

Metric Based Multi-Timescale Control For Reducing Power In Embedded Systems

Nitin Kataria, Forrest Brewer, João Hespanha, and Timothy Sherwood
Engineering I, University of California, Santa Barbara, CA 93106, USA
nitin@enr.ucsb.edu, forrest@ece.ucsb.edu, hespanha@ece.ucsb.edu, sherwood@cs.ucsb.edu

Abstract—Digital control for embedded systems often requires low-power, hard real-time computation to satisfy high control-loop bandwidth, low latency, and low-power requirements. In particular, the emerging applications of Micro Electro-Mechanical Systems (MEMS) sensors, and their increasing integration, presents a challenging requirement to embed ultra-low power digital control architectures for these lithographically formed micro-structures. Controlling electromechanical structures of such a small scale, using naive digital controllers, can be prohibitively expensive (both in power and cost for portable or battery operated applications.) In this paper, we describe the potential for control systems to be transformed into a set of co-operating parallel linear systems and demonstrate, for the first time, that this parallelization can reduce the total number of instructions executed, thereby reducing power, at the expense of controlled loss in control fidelity. Since the error tolerance of linear feedback control systems is mathematically well-posed, this technique opens up a new, independent dimension for system optimization. We present a novel Computer-Aided Design (CAD) method to evaluate control fidelity, with varying timescales on the controller, and analyze the trade-off between performance and power dissipation. A CAD Metric for control fidelity is proposed and we demonstrate the potential for power savings using this decomposition on two different control problems

I. INTRODUCTION

Digital feedback controllers make up a substantial part of modern embedded systems like ink-jet printers and anti-lock brake systems in cars. Their discrete-time realization, largely designed using classical unit-time sampling techniques, tends to dominate the design performance criteria because of real-time sampling and actuation constraints. These controllers are often implemented on micro-controllers for low bandwidth and digital signal processors or Field-Programmable Gate Arrays (FPGA) for higher bandwidth requirements. Their implementation exploits the linear systems approach for control realization by performing matrix-multiplication based updates of a set of state variables, or direct digital realization as an *Infinite Impulse Response* (IIR) digital filter.

In practice, however, such a strategy can lead to very inefficient, impractical, and expensive designs. For example, the performance of Atomic Force Microscopy (AFM) depends on the performance of the feedback control of the

scanning cantilever, which is based on distributed FPGA implementations of the control algorithm to maintain the required loop latency. Another example is the design of portable MEMS sensors, whose diminutive physical size makes very high demands on loop-bandwidth requirements from the controller [1], [2]. Typically, a linear size scaling of a MEMS device linearly increases the required control-loop bandwidth, and with added requirements of noise shaping and filtering, makes the control algorithms computation intensive, running at sample rates of the order of a few MHz. These strict design requirements seriously hamper the development of small, portable MEMS based systems. Indeed, all current closed-loop MEMS designs are based on analog controller designs.

A promising alternative is to use a custom low-power hardware multi-threaded architecture [3]. Such an architecture enables processes that can be interrupted and interleaved at the granularity of single instruction cycles to meet timing (jitter) obligations. Further, we can substantially reduce latency and power by partitioning the control system based on bandwidth requirements and by performing only the most critical computations on the fastest thread, letting the slower parts run on slower threads. This nontrivial and difficult to analyze technique lowers the total number of instructions executed per second, which translates directly to significant power savings.

It is also possible to attempt algebraic manipulation [4] of the control algorithm to reduce latency and power without losing significant control fidelity. However, this technique has several problems: First, even if the transformation is algebraically correct, round-off error and numerical stability issues often lead to unstable results [5]. Second, the space of possible control algorithms, which might be suitable for control decomposition, is very large – only a minuscule fraction of which are algebraically equivalent to yield the desired control law. Instead, we posit that it should be possible to use well defined controller performance metrics to evaluate the fidelity and power dissipation of a given embedded control design. This approach effectively increases the level of abstraction in the design since a candidate controller need not be equivalent in any sense other than

the evaluation of the performance metric.

In this paper, we take the first steps toward this approach by using a well defined metric for control stability and closed-loop performance. We use an exhaustive search based computer-aided design (CAD) method for exploring the design space by running a partitioned controller at different rates of computation. The metric allows us to explore trade-offs between performance and computation power and clearly shows that we can raise the level of abstraction for control design and optimization, beyond traditional methods. Rest of the paper is organized as follow: Section II present previous work; Section III introduces the H-infinity framework, a basic control design methodology and performance metric with two case studies; Section IV presents our analysis of computation cost relative to performance within a bounding envelope of deviation from an ideal continuous controller. Finally, to conclude, we also demonstrate the use of *Voltage Frequency Scaling* (VFS) to reduce computation power significantly.

II. RELATED WORK

Ad hoc techniques for low power multi-rate control have been used to implement a hard disk drive (HDD) controller in [6]. Ad hoc techniques for multi-rate control with interlacing of computations have been simulated and analyzed in [7], where the results show that the control performance deteriorates marginally with significant reduction in computation energy. Multi-rate digital control for motion control applications has been demonstrated, through various examples, in [8].

Software based power estimation for optimizing power in embedded systems is discussed in [9], [10]. An optimizing compiler, that minimizes execution time to save power has been proposed in [11]. A multiple clock and voltage domain, power-aware, tile-based embedded multi-processor architecture has been proposed and analyzed in [12]. Architectural trade-offs for MEMS closed-loop control have been presented in [13], and a case study of hardware multi-threaded architectures for control has been discussed in [3].

III. MULTI RATE DIGITAL CONTROLLER IMPLEMENTATION

A. Control Design

The H-infinity framework has been used as the basic control design methodology and the control performance metric [14]. In this type of design, one starts by determining the points at which disturbances and noise are most likely to enter the system dynamics and then designs a feedback controller that minimizes the effect of these disturbances on specific signals that one wants to regulate, typically tracking errors (i.e., the difference between a signal and its desired value) and control signals. Specifically, denoting by w the vector of exogenous noise/disturbance signals that affect the dynamics of the process, sensors, and actuators and by z the

vector of signals that one wants to regulate, an H-infinity controller minimizes the worst-case root-mean-square gain

$$J = \max_w \frac{(\int_0^\infty \|z(t)\|^2 dt)^{\frac{1}{2}}}{(\int_0^\infty \|w(t)\|^2 dt)^{\frac{1}{2}}}, \quad (1)$$

where the maximum is taken over all possible disturbance signals w . One can capture band-limited disturbances by introducing weighting pre-filters in w and can also introduce weighting post-filters in z to capture the fact that one may only want to regulate this signal tightly over specific bands of frequency. The value in equation (1) obtained by a specific controller is called the *H-infinity norm* of the closed-loop system.

For linear time-invariant continuous-time processes, the controller that minimizes the H-infinity norm is also linear time-invariant and operates in continuous time, thus requiring an infinite amount of computation. In practice, this cannot be implemented and thus one needs to approximate this continuous-time controller by a discrete-time controller that can be implemented digitally. Since this is not an optimal continuous-time controller, the discrete-time digital controller will not be able to achieve the minimum value for the H-infinity norm in equation (1), but hopefully does not increase it by too much. A side effect of this analysis is the generation of a lower bound for fidelity allowing derivation of sensible trade-offs.

Traditionally, a digital controller is obtained by discretizing a continuous-time controller using the *zero-order-hold* (ZOH) method with a fixed sampling interval T . For sufficiently large sampling rate, the resulting closed-loop exhibits an H-infinity norm very close to that of the continuous-time controller. However, if one has computational constraints that limit the sampling rate, then the performance may degrade significantly and may result in an unstable closed-loop system.

B. Multi-rate Digital Implementation

Here, we propose an alternative approach that is motivated by the observation that generally the optimal continuous-time controller exhibits a few high-frequency modes that need to be implemented at a high-sampling rate, while its low-frequency modes can be implemented at a lower sampling rate. In particular, suppose that the transfer function of the continuous-time controller is denoted by $K(s)$ and that we perform a partial fraction expansion to decompose this transfer function as a sum of two transfer functions as:

$$K(s) = K_1(s) + K_2(s)$$

where $K_1(s)$ is a first-order transfer function whose single pole is the fastest pole of $K(s)$ and $K_2(s) = K(s) - K_1(s)$. When the fastest pole of $K(s)$ has a non-zero imaginary part, then $K_1(s)$ is a second-order transfer function with the two fastest (complex conjugate) poles of $K(s)$. For digital implementation, we now have two degrees of freedom in selecting

the sampling rate, as we can discretize and implement $K_1(s)$ and $K_2(s)$ with two different sampling intervals T_1 and T_2 , respectively.

C. Case Studies

We use two case studies to demonstrate the benefits that can be drawn from this approach: the position control of Quanser's DC servo motor [15] and the control of a MEMS tunneling accelerometer [16].

a) *DC servo*: Quanser's DC servo motor can be modeled by the following second-order transfer function from the input voltage to the shaft angle as:

$$G(s) = \frac{500}{s(s+53.5)}.$$

For the purposes of H-infinity design, we considered the effect of an additive input disturbance with energy concentrated below 100 rad/s (which is roughly twice the system's bandwidth) and flat-spectrum measurement noise with an amplitude 100 times smaller than that of the disturbance. The signals to be regulated by the H-infinity controller are the shaft angle and the applied voltage. Through the use of a weighting post-filter we mostly penalized the control signal at frequencies above 1000 rad/sec to prevent high frequency noise to flow through the feedback loop. These design choices led to the following H-infinity controller:

$$K(s) = \frac{-7.51(s+1001)(s+92.01)(s+73.99)}{(s+124.3)(s^2+317.9s+4.62e04)}$$

which achieved an optimal H-infinity norm of $J_{\text{opt}} = .1505$. For the multi-rate implementation, this controller was decomposed as:

$$K(s) = K_1(s) + K_2(s)$$

$$K_1(s) = \frac{-7.51(s+945.8)(s+32.72)}{s^2+317.9s+4.62e04}, \quad K_2(s) = \frac{-482.33}{s+124.3}$$

It is interesting to note that although the poles of $K_1(s)$ (with absolute value 215rad/sec) are not much faster than those of $K_2(s)$, this decomposition can still lead to significant computational savings, as we shall see below.

b) *MEMS tunneling accelerometer*: The MEMS tunneling accelerometer described in [16] is analyzed similarly. In this type of accelerometer, the feedback controller should use the applied voltage to cancel acceleration and one effectively gets the readout equation:

$$W_{\text{acc}}(s) \approx -187.6V_{\text{app}}(s).$$

where $V_{\text{app}}(s)$ is the control voltage.

For the purpose of H-infinity design, the goal is to reject from the measured voltage the effect of acceleration, as well as noise. In constructing the vector w of exogenous inputs, noise terms were scaled to have unit standard deviation [16]. Additionally, a fictitious low-frequency noise term is added to the measured voltage, to make the controller insensitive to low frequency measurement biases such as 1/f amplifier

noise. These design choices led to the following H-infinity controller:

$$K(s) = \frac{-1.21e04(s+6.29e04)(s+5.78e04)(s^2+5239s+6.95e08)}{s(s^2+1.14e05s+4.41e09)(s^2+2.23e05s+2.01e10)}$$

which achieves an optimal H-infinity norm of $J_{\text{opt}} = 4.6298$. For the multi-rate implementation, this controller is decomposed as:

$$K_1(s) = \frac{-1.41e04(s+2.91e04)}{s^2+2.23e05s+2.01e10}$$

$$K_2(s) = \frac{2.01e03(s+4.09e04)(s-1.85e04)}{s(s^2+1.14e05s+4.42e09)}$$

In this case as well, the poles of $K_1(s)$ (141881 rad/sec) are not much faster than those of $K_2(s)$ (66478 rad/sec) and yet significant computational savings can be achieved with this decomposition.

IV. MODEL FOR COMPUTATION COST

The model for computation power cost focuses on the execution of the control algorithm and conservatively assumes that the cost of data acquisition and actuation is constant. Both examples are *Single Input Single output (SISO)* control systems, where the decomposed controllers share the same input, and their outputs are added together to generate the control output. The two controllers run as two separate threads of execution on two similar processors with no communication except the summation of the outputs, which is done in the faster thread. For the purpose of implementing digital control, the above two example controllers are discretized using *zero-order-hold (ZOH)* and implemented as shown in Figure 1.

A. Instruction Count

As a first order approximation of the computation cost, we determine the total number of processor instructions per second executed to implement a given control algorithm. For the purpose of our model it is convenient to implement the controller as cascaded, direct form II, structured second order sections (SOS), also known as *Biquads (Figure 2)* on a processor.

To evaluate instruction count, we used a Texas Instruments TMS320C2801 (Digital Signal Controller), which is a 32-bit, 8-level deep pipelined architecture, offering up to 100

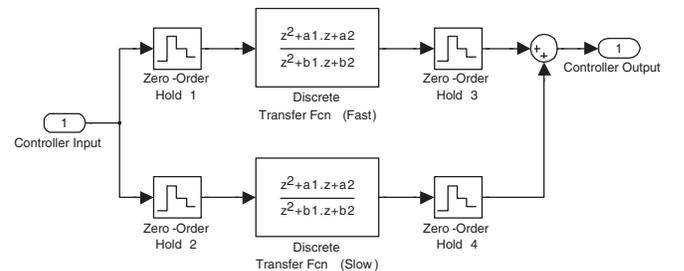


Figure 1. Parallel form decomposed controller.

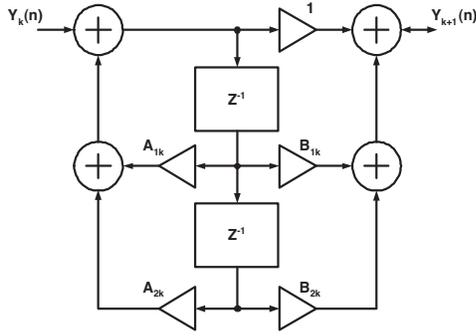


Figure 2. Structured second order biquad.

MIPS of performance and an ARM7TDMI, a 32/16-bit, RISC architecture offering up to 130 MIPS of performance. The total number of instructions required to implement a biquad section on each processor is determined through their individual library assembly routines [17], [18]. In Table I we list the total number of biquad sections needed to implement our control examples and the total number of instructions executed per sample time on the two processors. For a single-rate controller running at a frequency F_s and its decomposed version running at frequencies F_{s1} (fast) and F_{s2} (slow), the total number of instructions per second is determined as follows:

- Single-rate: F_s * instructions per biquad * number of biquad sections
- Multi-rate: F_{s1} * instructions per biquad * number of biquad sections (fast) + F_{s2} * instructions per biquad * number of biquad sections (slow)

Searching the sampling parameter space (sample time pairs for the fast and slow controllers) of the H-infinity framework, we generate a family of curves for the power/performance trade-off. It is then easy to determine Pareto optimal points suitable for controller realization. Figure 3 shows the computation-cost/performance trade-off for the DC servo motor control, where, the H-infinity norm is plotted against the total number of instructions/second (TMS320C2801) for a single-rate controller, and a multi-rate controller running at different rates of computation. The plot highlights the family of curves and the Pareto optimal points for single-rate and multi-rate implementation of control.

B. Power Estimation Model

The power estimation model measures power consumption of each thread, assumed to be running on individual pro-

Table I
TOTAL NUMBER OF BIQUAD SECTIONS AND INSTRUCTION COUNT PER BIQUAD

Example	Controller order	Biquad sections	Instr. count C2801	Instr. count ARM7TDMI
DC servo	3	2	56	54
MEMS Accel.	5	3	79	81

cessors in an architecture similar to *Synchroscalar* [12]. Although different strategies for realizing digital control would result in a different mix of instructions, and hence different power consumption, choosing the well studied biquad ensures that realistically optimized estimates can be determined. It is important to note that by decomposing the control, we benefit from reduced computation order in addition to enabling multi-rate computation. Therefore, once we know the total number of instructions/second and the instruction distribution for a control realization, we can use the results from a cycle accurate energy consumption measurement of a processor instruction set [19] to estimate the power dissipation. Figure 4 illustrates the Pareto optimal points for the case study of MEMS accelerometer control where H-infinity norm is plotted against the total power dissipation for an ARM7TDMI processor.

In Table II we show the computation cost for DC motor control, and, compare the results for the single-rate uniformly sampled controller against a few configurations of the multi-rate implementation, for near similar values of H-infinity norm. It is interesting to note that performance is non-monotone with sample rates and that careful selection of rates can lead to higher overall fidelity than is achievable by single-rate sampling. In Table III we show the power consumption and the H-infinity norm for the MEMS single-rate controller versus a few configurations of multi-rate controller, implemented on an ARM7TDMI.

C. Voltage Frequency Scaling

In this section we postulate that an architecture similar to *Synchroscalar* [12] can be used to implement decomposed control algorithms to further reduce power. Each thread of computation is mapped onto a different processor core running at its optimized voltage and frequency requirements. The voltage/frequency pairs are statically scheduled after determining the real-time requirement of control loops. This

Table II
COMPARISON OF ESTIMATED TOTAL NUMBER OF INSTRUCTIONS/SECOND EXECUTED BY SINGLE-RATE AND MULTI-RATE CONTROLLERS USING TMS320C2801

Single-rate			Multi-rate			
Ts (ms)	H-inf.	Instr. / sec.	Ts1(ms) (fast)	Ts2(ms) (slow)	H-inf.	Instr. / sec.
1	0.24	53000	1	4	0.25	41250
			1	6		38478
			1	14		35343
2	0.34	28000	2	4	0.33	24750
			2	8		20625
			2	11		18843
3	0.45	18648	3	5	0.41	17589
			3	16		10593
			3	18		10065
4	0.56	14000	4	8	0.51	12375
			4	14		10593
			4	18		10065
5	0.69	11200	5	8	0.64	10725

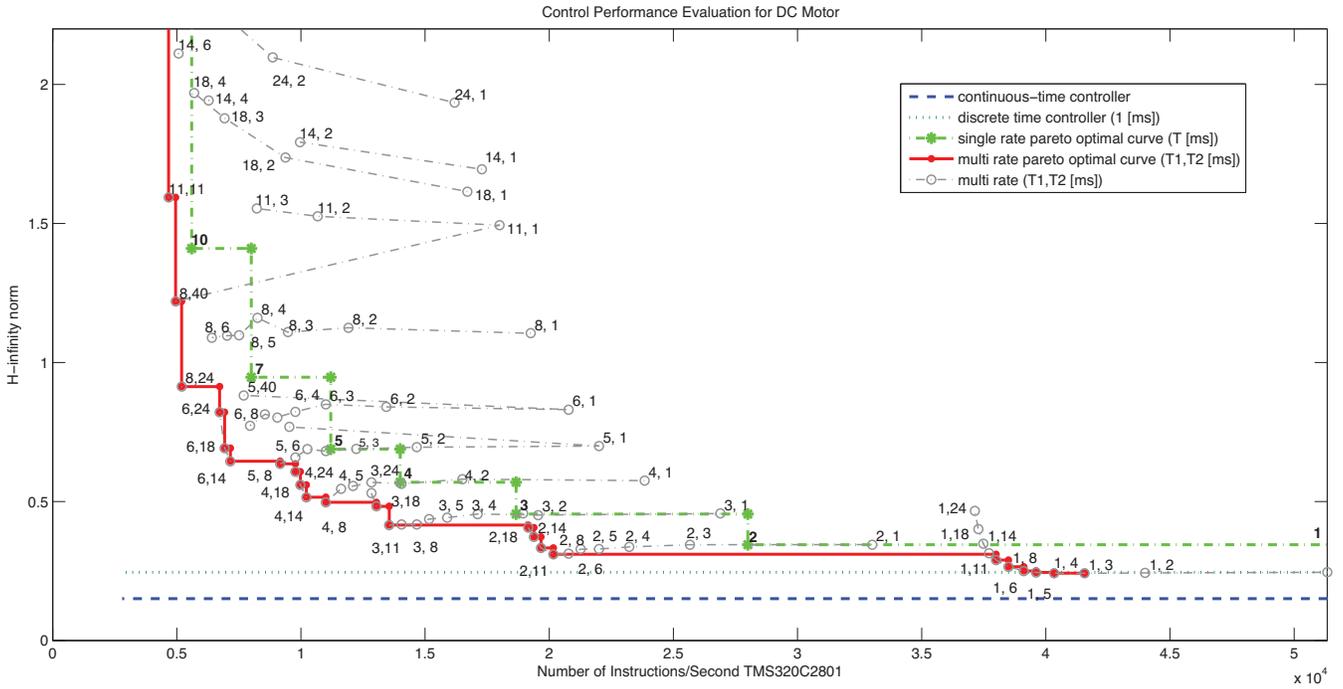


Figure 3. DC Motor performance versus instruction count trade-off. Increasing values of H-infinity norm indicate decreasing control performance.

Table III

COMPARISON OF ESTIMATED POWER CONSUMED BY COMPUTATIONS IN SINGLE-RATE AND MULTI-RATE CONTROLLERS USING ARM7TDMI

Single-rate			Multi-rate			
Ts (μ s)	H-inf.	Power (μ W)	Ts1 (μ s) (fast)	Ts2 (μ s) (slow)	H-inf.	Power (μ W)
5	6.08	10782	5	10	7.46	7188
			5	15	10.59	6090
10	11.46	5391	10	20	11.83	3594
			10	25	16.55	3292
			10	30	20.05	3234
15	30.65	3494	15	20	23.92	2994
			15	25	18.71	2635
			15	30	17.79	2395
75	43.37	718	75	100	44.85	638

approach is different from *Dynamic Voltage Frequency Scaling* (DVFS) [20].

Advantages of statically scheduled voltage frequency pairs are more prominent if we consider, for example, a computation architecture that needs to control a 3-Dimensional (3-D) MEMS accelerometer. From our MEMS control example, we needed a fifth-order controller for 1-Dimensional (1-D) control, therefore, for a 3-D MEMS accelerometer, we would need to execute at least three times the total number of instructions executed for the 1-D controller, within the same execution deadline. Such a controller would require a very fast processor, consuming a lot of power. Instead, we can partition each fifth-order controller, like before, into two separate threads, and run them individually as three fast and three slow threads on *Synchroscalar*. Table IV shows the

power consumption of a 3-D MEMS controller implemented as a single thread of execution versus six separate threads of execution, running at different rates on *Synchroscalar*.

V. CONCLUSIONS

In this work, we have shown that it is possible to raise the level of abstraction in the design power analysis for linear feedback controllers in a way that directly relies on well-known metrics for the design of such controllers. This is in contrast to algebraic methods which can reach only a small fraction of the same design space. The new techniques exploit simple multi-thread or multi-core architecture tricks to allow practical implementation where dependence of the partitioned controllers is minimal. Using these techniques, very substantial power savings are obtained in a way that is independent from conventional algebraic tricks such as sub-expression decomposition and related high-level synthesis techniques. Although the current approach is exhaustive, for practical scale designs it can be performed in a very reasonable amount of time, further work in this area will likely target more efficient exploration techniques as well

Table IV
ESTIMATED POWER CONSUMED BY A 3-D MEMS CONTROLLER

Controller type	Thread count	Total cycles	Cycle time	Volt. (V)	Freq. (MHz)	Power (mW)
Single-rate Ts (5 μ s)	1	531	9(η s)	0.8	120	23.17
Multi-rate Ts1 (5 μ s) Ts2 (10 μ s)	3 (fast)	117	85(η s)	0.7	10	14.14
	3 (slow)	354	85(η s)			

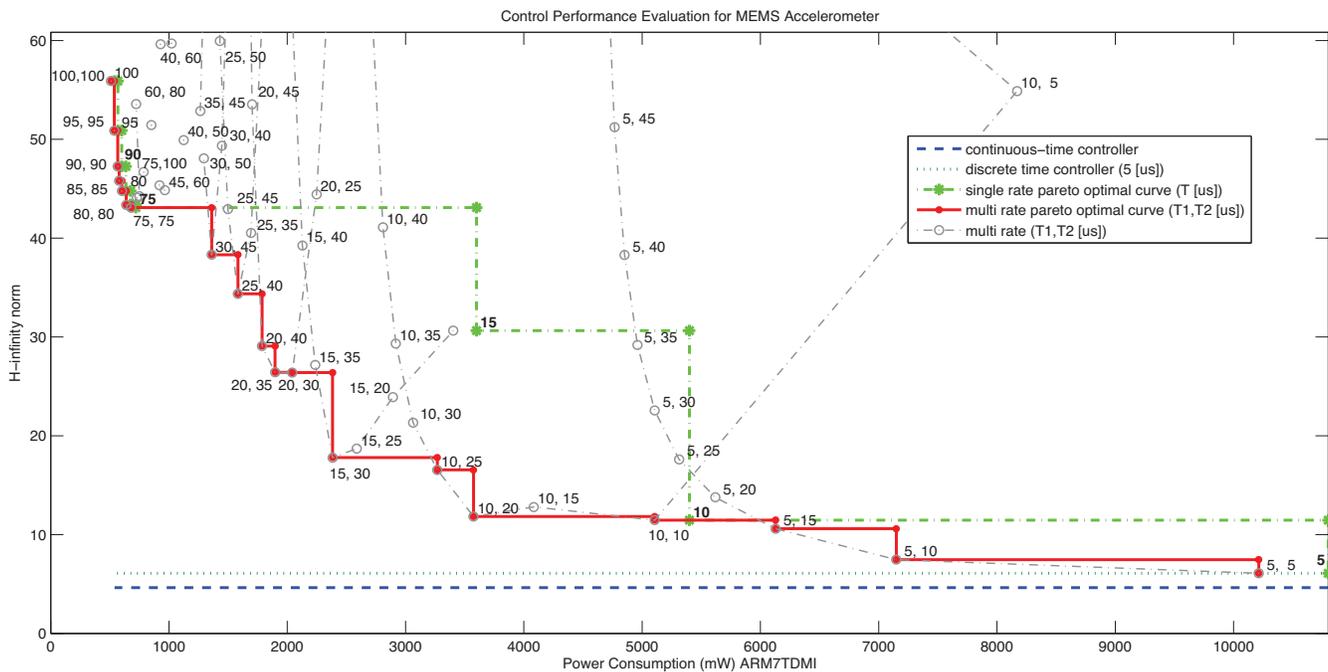


Figure 4. MEMS accelerometer performance versus power consumption trade-off. Increasing values of H-infinity norm indicate decreasing control performance.

as alternative design metrics suitable for sensors and other related designs.

REFERENCES

- [1] M. K. L. O.-R. K. Turner, "Robust feedback control design of an ultra-sensitive, high bandwidth tunneling accelerometer," *American Control Conference, 2005. Proceedings of the 2005*, vol. 6, pp. 4176–4180, June 2005.
- [2] J. Bryczek and E. Abbott, "Control issues for mems," in *Proc. of the 42nd IEEE Conference on Decision and Control*, 2003, pp. 3039–3047.
- [3] G. Hoover, T. Sherwood, and F. Brewer, "A case study of multi-threading in the embedded space," in *CASES '06: Proceedings of the 2006 international conference on Compilers, architecture, and synthesis for embedded systems*, 2006, pp. 357–367.
- [4] A. Hosangadi, F. Fallah, and R. Kastner, "Common subexpression elimination involving multiple variables linear dsp synthesis," in *ASAP '04: Proceedings of the Application-Specific Systems, Architectures and Processors, 15th IEEE International Conference*. Washington, DC, USA: IEEE Computer Society, Sept. 2004, pp. 202–212.
- [5] N. Sureshbabu, B. Powell, and M. Dunn, "An integrated procedure for fixed-point control implementation," *American Control Conference, 1998. Proceedings of the 1998*, vol. 5, pp. 3096–3100 vol.5, 21–26 Jun 1998.
- [6] J. Ding, F. Marcassa, S.-C. Wu, and M. Tomizuka, "Multirate control for computation saving," *Control Systems Technology, IEEE Transactions on*, vol. 14, no. 1, pp. 165–169, Jan. 2006.
- [7] S.-C. Wu and M. Tomizuka, "Multi-rate digital control with interlacing and its application to hard disk drive servo," *American Control Conference, 2003. Proceedings of the 2003*, vol. 5, pp. 4347–4352 vol.5, 4–6 June 2003.
- [8] M. Tomizuka, "Multi-rate control for motion control applications," *Advanced Motion Control, 2004. AMC '04. The 8th IEEE International Workshop on*, pp. 21–29, 25–28 March 2004.
- [9] V. Tiwari, S. Malik, and A. Wolfe, "Power analysis of embedded software: a first step towards software power minimization," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 2, no. 4, pp. 437–445, Dec 1994.
- [10] M. T.-C. Lee, V. Tiwari, S. Malik, and M. Fujita, "Power analysis and minimization techniques for embedded dsp software," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 5, no. 1, pp. 123–135, Mar 1997.
- [11] J. Russell and M. Jacome, "Software power estimation and optimization for high performance, 32-bit embedded processors," *Computer Design: VLSI in Computers and Processors, 1998. ICCD '98. Proceedings., International Conference on*, pp. 328–333, 5–7 Oct 1998.
- [12] J. Oliver, R. Rao, P. Sultana, J. Crandall, E. Czernikowski, I. Jones, L.W., D. Franklin, V. Akella, and F. Chong, "Synchrosalar: a multiple clock domain, power-aware, tile-based embedded processor," *Computer Architecture, 2004. Proceedings. 31st Annual International Symposium on*, pp. 150–161, 19–23 June 2004.
- [13] G. Hoover, T. Sherwood, and F. Brewer, "Towards understanding architectural tradeoffs in mems closed-loop feedback control," in *CASES '07: Proceedings of the 2007 international conference on Compilers, architecture, and synthesis for embedded systems*, 2007.
- [14] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. New Jersey: Prentice Hall, 1996.
- [15] "Rotary servo plant with encoder, srv02." [Online]. Available: <http://www.quanser.com/>
- [16] C. B. K. Å. F. B. L.A. Oropeza-Ramos, N. Kataria and K. Turner, "Noise analysis of a tunneling accelerometer based on state space stochastic theory," in *Solid-State Sensor, Actuator, and Microsystems Workshop, Hilton Head, SC June 2008*.
- [17] C. W. Andrew N. Sloss, Dominic Symes, "Arm system developer's guide designing and optimizing system software," <http://www.arm.com>, Advanced RISC Machines Ltd., Tech. Rep.
- [18] "Filter library, module user's guide, c28x foundation software," <http://www.ti.com>, Texas Instruments, Tech. Rep., 2002.
- [19] N. Chang, K. Kim, and H. G. Lee, "Cycle-accurate energy consumption measurement and analysis: Case study of arm7tdmi," *Low Power Electronics and Design, 2000. ISLPED '00. Proceedings of the 2000 International Symposium on*, pp. 185–190, 2000.
- [20] G. Magklis, G. Semeraro, D. Albonesi, S. Dropsho, S. Dwarkadas, and M. Scott, "Dynamic frequency and voltage scaling for a multiple-clock-domain microprocessor," *Micro, IEEE*, vol. 23, no. 6, pp. 62–68, Nov.-Dec. 2003.