QoE-Aware Dual Control System to Guarantee Battery Lifetime for Mobile Video Applications

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

***Abstract*—The use of consumer-electronics mobile terminals to decode and view digital video is constantly growing. Consumers expect from them a good quality of experience (QoE) during the complete viewing of the video content, which is not always possible due to the battery-operated nature of this type of devices. In this paper, a dual closed-loop control system is proposed to act on the CPU frequency such that, by default, it controls a QoE-related variable to be at its lowest valid level in order to save as much energy as possible. This energy savings are computed as an energy bonus (EB) that can be further used in moments of higher power consumption. Anyway, when the EB decreases below a dynamic threshold, the system switches to an exception mode in which it controls the EB to be enough to guarantee the expected playback lifetime at the expense of lowering QoE. Besides, the system can be configured by users to be more or less aggressive in risking lifetime vs QoE by adjusting the mode-switching threshold. Upon implementation in the operating system of a commercial development board, results show how the system is able to meet the desired lifetime with a reduction of up to 70% of the time with QoE restrictions with respect to a previous lifetime-guaranteeing approach. Furthermore, the worst-case computation-time overhead of the system is only a 4.2%.**

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

I. Introduction

IT IS a fact that the mobile technology is widely spread and in constant growth within the consumer-electronics market. Some data in this regard [1]: around 360 million smartphones were sold in Q3 2018, with a total number of mobile broadband subscriptions about 5.7 billion and being expected to reach 8.4 billion in 2024, 1.5 billion of which will already be 5G. On the other hand, video is currently the most significant traffic type consumed by smartphone users, since it represents the 60% of total traffic, whereas it is expected to grow up to a 74% by the end of 2024 [1]. This growth, enabled by higher-capacity networks, will be due not only to the increase of video-content range, video resolutions or video-format demands, but simply also to the increase in the viewing time spent by mobile users. In turn, this has implications in the time the battery should be supplying energy to the mobile device in order to cover that increasing video-viewing time. Since the increase in battery demands usually exceeds the increase in battery capacity, mostly in this constrained field of handheld devices [2], the power-consumption optimization is an important workhorse. When trying to optimize the power consumption of multimedia applications, a third-party issue that may be indirectly affected arises: the quality of experience (QoE). Indeed, reductions on power consumption frequently tend to worsen QoE and the other way round, improvements in QoE usually tend to increase power consumption. Therefore, a tradeoff has to be observed between those two variables. Moreover, a flexibility degree could be applied to that tradeoff because, depending on the type and/or situation of the user, he/she may prefer to extend the battery lifetime regardless of a likely worse QoE or to keep a good QoE regardless of the energy consumption.

In this context, this paper presents a control system to deal with a flexible tradeoff between QoE and battery lifetime in mobile multimedia applications. Particularly, a dual-output closed-loop architecture is proposed such that both a QoE-related output and an energy-related output are monitored and used as feedback signals for the control system during video decoding. According to an algorithm that will be explained below and depending on those feedback signals, the control system calculates an action signal which is able to affect both outputs of the control system, i.e., the QoE and the dynamic power consumption of the video decoder. This is achieved through the operating performance point (OPP) of the dynamic voltage and frequency scaling (DVFS) mechanism of the video processor. Indeed, a higher OPP, i.e., a pair of higher voltage and frequency values for the processor, increases its available resources to improve QoE but also increases its power consumption, whereas a lower OPP implies the inverse effects [3]. And here the tradeoff issue appears: which is the suitable OPP to be set by the control algorithm at each instant during video decoding in order to meet the user expectations in terms of battery lifetime and QoE? These user expectations to be met give rise to the two inputs of the control system. Both of them, as well as the rest of aforementioned variables, can be seen in Fig. 1, which shows a conceptual block diagram, above implementation details, of the proposed system.

In Fig. 1, two additional variables are shown. One of them is the disturbance representing the unpredictable variations in the video-decoding activity which, depending on issues like the video-decoding complexity or performance parameters, may

Manuscript received April 4, 2019; revised June 21, 2019; accepted July 29, 2019. Date of publication August 2, 2019; date of current version October 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.2019.2932778

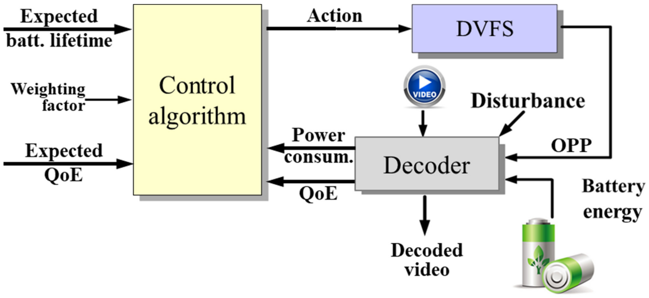

Fig. 1. Conceptual block diagram of the proposed system.

imply variations in the real-time power consumption or QoE for a fixed OPP. The existence of this disturbance itself is what mainly justifies the closed-loop topology proposed for the system, because a closed-loop system can be designed to adapt itself to and counteract unexpected variations in the controlled variables.

The other additional variable in the block diagram of Fig. 1 is a weighting factor which allows users to quantify the relative importance of the two opposed control-system inputs, i.e., to what extent the user is willing to sacrifice QoE in order to approach the expected battery lifetime or the other way round, to what extent the user is willing to sacrifice battery lifetime in order to enjoy a good QoE. This weighting factor is just the way in which the proposed system can manage the aforementioned flexibility of the tradeoff between QoE and battery lifetime.

The proposed dual-input dual-output (DIDO) system is wanted to be as generally applicable as possible, based on the operating-system (OS) support and without needing any special feature from the multimedia applications. With this aim, any information related to any of the two system outputs that is available from the OS can be used. Thus, for example, with respect to power consumption, an indirect estimation algorithm based on general-purpose processor performance monitoring counters (PMC) [4] is proposed as a general solution for those restricted cases in which there is not direct power-consumption information available. With respect to the other system output, the QoE, the proposal is to take advantage of the time-measurement possibilities of the OS to focus on the QoE component that has to do with the video temporal resolution, for example through the capability of the decoder to cope with the required frame rate. For this purpose, one option is to monitor the decoder slack time (ST) [5], i.e., the idle time of the decoder during its real-time processing. Calculated at a frame-basis rate as the difference between the absolute deadline of the frame and the moment in which the frame decoding is completed, positive values of ST, i.e., deadlines met, enable the expected real-time performance of the decoder, which contributes to keep a good QoE, whereas negative values of ST, i.e., deadlines missed, threaten the real-time performance of the decoder, which is a probable cause of QoE deterioration that would have to be mitigated by applying buffering techniques with their corresponding latency effects.

The structure of this paper is the following. After the introduction in Section I, some related work is summarized in Section II; Section III presents the method followed to achieve the aim; Section IV develops the mathematical model of the system which enables the controller design; Section V describes the test bench to implement and test the system; Section VI fixes the test conditions and show the test results, which are compared to other related approaches in Section VII; finally Section VIII concludes the paper.

II. Related Works

As already introduced above, the work presented in this paper deals mainly with a couple of issues very common in research areas related to battery-operated mobile multimedia devices: the energy consumption and the user's QoE. As such, a lot of related work results are published in the literature. However, as the application field is being restricted to the one of interest for this paper, the amount of related work is obviously reduced.

Regarding to energy-consumption optimization, it is a challenge that nowadays applies to every facet of computation [6], ranging from large systems [7] to small mobile ones [8]. More related to video decoding, one possibility is to tackle the challenge from an application-dependent lower level [9], [10], whereas other possibility is to act from an OS-based higher level, as in the case of this paper or others like the ones developed by Wei *et al.* [11] or Tang *et al.* [12]. Also as in this paper but with different approaches, references can be found that address the management of the battery lifetime. Thus, Cho *et al.* [13] proposed a task-scheduling scheme to guarantee battery lifetime for specific applications in mobile devices; Ravi *et al.* [14] proposed a monitoring algorithm able to estimate the remaining battery lifetime from previous consumption patterns; and Groba *et al.* [15] proposed a closed-loop system to guarantee battery lifetime in mobile video applications but without taking into account QoE issues. Besides, in this last reference, the power consumption was already estimated by means of a PMC-based algorithm, which, in turn, was adapted by Tang [16] from the initial version of Ren *et al.* [4].

With respect to QoE-related research, it can be addressed from different points of view, from those more general attending to the mobile-device responsiveness to other more application dependent, like in the case of video applications, and even including energy-optimization objectives. Dealing with this type of always-connected terminals, much effort is put in general into the optimization of the data transmission [17], [18], whereas in the particular field of mobile video, lot of work focuses also on scalable video coding (SVC) [19], [20]. A remarkable case within the latter set is the work of Jeong *et al.* [21], which proposed to decide the SVC layers to be decoded, as well as the DVFS OPP, depending on the remaining battery energy, such that a minimum user-defined QoE was guaranteed. The proposed algorithm forecasted the power consumption of next frames by observing the power consumption of previous ones. Although the work has philosophical similarities with the proposal in this paper, it is a kind of best-effort approach which simply tries to optimize the (low) remaining battery but without guaranteeing any user-defined lifetime. A similar best-effort approach to extend the video playback time for the remaining battery energy was proposed by Jo *et al.* [22] but from the network point of view, with an HTTP-based adaptive rate control streaming scheme.

Apart from SVC-based methods or specific QoE metrics such as the ones adopted by Jeong *et al.* [21] or Jo *et al.* [22], other more general indicators can also be used to get objective data about QoE. As proposed in this paper, one possibility is to take into account the ST of the video decoder, in the sense that if the ST is positive the decoder is able to work in real time, i.e., each frame is decoded within its deadline. In fact, this is a well-known concept in the real-time systems field [3], [23], [24]. Under this philosophy, Groba *et al.* [5] proposed a DVFS-based control system which, regulating the ST of video decoders, achieved energy savings in order to extend as much as possible the playback time in mobile devices. Other example of ST-based video-decoding energy-saving approach is the one proposed by Pal *et al.* [25], although in this case the ST was used to decide if any of the cores in multicore platforms could be switched off.

Other possible QoE indicator, which, in turn, is related to ST, is the system workload or utilization factor. Indeed, this factor is the percentage of time during which the processor is not idle during the execution of a task, which, for a video-decoding one, has much to do with the decoder ST. Apart from the also classic use of this concept in real-time systems research [23], [26], [27], some general-purpose OSs use workload data to tune the DVFS status. One common case is Linux with its *cpufreq* governors [28]. Among them, two dynamic ones like *Conservative* and *Ondemand* [29] try to adapt the OPP to the CPU workload such that the utilization factor is kept below 100% and the power consumption is kept low. In the former, the OPP tends to be conserved while the workload can be dispatched in time, whereas the latter is more aggressive, in the sense that it tends to set always the minimum OPP that allows the processor to satisfy its demand (QoE), thus achieving normally more energy savings than in the former case.

From all this related work, the results of this paper will be compared in Section VII to the ones obtained from references [5], [15], and [29].

III. Method

The idea is to provide the user with the possibility of reserving an amount of the available battery charge (which could be the complete remaining one or not) to last for a target period of time because he/she wants to ensure a video-content viewing during that time. Hereinafter, the amount of battery charge to be put into play will be referred to as C, in units of mAh, whereas the target lifetime for C will be referred to as TL, in units of hours. Besides, the QoE-awareness is proposed to be managed by controlling the decoder ST in closed loop. In fact, if ST, although positive, is kept low enough, then the dynamic power consumption is reduced while the decoding deadlines are met [5]. The problem that may arise and that motivates the utility of the proposed DIDO system is what happens if, by itself, the best-effort performance of the ST control with respect to power consumption is not enough to reach TL, which depends, among other things, on the unpredictable real-time values of the disturbance (see Fig. 1). Here again, the user should be provided with the possibility of deciding what to do, i.e., whether to sacrifice lifetime in favor of QoE or vice versa, as introduced above. For this purpose, the proposal considers a flexible weighting factor (see Fig. 1), α, which can be assigned any intermediate value between 0 and 1 such that if $\alpha = 0$ maximum priority is given to lifetime, whereas if $\alpha = 1$ maximum priority is given to QoE.

The proposal is to manage the DIDO control system introduced in Section I as a single-input single-output (SISO) one which, by default, controls the decoder ST to keep track of the set point. In that way, it is supposed that, during some time intervals, the power consumption could be lower than the average one needed to reach TL, which would give rise to a kind of energy bonus (EB) that could be used to compensate the higher power consumption that might appear during other subsequent time intervals due to disturbance variations. Therefore, while the SISO system is controlling ST, the EB has to be calculated and monitored in real time as indicated in (1), where $\Delta P(\tau)$ is the difference between the average power consumption needed to reach TL and the actual power consumption $P(t)$ at a time $t = \tau$. For the sake of simplicity, EB(t) is proposed to be managed in terms of battery charge, in units of mAh, whereas $P(t)$ is assimilated to the current drained from the battery, in units of mA. Besides, EB(t) is assumed to be 0 at the beginning of the control process ($t = 0$).

$$\mathrm{EB}(t) = \int_0^t \Delta P(\tau) d\tau = \int_0^t \left(\frac{C}{\mathrm{TL}} - P(\tau) \right) d\tau \text{ mAh}. \quad (1)$$

Furthermore, if at any instant EB(t) decreases below a predefined bonus threshold (B_{th}) under which it is considered that TL could not be guaranteed, the DIDO system switches to an exception status in which another SISO system replaces the default one such that it takes control of EB(t) in order to avoid the premature battery exhaustion, whereas ST becomes a monitored variable instead of the controlled variable. In this exception status, the decoder ST is likely to be shorter than its set point (it could even be negative, thus worsening the QoE) because the exceptional EB control system will restrict the resources available for the decoder, i.e., it will set a lower OPP in order to revert the battery discharge trend. In order to restrict the QoE as less as possible, the proposal is to control the EB such that it just follows the bonus threshold value, i.e., B_{th} will be the reference signal of the EB SISO control system. If at any moment during the exception interval EB(t) is not below B_{th} and the monitored ST is not shorter than its set point, i.e., its control will not need a higher OPP, the control system can go back again to its default (ST-based) status.

Finally, some details about B_{th} have to be given. First, it is worth noting that the room for maneuver is greater at $t = 0$ than when t tends to TL, i.e., as t approaches TL the probability that a negative EB can still be recovered in next more favorable periods is reduced. For this reason, B_{th} should be a not decreasing function of time, $B_{th}(t)$. Secondly, it also has to be noted that if EB(TL)>0 then this means that there is still energy left which perhaps could have been used to avoid (worse-QoE) exception intervals, whereas if EB(TL)<0 then

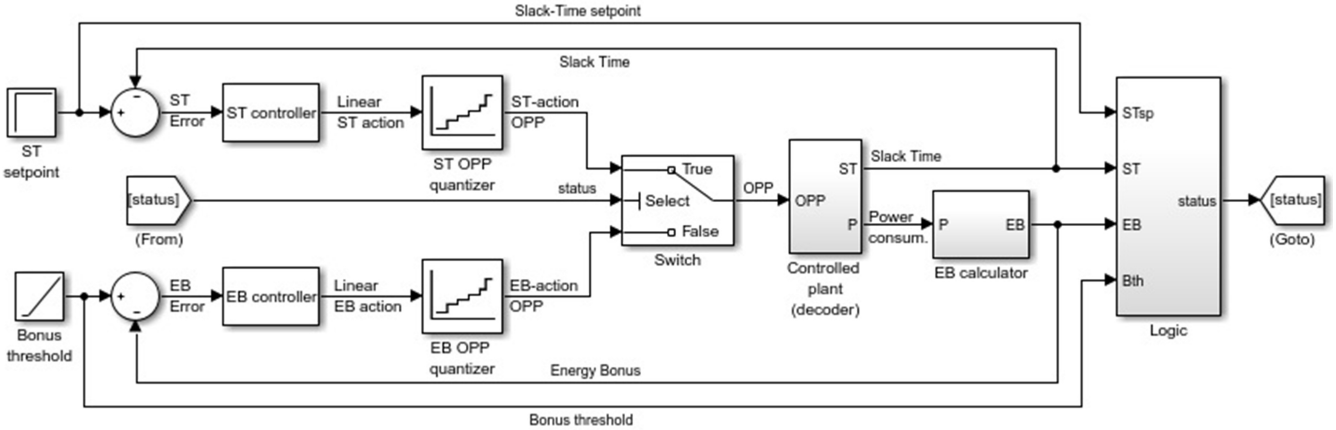

Fig. 2. Block diagram of the proposed method: a DIDO system based on two multiplexed SISO control systems.

this means that C has been exhausted before TL. According to this, $B_{th}(t)$ is proposed to be less than or equal to 0 for t<TL and strictly equal to 0 at t = TL. And thirdly, since QoE tends to be promoted while EB(t)>$B_{th}(t)$ (default status) whereas lifetime tends to be promoted while EB(t)<$B_{th}(t)$ (exception status), the weighting factor α is proposed to affect the value of $B_{th}(t)$. With all these premises into account, $B_{th}(t, \alpha)$ is proposed to be a linear function of time, i.e., a ramp, as expressed in (2), where both the slope s and the offset o are functions of α. Obviously, $B_{th}(t, \alpha)$ has to be managed in the same units as EB(t), i.e., mAh, which leads to that same units for $o(\alpha)$ and units of mA for $s(\alpha)$ if t is measured in hours:

$$B_{th}(t, \alpha) = s(\alpha) \cdot t + o(\alpha)\text{mAh}. \tag{2}$$

For the particular case of $\alpha = 0$, the promotion of battery lifetime over QoE is proposed to be achieved by rising $B_{th}(t, 0)$ to a horizontal line of null value, i.e., $s(0) = o(0) = 0$, which would represent the null EB associated to the average power consumption needed to reach TL. On the other hand, when $\alpha = 1$, the promotion of QoE over battery lifetime is proposed to be achieved by lowering $B_{th}(t, 1)$ to a minimum EB line which would allow to reach TL only under a best-case (low power) condition corresponding to the lowest OPP and disturbance values. Hence, naming P_0 the power consumption (assimilated to current in mA) under that best-case condition, the proposed expressions for the slope and offset of $B_{th}(t, 1)$ are $s(1) = C/\text{TL} - P_0$ and $o(1) = P_0\text{TL} - C$, respectively. Any other intermediate value of α between 0 and 1 will also lead to intermediate values of $s(\alpha)$ and $o(\alpha)$, with the following general expressions:

$$\begin{cases} s(\alpha) = \alpha(C/\text{TL} - P_0) \text{ mA} \\ o(\alpha) = \alpha(P_0\text{TL} - C) \text{ mAh.} \end{cases} \tag{3}$$

From (3), it is clear that an initial obvious premise must be met for the proposal to make sense, which is that the user has to choose a couple of values for C and TL such that C/TL>P_0, i.e., TL has to be shorter than the time at which C would be exhausted at the best-case consumption rate.

A global block diagram which describes graphically the proposed method is represented in Fig. 2. In that figure, both the ST and EB SISO control systems can be identified as being based on linear controllers, whereas a Boolean variable is used to set the default (ST-control) status when it is true and the exception (EB-control) status when it is false. On the other hand, the figures showing test results in Section VI will also help to understand better the system behavior.

IV. Model

As already explained, the DIDO control system of Fig. 1 is proposed to be managed by multiplexing two SISO systems, as depicted in Fig. 2. In order to design the two controllers represented in Fig. 2, a model of each SISO system is needed, and within them, a model of the controlled plant.

A. Controlled Plant

The Z transfer function defined in (4) can be used as the simplified linear model of the plant within the models of both SISO systems if it meets the following premises.

1) It linearly models the part of the system between the output of the controller and the output of the controlled plant, excluding the nonlinear OPP-quantizer effects (see Fig. 2).

2) It mathematically represents the quotient between the Z transforms of the controlled-plant output and the controller output.

3) The sampling period (T=0.1 s) is longer enough than the settling time of the controlled-plant output. This, in turn, lightens the computation overhead of the system.

4) It is dimensionless, i.e., its output units are the same as its input ones.

5) In the absence of disturbances, its output and input values are the same in steady state. This implies that the linear-controller implementation has to be complemented with an OPP quantizer (see Fig. 2) which sets the DVFS OPP that leads to the plant output closest to the linear-controller output value.

$$G(z) = \frac{1}{z}. \tag{4}$$

B. ST Control System

This is the SISO control system that operates in the default status of the DIDO control system. The work presented by

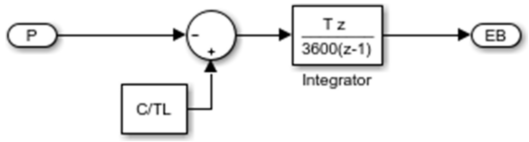

Fig. 3. Block diagram of the EB calculator.

Groba *et al.* [5] is applied here to control the decoder ST and will be further enhanced since it is only a part of the proposed DIDO system. In the ST control system, the ST samples are actually obtained in relative terms with respect to the frame period, i.e., the inverse of the frame rate, in percentage units. On the other hand, a linear discrete-time Tustin-rule integral (TR-I) controller is used such that for a sampling period T=0.1 s and the plant model (4), it yields good results with a gain $K_{TR} = 3.43$ in the following transfer function [5]:

$$F_{ST}(z) = \frac{K_{TR}T(z+1)}{2(z-1)} = \frac{0.172 \cdot (z+1)}{z-1}. \tag{5}$$

Indeed, as justified by Tang *et al.* [30], the TR-I controller leads to a good tradeoff between smoothness and settling time of the system transient time response. Furthermore, its integral nature makes the ST-control system type 1 with respect to the steady-state accuracy thus ensuring a null error for a constant ST set point, regardless of the data used to create the OPP look-up table (LUT) necessary to satisfy premise 5) in Section IV-A. However, it is worth noting that the nonlinear quantization effect of that LUT implies that the controller likely achieves a null average steady-state error by means of a kind of implicit PWM oscillation between 2 adjacent OPPs.

C. EB Control System

This is the SISO control system that operates in the exception status of the DIDO control system. As represented in Fig. 2, in this status, the output variable of interest from the decoder is no longer its ST but its power consumption. In any case, taking into account that the sampling period is common for the whole DIDO system, i.e., T=0.1 s, it also meets premise 3) in Section IV-A for the power output of the plant. Hence, the plant model (4) is also applicable to the EB control system with all the considerations included in Section IV-A.

However, since the controlled variable in this SISO control system is actually the EB, it has to be obtained from the power by following the expression (1). In order to calculate EB within a periodic real-time control system, a discrete-time version of (1) has to be used, like (6) and its corresponding Z transform (7), which define the 'EB calculator' block of Fig. 2 as depicted in Fig. 3.

$$\mathrm{EB}[k] = \mathrm{EB}[k-1] + \left(\frac{C}{\mathrm{TL}} - P[k]\right)\frac{T}{3600}\ \mathrm{mAh}. \tag{6}$$

$$\mathrm{EB}(z) = \frac{Tz}{3600(z-1)}\left[\frac{Cz}{\mathrm{TL}(z-1)} - P(z)\right]\ \mathrm{mAh}. \tag{7}$$

From (7) and Fig. 3, it can be deduced that, apart from a step Z transform (static offset), the dynamic relationship between $P(z)$ and $\mathrm{EB}(z)$ is

$$\frac{\mathrm{EB}(z)}{P(z)} = -\frac{Tz}{3600(z-1)}\ \mathrm{h}. \tag{8}$$

It leads to the following open-loop (OL) dynamic transfer function which, as a linear model, includes (4) without the nonlinear EB OPP quantizer, as stated in premise 1) in Section IV-A, as well as the transfer function of the EB controller, i.e., $F_{EB}(z)$. Naming $E_{EB}(z)$ the Z transform of the error of the EB control system:

$$\mathrm{OL}_{EB}(z) = \frac{\mathrm{EB}(z)}{E_{EB}(z)} = -F_{EB}(z)\frac{T}{3600(z-1)}. \tag{9}$$

From (9), it is clear that in order to determine completely the OL transfer function, $F_{EB}(z)$ has to be designed. In this sense and trying to lead to an EB control system as parallel to its counterpart ST control system as possible, a classic linear controller is also proposed here. In this case, the OL pole in $z = 1$ that can be seen in (9) ensures that the EB control loop is already type 1 with respect to the steady-state accuracy, regardless of $F_{EB}(z)$. Therefore, although in the ST control system an integral controller was necessary to make it type 1, this is not the case for the EB control system. For this reason and in order to keep the system complexity (and computation overhead) as low as possible, a simple proportional (P) controller is proposed with transfer function

$$F_{EB}(z) = K_P \mathrm{h}^{-1}. \tag{10}$$

The simplicity of a P controller, which, as shown by Groba *et al.* [15], is not incompatible with a good behavior, has simplicity advantages not only at implementation or execution but also at design time. Indeed, when dealing with the calculation of K_P, a basic time-response-based design criterion can be used from a twofold view, i.e., the transient state and the steady state. With respect to the former, from the expression of the system pole $z_{p(EB)}$, obtained from the equation $1 + \mathrm{OL}_{EB}(z) = 0$, i.e.,

$$z_{p(EB)} = 1 + \frac{K_P T}{3600}, \tag{11}$$

the value of K_P needed to place the system pole in a satisfactory position on the Z plane can be obtained. With respect to the steady state, it is important to remind that the reference input of the EB control system is $B_{th}(t, \alpha)$ which, in general terms, is a ramp, not a constant. This means that the steady-state error of the EB control system will not be null and has to be considered. In a first stage, that error can be calculated by superposing its dependence on the input reference ramp, i.e., $B_{th}(t, \alpha)$, and on the static offset step applied within the EB calculator, i.e., C/TL (see (7) and Fig. 3). Taking into account the system topology depicted in Fig. 2 and its closed-loop nature, the aforementioned dependence can be expressed as

$$E_{EB}(z) = \frac{B_{th}(z) - \frac{C}{\mathrm{TL}}\frac{Tz^2}{3600(z-1)^2}}{1 - \frac{K_P T}{3600(z-1)}}\ \mathrm{mAh}, \tag{12}$$

where

$$B_{th}(z) = \frac{s(\alpha)Tz}{3600(z-1)^2} + \frac{o(\alpha)z}{z-1}\ \mathrm{mAh}, \tag{13}$$

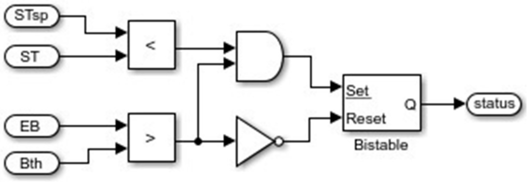

Fig. 4. Scheme of the status decision logic.

because $B_{th}(t, \alpha)$ can be modeled as a pure (without offset) $s(\alpha)$-slope ramp plus an $o(\alpha)$-amplitude step. And in a second stage, the steady-state value of the error ($e_{SS(EB)}$) is obtained from (12), (13) and (3) by applying the Z-transform final-value theorem:

$$e_{SS(EB)} = \lim_{z \to 1}(z-1)E_{EB}(z) = \frac{\frac{C}{\mathrm{TL}}(1-\alpha) + \alpha P_0}{K_P} \text{ mAh.} \quad (14)$$

By observing (11) and (14), it can be realized that while the relationship between K_P and $z_{p(EB)}$ depends only on a system-design parameter (T), the relationship between K_P and $e_{SS(EB)}$ depends, apart from another system parameter (P_0), on user-defined parameters (C, TL, α) which will vary with user needs. For this reason, the design proposal is to use the more general relationship (11) to obtain a value of K_P for a suitable transient time response of the system, which in turn tends to minimize the steady-state error. In this sense, from (14), it is clear that the larger the value of K_P, the smaller the value of $e_{SS(EB)}$. However, from (11), it can be deduced that $|K_P|$ has to be limited in order to prevent the system instability when $|z_{p(EB)}| \geq 1$. Moreover, $z_{p(EB)}$ has to be non-negative in order to avoid undesirable sample-to-sample oscillations in the system time response. Hence, from (11) and for T=0.1 s, the valid range of K_P for $0 \leq z_{p(EB)} < 1$ is $-36000 \leq K_P < 0$ in h^{-1}. From within this range, a value of -27000 h^{-1} is proposed for K_P, i.e., the midpoint of the higher half of the range, because a large value of K_P is preferred in order to reduce $e_{SS(EB)}$. In turn, this value provides also some kind of balance between smoothness and settling time of the system transient time response, in similar terms to those applied by Tang *et al.* [30] to fix the ST controller gain. The corresponding $z_{p(EB)}$ value is 0.25. Finally, it is considered that the value proposed for K_P is large enough as to achieve reasonably small values of $e_{SS(EB)}$ in (14), as it will be shown in Section VI.

D. Status Decision Logic

In the block diagram of Fig. 2, apart from each of the two SISO control systems modeled above, there is also a common logic block in charge of deciding when to switch from one system to the other, i.e., it decides the status of the DIDO system. Its scheme is depicted in Fig. 4 and responds to the decision criteria explained in Section III. The underlined 'Set' input label in the bistable block of Fig. 4 indicates that the default (initial) status value is 'true' in order to select the ST-control system in the switch block of Fig. 2.

V. Test Bench

It is worth highlighting that the model presented in Section IV has been developed with the aim of being as generally applicable as possible, which implies that, once the sampling period T is fixed, the whole system can be implemented, except the OPP quantizers (see Fig. 2), which are not part of the model and depend on the characteristics of the available DVFS support. Thus, these quantizers have been defined according to the DVFS features of the support platform used to implement the system. A description of this platform is given below, as well as some details about the video-decoding task used to test the system. With respect to that test, in order to achieve deterministic and enough representative results, it will follow the set of premises indicated below, which will be applied throughout the rest of the paper.

A. Premises for Practical Tests

1) The decoded video sequence alternates periodically from low to high power and QoE requirements and vice versa.
2) The low-requirements intervals correspond to the best-case (no disturbance) situation; the requirements increases in other intervals are understood as disturbances over the best case.
3) In time-response figures, non-disturbance intervals are represented in white whereas disturbance intervals are represented in grey.
4) In time-response figures, the time axis ends at TL.
5) In time-response figures, exception (EB-control) intervals are distinguished from default (ST-control) intervals by striped areas.
6) In time-response figures, the battery lifetime achieved for C mAh, i.e., AL, is marked with a thick dashed vertical line.

B. Support Platform

A commercial board for developing small multimedia consumer-electronics devices has been used to implement the system. Specifically, in order to enable a comparison consistent with the results obtained by Groba *et al.* [5], [15], the same development board used there has been also used here. As a brief summary, its hardware can be described as being based on a single-core main processor with an assistant DSP and a SIMD coprocessor, 256-MB NAND and 256-MB SDRAM memories and various I/O ports and multimedia interfaces. With respect to the OS, the Linux 3.8.0 kernel has been used, including the *cpufreq* driver as a means to access the DVFS subsystem, which has been configured to manage 27 OPPs with different voltages (from 0.98 V to 1.31 V) and different frequencies (from 125 MHz to 720 MHz).

It is, hence, a low-performance board in which the decoder ST feedback information has been obtained through Linux time-measurement functions, whereas the decoder consumption information has been estimated without needing any specific power monitor but using the general purpose PMCs of the processor. Indeed, as advanced in Sections I and II, the PMC-based estimation method developed by Tang [16] and already applied by Groba *et al.* [5], [15] has been also the one included in this new test bench due to its good results in estimating the current consumption of the video decoder.

C. Video-Decoding Activity

The video decoder, represented as the controlled plant with its ST and power outputs in Fig. 2, has been implemented as an MPEG4-Part2 decoder. On the other hand, in order to meet the alternating requirements level of the video sequence stated in premise 1) in Section V-A, a couple of standard-conformance video clips have been taken from the Open RVC-CAL Compiler repository [31] such that one of them (mat001.m4v) needs far fewer computation resources than the other (mat000.m4v) to be decoded. Besides, three different frame rates have been considered in order to vary also the frame deadline, which affects the decoder ST: 25, 30 and 50 frames/s (fps). Therefore, the video sequence to be decoded during the tests has been built by concatenating a series of the two aforementioned video clips with various performance requirements (temporal resolutions), as it will be specified in Section VI.

Furthermore, in order to meet premise 2) in Section V-A, the non-disturbance situation has been associated to the decoding of the simplest video clip with the lowest temporal resolution, i.e., mat001.m4v at 25 fps. Since the rest of possibilities of the video sequence will be treated as disturbances over this default situation, the LUTs values needed to quantize the controllers output in OPP steps have been obtained by characterizing the plant outputs in open loop while decoding mat001.m4v at 25 fps, aiming to meet premise 5) in Section IV-A. For this purpose, the corresponding monotonically increasing relationship between OPP and average values of decoder ST and power consumption in such default situation had to be acquired. It is worth recalling that the type-1 nature of both SISO control loops makes that they tend to adapt automatically the OPP to achieve the lowest steady-state error, regardless of the accuracy of the LUTs values. Note that LUTs accuracy errors will have the effect of additional disturbances over the control loops. What is more important is to consider all the available OPP steps in order to minimize the quantization step size, as well as to apply an anti-windup mechanism [32] to the ST TR-I controller in order to avoid the undesirable increase of its output beyond the LUT limits during saturation situations. In similar terms, the ST TR-I controller output has to be reset, i.e., set to the lowest limit of the LUT, when the DIDO system switches from the exception to the default status in order to minimize transient oscillations after the status change.

VI. Practical Test

A. Test Conditions

Before presenting the test results, the conditions under which the system test has been carried out are indicated here. First of all, it has been supposed that the user needs are to devote a maximum battery charge of C=15 mAh to view (decode) a video content for at least TL=330 s (0.092 h). These values have been chosen so artificially low simply to shorten the tests duration. Furthermore, three test cases have actually been defined by considering three values (low, medium and high) of the user-preference weighting factor. Thus, from an initial low one, i.e., $\alpha_1 = 0.15$, the other two factor values have been calculated as $\alpha_i = 2\alpha_{i-1} + 0.05(i-1)$ for $i = 2$ and $i = 3$, which leads to $\alpha_2 = 0.35$ and $\alpha_3 = 0.80$. Finally, the set point of the ST control system has been fixed in the same value as that considered by Groba *et al.* [5], i.e., 5%, as an energy-saver one but with a safety distance to negative ST values.

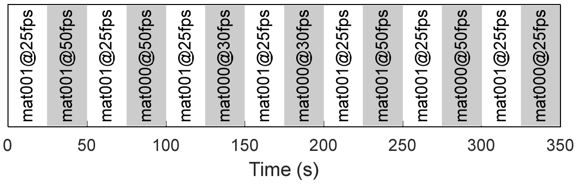

Fig. 5. Structure of the test video sequence.

On the other hand, the video content that the user is supposed to want to view (decode) is composed of the specific sequence of video clips and temporal resolutions (QoE requirements) indicated in Fig. 5, which meets premises 1) to 3) in Section V-A.

B. Test Results

The results obtained from the actual system implemented in the support platform are presented here. In this sense, it is important to highlight, first of all, that the computation-overhead measurements of the DIDO control system yielded values between 4.1% and 4.3% of time in the worst case, i.e., at the lowest OPP, which confirms its expected low value. Moreover, the evolution of EB throughout time under the test conditions for the three values of α stated above and following the premises in Section V-A is shown in Fig. 6. In this figure, B_{th} is also included as a reference to understand the system behavior. Thus, in the default status, the ST control system adapts the OPP to the QoE requirements (disturbances) of the video-sequence decoding, which translates into EB increases during non-disturbance (white) intervals due to low power consumption and more or less sharp EB decreases during disturbance (grey) intervals due to the higher power consumption implied by the higher OPPs needed. When EB reaches B_{th}, the system switches to the exception (EB-control, striped areas in Fig. 6) status and the OPP is adjusted such that EB tries to follow the B_{th} reference line until a decrease in the QoE requirements allows the system to return to its default (ST-control, non-striped areas in Fig. 6) status. On the other hand, as α increases, the range for EB to be controlled in the default status increases too and this leads to fewer exception intervals of restricted QoE. However, this implies also a higher risk of not achieving the desired TL. As a means to identify graphically the achieved lifetime (AL), the battery state of charge throughout time, i.e., SoC(t), is also represented in Fig. 6, such that SoC(AL)=0 once C mAh have been drained from the battery. Indeed, SoC(t) is calculated at a discrete-time basis as

$$\mathrm{SoC}[k] = \mathrm{SoC}[k-1] - P[k]\frac{T}{3600}\ \mathrm{mAh}, \tag{15}$$

being $\mathrm{SoC}[0] = C$. Hence, case (c) in Fig. 6, is the only one in which AL is shorter than TL, although only slightly, i.e.,

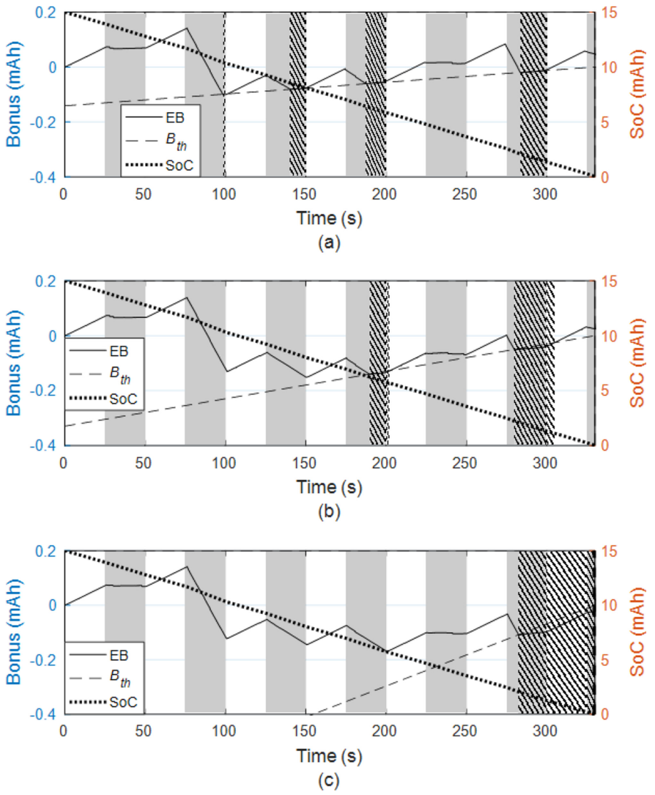

Fig. 6. EB, *Bth* and SoC for the three values of α. (a) 0.15. (b) 0.35. (c) 0.80.

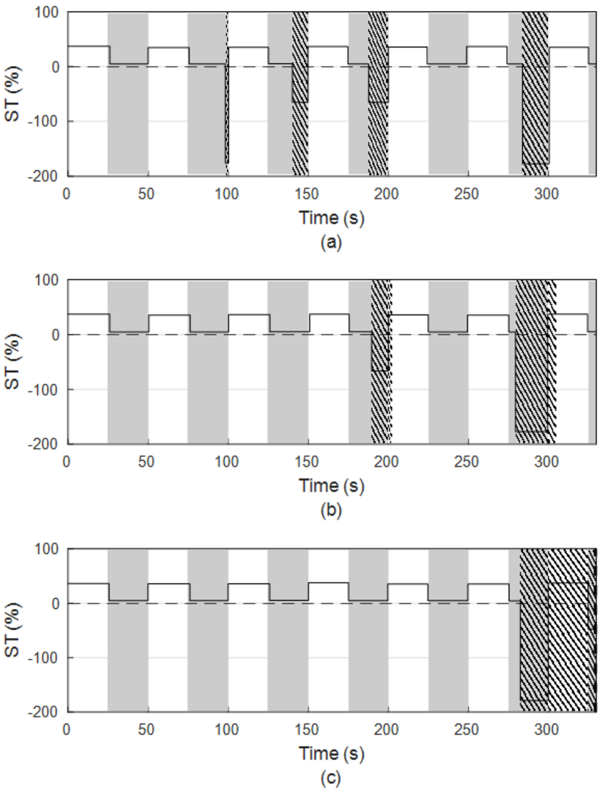

Fig. 7. Average ST for the three values of α. (a) 0.15. (b) 0.35. (c) 0.80.

$\text{AL}_{\alpha 3} = 329.5$ s. This can be also identified by realizing that EB(TL)<0 in Fig. 6 (c). Indeed, the slope of $B_{th}(t, 0.8)$ is so high that, due to (lowest) OPP saturation, the EB control system cannot keep track of it during the disturbance intervals of the exception status in Fig. 6 (c), whereas the last non-disturbance interval is neither long enough as to allow to recover enough EB before TL. Furthermore, from (14) and while the EB controller is not saturated, the value of $e_{SS(EB)}$ is around -6 μAh for the three values of α, which confirms the expected low error values. Besides, since $e_{SS(EB)}$ is negative, EB>B_{th} in steady state, which benefits the system behavior.

As a means of evaluating QoE for the three cases represented in Fig. 6, Fig. 7 shows the average value of ST in each interval compared to the minimum threshold, i.e., the null-ST reference dashed lines marked in Fig. 7. A QoE evaluation metric is proposed such that the shorter the cumulative duration of negative-ST intervals of a test case, the better its QoE. From Fig. 7, it can be realized that the average ST is positive in the default status, being around the set point (5%) during disturbance intervals and higher during non-disturbance intervals due to (lowest) OPP saturation. On the other hand, it can be realized also that, as expected, the average ST becomes negative during disturbance intervals in exception status. Hence, since the cumulative duration of this type of intervals decreases as α increases, the best case with respect to the proposed QoE metric is (c), though TL is not reached. Although it depends on user preferences, perhaps the best tradeoff between lifetime guarantee and QoE is achieved for the intermediate value of α, in case (b) of Fig. 6 and Fig. 7, because TL is reached and QoE is better than in case (a).

VII. Comparative Results

In this section, the results obtained from other related approaches are shown and compared with the ones obtained from the proposed system and presented in the previous section. Specifically, three alternative approaches are considered, all of which were introduced in Section II: The Linux *Ondemand* governor [29], the approach of Groba *et al.* [5] and the approach of Groba *et al.* [15]. Hereinafter, they will be referred to as *Ondemand, ST_control and TL_control*, respectively. While decoding the video sequence described in Fig. 5, the *Ondemand* approach has been tested with the default *cpufreq* parameters, the *ST_control* approach has been tested for a set point of 5% and the *TL_control* approach has been tested with the same values of *C* and TL than those defined in Section VI-A. For comparison purposes, a couple of figures analogous to Fig. 6 and Fig. 7 are used here for the three alternative approaches. Thus, Fig. 8 represents the EB and battery SoC calculated on the basis of (6) and (15), respectively, and registered throughout the test of each approach. In that figure, it can be seen how the two QoE-oriented approaches, i.e., *Ondemand* and *ST_control*, do not reach the desired TL, whereas AL=TL in the *TL_control* approach. The behavior of the latter approach is quite different from the other two because it is designed to achieve a constant consumption of *C*/TL mA, which would lead to a constant null EB except for the small value accumulated during the initial transient period of the control system. It is worth noting that case (b) in Fig. 8 matches the behavior of the proposed system until an exception status arises (see Fig. 6).

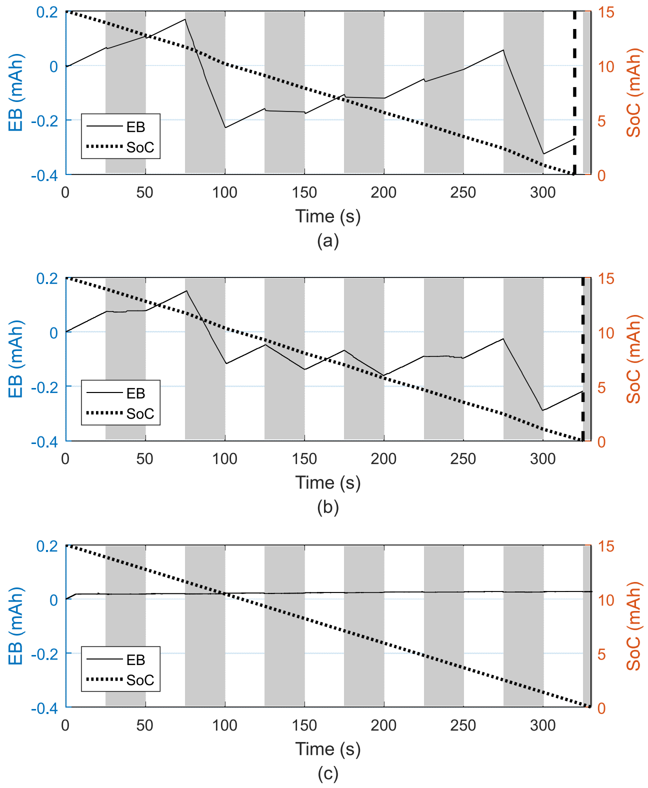

Fig. 8. EB and SoC for the three alternative approaches. (a) *Ondemand*. (b) *ST_control*. (c) *TL_control*.

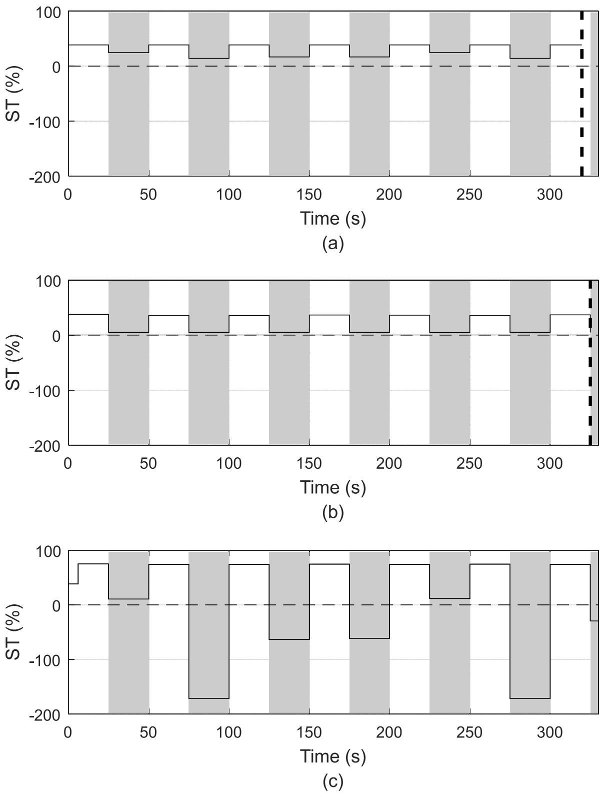

Fig. 9. Average ST for the three alternative approaches. (a) *Ondemand*. (b) *ST_control*. (c) *TL_control*.

TABLE I
ASSESSMENT OF COMPARATIVE RESULTS

System	AL≥TL?	ST<0 [a]
Proposed with α_1	Yes	12.3%
Proposed with α_2	Yes	9.3%
Proposed with α_3	No	6.8%
Ondemand	No	0.0%
ST_control	No	0.0%
TL_control	Yes	31.8%

[a] Percentage of time during which average ST is negative.

On the other hand, Fig. 9 shows the average ST in each interval. As expected, the two QoE-oriented approaches keep the average ST positive, whereas the *TL_control* approach leads to negative values of the average ST in some of the disturbance intervals.

As a summary, Table I assesses the results obtained from the proposed system, with the three values of α, as well as those obtained from the three alternative approaches. The assessment is stated in terms of whether each case achieves or not the TL expected by the user and of the proposed QoE metric, i.e., the percentage of time during which the average ST is negative. From that table, it turns out again that the proposed system with an intermediate value of α, i.e., $\alpha_2 = 0.35$, seems the best option since it is the one that meets TL with the lowest QoE losses. In fact, it improves in around a 25% the time percentage of negative average ST with respect to $\alpha_1 = 0.15$ and a 70% with respect to *TL_control*, the other two cases that meet TL. Anyway, in front of the a priori unknown disturbances that any video decoding is going to produce, the proposed system provides the user with the possibility of increase more or less the value of α in order to put more or less risk on the lifetime to be achieved or on the expected QoE, which is not possible in the alternative approaches.

VIII. CONCLUSION

This paper has presented the modeling, implementation and test results of a dual-loop control system able to guarantee battery lifetime while keeping a tradeoff with QoE during video viewing sessions. Working in two possible statuses, the DVFS-based DIDO control system allows mobile-video consumers to decide up to what extent they prefer to sacrifice QoE or to risk battery lifetime. In the default status, the system controls the QoE in the form of the video-decoder ST such that it keeps track of a positive set point while computes the EB generated. When, due to an increase in the decoder demand, the EB crosses below B_{th}, whose value depends on the aforementioned user decision, the system switches to the exception status in which it controls the EB to keep track of B_{th} at the expense of even canceling ST thus restricting QoE. When, due to a decrease in the decoder demand, ST increases above its set point and EB is not lower than B_{th}, the system switches back to its default status. Based on both basic integral and proportional controllers, the DIDO system implemented in the OS of a commercial development board works transparently to the video decoder with a worst-case computation-time overhead of about only 4.2%. Under a demanding test case, with changing multimedia features, the system is compared to other related approaches from which only one is able to meet the expected

TL but with more than three times of restricted-QoE intervals than the proposed system for an intermediate B_{th} reference.

REFERENCES

[1] *Ericsson Mobility Report*, Ericsson, Stockholm, Sweden, 2018. [Online]. Available: https://www.ericsson.com/assets/local/mobility-report/documents/2018/ericsson-mobility-report-november-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. doi: 10.1109/MCE.2014.2361266.

[3] A. Martins, M. Ruaro, A. Santana, and F. G. Moraes, "Runtime energy management under real-time constraints in MPSoCs," in *Proc. IEEE Int. Symp. Circuits Syst.*, Baltimore, MD, USA, 2017, pp. 1–4. doi: 10.1109/ISCAS.2017.8050947.

[4] R. Ren, J. Wei, E. Juárez, M. Garrido, C. Sanz, and F. Pescador, "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. doi: 10.1016/j.image.2013.08.014.

[5] 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. doi: 10.1109/TCE.2018.2843284.

[6] 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. doi: 10.1109/ETIICT.2017.7977013.

[7] 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. doi: 10.1109/ICCW.2015.7247442.

[8] 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. doi: 10.1109/ICCCBDA.2018.8386525.

[9] R. Ren, E. Juárez, C. Sanz, M. Raulet, and F. Pescador, "Energy-aware decoder management: A case study on RVC-CAL specification based on just-in-time adaptive decoder engine," *IEEE Trans. Consum. Electron.*, vol. 60, no. 3, pp. 499–507, Aug. 2014. doi: 10.1109/TCE.2014.6937336.

[10] N. Sidaty, J. Heulot, W. Hamidouche, E. Nogues, M. Pelcat, and D. Menard, "Reducing computational complexity in HEVC decoder for mobile energy saving," in *Proc. Eur. Signal Process. Conf.*, 2017, pp. 1026–1030. doi: 10.23919/EUSIPCO.2017.8081363.

[11] 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. doi: 10.1109/TCE.2014.6852003.

[12] 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. doi: 10.1109/ISCE.2015.7177807.

[13] 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. doi: 10.1109/TCE.2014.6780938.

[14] 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. doi: 10.1109/PERCOM.2008.108.

[15] A. M. Groba, P. J. Lobo, and M. Chavarrías, "Closed-loop system to guarantee battery lifetime for mobile video applications," *IEEE Trans. Consum. Electron.*, vol. 65, no. 1, pp. 18–27, Feb. 2019. doi: 10.1109/TCE.2019.2891178.

[16] Q. Tang, "Control algorithms for energy optimization in multimedia hand-held devices," Ph.D. dissertation, Dept. Telematic Electron. Eng., UPM, Madrid, Spain, 2017. doi: 10.20868/UPM.thesis.46893.

[17] F. Kaup and D. Hausheer, "Optimizing energy consumption and QoE on mobile devices," in *Proc. IEEE Int. Conf. Netw. Protocols*, Göttingen, Germany, 2013, pp. 1–3. doi: 10.1109/ICNP.2013.6733641.

[18] H. K. Yarnagula, R. K. Vooda, and V. Tamarapalli, "A measurement study of energy consumption and QoE trade-offs for DASH in mobile devices," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst.*, Bengaluru, India, 2016, pp. 1–3. doi: 10.1109/ANTS.2016.7947817.

[19] J. Cheng, X. Wen, Z. Lu, and Y. Chen, "Joint optimization of energy and video quality for scalable video in mobile devices," in *Proc. Int. Conf. Inf. Sci. Security*, Seoul, South Korea, 2015, pp. 1–5. doi: 10.1109/ICISSEC.2015.7371012.

[20] S. Zhou, M. Ran, and Z. Lu, "Adaptive energy-efficient and QoE-aware optimization method for mobile video services," in *Proc. Int. Symp. Commun. Inf. Technol.*, Qingdao, China, 2016, pp. 388–392. doi: 10.1109/ISCIT.2016.7751657.

[21] 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. doi: 10.1109/TCE.2016.7613197.

[22] S.-W. Jo, W. Yoo, and J.-M. Chung, "Video quality adaptation for extended playback time on mobile devices with limited energy," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1260–1263, Jun. 2018. doi: 10.1109/LCOMM.2018.2820685.

[23] J. W. S. Liu, *Real-Time Systems*. Upper Saddle River, NJ, USA: Prentice-Hall, 2000.

[24] X. Li, Z. Jia, and L. Ju, "Slack-time-aware energy efficient scheduling for multiprocessor SoCs," in *Proc. IEEE Int. Conf. Embedded Ubiquitous Comput.*, 2013, pp. 278–285. doi: 10.1109/HPCC.and.EUC.2013.48.

[25] R. K. Pal, K. Paul, and S. Prasad, "Energy efficient dynamic core allocation for video decoding in embedded multicore architectures," in *Proc. IEEE Int. Conf. Embedded Softw. Syst.*, Paris, France, 2014, pp. 653–660. doi: 10.1109/HPCC.2014.95.

[26] 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. doi: 10.1109/TII.2010.2051232.

[27] G. Chen *et al.*, "Utilization-based scheduling of flexible mixed-criticality real-time tasks," *IEEE Trans. Comput.*, vol. 67, no. 4, pp. 543–558, Apr. 2018. doi: 10.1109/TC.2017.2763133.

[28] D. Brodowski, N. Golde, R. J. Wysocki, and V. Kumar. *Linux CPUFreq Governors*. Accessed: Apr. 3, 2019. [Online]. Available: https://www.kernel.org/doc/Documentation/cpu-freq/governors.txt

[29] V. Pallipadi and A. Starikovskiy, "The ondemand governor," in *Proc. Linux Symp.*, vol. 2. Ottawa, ON, Canada, 2006, pp. 215–230.

[30] 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. doi: 10.1109/TCE.2016.7838088.

[31] SourceForge. *Open RVC-CAL Compiler, Sequences*. Accessed: Apr. 3, 2019. [Online]. Available: http://sourceforge.net/projects/orcc/files/Sequences/MPEG4.zip

[32] F. Golnaraghi and B. C. Kuo, *Automatic Control Systems*, 9th ed. Hoboken, NJ, USA: Wiley, 2009.

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 UPM, where he is currently a member of the Research Center on Software Technologies and Multimedia Systems for Sustainability. His current research interests focus on control systems for energy optimization in multimedia systems.

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 a Teaching and Research Assistant with UPM, where he has been with 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.