On the Estimation-Based Closed-Loop Power Consumption Control in Multimedia Mobile Devices

Qiong Tang, Ángel M. Groba, Eduardo Juárez and César Sanz
Centro de Investigación en Tecnologías del Software y Sistemas Multimedia para la Sostenibilidad (CITSEM)
Universidad Politécnica de Madrid (UPM)
Madrid, Spain
e-mail: qiong.tang@alumnos.upm.es; {angelmanuel.groba, eduardo.juarez, cesar.sanz}@upm.es

***Abstract*—In this paper, a closed-loop approach to control the power consumption of multimedia mobile devices is presented, such that the feedback signal is an estimation based on monitored system events. First, the power estimation method is presented and validated. Afterwards, prior to the implementation of the real-time control system, off-line estimation data are used to get a system model, which enables the application of classic control-theory methods to analyze and design an integral controller whose behavior is then simulated. The target system is a video decoder running in an embedded development platform. The simulation results show how the controller achieves null average steady-state error with short settling times, even in the presence of estimation noise or disturbance, thus predicting promising results for the closed-loop approach to the final real-time system implementation.**

Keywords—multimedia; embedded; power estimation; modeling; PMC; DVFS; closed-loop control.

I. INTRODUCTION

Embedded and mobile multimedia systems require, like others, the optimization of the quality of experience (QoE) they offer to the user. However, their common battery dependency makes also necessary the optimization of their energy consumption. Indeed, for example, the wide spectrum of usual available applications for current smart phones make them to have quite limited operating times, especially when they execute common video encoding, decoding and/or presentation applications. Furthermore, the increasing complexity of emerging video standards, such as High Efficiency Video Coding (HEVC) [1], will probably increase this limitation with respect to other previous ones, like H.264/AVC [2]. Therefore, there is an increasing effort into trying to reduce the energy consumption of this kind of systems from different points of view. Particularly, we are interested in optimizing their energy consumption in relation with applications of video decoding, obviously keeping a reasonable trade-off with QoE. Although our research group has been already working on these issues since several years ago [3][4][5], we are starting now a new research branch based on a less heuristic and more systematic approach, whose validity and efficiency is wanted to be tested.

As one of the first-stage results of this new approach, in this paper, we present the formal application of classic closed-loop control techniques to the power-consumption regulation of a video decoding application running in an embedded multimedia platform. This will be based on power estimations from available system indicators in order to avoid the need of a power monitor subsystem.

For this purpose, in general terms, the system should be modeled as a real-time closed-loop control system (see Figure 1), in which the *controlled output* follows the *target* signal (often called *set-point* in the Control jargon). This is achieved by a *controller*, which processes the system *error* between the target and the *feedback* information coming from a sensor and generates the *action* signal to the device under control (often called *plant* in the Control jargon). In a typical industrial process control, the plant is normally designed to be controlled in this way, so it usually offers action inputs able to vary the plant outputs and even sensors for feeding the output values back. In our case, we face the previous problem of adapting our plant (the embedded multimedia platform) to this topology, because it is not initially thought to be controlled in this way. Therefore, the first task is to identify and set up both an action and a feedback signal in the plant. For the former, we have identified and used the dynamic voltage and frequency scaling (DVFS) mechanism, present in many commercially available platforms and able to act on the system consumption by varying the operating performance point (OPP) of the microprocessor unit (MPU). For the later, what we have identified is a lack of direct consumption sensors in the majority of present commercial embedded platforms. Hence, we have decided to adopt an intermediate solution, which is to estimate the power consumption from other available signals in the plant, i.e., event counts. This leads to a structure like the one shown in Figure 1, which decouples the consumption optimization infrastructure from specific instrumentation needed to monitor actual power consumption, thus increasing the platform autonomy and the control-system applicability.

The power consumption estimation we present is based on previous work [3][4] in which static energy estimations are mathematically calculated for video decoding tasks for a fixed OPP by off-line correlation between actual energy measurements and significant-events counts taken from the processor performance monitoring counters (PMCs). Now, this estimation method is extended to a system in which the OPP is variable. In a next step, the estimations will be periodically calculated in real time and the estimator will feed consumption samples back to the controller, which, in turn, will drive the DVFS system to set the suitable OPP.

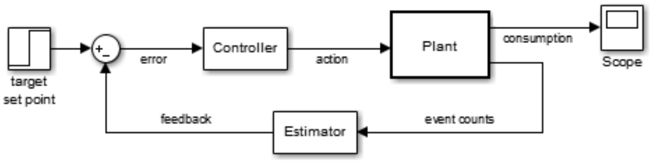

Figure 1. Diagram of a closed-loop consumption control system based on estimation feedback.

But before the control system is finally implemented to work in real time, a previous work of modeling and analysis is done to design and validate a first controller based on simulation results. This is the work presented in this paper, with the following structure: in Section II previous related work is presented; Section III describes the test bed used; in Section IV the power estimation process is sketched; in Section V classical control theory is applied to model the system, design the controller and get a system simulator; Section VI outlines the main results; and Section VII presents conclusion and future work.

II. Related Work

Given the increasing concern about saving energy whenever is possible, a great number of research developments can be found related to energy consumption optimization in microprocessor-based systems. They range from small battery-operated systems [4][6] to larger data centers or web servers [7][8], focusing also in multimedia applications [3][5][9]. Furthermore, the DVFS method is being used to act on the system consumption since a number of years ago [10][11][12]. On the other hand, the application of closed-loop techniques also appears in the literature of all these fields [6][7][9], also with widespread use of DVFS. However, among this group of solutions, there is not a clear proposal of how to feed back the closed-loop system, mainly because there is not an obvious feedback signal available in most conventional platforms, as mentioned above.

For example, in [13], [14] and [15], the controlled variable is the processor utilization factor (U), which is modulated through the DVFS system by means of P [13], PI [14] and PID [15] controllers. The energy savings increase as U approaches 100%, while meeting the task deadlines.

Another set of approaches are found in which the controlled variable is the occupancy level of certain system queues, given that keeping it constant implies that just the needed energy is being consumed. A couple of examples based also on DVFS are [16], in which authors explore the benefits of a control architecture on the throughput and energy consumption of a video encoder executed on a multiple-processor system on chip (MPSoC); and [17], where a nonlinear controller is used in queue-based streaming applications.

Also, there are approaches in which the control loop adapts the OPP to the just needed frequency by estimating the processor workload, like [18], where a Kalman filter estimates the computation time needed by MPEG-2 decoded frames.

In some specific cases, the target system includes a power monitor unit able to feed actual consumption data back to the closed-loop controller, as in [19] for a chip multiprocessor with an optimal controller. However, our aim is to reach a control system which can regulate the power consumption of an embedded multimedia system without the need of added power monitors but basing it on power estimations derived from commonly available information. It is an approach similar to the one used in [20], where the use of PMCs is proposed for estimating L2 cache consumption, but this is extended in our case to the processor and based on its DVFS mechanism.

Combining PMCs with multivariate adaptive regression splines (MARS) method to build an energy model has been successfully used in different cases [21]-[24]. Now, we apply this methodology to control video decoder consumption in closed-loop.

III. Test Bed

Our test bed is based on a single-core hardware development platform for multimedia embedded systems: BeagleBoard [25]. It features, apart from a number of peripherals, 2Gb NAND and 2Gb SDRAM of memory and an OMAP3530 processor system [26]. This system includes a MPU based on an up to 720-MHz ARM Cortex-A8 core, a digital signal processor (DSP) core and other coprocessors. The Cortex-A8 architecture includes 4 PMCs and one specific counter for CPU clock cycles. Related to the BeagleBoard peripherals, it is worth mentioning the possibility a subset of them offers for changing the MPU supply voltage and clock frequency (DVFS subsystem).

This development platform allows us to execute video decoding applications while monitoring their power consumption in order to tune the power estimator. An Agilent set of programmable power supply, battery emulator and PC-based GPIB-linked acquisition system [27] has been used to supply the board and acquire records of its current consumption (see Figure 2). To simplify the work and focus on the energy consumption caused by the MPU, the memory subsystem and the related I/O buses, the board has been configured as a minimal system that disables the unnecessary components.

With respect to the software part, a Linux 3.8.0 kernel, patched to support the platform DVFS mechanism, is running in the processor. On the other hand, taking advantage of our expertise in the Open RVC-CAL Compiler (Orcc) [3], a MPEG4-Part2 decoder is built from [28] to be used as the power-consuming video application. This user-level application is suitably modified to include the estimation module (see Figure 2). Besides, several video sequences [29], accessed through the platform SD card, are used to test the system.

The DVFS subsystem is managed through the *cpufreq* Linux driver. This driver includes four predefined governors to fix the MPU OPP, two static and two dynamic, which react to the system load. This is achieved by a function called *cpufreq_driver_target*, one of whose input parameters is the target frequency of the desired OPP to switch to. This function searches the target frequency among the ones of the OPPs defined in an internal table and selects the appropriate one by applying a ceil- or a floor-rounding algorithm, depending on another input parameter. The function then sets the frequency and the voltage corresponding to the selected OPP. The default *cpufreq* definitions for the BeagleBoard only consider 6 OPPs.

In order to decrease this strong nonlinearity in the DVFS-based plant input, additional valid OPPs were searched, verified and included into the *cpufreq* table, reaching a total of 27.

In this first stage in which the power estimator has to be validated prior to the final control-system implementation, a third-party tool, Performance Application Programming Interface (PAPI) [30], is used to access PMCs from the application level, offering both a high- and a low-level interface. The high-level interface includes start, stop and read sets of PMCs and other simple operations which can obtain accurate measurements of applications. The fully programmable low-level interface provides the possibility to control the counters. PAPI has been implemented on a number of Linux platforms and the latest release now provides support for ARM Cortex A8. In our work, PAPI is employed as the PMC driver, which is used from the application to take PMC event samples after decoding every video frame.

During decoding, the application sends signals to a modified *cpufreq* governor into the operating-system kernel to set the OPP, thus achieving test-oriented OPP changes, such as increase, decrease or jump among OPP table values. Meanwhile, the application also selects the suitable power estimation model depending on the active OPP. After that, the estimation procedure calculates a power estimation sample and then writes it into a SD-card file for off-line validation purposes. Figure 2 shows an explanatory block diagram of the test bed.

IV. POWER CONSUMPTION ESTIMATION

Our work is based on that presented in [3] and [4], where static estimations of CPU and memory energy consumption are mathematically calculated following the MARS methodology while executing video decoding tasks in a non-DVFS scenario. This is achieved by correlating actual energy measurements with significant-events counts taken from the processor PMCs. These previous approaches are enhanced in this paper to work in a scenario in which the OPP is variable and power estimation samples are successively required in execution time.

The first phase of the estimator implementation leads to identify the set of events which are most significant with respect to the power estimation. This is achieved by a filtering procedure. Thus, the Spearman's rank correlation ρSi is computed between each event and power consumption. After this step, a threshold α is established to eliminate any events below this threshold. On the other hand, to reduce event redundancy, correlations ρ(i,j) between each pair of events 'i' and 'j' are computed to identify the event relationship. The purpose is to eliminate those events whose information can be obtained from other events. Hence, starting from the event 'a' with the largest correlation ρSa, those events 'j' whose correlation ρ(a,j) exceeds certain threshold β are eliminated. Then, the procedure continues with event 'b', with correlation value ρSb, to eliminate the events 'j' whose ρ(b,j) exceeds β. This process is repeated until there are no more events to eliminate. Finally, the remaining events are highly related with power consumption and orthogonal among them. In this work, α is set to 0.5 and β is set to 0.9. TABLE I lists the events resulting from the filter procedure.

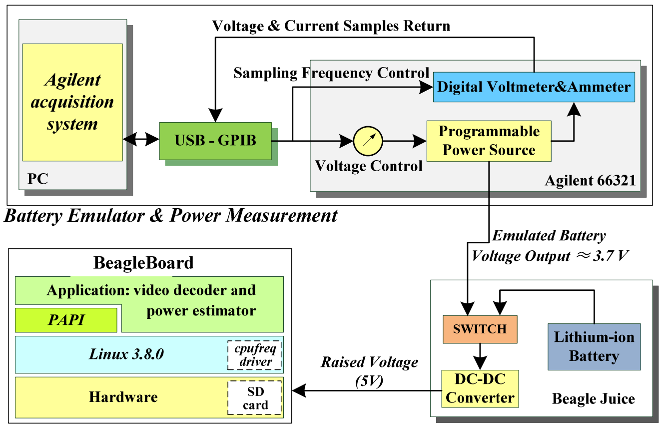

Figure 2. Block diagram of the test bed.

Once the list of significant events is obtained, the MARS method is applied to get an estimator able to estimate power consumption from event counts read from PMCs. Combining PMCs with MARS method to build an energy model has been successfully used in different cases [21]-[24]. Previous work proves that this methodology is also suitable for video decoders [22]. In our work, power estimation models have been tuned by combining 78 video sequences belonging to two resolution groups (CIF and QCIF) and by off-line correlation between event counts and power measurements. Since the measurement system captures current values of the BeagleBoard supply, the power is calculated multiplying them by the board supply voltage. Anyway, the acquired current signal is filtered to remove sporadic consumption spikes not due to the video decoding task itself. Models have been tested in this way with all the available test sequences against 27 OPPs and approximately 95% of the models obtained from a combination of training sequences have an average relative estimation error less than 5%.

Once implemented the estimator as explained above and in order to model the plant for designing the closed-loop controller, a subset of 8 sequences has been used to average the estimated power consumption for each OPP. TABLE II shows average values of estimated power, measured power consumption and relative error between them for all OPPs. The total mean relative estimation error is -0.37%.

V. SYSTEM MODELING, ANALYSIS AND DESIGN

One means of designing the system controller is to base it on a suitable model of the plant. As a first approach to the problem, a simplified model is used to facilitate the application of the classic control theory. Later, those models could be refined and sophisticated and different advanced closed-loop control strategies could be applied.

The records of consumption estimation used to validate the estimator module are also useful for modeling purposes. For example, if the estimator is considered to be included into the plant itself, the analysis of how it responds to a change in the OPP enables the plant dynamics modeling. Thus, for example, Figure 3 shows the estimator output for the OPP changing from number 26 to number 27 during a certain video sequence decoding. Since the estimation period is the video-frame

TABLE I. SELECTED EVENTS AND FUNCTIONALITY

PAPI events	Description
L2_TCM	Level2 total cache misses
TLB_IM	Instruction translation look aside buffer misses
BR_TKN	Conditional branch instruction taken
SR_INS	Store instructions executed
TOT_CYC	Total cycles

TABLE II. POWER ESTIMATION DATA FOR ALL OPPS

No.[a]	C[b]	E[c]	e[d]
1	0.930	0.926	-0.34
2	0.985	0.982	-0.26
3	0.995	0.990	-0.46
4	1.005	1.005	0.05
5	1.025	1.020	-0.46
6	1.035	1.031	-0.40
7	1.055	1.046	-0.78
8	1.075	1.074	-0.05
9	1.095	1.092	-0.20
10	1.120	1.112	-0.70
11	1.140	1.137	-0.20
12	1.165	1.160	-0.36
13	1.180	1.173	-0.60
14	1.205	1.197	-0.63

No.[a]	C[b]	E[c]	e[d]
15	1.225	1.222	-0.25
16	1.295	1.288	-0.47
17	1.315	1.313	-0.10
18	1.330	1.326	-0.26
19	1.350	1.347	-0.16
20	1.370	1.361	-0.66
21	1.389	1.381	-0.60
22	1.409	1.400	-0.70
23	1.439	1.438	-0.13
24	1.449	1.447	-0.16
25	1.484	1.479	-0.38
26	1.504	1.502	-0.16
27	1.634	1.625	-0.58

[a] OPP number; [b] Average consumption (W)
[c] Average estimation (W); [d] Relative error (%)

decoding period, it is long enough as to allow the estimator to complete its OPP switch from one sample to the next, as it can be seen at t=260s in Figure 3. Hence, a simplified discrete transfer function model of the plant could be G(z)=1/z [31], which relates the power estimation with OPP average power level. This simple model is valid while the system sample period is much longer than the settling time of the analog power consumption process or, correspondingly, than the time resolution of the PMCs used to estimate the power. The frame period applied to the estimations for open-loop tuning and modeling purposes is obviously long enough. Furthermore, although the sample period to be used in the real-time closed-loop control system will be shorter than the frame period, it will still be much longer than consumption settling time, otherwise the system overhead would be unbearable.

Thinking on implementing a closed-loop automatic power regulation system, its stability is one of the characteristics that must be ensured apart from other technological issues. As a first approach, the simplest controller that can be used in closed loop is a P one [32]. If we call K the gain of this controller, the transfer function of the closed-loop system is M(z)=K/(z+K). Therefore, the critical gain which leads the system to instability is K_c=1. Then, the lower bound for the closed-loop system error in steady state is min(e_{ss})=100/(1+ K_c)=50% [33], which is too high. In order to avoid this limitation and still keeping a classic linear controller, an integral action can be added to it.

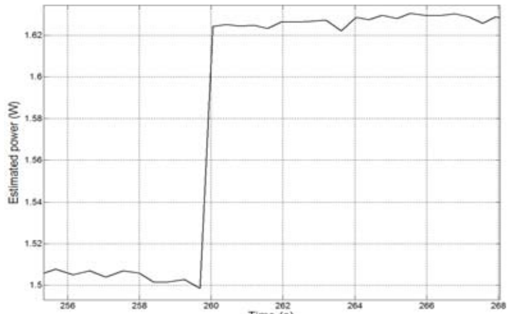

Figure 3. Estimated consumption for an OPP26 to OPP27 step.

As a first and simple approach to the integral action, one can choose between a forward and a backward rectangular rule (FRR and BRR, respectively) [31]. Their corresponding Z transfer functions differ only on a zero in z=0, which appears in the second case. Hence, the zero-pole cancellation in a series of a BRR I controller and G(z) will enable shorter settling times than with a FRR I because the system dominant pole can be closer to z=0.

Thus, considering the BRR option, the closed-loop pole of the system is p_{CL}=1-KT, being again K the controller gain. Let us consider a sample period T of 100ms, which seems to be a good trade-off value for keeping reasonable overhead, immunity to jitter effects and frequency of control actions. For this period, the critical gain which leads the system to instability (p_{CL}=-1) is K_c=20, whereas the gain for the shortest settling time (p_{CL}=0) is K=10. This is the gain used for the I controller, which can be seen included into the *Simulink* block diagram of Figure 4.

The block in Figure 4 between the controller and the plant is modeling the nonlinearity implied by the plant interface, which only admits 27 different levels, i.e., the 27 available OPPs. It is modeled as a quantization process, whose steps are defined by the average power estimation values shown in TABLE II. In order to limit the maximum quantization error to ±step/2, the breakpoints are set in the midpoint between estimation levels. Since, for simplifying purposes, the plant model expects power levels as input values, the final implementation will include a module which translates each of the 27 average power estimation values into its corresponding OPP frequency as the input parameter for the *cpufreq* interface.

The closed-loop diagram of Figure 4 also includes a disturbance input in order to test the capability of the system to react to disturbances on its controlled output. The disturbance input would simulate the effect of a consumption variation when the system is following the set point in steady state, due, for example, to a variation in the processor load.

The disturbance input of Figure 4 can also be used to inject real *estimation noise* into the ideal signals of the *Simulink* model. For example, the third column of TABLE I represents average values of power estimation for each OPP but the actual estimation values do fluctuate around those averages, as can be distinguished in Figure 3. The estimation signal can be modeled in each OPP as a constant (its corresponding average value) plus a random-like "noise" with a maximum peak close to 9mW and zero mean. Figure 5 shows an example of this noise.

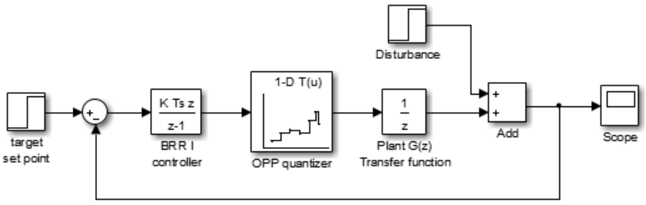

Figure 4. Diagram of the system simulator with disturbance input.

VI. RESULTS

The system simulator of Figure 4 has been tested for a number of set points and disturbances, mainly step shaped. It is worth noting that a step-shaped input would simulate a constant power desired for the system consumption (set point) or a sudden change in the system power consumption (disturbance). Therefore, the following results must be understood within a scenario in which the system is continuously consuming power, for instance in a video-streaming application, which in turn is desired to be constant, regardless now of energy or QoE issues.

As a summary, Figure 6 shows the system time response for a set-point step from OPP1 level to OPP15 level in t=0 and a disturbance step (undesirable and unexpected increase of consumption) of a 40% of the input step in t=0.5s. In that figure, it can be seen, on one hand, how if the OPP is changed at t=0, the power estimator would not reflect the corresponding consumption change until next sample period (t=0.1s). On the other hand, with the system consumption stable at OPP15 level, the power estimator outputs a sudden consumption increase at t=0.5s, which keeps until the next sample time at t=0.6s. At that moment, the I controller detects the anomaly, i.e., a power consumption higher than desired, and corrects it immediately by decreasing its output to the plant (i.e., by setting a lower OPP). However, since the disturbance value implies that there is not any OPP which cancels exactly its effect, i.e., none OPP applied to the plant reaches a consumption equal to the target, the I controller makes the response oscillate. This oscillation in the system power consumption can be seen, on one hand, as an undesirable behavior of the system, given that the set point is not oscillating, or, on the other hand, as the only way the nonlinear control system can satisfy the set-point requirement, in average.

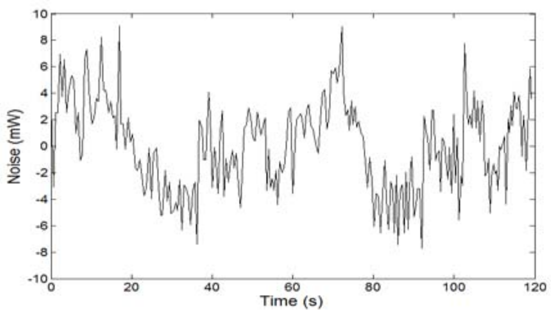

Figure 5. Simulated noise to be added to the estimation signal.

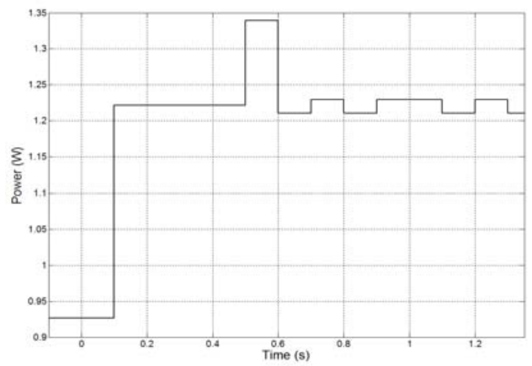

Figure 6. Closed-loop time reponse for an OPP1 to OPP15 input step and disturbance of 40% at t=0.5s.

If the disturbance input of Figure 4 is used to inject the characteristic noise of the power estimation (see Figure 5) and we let enough response time, the system output for an example set point equivalent to OPP15 level is the one shown in Figure 7. In that figure, it can be seen how the system reaches a consumption with the average value of OPP15 (see TABLE II), as desired, but some glitches appear occasionally. What happens is that, due to the quantized input to the plant, the noise deviations cannot be corrected until the accumulated error in the I controller reaches an OPP breakpoint threshold. In that moment, perhaps the minimum action change, i.e., the next OPP level, is greater than needed, thus triggering the aforementioned glitches, whose levels correspond to the neighboring OPP values. This is one of the disadvantages of the I controller, which should be solved or minimized in the implemented system by applying specific corrective techniques or other types of controllers.

VII. CONCLUSION AND FUTURE WORK

A power consumption estimator, which estimates the power consumption of a video decoding application running in a commercial embedded development platform has been used to get a mathematical model of the consumption process. The estimator, which is based on PMC values and is sensitive to the active OPP, could be deployed in a wide range of systems without needing specific power monitoring hardware. From the mathematical model, classic analysis and design techniques have been applied to get a suitable controller able to keep track of the power consumption in closed loop. Prior to the implementation of the real-time control system, simulation results have proved the system stability and average regulation of power consumption according to the set point and in the presence of consumption variations and estimation noise. This paves the way for applying such closed-loop techniques to optimize the power consumption of multimedia hand-held devices.

From now on, the real-time control system has to be implemented by feeding power estimations back. Then, its response will be contrasted with the simulation results presented in this paper. This will open the door to further

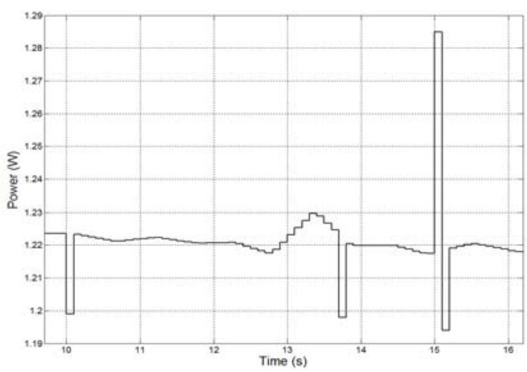

Figure 7. Closed-loop reponse for OPP15 target and noise disturbance.

improvements, such as the test of different controllers and design methods, or the suitable programming of a variable set point to achieve different objectives involving battery life time, QoE or performance parameters, among others, all related to the power optimization problem. Also, the test bed can be moved from uniprocessor to multiprocessor platforms, managing issues like load distribution or multiple power sinks.

ACKNOWLEDGMENT

This work has been supported by the Spanish Ministry of Economy and Competitiveness under grant TEC2013-48453-C02-2-R. Authors wish also to acknowledge the support of the UPM Department of Telematic Engineering and Electronics.

REFERENCES

[1] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Trans. on Circuits and Systems for Video Technology, vol. 22, issue 12, Dec. 2012, pp.1649-1668.

[2] ITU-T and ISO/IEC JTC 1, "Advanced Video Coding for Generic Audiovisual Services," ITU-T Rec. H.264 & ISO/IEC 14496-10.

[3] R. Ren et al., "A PMC-driven methodology for energy estimation in RVC-CAL video codec specifications," Signal Processing: Image Communication, vol. 28, issue 10, Nov. 2013, pp. 1303–1314.

[4] R. Ren, E. Juárez, F. Pescador, and C. Sanz, "A stable high-level energy estimation methodology for battery-powered embedded systems," Procs. of 16th IEEE Intl. Symposium on Consumer Electronics (ISCE 2012), June 2012, pp. 1-3.

[5] E. Juárez, F. Pescador, P. Lobo, A. Groba, and C. Sanz, "Distortion-energy analysis of an OMAP-based H.264/SVC decoder," Mobile Multimedia Communications, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Vol. 77, 2012, pp 544-559.

[6] D. Le and H. Wang, "An effective feedback-driven approach for energy saving in battery powered systems," Procs. of 18th IEEE Intl. Workshop on Quality of Service, June 2010, pp. 1-9.

[7] X. Wang and Y. Wang, "Co-Con: coordinated control of power and application performance for virtualized server clusters," Procs. of Intl. Workshop on Quality of Service, 2009, pp. 1-9.

[8] V. Petrucci, E. V. Carrera, O. Loques, J. C. B. Leite and D. Mossé, "Optimized management of power and performance for virtualized heterogeneous server clusters," Procs. of Intl. Symp. on Cluster, Cloud and Grid Computing, 2011, pp. 23-32.

[9] G. Cao, A. Ravindran, S. Kamalasadan, B. Joshi and A. Mukherjee, "A cross-stack predictive control framework for multimedia applications," Procs. of Intl. Symposium on Multimedia, 2013, pp. 403-404.

[10] D. C. Snowdon, S. M. Petters, and G. Heiser, "Accurate on-line prediction of processor and memory energy usage under voltage scaling," Procs. of 7th Intl. Conf. on Embedded Software, Sep. 2007, pp. 84-93.

[11] M. E. Salehi, M. Samadi, M. Najibi and A. Afzali-Kusha,, "Dynamic voltage and frequency scheduling for embedded processors considering power/performance tradeoffs," IEEE Transactions on Very Large Scale Integration Systems, vol. 19, issue 10, Aug. 2010, pp. 1931–1935.

[12] X. Lin, Y. Wang, Q. Xie and M. Pedram, "Task scheduling with dynamic voltage and frequency scaling for energy minimization in the mobile cloud computing environment," IEEE Transactions on Services Computing, in press.

[13] X. Wang, X. Fu, X. Liu and Z. Gu, "PAUC: power-aware utilization control in distributed real-time systems," IEEE Trans. on Industrial Informatics, vol. 6, no. 3, Aug. 2010, pp. 302–315.

[14] A. S. Ahmadian, M. Hosseingholi, and A. Ejlali, "A control-theoretic energy management for fault-tolerant hard real-time systems," Procs. of IEEE International Conference on Computer Design (ICCD), Oct. 2010, pp. 173-178.

[15] A.K. Mishra, S. Srikantaiah, M. Kandemir and C.R. Das, "CPM in CMPs: coordinated power management in chip-multiprocessors," Procs. of Intl. Conf. for High Performance Computing, Networking, Storage and Analysis, 2010, pp. 1-12.

[16] S. Garg, D. Marculescu and R. Marculescu, "Custom feedback control: enabling truly scalable on-chip power management for MPSoCs," Procs. of International Symposium on Low-Power Electronics and Design, 2010, pp. 425 - 430.

[17] A. Alimonda, S. Carta, A. Acquaviva, A. Pisano, and L. Benini, "A feedback-based approach to DVFS in data-flow applications," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 28 , issue 11, Nov. 2009, pp. 1691–1704.

[18] S. Y. Bang, K. Bang, S. Yoon, and E. Y. Chung, "Run-time adaptive workload estimation for dynamic voltage scaling," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 28, issue 9, Sep. 2009, pp. 1334–1347.

[19] Y. Wang, K. Ma, and X. Wang, "Temperature-constrained power control for chip multiprocessors with online model estimation," Procs. of 36th International Symposium on Computer Architecture (ISCA), June 2009, pp. 314-324.

[20] X. Wang, K. Ma, and Y. Wang, "Achieving fair or differentiated cache sharing in power-constrained chip multiprocessors," Procs. of 39th Intl. Conf. on Parallel Processing, 2010, pp. 1-10.

[21] B. Goel et al., "Portable, scalable, per-core power estimation for intelligent resource management," Procs. of International Green Computing Conference, Aug. 2010, pp. 135-146.

[22] Y. Xiao et al., "A system-level model for runtime power estimation on mobile devices," Procs. of IEEE/ACM Intl. Conf. on Green Computing and Communications, 2010, pp. 27-34.

[23] K. Singh, M. Bhadauria, and S. A. McKee, "Real time power estimation and thread scheduling via performance counters", ACM SIGARCH Computer Architecture News, vol. 37, issue 2, July 2009, pp. 46-55.

[24] C. Lively et al., "Power-aware predictive models of hybrid (MPI/OpenMP) scientific applications on multicore systems", Computer Science-Research and Development, vol. 27, issue 4, Nov. 2012, pp. 245-253.

[25] BeagleBoard.org Foundation. *BeagleBoard*. [Online]. Available from: http://beagleboard.org/beagleboard, 2015.02.20.

[26] Texas Instruments. *OMAP3530 Applications Processor*. [Online]. Available from: http://www.ti.com/product/omap3530, 2015.02.20.

[27] Agilent Technologies. *Agilent 14565B Software and 66319B/D and 66321B/D Mobile Communications DC Sources*. [Online]. Available from: http://cp.literature.agilent.com/litweb/pdf/5990-3503EN.pdf, 2015.02.20.

[28] GitHub ORCC. *Orc-apps, Mpeg4 Part2*. [Online]. Available from:https://github.com/orcc/orc-apps/tree/master/RVC/src/org/sc29/wg11/mpeg4/part2, 2015/02/23.

[29] SourceForge. *Open RVC-CAL Compiler, sequences*. [Online]. Available from: http://sourceforge.net/projects/orcc/files/Sequences, 2015.02.23.

[30] PAPI. *Performance Application Programming Interface*. [Online]. Available from: http://icl.cs.utk.edu/papi/, 2015.02.23.

[31] G. F. Franklin and J. D. Powell, "Digital control of dynamic systems," 3rd edition, Addison Wesley, 1997.

[32] C. L. Phillips and J. M. Parr, "Feedback control systems," 5th edition, Prentice Hall, 2010.

[33] K. Ogata, "Modern control engineering," 5th ed., P. Hall, 2010.