power is 50%-60% of the peak power

UNC CHARLOTTE

Energy Efficient Computing



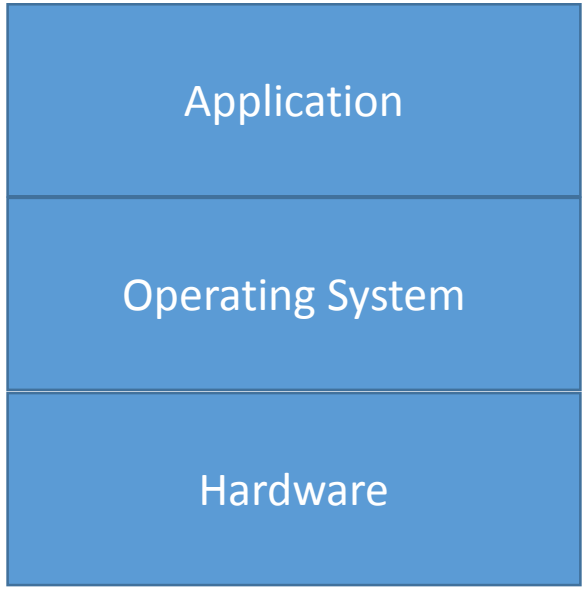
UNC CHARLOTTE

- Motivation
- Related Work
- Cross-Layer Control Framework
- Evaluation Methodology
- Experimental Results
- Future Directions

UNC CHARLOTTE

What we mean by cross layer…

From a computing systems point of view…



UNC CHARLOTTE

Cross layer optimization and control

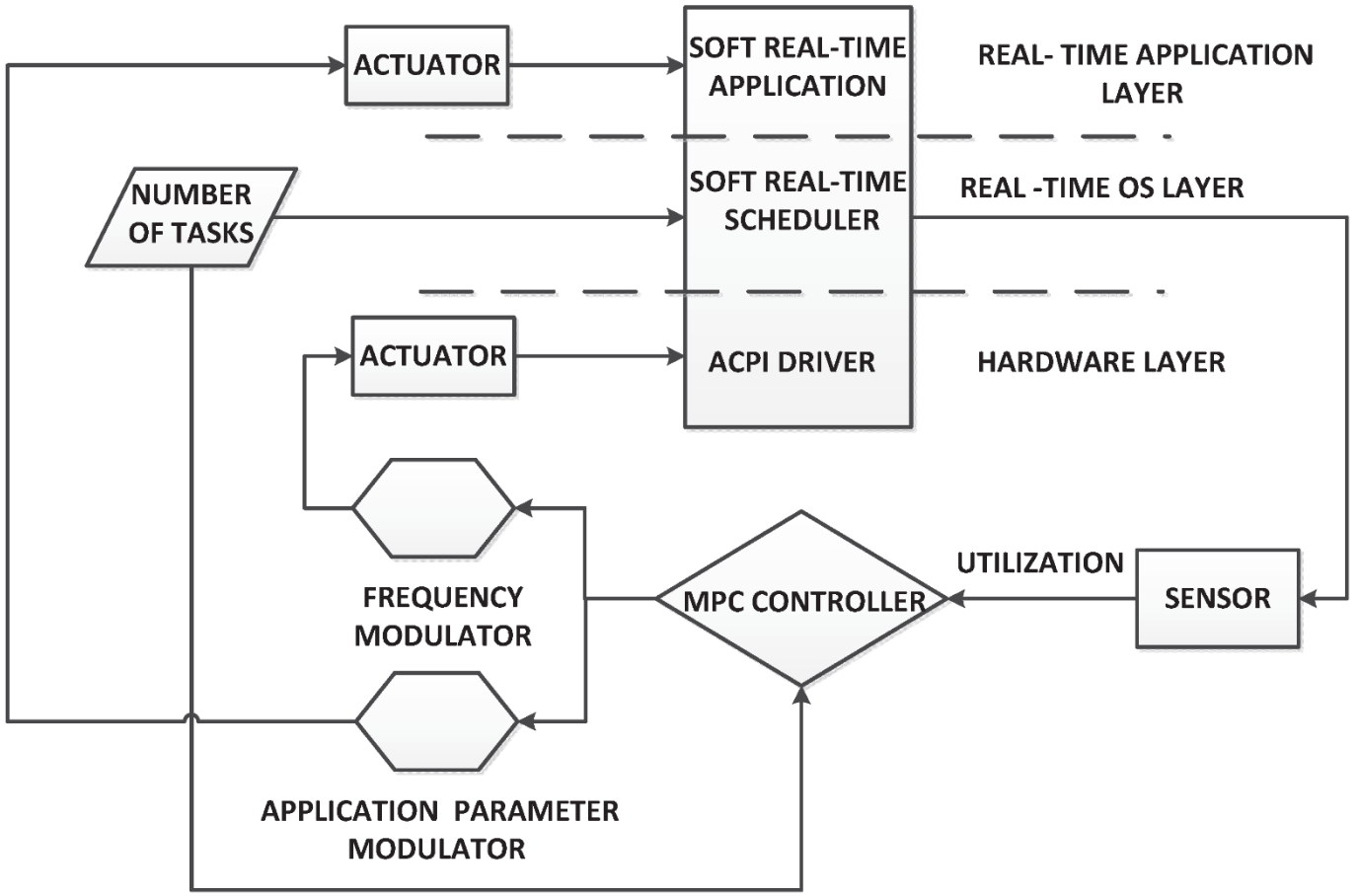
- Several work on single layer feedback control
- Fu et. al. (2011) used Model Predictive Control for cache aware utilization control
- Hoffman et. al. (2013) proposed a control framework for controlling multiple hardware parameters
- Reed et. al. (2013) proposed an application level controller for Apache webserver
- Among cross layer approaches that influenced our work-
  - Illinois GRACE project (2006)
    - DVFS, CPU budget, frame rate and dithering for video decoding
    - Hierarchical optimization
  - Cucinotta et. al. (2010)
    - Cross-layer feedback approach with separate feedback loops
    - Internal loop for resource allocation by controlling scheduling parameters
    - External loop for application quality

UNC CHARLOTTE

- Motivation
- Related Work
- Cross-Layer Control Framework
- Evaluation Methodology
- Experimental Results
- Future Directions

UNC CHARLOTTE

Control Framework



UNC CHARLOTTE

Soft Real Time Schedulers

- Multiprocessor Earliest Deadline First Algorithm
- Previous research (Devi and Anderson) have shown that for soft real-time tasks, bounded tardiness with utilization of m (# of cores) is possible for multi-processor EDF

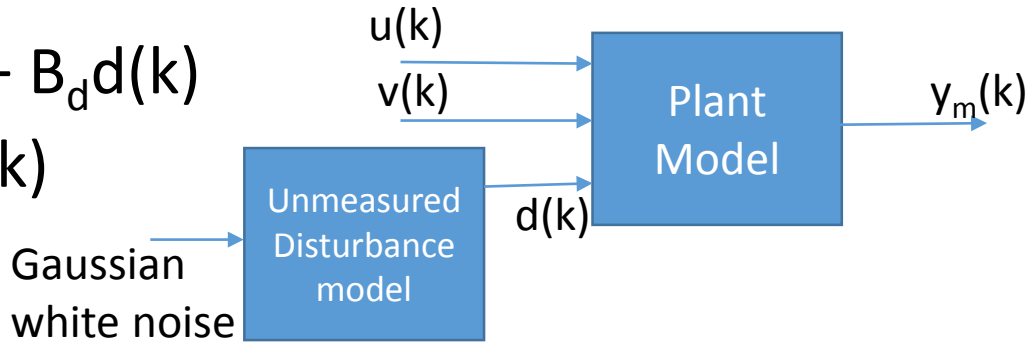
UNC CHARLOTTE

System Model

- LTI State space model

$$x(k+1) = Ax(k) + B_u u(k) + B_v v(k) + B_d d(k)$$

$$y_m(k) = C_m x(k) + D_{vm} v(k) + D_{dm} d(k)$$



$x(k)$ is the n_x-dimensional state vector of the plant

$u(k)$ is the n_u-dimensional vector of manipulated variables

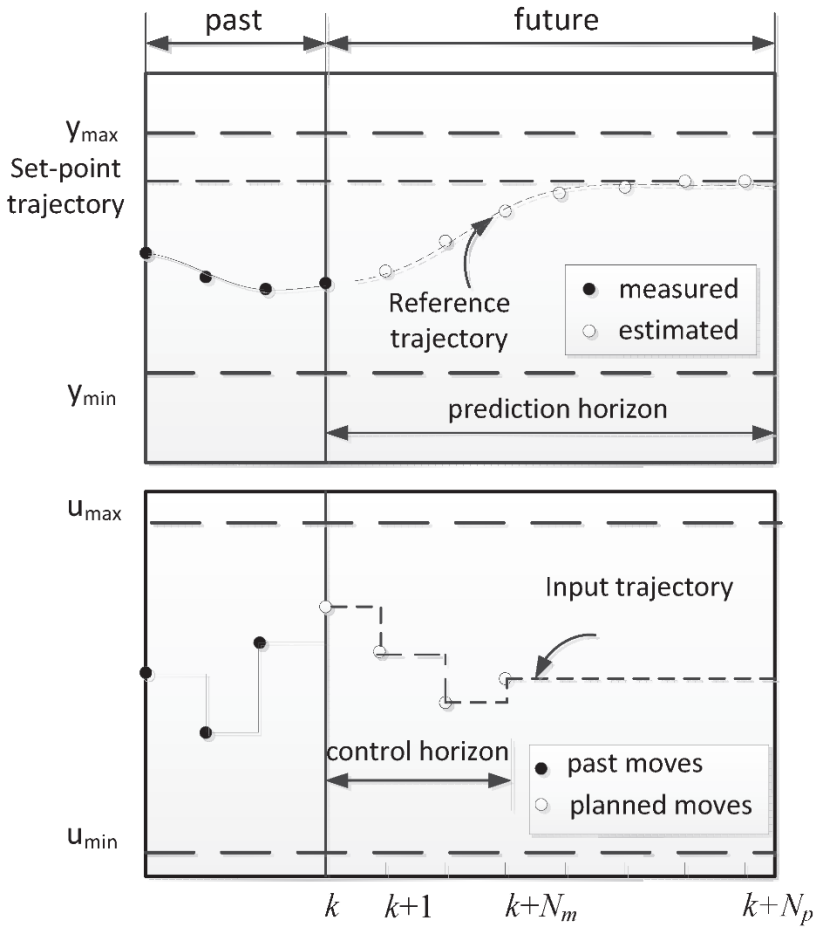
$v(k)$ is the n_v-dimensional vector of measured disturbances

$d(k)$ is the n_d-dimensional vector of unmeasured disturbances

$y_m(k)$ is the n_y-dimensional vector of measured outputs

UNC CHARLOTTE

Model Predictive Control



Source: Bemporad, Morari and Ricker, "Users Guide, Model Predictive Control Toolbox – For use with Matlab"

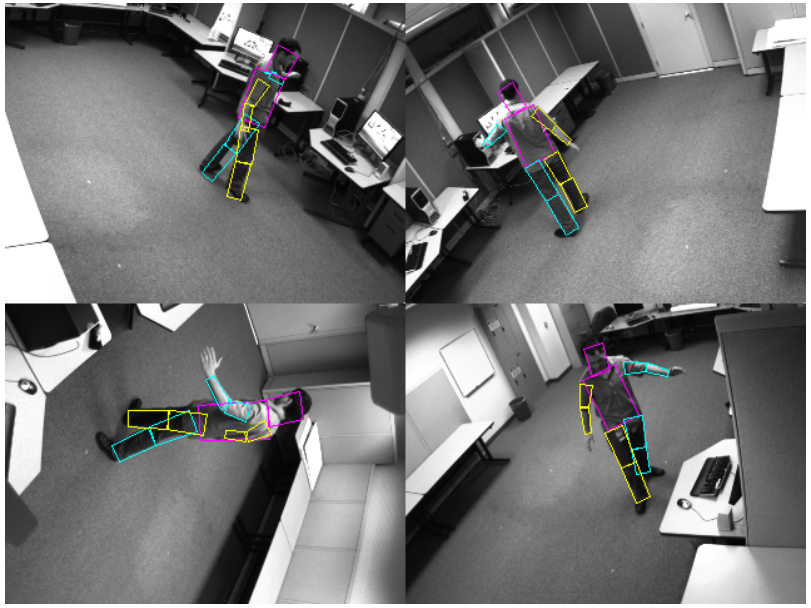
UNC CHARLOTTE

- Motivation
- Related Work
- Cross-Layer Control Framework
- Evaluation Methodology
- Experimental Results
- Future Directions

UNC CHARLOTTE

Benchmarks

- x264 video Encoder (from FFMEPG)
  - Application quality control variable – per frame video resolution
- Bodytrack track human movement (from Parsec benchmark)
  - Application quality control variable – annealing layers and number of particles
  - Visual quality determined the relative mean square error in the magnitude of position vectors
- Benchmarks modified to satisfy Soft Real-Time task model and allow for application quality control



UNC CHARLOTTE

Experimental Setup

- Dual socket Intel Clovertown (X5365) quadcore
- DVFS levels: 2.0 GHz, 2.33 GHz, 2.67 GHz, and 3.0 GHz
- Application quality levels: 4 each for x264 encoder and bodytrack
- Linux 2.6.36 kernel patched with Litmus-RT-2011

UNC CHARLOTTE

Sensors and Actuators

- DVFS (actuator)
  - Low transition latency (~ 10 us)
  - Cpufreq used to dynamically scale operational frequency
  - Modulated using a delta-sigma modulator (uses feedback)
- Application quality (actuator)
  - Higher transition latency (~ 500 us)
  - Global variables protected by FMLP read-write lock
  - Modulated using a pulse-width modulator (no feedback)
- Utilization (sensor)
  - custom system call that aggregates average per-core execution time measured using a high resolution timer, and divides it by the control period

UNC CHARLOTTE

Controller Design

- System Identification – MATLAB SI toolbox
  - First order model – fit 84.8% for x264 and 87.4% for bodytrack
  - $n_x = 1$, $n_u = 2$, $n_v = 1$, and $n_d = 1$
- Controller design – MATLAB MPC toolbox
- C code generation – MATLAB Embedded Coder

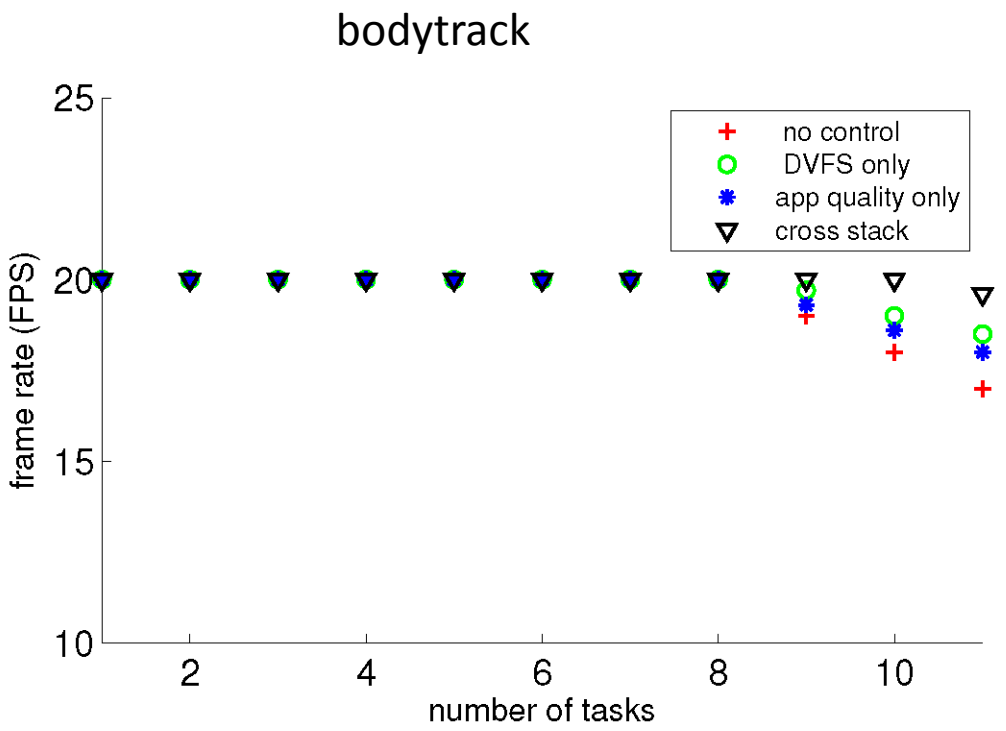
	x264	bodytrack
Control horizon	2	4
Prediction horizon	10	12
Input weight	0, 0	0, 0
Output weight	1	1
Blocking step	5	3
Disturbance model	$\frac{1}{s+1}$	$\frac{1}{s+10}$

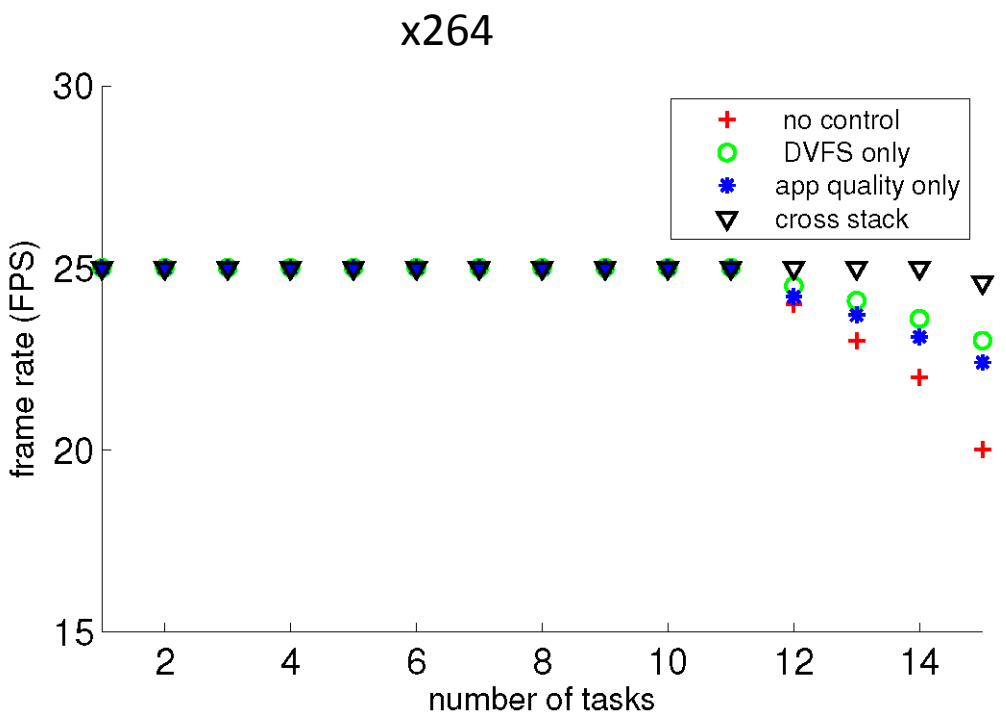
UNC CHARLOTTE

- Motivation
- Related Work
- Cross-Layer Control Framework
- Evaluation Methodology
- Experimental Results
- Future Directions

UNC CHARLOTTE

Avg. FPS vs Number of Tasks

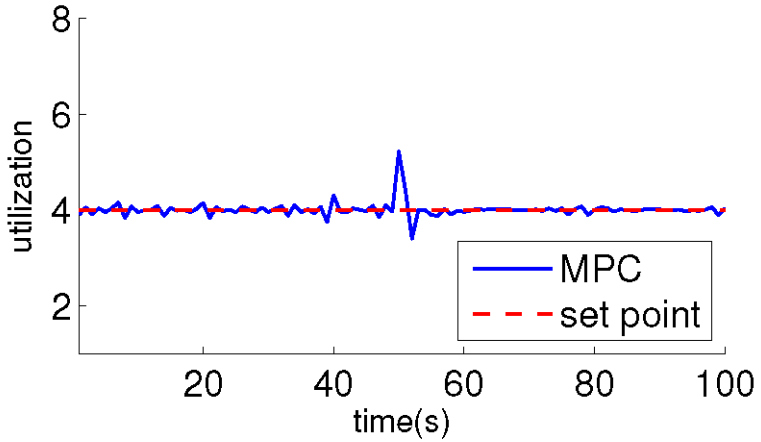


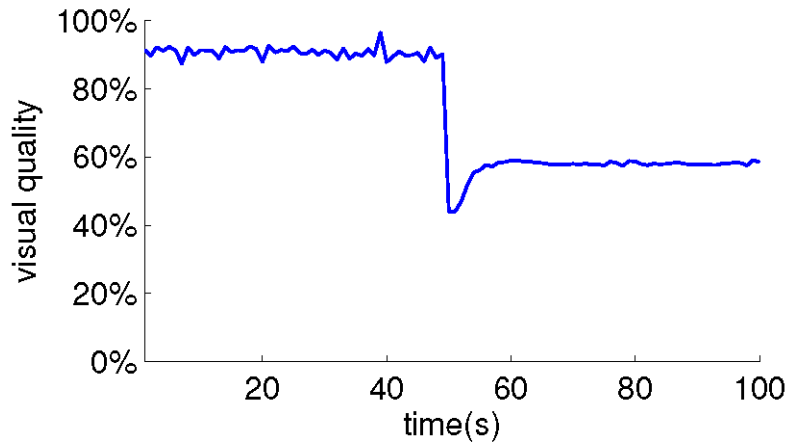


UNC CHARLOTTE

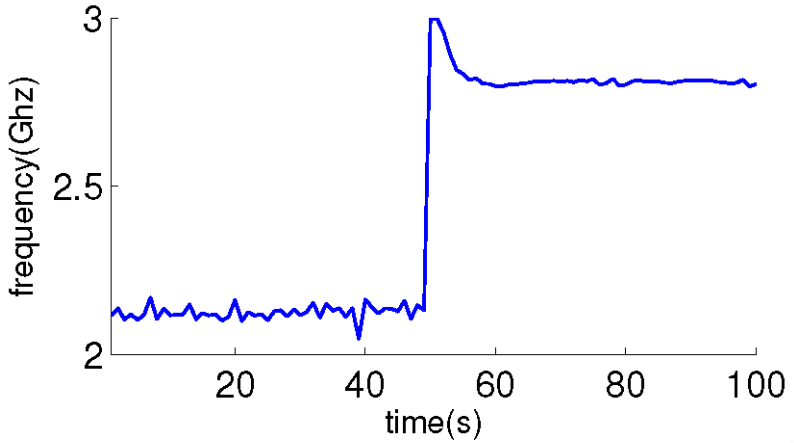
Controller Step Response – Input step

Step change in the number of tasks from 5 to 9 at t = 50s for bodytrack





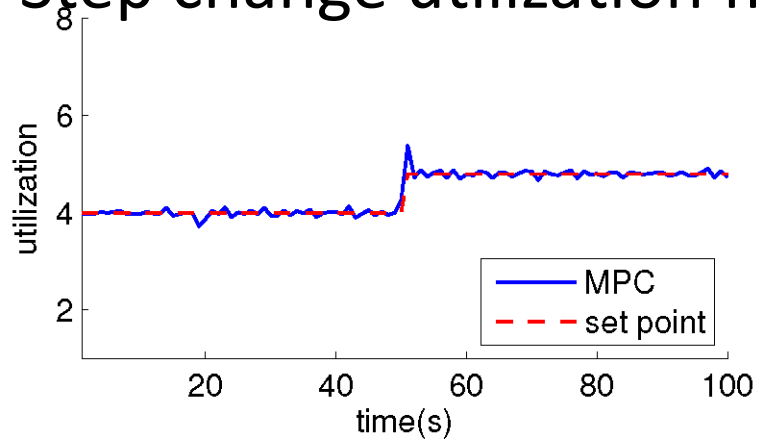
% steady state error	5%
% peak overshoot	30%
settling time	3.8 seconds

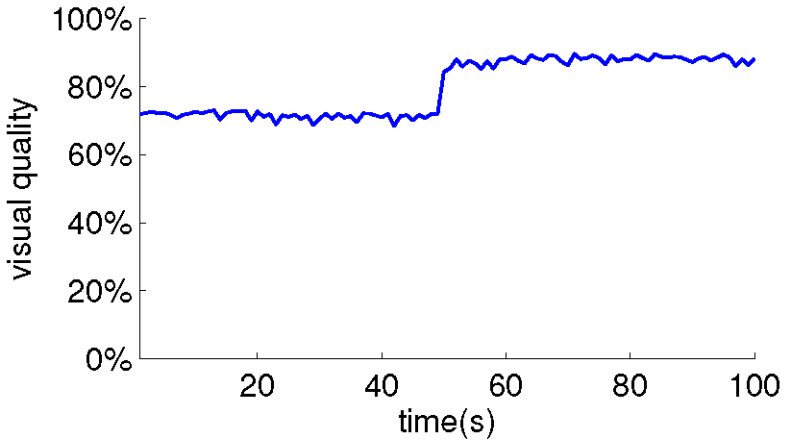


UNC CHARLOTTE

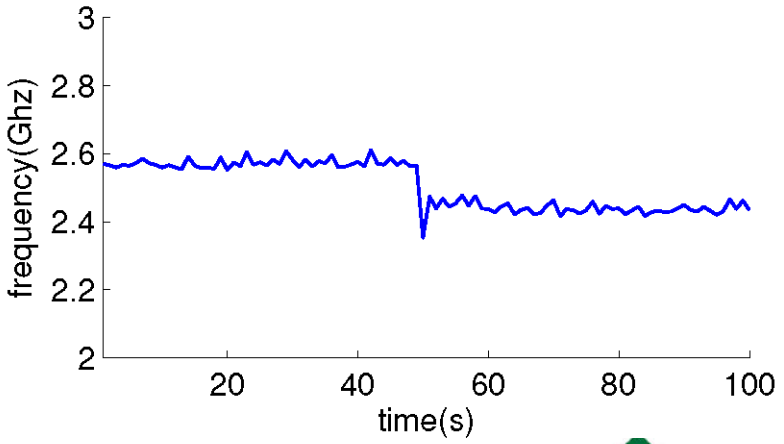
Controller step response – output step

Step change utilization from 4 to 5 at t = 50s for bodytrack





% steady state error	5%
% peak overshoot	22%
settling time	1.8 seconds



UNC CHARLOTTE

Other benefits

- For light task load potential to save power while meeting performance goals
- $P \alpha f^3$
- To evaluate power savings, we compare the cross-layer control vs. the non-control case for different tasks loads from ranging to light to heavy and calculate the average.
- Average power saving is 31% for x264 and 21% for body track
  - Obtained at average application quality of 70% for x264 and 65% for bodytrack
- Fault tolerance

UNC CHARLOTTE

Task Heterogeneity and Scheduling

Number of tasks		FPS of x264		FPS of bodytrack	
x264	bodytrack	C-EDF	G-EDF	C-EDF	G-EDF
2	2	25	25	20	20
2	8	25	25	15.8	20
10	2	20.1	25	20	20
8	6	25	23.1	20	18.3

- C-EDF vs G-EDF
  - C-EDF better data locality
  - G-EDF better load balancing
- G-EDF performs better when one application has much more tasks than other
- C-EDF performs better when both applications are more evenly matched
- Scheduling algorithm – potentially another control variable ?

UNC CHARLOTTE

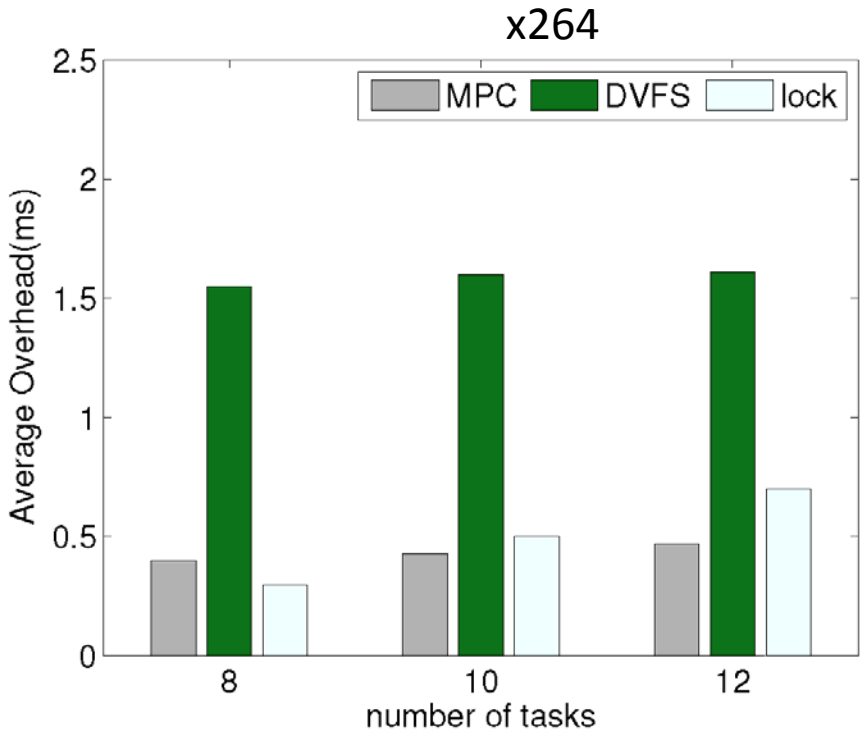
How good is the LTI model?

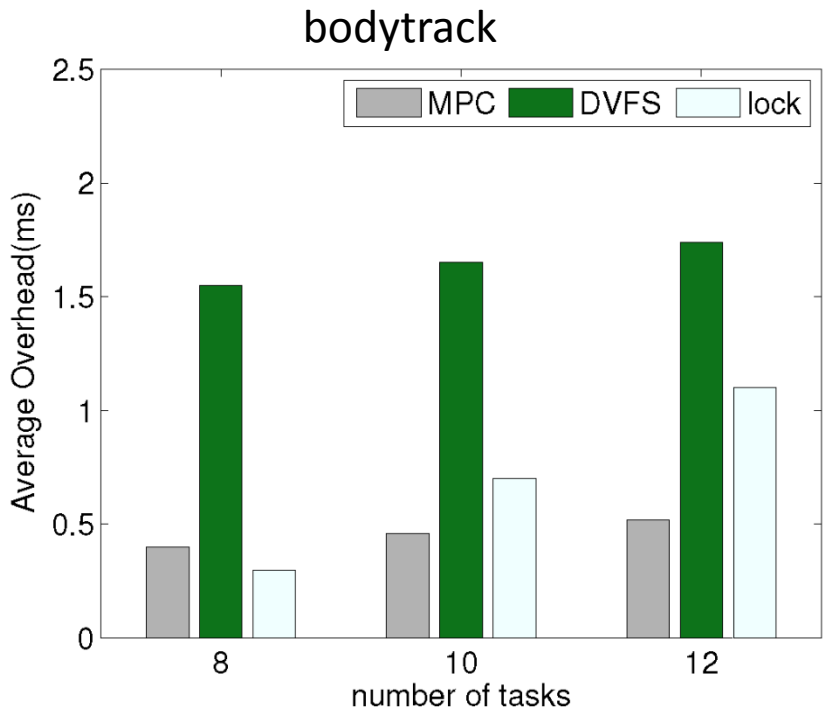
Video index	% steady state error	Significance level of K-S test
1 music video	8.6%	31.3%
2 music video	7.5%	36.7%
3 news report	9.1%	28.9%
4 photography hacks	22.5%	0.015%
5 cooking	8.2%	32.5%
6 sports	25.7%	0.006%
7 news report	9.7%	24.3%
8 hiring program	8.9%	29.4%
9 movie clip	11.2%	19.4%
10 about champagne	9.5%	24.1%

- X264 controller built with the “Hubble video” input
- Evaluate performance of controller against other popular videos drawn from YouTube
- Found to perform well if Kolmogorov-Smirnov test of distribution of average execution times returns a high significance level

UNC CHARLOTTE

Controller overheads





- About 0.5% of one control period

UNC CHARLOTTE

- Motivation
- Related Work
- Cross-Layer Control Framework
- Evaluation Methodology
- Experimental Results
- Future Directions

UNC CHARLOTTE

What next?

- Non-linear control
- Adaptive control
- Power models
- Increased Control variables
- User space control
- Scalability

UNC CHARLOTTE

Questions and Suggestions?



UNC CHARLOTTE