Closed-Loop System to Guarantee Battery Lifetime for Mobile Video Applications

Ángel M. Groba, Pedro J. Lobo, and Miguel Chavarrías

***Abstract*—The pervasive utilization of mobile technology for an increasing number of utilities and applications enhances the importance of the battery-energy optimization. Among those applications, the ones related to digital video are very popular. When using this type of applications, users may want to ensure a certain battery lifetime in order to complete the viewing of a given video content. This paper presents the design of a closed-loop control system that, from the user-desired battery lifetime, regulates the battery discharge rate in order to meet the user expectations regardless of the real-time dynamic power variations of the video decoding activity. The system is validated first by simulation and then by real tests, since it has been implemented in the operating system (OS) of a commercial development board. A simple proportional controller is able to limit the system error to an equivalent difference of only 0.4 s between the target and the achieved lifetimes, with very little computational overhead. Besides, its OS-based implementation makes it transparent to the user-level applications. Some tests show how the system is able to achieve the same lifetimes as the ones achieved with a couple of Linux-based governors, but with the determinism that the user target lifetime sets from the beginning of each test. Moreover, further tests show that the deterministic lifetimes can be extended up to 12% and 4% beyond those two governors, respectively.**

***Index Terms*—Battery-lifetime guarantee, dynamic-power regulation, dynamic voltage and frequency scaling (DVFS), mobile video.**

I. Introduction

ENCOURAGED by the expected increase in the number of 5G subscriptions, the possibilities of new and already available mobile services will skyrocket in next years [1]. All these services, with evident energy needs, have to be kept working by means of the batteries included into the mobile terminals, which are wanted to be lightweight and manageable as well as to have a long time of autonomy. Since the quick development of mobile-systems capabilities is followed by a not so quick development of the capacity/volume ratio of the batteries [2], much effort is put into the power-aware design of systems and applications, as outlined in Section II.

For the particular case of mobile video applications, with viewing times and resolutions that grow day by day [1], general power-saving strategies may not be enough and some other specific power-aware algorithms are needed. For instance, one clear expectation of the mobile-video consumers is that the available battery energy can be managed to last, at least, until the video-content viewing is completed. Sometimes this is considered by users even more important than the quality which the video is viewed with. It can be motivated by the finite and known duration of the video content (e.g., a sporting match or a series episode) or the finite and known duration of the entertainment time interval that is wanted to be covered (e.g., a ride on public transport).

The work presented in this paper proposes a closed-loop control system to fulfil that expectation, i.e., to make the battery last for a certain desired period of time during which the mobile terminal is decoding video. What the system does is to regulate the battery discharge rate according to a certain reference (set-point or input signal) in the form of which the battery state of charge (SoC) should be at every instant. This reference is linearly decreased in real time to achieve the user expectation, which will have been set at a given moment, either at the start or during the video decoding, in the form of the desired (target) battery lifetime. As it will be explained later in more detail, the system feedback signal is the real battery SoC, such that, depending on the difference between it and the reference one, i.e., the system error, the controller calculates a suitable action signal to adapt automatically the power consumption of the video decoder. This is implemented through the common infrastructure of dynamic voltage and frequency scaling (DVFS), which selects one of the available operating performance points (OPP), i.e., a voltage-frequency pair. This description can be better understood by observing the block diagram depicted in Fig. 1. The disturbance arrow represents the uncontrolled and unavoidable variations of the decoder power consumption, which are due to workload variations inherent to the decoding activity. Indeed, this fact is just what justifies the usefulness of the closed-loop proposal, given that such a control system is expected to be robust against disturbances. Finally, it is worth mentioning that the system is proposed to be implemented within the operating system (OS) of the mobile platform, so that user-level video applications do not need to be aware of this infrastructure. Anyway, video-decoding applications are usually provided with a certain capability to adjust their performance to system restrictions by, for instance, dropping frames or lowering resolution.

The work has covered model, design, simulation, implementation and experimentation phases, which are described in the paper with the following structure. Section II summarizes

Manuscript received September 14, 2018; revised November 28, 2018; accepted December 31, 2018. Date of publication January 7, 2019; date of current version January 24, 2019. This work was supported by the Spanish Government under Grant TEC2016-75981-C2-2-R. *(Corresponding author: Ángel M. Groba.)*

The authors are with the Research Center on Software Technologies and Multimedia Systems for Sustainability, Universidad Politécnica de Madrid, 28031 Madrid, Spain (e-mail: angelmanuel.groba@upm.es; pedro.lobo@upm.es; miguel.chavarrias@upm.es).

Digital Object Identifier 10.1109/TCE.2019.2891178

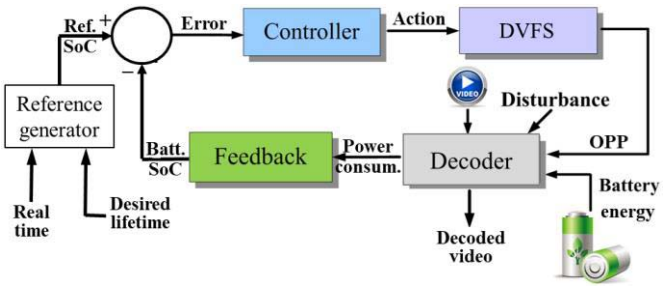

Fig. 1. Block diagram of the proposed system.

other related works; Section III raises the system model and, based on it, presents the controller design; Section IV highlights the relevant features of the test bench used to test the proposal; Section V outlines the simulation model used to carry out preliminary tests and advances some promising results; Section VI presents results from the implemented system and compares them to a couple of general Linux-based power-saving approaches; finally Section VII includes a conclusion.

II. Related Work

The growing demand for energy and the problems associated with its production and consumption make it necessary to invest and research in methods that minimize the negative impact of this fact. Indeed, there is a general sustainability need that affects many technological fields. With respect to electronic computing systems, apart from the general energy-saving constraints, when dealing with battery-operated devices, an additional issue appears which is the battery lifetime or the time between battery recharges, which users want to prolong as much as possible. Thus, a great number of references can be found in this research area, from general approaches [3] to specific high-energy-consumption applications [4] or low-energy-consumption applications for mobile devices [5].

Dealing with this latter application field of mobile computing, there are many approaches that, addressing different aspects of the systems, try to save battery energy. For example, Wei *et al.* [6], [7] proposed and implemented a system-level energy-aware task-scheduling algorithm for battery-limited mobile systems. Or with respect to mobile video applications, Jeong *et al.* [8] and Sidaty *et al.* [9] applied low-level video-specific techniques to reduce the energy consumption of the decoding task.

As in the work presented in this paper, there are approaches like [9] and [10] which use the DVFS mechanism as a means to influence the power consumption of video decoders. Both of them are based on models of the expected decoding complexity to decide the minimum needed OPP, the former in a video-segment basis using scalable video coding and the latter in a video-frame basis using MPEG metadata. Besides, the latter compares its results to the *Ondemand* Linux governor. Indeed, the classic dynamic *cpufreq* Linux governors, i.e., *Conservative* [11] and *Ondemand* [12], are other examples of DVFS-based energy-saving algorithms. In particular, they are closed-loop approaches in which the feedback signal is an estimation of the processor workload, while the action variable is set through the DVFS OPP. Following different control strategies, more aggressive in the *Ondemand* case than in the *Conservative* one, but nonlinear and based on hysteretic cycles in both cases, these governors tend to decrease the OPP as much as possible, reducing then the dynamic power consumption.

Other previous works in which closed-loop systems are used to manage some system parameters that influence the power consumption can be highlighted. Le and Wang [13] proposed a feedback-based model to achieve power reduction when scheduling I/O tasks in battery-operated systems and taking into account quality-of-service constraints. Another couple of feedback-based proposals are [14] and [15], which use DVFS techniques to adapt the power consumption of multiprocessor (MP) systems in a two-level strategy, first, assigning workload-based references to the processors and, second, controlling the utilization of each processor according to the assigned references. In similar terms, Garg *et al.* [16] applied DVFS-based closed-loop techniques to optimize the energy consumption of a video encoder in MP architectures by regulating the status of some relevant system queues.

With respect to video-decoding applications, Tang *et al.* [17] proposed a closed-loop system to control the power consumption of a video decoder in a consumer-electronics mobile platform based on power-estimation feedback [18], thus avoiding the need of a power-consumption sensor. The system acted on the DVFS mechanism of the processor to adjust the power consumption to the set-point input in real time, regardless of the power demand of the video decoder. This control system was later enhanced with a power-aware governor [19], which used the closed-loop system feedback itself to estimate the energy consumed and decide in real time a suitable set point for the control system. Several set-point profiles were tested in order to evaluate their energy-saving capabilities. Furthermore, since some kind of tradeoff can be established between dynamic power consumption and quality of experience in video decoders, Groba *et al.* [20] adapted the previously proposed power-consumption control system to a new application in which the controlled (and feedback) variable was the decoder slack time while maintaining the DVFS mechanism as the action variable. Here the control system decided the OPP to keep the decoder slack time close to the desired (set point) slack time, regardless of the real-time computational needs of the decoder. Setting low set-point slack times, the processor worked close to the minimum required performance thus lowering the dynamic power consumption.

The references outlined above are examples of the great effort put in the necessary general aim of optimizing the energy consumption of different applications in different platforms and for different optimization criteria. However, as mentioned above, for the specific case of battery-powered systems, users may have sometimes the particular need of ensuring a certain battery lifetime to complete some task, even sacrificing features that in other situations can be considered essential. Obviously, there is not so much research work in this concrete issue. The work of Cho *et al.* [21] can be mentioned

as an example in which a scheduling scheme is proposed to guarantee battery lifetime for a set of selected applications in a multitasking environment. Apart from other monitoring approaches which estimate the remaining lifetime depending on previous consumption data [22], the novelty of the system proposed in this paper lies in the fact that it allows mobile-video consumers to specify a desired lifetime for all or part of the remaining battery energy at a given moment from which a certain multimedia content is going to be processed. From this premise, the proposed closed-loop control system regulates the battery discharge rate acting on the decoder power consumption according to the real-time battery SoC feedback in order to achieve the user requirement.

III. Method and Model

As previously anticipated, the proposed control system is intended to allow video users to set a target lifetime (TL) for a certain percentage of the remaining battery energy at a given moment, which could be the 100% by default. That portion of the battery energy that is wanted to be consumed during TL will hereinafter be referred to as C, since it will be measured in terms of battery charge (in mAh). Hence, without having a priori knowledge about what the power needs of the decoder will be throughout time, a linear discharge pattern is considered, with a rate of C/TL mAh/s. Therefore, a closed-loop system is proposed to control this discharge rate regardless of the real-time power demands of the decoder. In a first approach, for the sake of simplicity, the system design will be based on a linear model. In a further approach, the main nonlinear issues will be considered in a simulation phase (see Section V) in order to check how they affect the designed system.

A. Process Model

Taking into account the good behavior achieved in other previous closed-loop proposals [17], [20], the same DVFS-based action signal is going to be used here in order to act on the decoder performance. As in those previous approaches, the control period T will be set to a value low enough to ensure a valid monitoring process, but high enough to avoid unnecessary overload. In current DVFS-capable systems [23], this last premise implies that after a change in the OPP, the new system condition (power consumption) is already stable when the next control instant comes (after T s). Thus, as in [17] and [20], a simplified Z-transform-based linear model of the decoder power-consumption dynamics can be the one shown in (1), where $G(z)$ represents the one-period delay between the power consumption dictated by the DVFS action $A(z)$ and its actual effect on the decoder power consumption $P(z)$

$$G(z) = \frac{P(z)}{A(z)} = \frac{1}{z}. \tag{1}$$

However, unlike in the aforementioned closed-loop approaches, where power consumption [17] or slack time [20] were the controlled variables used as system feedback, this new proposal aims to control the battery SoC throughout time, and then, this will be the feedback variable. To do so, and knowing that the DVFS-based action affects the power consumption of the decoder, an estimation of the battery SoC at a time t, in hours, can be obtained with (2) from a simplified battery model:

$$\mathrm{SoC}(t) = C - \int_0^t i(\tau)\cdot d\tau = C - \int_0^t \frac{p(\tau)}{v(\tau)}\cdot d\tau \,\mathrm{mAh}, \tag{2}$$

where $i(\tau)$ is the instantaneous current intensity draining the battery at an instant $t = \tau$, in mA; $p(t)$ is the instantaneous power consumption, in mW; $v(t)$ is the instantaneous supply voltage, in V; and being $t = 0$ the initial instant in which the user launches the controlling task.

Nevertheless, since the implementation of that estimation will run periodically (every T s) within the control system, a discretized version of (2) is needed. Approximating the supply voltage to a constant value V, the Z transform of a discretized version of (2) is

$$\mathrm{SoC}(z) = C - \frac{K_i\cdot z}{z-1}P(z)\ \mathrm{mAh}, \tag{3}$$

where K_i is an integration factor with the following expression

$$K_i = \frac{T}{3600\cdot V}\frac{\mathrm{h}}{\mathrm{V}}. \tag{4}$$

The mathematical model of the process can be described graphically as in Fig. 2.

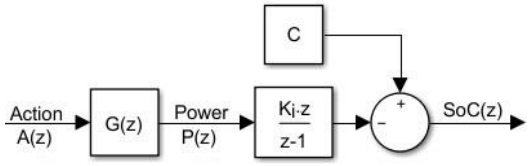

Fig. 2. Graphical description of the process model.

Understanding C as a constant offset in (2) and (3) and starting from (1), a Z transfer function can be used to model the process dynamics (PD) as the quotient between the Z transforms of SoC and DVFS action

$$\mathrm{PD}(z) = \frac{\mathrm{SoC}(z)}{A(z)} = -\frac{K_i}{z-1}\frac{\mathrm{mAh}}{\mathrm{mW}}. \tag{5}$$

B. Controller Design

1) Accuracy Focus: From the first-order pole in $z = 1$ that appears in (5), it is clear that the process dynamics within a closed-loop system would make it to behave as a type-1 control system with respect to the steady-state accuracy. This implies that such a control system, in a linear approach, would have no steady-state error for constant set points. However, since the proposed system will control the battery SoC and it is not realistic that the battery SoC can be kept constant during video decoding, no constant set points can be applied to the control system. In fact, as explained above, the set point should mark a constant rate of battery discharge, i.e., it should be a ramp-function reference with a slope R as follows

$$R = -\frac{C}{\mathrm{TL}}\frac{\mathrm{mAh}}{\mathrm{s}}. \tag{6}$$

A type-1 control system has non-null steady-state error for a ramp-function reference input (e_{ssr}), which can be calculated

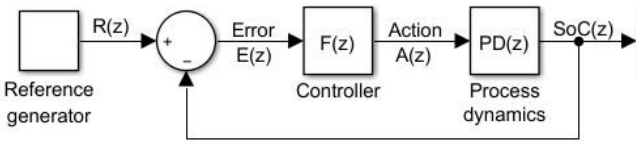

Fig. 3. Graphical description of the linear control-system model.

as in (7) for the expression of R in (6):

$$e_{ssr} = \frac{R \cdot T}{K_r} = -\frac{C \cdot T}{\mathrm{TL} \cdot K_r}\mathrm{mAh}, \tag{7}$$

where K_r is the ramp-error constant [24] of the system. The following expression shows the calculation of K_r for the process dynamics defined in (5):

$$K_r = \lim_{z\to 1}(z-1)[F(z)\cdot \mathrm{PD}(z)] = -\lim_{z\to 1}[F(z)\cdot K_i]. \tag{8}$$

F(z) in (8) is representing the Z transfer function of the controller, understood as the system block that calculates the action values from the system error ones, as its position in the block diagram of Fig. 3 indicates. If a low enough value is admitted for e_{ssr}, then a proportional (P) controller can be initially considered, due to the clear advantage of its simplicity. Indeed, the transfer function of this type of controllers is simply a gain (K_P), like this:

$$F(z) = \frac{A(z)}{E(z)} = K_P\frac{\mathrm{mW}}{\mathrm{mAh}}, \tag{9}$$

where $E(z)$ is the Z transform of the error sequence, i.e., $E(z) = R(z) - \mathrm{SoC}(z)$, which is the difference between the reference and feedback values, as depicted in Fig. 3. Since the steady-state value of $E(z)$ for a ramp-function input is calculated with (7), if (9) and (4) are applied to (8), and (8) is applied to (7), the value of e_{ssr} for a P controller is

$$e_{ssrP} = \frac{C \cdot T}{\mathrm{TL} \cdot K_P \cdot K_i} = \frac{3600 \cdot C \cdot V}{\mathrm{TL} \cdot K_P}\mathrm{mAh}. \tag{10}$$

From (10), it is clear that e_{ssrP} is constant, which implies that the controlled battery SoC is a ramp function of time, parallel to the reference ramp but at a constant distance equal to e_{ssrP}. This distance in magnitude has a time counterpart in the sense that the really achieved battery lifetime (AL) will be different from TL. Indeed, AL can be calculated from (6) and (10) as

$$\mathrm{AL} = \mathrm{TL} + \frac{e_{ssrP}}{R} = \mathrm{TL} - \frac{3600 \cdot V}{K_P}\mathrm{s}. \tag{11}$$

It is worth noting a couple of issues that turn out from (11). On one hand, the difference between TL and AL does not depend on TL, in fact, it is a constant so that the relative difference decreases as TL increases. On the other hand, given that the gain K_P has to be negative in order to neutralize the minus sign of PD(z) [see (5)] and thus ensure negative feedback in the closed loop represented in Fig. 3, AL will always be longer than TL. This is a point in favor of the P controller, although at the cost of lowering performance. In any case, as verified in next simulation- and empirical-results sections, the value of e_{ssrP}, and the corresponding extension of AL beyond TL or the performance lowering are practically negligible for the value of K_P to be chosen.

2) Stability Focus: Although the aim of the proposed control system has more to do with the steady state in the sense that, at the end, a certain TL is wanted to be reached, other more transient-related issues cannot be neglected when designing the controller. In fact, K_P is really calculated based on them and then checked in terms of the steady-state accuracy achieved with it. The calculation can be addressed from the system-stability point of view in a twofold way. Firstly, the absolute stability of the closed-loop system has to be ensured. For example, as said above, the controller gain has to be negative in order to ensure negative system feedback. Secondly, a suitable relative stability should be fixed such that the system response is not too slow in reaching the steady state, but neither too fast in order to avoid potential frequent oscillations.

Although systematic techniques could be applied, for a simplified first-order model such as the one proposed above, it is easier and more flexible to fix directly the system pole in the right place. Thus, to meet the premises stated above, the system pole should be placed in an intermediate point between 0 and 1 in the Z plane. Trying to promote a bit the response smoothness over the shortening of transient duration, a value of 0.75 (closer to 1 than to 0) is proposed for the system pole, which, in turn, enables the calculation of the needed controller gain. Indeed, the pole is the solution of the characteristic equation of the system represented in Fig. 3, i.e.,

$$1 + F(z)\cdot \mathrm{PD}(z) = 1 - \frac{K_P \cdot K_i}{z-1} = 0 \Rightarrow z - 1 - K_P \cdot K_i = 0. \tag{12}$$

Therefore, from (12) and (4), the needed K_P is

$$K_P = \frac{0.75-1}{K_i} = -\frac{900 \cdot V}{T}\frac{\mathrm{mW}}{\mathrm{mAh}}. \tag{13}$$

IV. Implementation Details

The control system proposed in the previous section needs to be validated. In this paper, it is not only done by simulation but also by implementation in a real multimedia system. For this purpose, the ideal linear model considered in Section III has to be particularized and completed with specific details of the real system in which it is going to be implemented. It consists of a commercial development board with a LinuxOS in which the control system has been integrated. Besides, a video decoder is executed in the board processor to decode some test video clip, thus consuming energy from the battery. A brief description of these parts is given below.

A. Development Board

It is a commercial one that has been chosen for its simplicity and low cost in order to be close to the consumer-electronics market and to minimize the implementation variables of the proposed linear model. Thus, although it has an assistant DSP and other coprocessors, only its single-core main processor with its PoP memory (256-MB NAND and 256-MB SDRAM) have been used to implement and test the system. The DVFS infrastructure has been configured and is accessed through the *cpufreq* Linux driver (Linux 3.8.0 kernel) to put the CPU to work in one of 27 available OPPs, ranging from 125 MHz at

0.98 V (OPP1) to 720 MHz at 1.31 V (OPP27). This will be the action interface of the proposed control system.

With respect to the feedback interface of the control system, not having a direct signal of power consumption or battery SoC in the development board, an alternative solution is proposed for this limitation. It consists on estimating the power consumption during video decoding and calculating the battery SoC with (2), or its equivalent discrete Z-transform version (3), from the estimated power. For the power-consumption estimation, an estimator based on counts of some events relevant for the power consumption of video decoders [25], and obtained from the processor PMCs (Performance Monitoring Counters) [18], has been used due to its good behavior in previous implementations [17], [26]. The estimator is first tuned offline against actual current measurements of the board, with all its unnecessary modules switched off in order to ensure that the bulk of the current is due to the decoding activity. The real-time power-consumption samples are obtained by multiplying the board current estimations by the board supply-voltage value. In this respect, it is worth saying that, for the sake of experiment convenience, the board is not really battery operated but supplied with a 5 V ac/dc power adapter. This means that $V = 5$ V.

The whole control system, including the power estimator and the SoC calculator, is implemented as an additional *cpufreq* governor, which is activated when a video decoding activity is launched. The benefit of this option is twofold: the *cpufreq* DVFS interface is easily accessible and the user multimedia tasks can be run transparently to the underlying control system. As in other previous related implementations [17], [19], the execution period of the system is fixed in $T = 100$ ms by taking into account the related remarks in Section III, i.e., it is long enough for $G(z)$ to be valid but short enough as to keep good track of the battery-discharge process.

B. Testing-Purpose Video Decoder

Since the proposed control system is intended for regulating the battery discharge during video decoding activity, a user-level video decoder has to be executed while the OS-level control system is being tested. Given the medium/low performance features of the selected board, an MPEG4-Part2 decoder [27] has been chosen. On the other hand, the video clips to be decoded have been taken among standard conformance ones [28]. Particularly, in order to ease the analysis of the control-system behavior, specific video sequences have been constructed by concatenating in series a couple of those clips with very different characteristics. One of them, "mat001.m4v", contains frames with little movement or change, whereas the other, "mat000.m4v", contains much more movement or change from frame to frame. Moreover, both clips are decoded according to two different frame rates, i.e., 25 and 30 frames/s (fps). The reason for that is that the control system has to be immune to the variations in the power demands of the decoder and hence, for testing purposes, the power demand of the decoder is made to vary by swapping the video clip and the required frame rate. In fact,

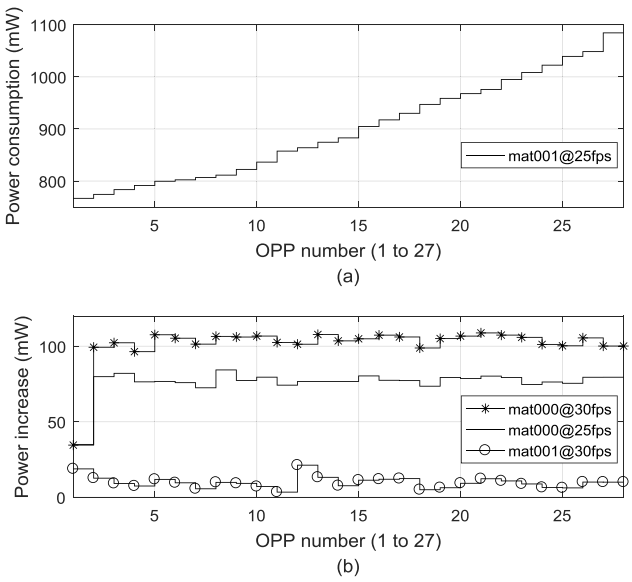

Fig. 4. Average decoder power consumption in each OPP. (a) Default case. (b) Increase (disturbance) with respect to the default case.

considering the decoding of "mat001" clip at 25 frames/s as the default case, its corresponding average power consumption for each OPP is represented in Fig. 4(a), whereas the average power increase in the rest of cases with respect to the default one is represented in Fig. 4(b). From the control-system point of view, these power increases are considered as disturbances over the default situation. Therefore, as introduced above, a test video-sequence has been constructed by alternating a couple of video clips every 50 s in this way: first, the default-case one and, second, a disturbance-case one, as represented in Fig. 5. In order to modulate the amount of disturbance applied to the system, a duty-cycle parameter has been defined as the percentage of time during which the system is being disturbed. Thus, as represented in Fig. 5 with shaded intervals, disturbance duty cycles of 30%, 50% and 70% have been considered.

It is worth saying that the video decoder takes the encoded frames from the test video sequence stored in the SD card of the board and that the decoded frames are simply dropped, i.e., no display task is implemented because the target application is the video decoder. Besides, some performance measurements have been taken which confirm that the video decoder is able to work in real time in all cases, except when trying to decode the "mat000" video clip in OPP1. For the purpose of this work, these are considered valid conditions for the experiments, and, anyway, the system could be further enhanced by adding some performance controller, like the one proposed in [20].

V. Simulation Test

As mentioned above, previous to the real implementation and test of the proposed system, some simulation experiments have been carried out to confirm the validity of the general system design presented in Section III for the particular details of the target platform presented in Section IV.

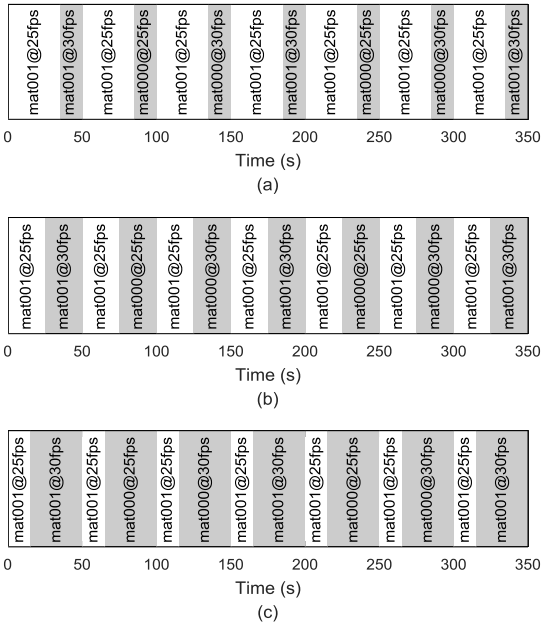

Fig. 5. Test video-sequence time pattern with different disturbance duty cycles. (a) 30%. (b) 50%. (c) 70%.

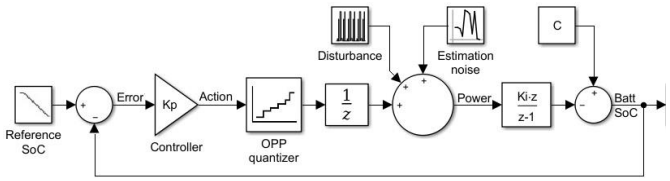

Fig. 6. Simulation block diagram.

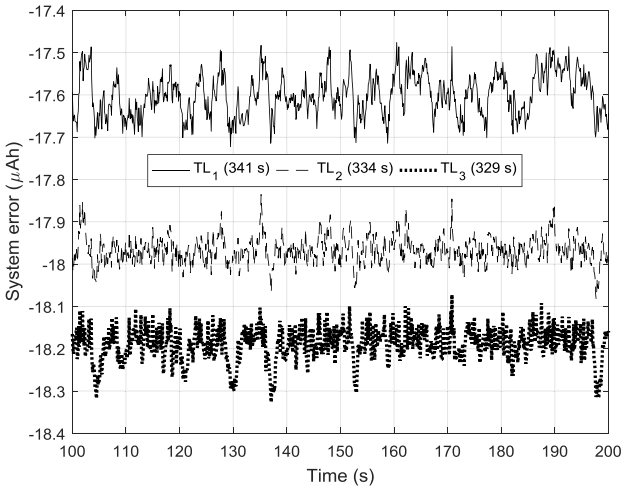

Fig. 7. Excerpt of system error in simulation, without disturbances, for the three considered target lifetimes.

A. Simulation Model

A commercial dynamic-system simulator has been used, such that it enables the graphic development of block diagrams and the mathematical definition of each block. Hence, the general block diagram of Fig. 3, with the particular detail of Fig. 2, has been defined into the simulator, along with the design details explained in Section III and the implementation details described in Section IV. Thus, for example, the controller is defined, from (13), as a gain $K_P = -4.50 \cdot 10^4$ mW/mAh, whereas the integration factor is defined, from (4), as $K_i = 5.56 \cdot 10^{-6}$ h/V. Besides, it is assumed that $C = 15$ mAh for all the experiments carried out, a small value to shorten them. Moreover, as it can be seen in Fig. 6, three new blocks have been added to the simulation model:

1) *OPP-quantizer block:* The output of the controller represents the power, in mW, that should correspond to the needed OPP. However, since the relationship between power and OPP is not linear [see Fig. 4(a)], a quantization function is inserted in order to simulate the 27-level DVFS interface.
2) *Disturbance block:* It allows to inject into the power-consumption signal the variations implied by the different video decoding conditions, according to Fig. 4(b) and depending on the time patterns defined in Fig. 5.
3) *Estimation-noise block:* Since the simulator works based on the average values represented in Fig. 4, this block injects into the power-consumption signal noisy variations similar to those that, at the sample-rate basis, can be observed in a real system, either with power measurements or with PMC-based power estimations, as in this case. The noise has been defined as a normally-distributed random signal with null mean and 15 mW of variance.

B. Simulation Results

Some simulations have been done in order to test the system under different circumstances, such as various target lifetimes, with and without disturbances. The results achieved confirm, apart from the necessary stability of the system, that the disturbances have little effect on the system time response, which proves the robustness of the system. This is true as long as the system is not saturated during long (unrecoverable) periods of time, because, obviously, there are some physical limits that cannot be exceeded. For example, for the proposed value of C, i.e., 15 mAh, the system could not guarantee average battery lifetimes longer than 352 s, which can be obtained from (2) for a constant power consumption of 767 mW [the minimum from Fig. 4(a)] and with a constant voltage supply of 5 V. Any attempt to use the system to prolong that battery lifetime, or even shorter as long as power disturbances appear, will fail due to system saturation.

For the reasons that will be indicated in next section, three specific values of TL have been considered for the experiments, i.e., $TL_1 = 341$ s, $TL_2 = 334$ s and $TL_3 = 329$ s, all of them shorter than the aforementioned upper bound of 352 s. For these values of TL and using (10), the theoretical steady-state error can be calculated as $e_{ssrP1} = -17.6$ μAh, $e_{ssrP2} = -18.0$ μAh and $e_{ssrP3} = -18.2$ μAh, which are considered low enough values. These calculations seem to match well the average simulation results, without disturbances, as it can be seen in Fig. 7 from $t = 100$ s to $t = 200$ s as an example time interval.

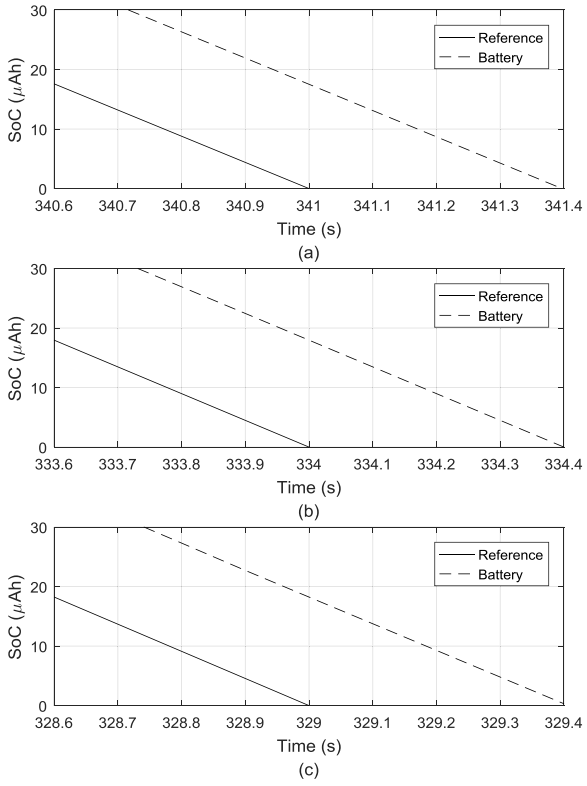

Fig. 8. Lifetime detail in simulation, without disturbances. (a) TL_1. (b) TL_2. (c) TL_3.

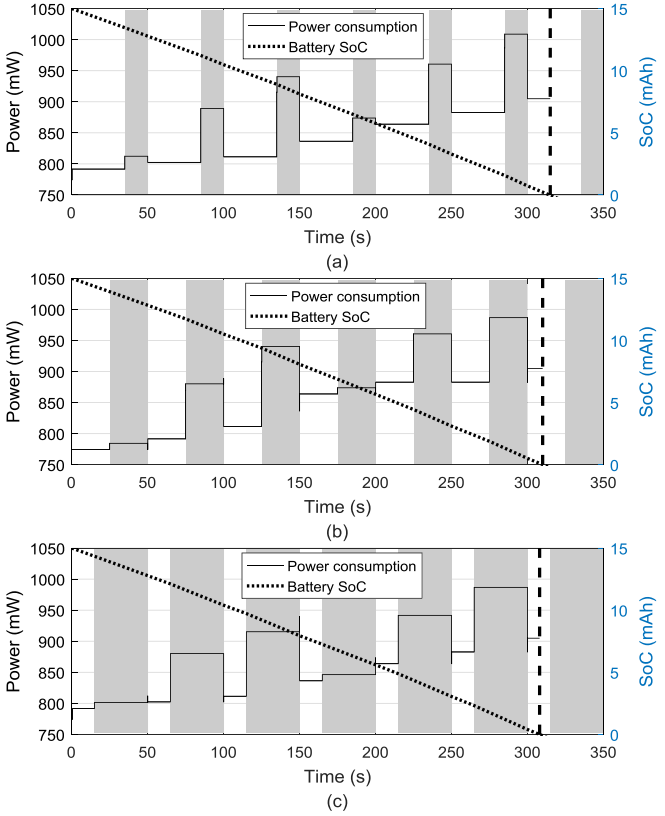

Fig. 9. Power consumption, and battery SoC and lifetime estimated for the Linux *Conservative* governor during the decoding of the test video sequence with different disturbance duty cycles. (a) 30%. (b) 50%. (c) 70%.

As explained above, due to the system error, the corresponding achieved lifetimes are slightly longer than the target ones. In particular, from (11), they have the following theoretical values: $AL_1 = 341.4$ s, $AL_2 = 334.4$ s and $AL_3 = 329.4$ s, as confirmed by the non-disturbed simulation results detailed in Fig. 8. It is worth noting that, although from the short period of time represented in Fig. 8 the battery SoC should appear with discrete steps every 100 ms, a graphic plot command has been used such that the discrete values are joined with a continuous dashed line. As expected, AL is always 0.4 s longer than TL, which leads to relative differences of about only 0.12% for the three considered TL values. This is considered a good enough result as to avoid the need for further research in the application of other more complex controllers that would increase the system overhead.

The simulation tests show, then, promising results, which are confirmed in real experiments, as follows, even in comparison with other related approaches.

VI. Empirical Results

The proposed system has been implemented in a commercial development platform, as outlined in Section IV, with the aim of confirming the preliminary good results obtained from the simulation tests and assessing the relative advantages of the proposal. Thus, when trying to choose realistic values of TL for the tests, the battery lifetimes reached with some existing energy-saving DVFS-based algorithms have been taken as such. In this way, it is supposed that if the proposed system is able to achieve such TL values, it proves not only the validity of the system to regulate the battery discharge but also its capacity to reach battery lifetimes in the range achieved by energy-saving algorithms, and with a clear advantage of the former over the latter: the proposed system is deterministic, in the sense that users can know in advance which the foreseeable battery lifetime is going to be, whereas this cannot be known with conventional energy-saving algorithms.

Two existing energy-saving algorithms have then been considered to fix the aforementioned TL values. They are two well-known dynamic *cpufreq* Linux governors, *Conservative* and *Ondemand*, already addressed as related work in Section II. What has been done before testing the proposed system, even in simulation, is to carry out six previous experiments consisting of running the decoding of the three video sequences represented in Fig. 5, first with the *Conservative* governor active and then with the *Ondemand* one. The results of these six experiments can be explained with Fig. 9 and Fig. 10. To obtain those figures, the OPP set by the Linux governors has been monitored, and from it and its associated average consumption data (see Fig. 4), the power consumption curves have been depicted. Then, the SoC curves have been drawn from the power consumption by applying (2). In those figures, as in Fig. 5 and other subsequent ones, the disturbance intervals are highlighted by shaded areas. By analyzing Fig. 9 and Fig. 10, it can be identified a more aggressive behavior of the *Ondemand* governor, as

TABLE I
BATTERY LIFETIMES ACHIEVED BY LINUX GOVERNORS

Disturbance duty cycle	*Conservative* governor	*Ondemand* governor
(a) 30%	315 s	341 s (TL_1)
(b) 50%	310 s	334 s (TL_2)
(c) 70%	308 s	329 s (TL_3)

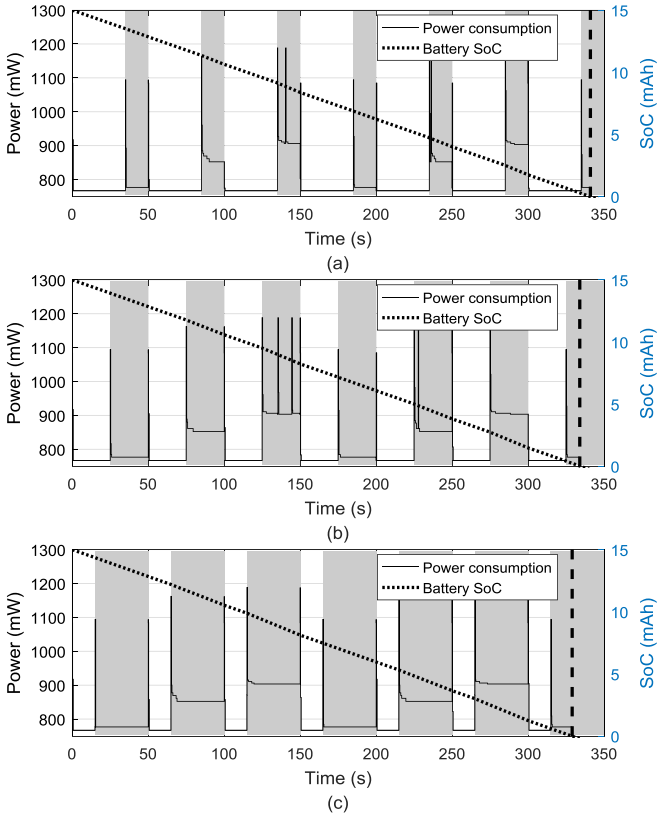

Fig. 10. Power consumption, and battery SoC and lifetime estimated for the Linux *Ondemand* governor during the decoding of the test video sequence with different disturbance duty cycles. (a) 30%. (b) 50%. (c) 70%.

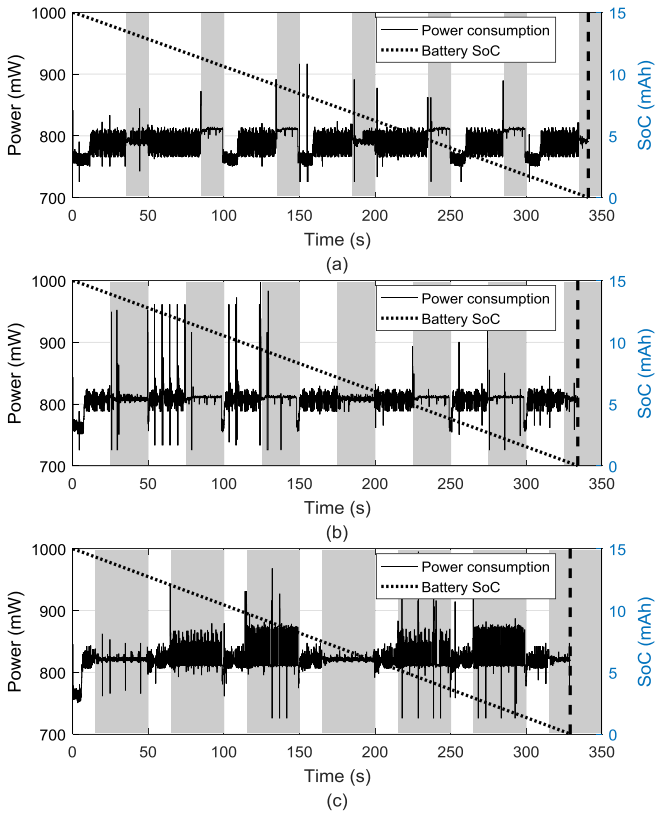

Fig. 11. Power consumption, battery SoC and lifetime achieved with the proposed system during the decoding of the test video sequence with different disturbance duty cycles and different values of TL. (a) 30% and TL_1. (b) 50% and TL_2. (c) 70% and TL_3.

already advanced in Section II. Indeed, it shows a profile more responsive to the disturbance evolution, which, according to its workload-based energy-saving nature, leads to the minimum power consumption during non-disturbance intervals. However, the more progressive behavior of the *Conservative* governor leads to higher mean values of power consumption and, then, shorter battery lifetimes, which are marked in Fig. 9 and Fig. 10, as well as in other subsequent figures, with a thick and dashed vertical line. Table I summarizes the battery lifetimes achieved in each of the six experiments represented in Fig. 9 and Fig. 10, which, obviously, decrease as the disturbance duty cycle increases. As it can be deduced, the better lifetimes, obtained by the *Ondemand* governor, have been the ones taken as TL values for testing the proposed control system, i.e., TL_1, TL_2 and TL_3, as already introduced in Section V.

The use of those three TL values to test the proposed control system, implemented in the development board outlined in Section IV, during the decoding of the corresponding three video sequences depicted in Fig. 5, yields the results represented in Fig. 11, with the same graphic criteria used in Fig. 9 and Fig. 10. From Fig. 11 it can be realized that the target lifetimes are achieved, with an imperceptible error by excess of less than half a second, as stated above by theoretical and simulation results, which results again in the validation of the proposed P controller. Furthermore, it can also be realized that, since the aim of the proposed system is not to save as much energy as possible, the power consumption does not tend to be set to the minimum value, but to be close to the average value that leads to the corresponding target lifetime. For this reason, the mean value of the power consumption can be identified to be slightly higher in Fig. 11 as the TL value decreases. The control system manages itself to counteract the effect of disturbances and then achieve the expected battery lifetime. In comparison with Fig. 9 and Fig. 10, it is worth noting that the curves of Fig. 11 are drown directly from the control-system data, with a sample period of 100 ms, which results in a much more noisy appearance.

Once proven that the proposed system is able to deterministically guarantee battery lifetimes as long as those achieved with the battery-saving-oriented *Ondemand* governor and, therefore, even longer than the ones achieved with the *Conservative* governor, some more ambitious experiments have been carried out in order to check the system behavior with TL values beyond the ones taken from Table I. In fact,

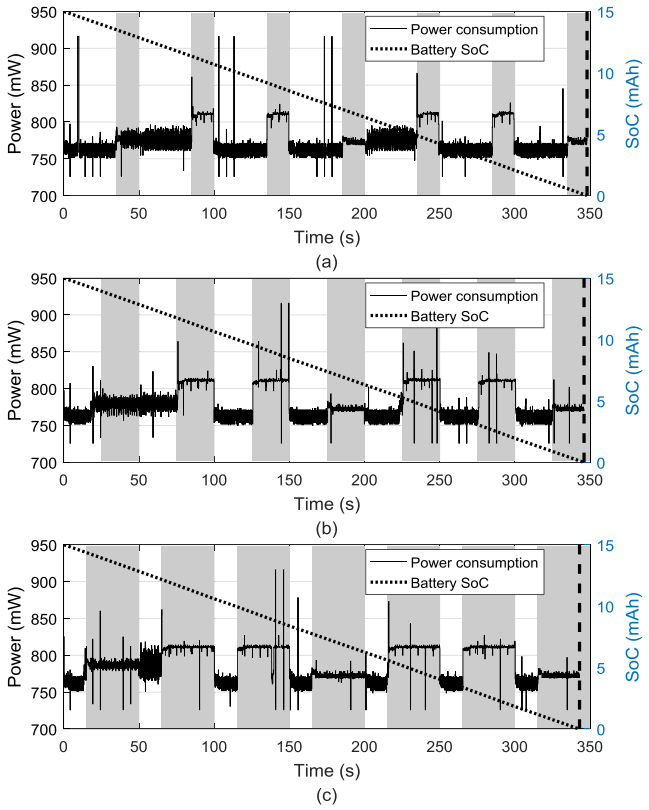

Fig. 12. Power consumption, battery SoC and lifetime achieved with the proposed system during the decoding of the test video sequence with different disturbance duty cycles and different values of TL. (a) 30% and TL = 348 s. (b) 50% and TL = 346 s. (c) 70% and TL = 343 s.

for each of the three video sequences depicted in Fig. 5, a TL value slightly shorter than the theoretical upper bound has been considered. This upper bound is iteratively estimated from (2) by taking into account the minimum average power consumption and disturbance-due increases represented in Fig. 4, as well as the duration of each of the involved white and shaded intervals represented in Fig. 5. Thus, the new selected values of TL for each of the three video-decoding experiments depicted in Fig. 5 are 348 s (a), 346 s (b) and 343 s (c). Note how these new TL values are shorter than the aforementioned absolute limit of 352 s (without disturbances) but longer than the ones achieved with the *Ondemand* governor (see Table I), as well as orderly shorter as the disturbance duty cycle increases from (a) to (c). The results obtained with these new target lifetimes are represented in Fig. 12, which proves the system capability for achieving them. What happens now is that, since the system is working near to its permanent saturation, there is more difference between the power consumption during disturbance intervals and the power consumption during non-disturbance intervals, which is around the minimum possible in the three cases of Fig. 12. Indeed, the saturation occurring in these experiments has the effect of reducing the AL values with respect to a non-saturated situation, but in such a small amount that the reduction simply compensates, approximately, the intrinsic excess error of the system, so that the AL values are now even closer to the TL ones.

TABLE II
RELATIVE LIFETIME EXTENSION OVER LINUX GOVERNORS

Disturbance duty cycle	*Conservative* governor	*Ondemand* governor
(a) 30%	10.5%	2.1%
(b) 50%	11.6%	3.6%
(c) 70%	11.4%	4.3%

Therefore, although the main goal of the proposed system is to allow battery-operated multimedia device users to have a deterministic control of the battery lifetime, the experiments carried out confirm not only the validity of the system for that goal but also its capability to extend the lifetime even beyond other common algorithms do. In particular, Table II shows the relative lifetime extension achieved with respect to Linux *Conservative* and *Ondemand* governors.

VII. CONCLUSION

This paper has presented the design, implementation and test of a closed-loop control system that allows mobile-video consumers to deterministically guarantee a desired battery lifetime in order to enable the complete decoding of a certain video content. From the user target lifetime, the system regulates the battery discharge rate in order to meet the user expectations regardless of the real-time dynamic power variations of the video decoding task. It acts on the decoder power consumption through DVFS techniques and has been implemented in a low-cost commercial development board on which the power consumption and battery SoC are estimated because no sensors are available. The system has been validated by both simulation and real tests, being the achieved lifetimes only 0.4 s longer than the target ones in the worst case, i.e., about a 0.12% of TL. Besides, the system has been able to deterministically ensure the same lifetimes as the ones achieved with a couple of Linux-based governors. Furthermore, it has even reached lifetimes up to around 11% and 3% longer than those two governors, respectively. Moreover, these satisfactory results have been obtained by means of a proportional controller, whose simplicity minimizes the system overhead, and since the control system is implemented at OS level, the user-level video applications have not to be aware of it.

REFERENCES

[1] *Ericsson Mobility Report*, Ericsson, Stockholm, Sweden, 2018. [Online]. Available: https://www.ericsson.com/assets/local/mobility-report/documents/2018/ericsson-mobility-report-june-2018.pdf
[2] T. Coughlin, "A Moore's law for mobile energy: Improving upon conventional batteries and energy sources for mobile devices," *IEEE Consum. Electron. Mag.*, vol. 4, no. 1, pp. 74–82, Jan. 2015.
[3] N. S. More and R. B. Ingle, "Challenges in green computing for energy saving techniques," in *Proc. Int. Conf. Emerg. Trends Innov. ICT*, Pune, India, 2017, pp. 73–76.
[4] M. Shojafar, N. Cordeschi, D. Amendola, and E. Baccarelli, "Energy-saving adaptive computing and traffic engineering for real-time-service data centers," in *Proc. IEEE Int. Conf. Commun. Workshop*, London, U.K., 2015, pp. 1800–1806.
[5] R. Liang, Y. Zhong, and Q. Xia, "Energy-saved data transfer model for mobile devices in cloudlet computing environment," in *Proc. IEEE Int. Conf. Cloud Comput. Big Data Anal.*, Chengdu, China, 2018, pp. 271–274.

[6] J. Wei, E. Juarez, M. J. Garrido, and F. Pescador, "Maximizing the user experience with energy-based fair sharing in battery limited mobile systems," *IEEE Trans. Consum. Electron.*, vol. 59, no. 3, pp. 690–698, Aug. 2013.
[7] J. Wei, R. Ren, E. Juarez, and F. Pescador, "A Linux implementation of the energy-based fair queuing scheduling algorithm for battery-limited mobile systems," *IEEE Trans. Consum. Electron.*, vol. 60, no. 2, pp. 267–275, May 2014.
[8] H. Jeong, J. Yang, and M. Song, "Video quality adaptation to limit energy usage in mobile systems," *IEEE Trans. Consum. Electron.*, vol. 62, no. 3, pp. 301–309, Aug. 2016.
[9] N. Sidaty *et al.*, "Reducing computational complexity in HEVC decoder for mobile energy saving," in *Proc. Eur. Signal Process. Conf.*, Kos, Greece, 2017, pp. 1026–1030.
[10] Y. Benmoussa, E. Senn, N. Derouineau, N. Tizon, and J. Boukhobza, "Joint DVFS and parallelism for energy efficient and low latency software video decoding," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 4, pp. 858–872, Apr. 2018.
[11] D. Brodowski, N. Golde, R. J. Wysocki, and V. Kumar. *Linux CPUFreq Governors*. Accessed: Sep. 3, 2018. [Online]. Available: https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt
[12] V. Pallipadi and A. Starikovskiy, "The ondemand governor," in *Proc. Linux Symp.*, vol. 2. Ottawa, ON, Canada, 2006, pp. 215–230.
[13] D. Le and H. Wang, "An effective feedback-driven approach for energy saving in battery powered system," in *Proc. Int. Workshop Qual. Service*, Beijing, China, 2010, pp. 1–9.
[14] X. Wang, X. Fu, X. Liu, and Z. Gu, "PAUC: Power-aware utilization control in distributed real-time systems," *IEEE Trans. Ind. Informat.*, vol. 6, no. 3, pp. 302–315, Aug. 2010.
[15] A. K. Mishra, S. Srikantaiah, M. Kandemir, and C. R. Das, "CPM in CMPs: Coordinated power management in chip-multiprocessors," in *Proc. Int. Conf. High Perform. Comput. Netw. Storage Anal.*, New Orleans, LA, USA, 2010, pp. 1–12.
[16] S. Garg, D. Marculescu, and R. Marculescu, "Custom feedback control: Enabling truly scalable on-chip power management for MPSoCs," in *Proc. Int. Symp. Low Power Electron. Design*, Austin, TX, USA, 2010, pp. 425–430.
[17] Q. Tang, A. M. Groba, E. Juárez, C. Sanz, and F. Pescador, "Real-time power-consumption control system for multimedia mobile devices," *IEEE Trans. Consum. Electron.*, vol. 62, no. 4, pp. 362–370, Nov. 2016.
[18] R. Ren *et al.*, "A PMC-driven methodology for energy estimation in RVC-CAL video codec specifications," *Signal Process. Image Commun.*, vol. 28, no. 10, pp. 1303–1314, Nov. 2013.
[19] Q. Tang, A. M. Groba, E. Juárez, and C. Sanz, "Closed-loop power-control governor for multimedia mobile devices," *IEEE Trans. Consum. Electron.*, vol. 63, no. 2, pp. 153–161, May 2017.
[20] A. M. Groba, P. J. Lobo, and M. Chavarrías, "Slack-time closed-loop control system for multimedia mobile devices," *IEEE Trans. Consum. Electron.*, vol. 64, no. 2, pp. 162–170, May 2018.
[21] J. Cho, Y. Woo, S. Kim, and E. Seo, "A battery lifetime guarantee scheme for selective applications in smart mobile devices," *IEEE Trans. Consum. Electron.*, vol. 60, no. 1, pp. 155–163, Feb. 2014.
[22] N. Ravi, J. Scott, L. Han, and L. Iftode, "Context-aware battery management for mobile phones," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun*, Hong Kong, 2008, pp. 224–233.
[23] M. Själander, M. Martonosi, and S. Kaxiras, *Power-Efficient Computer Architectures, Recent Advances*. San Rafael, CA, USA: Morgan & Claypool, 2015.
[24] F. Golnaraghi and B. C. Kuo, *Automatic Control Systems*, 9th ed. Hoboken, NJ, USA: Wiley, 2009.
[25] Q. Tang, "Control algorithms for energy optimization in multimedia hand-held devices," Ph.D. dissertation, Dept. Telematics Electron. Eng., UPM, Helsinki, Finland, 2017.
[26] Q. Tang, A. M. Groba, E. Blázquez, and E. Juárez, "OS-level power consumption estimator for multimedia mobile devices," in *Proc. IEEE Int. Symp. Consum. Electron.*, Madrid, Spain, 2015, pp. 1–2.
[27] GitHub ORCC. *Orc-Apps, MPEG4 Part2*. Accessed: Sep. 3, 2018. [Online]. Available: https://github.com/orcc/orc-apps/tree/master/RVC/src/org/sc29/wg11/mpeg4/part2
[28] SourceForge. *Open RVC-CAL Compiler, Sequences*. Accessed: Nov. 18, 2018. [Online]. Available: http://sourceforge.net/projects/orcc/files/Sequences/MPEG4.zip

Ángel M. Groba received the Ph.D. degree in telecommunication engineering from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 2004.

Since 1991, he has been an Associate Professor with the Department of Telematic and Electronic Engineering, UPM. Since 2004, he has been a member of the Electronic and Microelectronic Design Group, integrated with the Research Center on Software Technologies and Multimedia Systems for Sustainability since 2012, where he was the Secretary from 2016 to 2017. His current research interests include control systems and multimedia systems, particularly the application of the former to the energy optimization of the latter.

Pedro J. Lobo received the B.Sc. and M.Sc. degrees in telecommunication engineering from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 1993 and 2004, respectively, where he is currently pursuing the Ph.D. degree.

Since 2001, he has been with the Electronic and Microelectronic Design Group, UPM, where he is currently with the Research Center on Software Technologies and Multimedia Systems for Sustainability. His current research interests include multiprocessor architectures and multiprocessor programming methodologies.

Miguel Chavarrías received the Ph.D. degree in telecommunication engineering from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 2017.

Since 2016, he has been an Assistant Professor with the Department of Telematic and Electronic Engineering, UPM, where he has been with the Electronic and Microelectronic Design Group since 2011, and is currently with the Research Center on Software Technologies and Multimedia Systems for Sustainability. His current research interests include HEVC, power efficiency techniques for multiprocessor SoCs, multiprocessor architectures for video coding, and reconfigurable video coding.