Leveraging Flexibility in Reconfigurable Baseband Processors for Resource Management in Software-Defined and Cognitive Radios

Joseph Gaeddert and Jeffrey H. Reed
Wireless@Virginia Tech
Bradley Department of Electrical and Computer Engineering, Virginia Tech
432 Durham Hall, MS 0350, Blacksburg, VA 24061, USA
E-mail: jgaedder@vt.edu

***Abstract*—Software-defined radio (SDR) technologies promise significant enhancements over traditional radio design in terms of waveform flexibility, reconfigurability, and modular component reuse. These benefits are typically realized through the deployment of modular software processing blocks on a variety of processing platforms, however, the processing complexity of these target platforms severely limits their capacity and, in the case of mobile radios, their battery life. Because a considerable percentage of power consumed in SDR is in baseband processing, reducing its computational load has significant implications on its power consumption, processing bandwidth, and capable throughput.**

This paper provides analytical analysis for modeling power consumption on SDR platforms while preserving a necessary service quality for packet radios by leveraging software flexibility. We propose using computational complexity of the software-defined baseband processor as a metric for comparing resource consumption on such platforms. Furthermore the paper discusses strategies for managing power consumption by adapting link-level algorithms to reduce computational complexity, validated by hardware simulation of running a demodulator and forward error-correction decoder on an embedded processor.

I. Introduction

Despite their apparent benefits, software-defined radios tend to consume considerably more power than their hardware counterparts. The energy efficiency lost through the flexibility of baseband processing on a reconfigurable platform severely limits their computational capacity and has considerable implications on their physical size, weight, and battery life [1]. Resource management on reconfigurable platforms is therefore a critical step in deploying mobile reconfigurable radios, particularly those operating in dynamic spectrum environments. Many of the traditional methods for reducing energy consumption on mobile radios, however, are inapplicable to reconfigurable platforms because hardware and software are no longer strictly bound to one another.

Many SDR implementations seek to optimize DSP algorithms for a range of operating conditions, however because the performance of mobile radios depends so significantly upon channel conditions, it is not possible to guarantee minimal power consumption for a specific platform unless the dynamics of the algorithm itself are designed to be energy-scalable. Therefore the algorithm which minimizes the energy consumed for a particular application while achieving a certain quality of service cannot be predetermined without considering the complex dynamics of the wireless environment in addition to the platform upon which the application is running.

In contrast, the flexibility within the SDR platform can be leveraged to actually mitigate its additional power demands. The immense variability in re-programmable software components permits the system to be dynamically rebalanced for new scenarios, allowing for more efficient use of hardware. Power management in cognitive radios at the hardware level was discussed by Khajeh *et al.* in [2], demonstrating energy savings exceeding 20% while running a 3G WCDMA modem and H.264 video decoder. When not constrained to any particular standard, Bougard *et al.* in [3] demonstrated though simulation an energy consumption savings of a factor of 5 to 20 over the worst-case context scenario–a typical strategy carried over from legacy designs– while considering system level parameters only. In all these examples, the radio was given information about how its link-level parameters affected both is power consumption and its performance.

Some of the most immediate and obvious applications to this concept are cognitive radios, by which power management can be achieved through means other than just feedback in the protocol, namely a self-adapting engine. Despite its promises, cognitive radios (and otherwise adaptive radios) still require measureable system-level performance feedback in order to appropriately switch modes. It is therefore desireable to allow the radio to monitor its own dynamic resource consumption in addition to its link-level performance. The challenge therefore becomes an issue of how to efficiently model or measure the platform's consumption of finite resources; a particularly difficult task as software becomes portable to a growing population of heterogenous platforms. Highly parallelized architectures (such as FPGAs and multi-processor platforms) exacerbate the difficulty in estimating the processing time and complexity of executing particular algorithms.

This paper takes a renewed look at resource management on software-defined radio platforms by investigating the inherent tradeoff between spectrum and processing efficiencies which relate strongly to both the power consumed by the processor

and the complexity of the algorithm which it can support. The analysis is specifically concentrated on DSP implementations of digital demodulators in conjunction with forward error correction (FEC) codes in slowly-fading noisy channels.

II. Energy Consumption in SDR

For a wireless communications link, the energy and power loads can be split into those consumed by the transmitter and the receiver. The transmitter is typically burdened by the radio frequency front end, of which most of the power drain is attributed to the power amplifier.The receiver, however, is limited by its signal processing hardware. Lien et al. demonstrated that power consumed by the baseband processor increases non-linearly as a function of processor usage [4, Eq. (2)]. Power consumption typically increases with a linear relation to sampling frequency. This is apparent in analog converter (ADC) chips which demonstrate a strong relationship between clock frequency and power source current draw [5, TPC 27]. Both the receiver's computational complexity and power consumption are strongly related to its occupied signal bandwidth.

A. Energy Quality Scalability

The notion of energy quality scalability was introduced in [6] for VLSI systems. Conceptually, quality of service can be minimally compromised by dramatically reducing the energy consumed through processing. The tradeoff between accuracy and energy shows diminishing returns as considerable amount of energy is wasted reducing error below the tolerance. The optimal solution is one that achieves a sufficient quality measure without expending unnecessary resources attaining it. At a higher level, Khajeh *et al.* in [2] acknowledge that not all applications require 100% data correctness; a broad family of fault-tolerant applications exist, particularly in the wireless multimedia realm. For the case of fault-intolerant applications, erroneous packets are simply detected with a moderate-length cyclic redundancy check, and re-transmitted. The probability of the receiver being unable to decode a packet is kept sufficiently low through waveform adaptation.

B. Performance Metrics

Communication link quality is measured through application-driven metrics, such as bit-error rate (BER), data throughput, and latency. Link quality is contingent upon many controllable factors (modulation scheme, transmit power, forward error-correction coding) and uncontrollable factors (noise power spectral density, interference levels, hardware limitations). Some of these factors in particular can be applied uniquely to the receiver and do not require link-level adaptation. Equalization is a strong example of this; the length of the equalizer directly impacts both the receiver's computational complexity and its ability to reduce inter-symbol interference, thus reducing its error rate. Other factors require radios on both sides of the link to adapt, such as switching modulation and forward error-correction schemes.

The relationship between link-level QoS metrics and radio parameters for a given channel state is well-known, however quantification of processing complexity and its impact on these metrics is not. Furthermore, many of these algorithms' energy consumption significantly depends on the implementation and the target platform.

The maximum throughput for a link constrained by transmit power is well-known and is achievable as the occupied bandwidth approaches infinity, viz.

$$C = W \log_2 \left(1 + \frac{\bar{P} L_P}{W N_0}\right) \tag{1}$$

where $\bar{P}$ is the average transmit power, L_P is the path loss, W is occupied bandwidth, and N_0 is the noise power spectral density. However increasing bandwidth has considerable implications on the power consumed by the baseband processor at the receiver. We can therefore define the receivers processing channel efficiency as

$$\eta_c = \frac{P_c}{C} = \frac{P_c(W)}{W \log_2 \left(1 + \frac{\bar{P} L_P}{W N_0}\right)} \tag{2}$$

where $P_c(W)$ is the power required by the processor to recover the bits at the receiver. Note that η_c is measured in bits per Joule. Although theoretical channel throughput increases with bandwidth for a given average transmit power, the efficiency eventually degrades due to an increase in processor complexity and energy consumption at the receiver. The link can therefore sacrifice throughput for bandwidth efficiency and energy consumption. Several publications make reference to measure bits per expended energy as a metric for power efficiency is software-defined radios [7, Fig. 15], [8, Fig. 8]. Figure 1 demonstrates this strong relationship between processing efficiency and bandwidth.

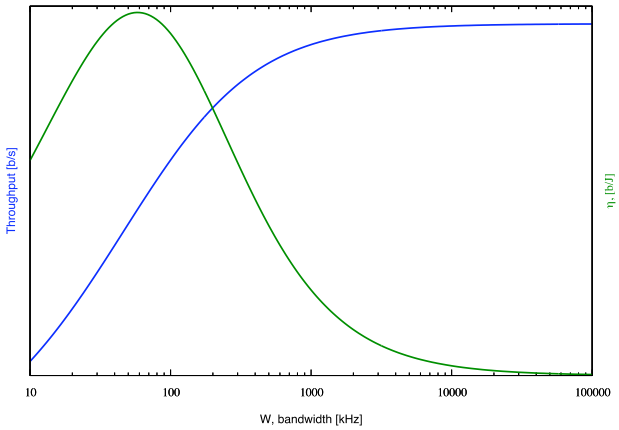


Fig. 1. Channel throughput [b/s] and energy efficiency [J/b] as a function of bandwidth. Energy efficiency assumes power consumption to be a linear function of occupied bandwidth.

C. Monitoring and Management

Understanding how algorithm adaptions affect energy consumption in processors it is necessary for an energy-scalable

system to adapt itself to changing environments. This section describes several methods of monitoring real-time energy consumption on DSP platforms.

D. Modeling

Real-time power monitoring can be modeled through processing complexity as in [4], [9] with streaming multimedia applications. Furthermore, it has been shown that significant power saving can result from selectively reducing processing to meet a target quality of service for such applications [10]. It is apparent from these results that the same benefits can be realized in SDR platforms which provide flexibility over baseband processing blocks (such as filters, oscillators, and synchronizers), and cognitive radios which seek to optimize link parameters (such as modulation depth and forward error-correction codes). Typical quality of service (QoS) metrics for data radios measure throughput, latency, and error probabilities: all highly sensitive to these baseband processing algorithms. Processing complexity therefore can be minimized while preserving a pre-defined QoS.

Perhaps the most obvious method is to physically measure the power directly from the source as the system runs. Several target platforms support real-time power monitoring. An excellent example of this is the Lyrtech Small Form Factor SDR development platform [11] which supports embedded, independent power monitoring for each processor in real time. Power can be measured simultaneously with the algorithms running on the different processors.

If no on-board power-monitoring component is available, energy consumption can be modeled through measurable quantities. For example, [4] demonstrate the strong relationship with CPU usage for general purpose processors and power. Many platforms can monitor CPU usage in real-time, making this a tractable solution to power monitoring and energy management. Following the results from [12] in which software energy consumption is modeled through processor hardware (supply voltage, clock frequency, static/dynamic current drain, transistor capacitance, and other parasitics), we propose using computational complexity as a baseline metric for comparing resource consumption. Furthermore, [13] demonstrated that while the energy consumed varied by the specific machine-level instruction being executed by the processor, this variation was not a considerable factor. To this end, we collapse our assessment to receiver complexity by observing the number of clock cycles required to complete one iteration of a certain task. DSP algorithms with flexible implementations must be able to be deployed on a variety of hardware. These implementations, however, might not consume comparable amounts of energy on different platforms. Energy consumption of such processing blocks can be measured offline on the target platform such that they can be efficiently partitioned when the application is to be executed. The energy consumption of each processing block can be either stored in a look-up table or modeled using simple curve-fitting techniques for a variety of typical operating ranges of the algorithm. The baseband receiver chain running in software is comprised of a number of signal processing algorithms which can be reconfigured at their execution time depending upon the link protocol, typically governed by channel conditions. The total computational load of the receiver in terms of clock cycles can be estimated by offline benchmarks, and the corresponding voltage/frequency operating point of the processor can be chosen. Assuming the supply voltage and clock frequency of the hardware can be dynamically scaled, the power consumed by the processor can be modeled quite accurately. This greatly simplifies our analysis of receiver complexity, and can easily be extended to processors with multiple, independent cores.

III. Methodology

We propose to extend the typical SDR model to incorporate real-time resource monitoring modules necessary for reducing energy consumption, and valuable computational clock cycles. SDR frameworks consist primarily of a set of inter-connected processing components perhaps running on independent processors. We formalize the situation as a classic non-linear optimization problem with the following axioms:

1) Both QoS and energy consumption are measurable quantities and are affected by digital processing blocks' deployment, allocation, and configuration;
2) QoS performance is measured through the system as a whole. Any solution is considered valid so long as the constraints of the system are met;
3) Processing complexity and power consumption performance is measured on independent processors, the sum of which is the true metric to minimize.

Therefore the performance monitor and radio control components must be aware of the impact energy-quality tradeoff one processing block has on the entire system performance. For example, power savings through reducing the length of the equalizer might degrade the uncoded BER to a point where forward error-correction codes cannot sufficiently recover the packet data.

IV. Experimentation

The benefits of SDR are typically realized through the deployment of modular software processing blocks on a variety of processing platforms. In a heterogenous cognitive network, however, this presents a problem with communication as now protocol negotiation must additionally incorporate resource management considering the different hardware capabilities of each node. Because the burden of computation is much greater at the receiving node, this negotiation must subsume its hardware and software capabilities: what decoding algorithms are available? can the hardware support running the decoder at this speed? are the power requirements for the decoder too great for the hardware to support?

Consider a scenario where two software radios in a network need to establish a link using packets capable of adapting modulation and forward error-correction coding schemes. While both nodes are limited by their platform capabilities, one in particular is crippled by its hardware clock frequency. Now they must negotiate an appropriate protocol by which

to communicate such that the complexity at the receiver is minimized.

A. Adaptive Modulation/Error-correction in Slowly-varying Fading Channels

Adaptive switching modulation and coding schemes based on instantaneous received power is a technique which as been incorporated in a wide population of wireless radios, however the impact these switching mechanisms have on baseband processors implemented on reconfigurable platforms has garnered little investigation. In this section we take a renewed look at adaptive switching modulation/coding schemes not simply for the sake of improving spectral efficiency, but how appropriately choosing a set of schemes can significantly reduce the complexity of the receiver with unnecessarily sacrificing capacity.

B. System Model

Given a fading distribution $f_\gamma(\gamma;\Omega)$ for instantaneous received signal power γ and average power Ω, an upper bound on the average bit error rate for a modulation/coding scheme pair can be computed by averaging its instantaneous BER performance over the distribution, viz. [14]

$$\bar{P}_b^{m,f}(\Omega) = \int_0^\infty P_b^{m,f}(\gamma) f_\gamma(\gamma;\Omega) d\gamma \tag{3}$$

For slowly-varying channels, the received signal power is assumed to be flat over the duration of a data packet, and thus the switching procedure chooses the "optimal" modulation/coding scheme pair which matches the estimated target error probability to the received signal power. The appropriate pairs for subsequent packets are chosen on the basis of signal strength alone. The average BER of an adaptive switching scheme is therefore computed by summing the marginal BER over the partitioned distribution:

$$\bar{P}_b(\Omega) = \frac{\sum_{i=0}^{N-1} k_i r_i \int_{\gamma_i}^{\gamma_{i+1}} P_i(\gamma) f_\gamma(\gamma,\Omega) d\gamma}{\sum_{i=0}^{N-1} k_i r_i \int_{\gamma_i}^{\gamma_{i+1}} f_\gamma(\gamma,\Omega) d\gamma} \tag{4}$$

where Ω is the average signal power, $k_i = \log_2(M_i)$ bits per symbol for the modulation type, r_i coding rate, γ_i is the threshold SNR, and $P_i(\gamma)$ is the BER distribution for the i^{th} modulation/coding pair type under an AWGN channel. Note that the denominator is the average spectral efficiency, $\bar{\eta}(\Omega)$, viz

$$\bar{\eta}(\Omega) = \sum_{i=0}^{N-1} k_i r_i \int_{\gamma_i}^{\gamma_{i+1}} f_\gamma(\gamma,\Omega) d\gamma \tag{5}$$

In a similar fashion, we define the average receiver complexity $\bar{\mathcal{K}}$ as

$$\bar{\mathcal{K}}(\Omega) = \sum_{i=0}^{N-1} \mathcal{K}_i \int_{\gamma_i}^{\gamma_{i+1}} f_\gamma(\gamma,\Omega) d\gamma \tag{6}$$

where $\mathcal{K}_i$ is the computational complexity of the modulation/coding pair measured in terms of CPU clock cycles per uncoded bit.

V. Results

In order to gain insight into the tradeoffs between computational complexity and link quality, we adopt the Rayleigh fading model, a typically pessimistic assumption about the channel conditions. The Rayleigh fading model for instantaneous received signal power γ is

$$f_\gamma(\gamma;\Omega) = \frac{1}{\Omega} e^{-\gamma/\Omega} \tag{7}$$

where Ω is the average power.[1] Error rates were generated by encoding data packets using a discrete set of modulation/coding scheme pairs, pushing the resulting symbols through an AWGN channel, and measuring the resulting bit errors. For each data point, a minimum of 10,000 trials were performed ensuring at least 2000 observed errors. A block interleaver was inserted after the encoder to help randomize bit errors within the data packet to aid the decoder. Computational complexity measurements were conducted by running 100,000 iterations of each demodulator and FEC decoder on an embedded 867 MHz PowerPC processor with AltiVec SIMD extensions.

Two sets of adaptive modulation/coding switching schemes were compared; the first set contains typical half-rate convolutional codes with constraint lengths $K = 7$, $K = 9$, and punctured codes with rates from ranging from $2/3$ to $7/8$; the second contains either no error-correction (uncoded) or only a computationally efficient Hamming$(7,4)$ block code. These pairs are presented in Table I along with the average computational complexity to receive each coded bit, and the required instantaneous SNR to achieve a BER of 10^{-3} in an AWGN channel.

Given the simplicity of the distribution of signal power for the Rayleigh fading model, the expressions for average spectral efficiency in (5) and average computational complexity (6) can be rewritten as $\bar{\eta}(\Omega) = \sum_{i=0}^{N-1} k_i r_i \left(e^{-\gamma_i/\Omega} - e^{-\gamma_{i+1}/\Omega}\right)$ and $\bar{\mathcal{K}}(\Omega) = \sum_{i=0}^{N-1} \mathcal{K}_i \left(e^{-\gamma_i/\Omega} - e^{-\gamma_{i+1}/\Omega}\right)$ respectively. The values for average error probability given by (4), were computed through numerical integration. The switching levels for adaptation are given in Table I.

Figure 2 depicts the results of the simulations. While it is hardly surprising that the stronger convolutional codes provide a higher capacity than the uncoded/weakly-coded set, the actual spectral efficiency does not suffer a significant hit. The strongest discrepancy of $\bar{\eta}$ between the two sets exists at $\Omega \approx 23$dB where the spectral efficiency is only about 0.9 b/s/Hz (a 16% degradation) as seen in Figure 2(a). In contrast, Figure 2(b) demonstrates that the difference in computational complexity between the two sets is greater than a factor 5 for low SNR, and nearly a factor 2 for high SNR in favor of the weaker codes. Analysis could easily be run for any number of modulation/coding combinations, with appropriate sets sought to maximize spectral efficiency, minimize complexity, or some

[1] It is important to note that while fading models typically are expressed in terms of signal amplitude, (7) denotes fading in terms of signal *power*. This is a trivial variable transformation necessary for analysis.

TABLE I
AVAILABLE MODULATION/FEC SCHEME PAIRS, BROKEN INTO 2 SETS

Modulation scheme	FEC scheme	$\eta, [b/s/Hz]$	$\mathcal{K}_i, [cycles/bit]$	γ_i [dB]
QPSK	conv. r=1/3, K=9	0.67	612.5	6.45
QPSK	conv. r=1/2, K=9	1.00	492.9	7.84
QPSK	conv. r=2/3, K=9	1.33	521.2	9.57
QPSK	conv. r=3/4, K=9	1.50	511.5	10.34
16-QAM	conv. r=1/2, K=9	2.00	519.1	13.52
16-QAM	conv. r=2/3, K=9	2.67	540.8	15.32
16-QAM	conv. r=3/4, K=7	3.00	211.7	16.61
16-QAM	conv. r=4/5, K=9	3.20	526.0	17.02
16-QAM	conv. r=7/8, K=9	3.50	501.3	18.06
64-QAM	conv. r=2/3, K=9	4.00	520.7	20.39
64-QAM	conv. r=3/4, K=9	4.50	511.0	21.48
64-QAM	conv. r=4/5, K=9	4.80	509.2	22.23
64-QAM	conv. r=7/8, K=9	5.25	485.9	23.40
BPSK	Hamming (7,4)	0.57	104.8	8.92
BPSK	uncoded	1.00	56.2	10.07
QPSK	Hamming (7,4)	1.14	66.3	10.49
QPSK	uncoded	2.00	34.2	12.77
8-PSK	uncoded	3.00	141.4	17.28
16-QAM	uncoded	4.00	47.3	18.90
64-QAM	uncoded	6.00	33.8	24.28

middle-ground combination of the two; the particular sets were chosen specifically to accentuate this tradeoff.

VI. CONCLUSIONS

Modeling processing complexity for DSP algorithms and the power they consume on a variety of platforms is a difficult problem to solve. When mapping platform-independent to platform-specific models one must incorporate a host of hardware design considerations. In this paper we have highlighted the advantages in managing resources on software-defined radio platforms by permitting the radio to be cognizant of its own power consumption and processing complexity. As are result, significant computational savings can be gained at the receiver at the expense of either spectral efficiency of the link or transmit power.

REFERENCES

[1] M. Uhm, "The Power of Software Defined Radio: A Holistic Approach to Designing SDRs for Power," in *COTS Journal*, 1 2007.

[2] A. Khajeh, S.-Y. Chen, A. M. Eltawil, and F. J. Kurdahi, "Power Management for Cognitive Radio Platforms," in *IEEE Global Telecommunications Conference*, 11 2007, pp. 4066–4070.

[3] B. Bougard, G. Lenoir, F. Catthoor, and W. Dehaene, "A New Approach to Dynamically Trade Off Performance and Energy Consumption in Wireless Communication Systems," in *IEEE Workshop on Signal Processing Systems*, 8 2003.

[4] C.-H. Lien, Y.-W. Bai, and M.-B. Lin, "Estimation by Software for the Power Consumption of Streaming-Media Servers," *IEEE Transactions of Instrumentation and Measurements*, vol. 56, no. 5, pp. 1859–1870, October 2007.

[5] "Mixed-Signal Front-End (MxFE^{TM}) Processor for Broadband Communications," Analog Devices Product Datasheet, 2002.

[6] A. Sinha, A. Wang, and A. Chandrakasan, "Energy scalable system design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 2, pp. 135–145, April 2002.

[7] B. Bougard, S. Pollin, A. Dejonghe, F. Catthoor, and W. Dehaene, "Cross-layer power management in wireless networks and consequences on system-level architecture," *EURASIP Signal Processing Journal, Special Issue on Advances in Signal Processing*, vol. 86, no. 8, pp. 1792–1803, 2006.

[8] A. Dejonghe, J. Declerck, F. Nassens, M. Glassée, A. Dusa, E. Umans, A. Ng, B. Bougard, G. Lenoir, J. Craninckx, and L. V. der Perre, "Low-power SDRs through cross-layer control: concepts at work," in *Mobile and Wireless Communications Summit*, 2007, pp. 1–6.

[9] Y. Hu, Q. Li, and C.-C. J. Kuo, "Run-time Power Consumption Modeling for Embedded Multimedia Systems," in *Proceedings of the 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, 2005.

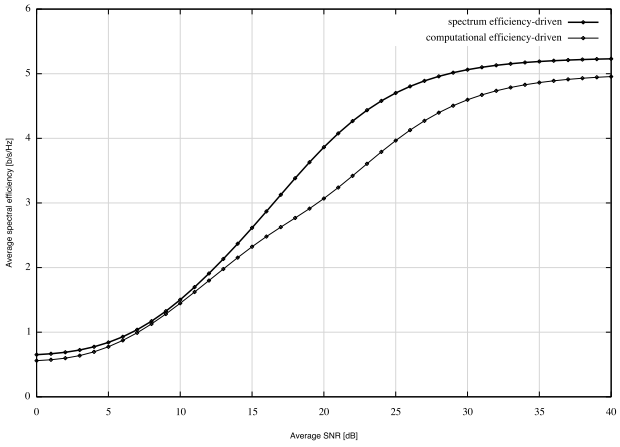
[10] S. M. Yardi, M. S. Hsiao, T. L. Martin, and D. S. Ha, "Quality-Driven Proactive Computation Elimination for Power-Aware Multimedia Processing," *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, 2005.

[11] "Lyrtech SFF SDR Development Platform," http://www.lyrtech.com/DSP-development/dsp_fpga/sffsdrdevelopmentplatform.php.

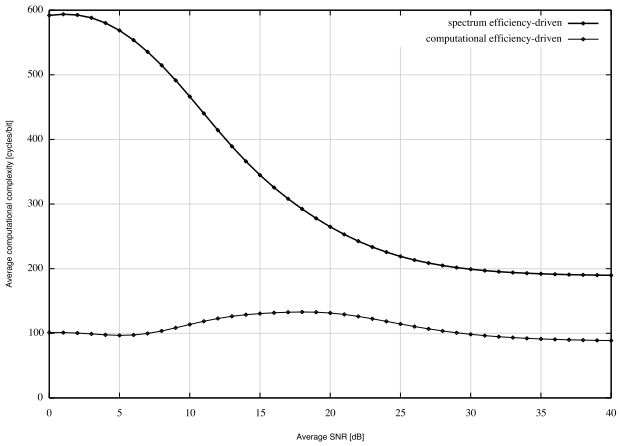
[12] A. Sinha and A. P. Chandrakasan, "Energy Aware Software," in *Thirteenth International Conference on VLSI Design*, 2000, pp. 50–55.

[13] ——, "JouleTrack–A web-based tool for software energy profiling," in *Proceedings of Design Automation Conference*, 2001, pp. 220–225.

[14] J. Torrance and L. Hanzo, "Upper bound performance of adaptive modulation in a slow Rayleigh fading channel," *IEEE Electronics Letters*, vol. 32, no. 8, pp. 718–9, April 1996.



(a) Average spectral efficiency, $\bar{\eta}(\Omega)$



(b) Average computational complexity, $\bar{\mathcal{K}}(\Omega)$

Fig. 2. Performance of adaptive switched modulation/coding scheme sets in a slowly-fading Rayleigh channel.