Microelectronics Reliability 119 (2021) 114071

Contents lists available at ScienceDirect

Microelectronics Reliability

journal homepage: www.elsevier.com/locate/microrel

Remaining useful life prediction in embedded systems using an online auto-updated machine learning based modeling

Oussama Djedidi [a,*], Mohand A. Djeziri [a], Samir Benmoussa [b]

[a] Aix-Marseille University, Université de Toulon, CNRS, LIS, Marseille, France
[b] Laboratoire d'Automatique et de Signaux de Annaba (LASA), Université Badji Mokhtar, Annaba, Algeria

ARTICLE INFO

Keywords:
Machine learning
Remaining useful life
System on Chip
Failure prognosis

ABSTRACT

Systems on Chips are increasingly involved in critical equipment in the fields of aeronautics, transportations, and energy. Therefore, monitoring their life cycle is a crucial issue for safety and hazard-prevention. This paper deals with a data-driven method for online prediction of the Remaining Useful Life (RUL) of the safety-critical System-on-Chips (SoC). This method is based on the detection and prediction of drifts in their operating temperatures. The work starts with a description of the formal relationships between temperature drifts and the degradation process of SoCs to justify the choice of the temperature as an indicator of the level of the degradation in the system. Then, temperature-based physical health indicators are constructed using data-driven analytical redundancy. Since temperature varies not just according to the degradation state of the system, but also according to its various normal operating points, data-driven analytical redundancy makes it possible to obtain a health indicator that has a well-defined physical meaning, and which is only sensitive to the SoC degradation process. To predict the remaining useful life of the chip, the trend of the drift is modeled using an auto-regressive neural (NAR) network. The latter is updated online according to the evolution of the temperature drift and the state of the system. Finally, forecasts of the remaining useful life of the SoC are obtained using a combination of temporal projection and threshold data. Simulations and experimental results highlight the effectiveness and accuracy of the proposed approach.

1. Introduction

Systems-on-Chips (SoCs) are increasingly embedded in safety-critical systems and undergoing strict certifications to ensure their reliability. The latter is generally characterized by the Mean Time To Failure (MTTF) provided by the manufacturers, and calculated in the laboratory testing conditions using data from a battery of tests [1]. However, the volumes of SoCs that are manufactured, the complexity of the manufacturing processes, as well as the great variability of the conditions of use, which are difficult to reproduce in a laboratory setting, require the development of algorithms for the online monitoring of SoCs used in safety-critical systems.

This work offers a solution to this issue by developing a data-driven method for the failure prognosis of SoCs, based on detecting and trend-modeling of temperature related drifts. In previous works, Djedidi et al. [2] built and validated a modeling framework and a simulator that estimates a set of key variables of the SoC including the temperature of the SoC. These key variables will be used in this work to detect drifts in the behavior of the system through analytical redundancy. The drift is then modeled in order to estimate the Remaining Useful Life (RUL).

Fault diagnosis and failure prognosis have been the subject of several research works and reviews [3,4], where a classification divided existing approaches into three categories: Model-based, Data-driven [5,6] and hybrid approaches [7,8]. However, in the field of fault diagnosis of embedded systems, SoCs are never studied independently. The electronic card is studied as a discrete system whose functioning depends on the proper functioning of all the sub-components [9]. In this area, causal models such as the multi-signal flow graph [10], information flow models [11], Direct graphs [12], Dependency graphical model [13], are the ones used the most. A classification and a detailed study of existing online error detection techniques applied to multicore processor architectures are presented by Gizopoulos et al. [14]. The authors propose categorized existing methods into four main categories: redundant execution [15–17], periodic Built-In Self-Test, BIST [18], dynamic verification [19], and anomaly detection approaches [20]. Löfwenmark and Nadjm-Tehrani [21] published a review which brought together

* Corresponding author.
E-mail addresses: oussama.djedidi@lis-lab.fr (O. Djedidi), djeziri@lis-lab.fr (M.A. Djeziri), samir.benmoussa@univ-annaba.dz (S. Benmoussa).

https://doi.org/10.1016/j.microrel.2021.114071
Received 10 June 2020; Received in revised form 4 February 2021; Accepted 18 February 2021
Available online 3 March 2021

Table 1
The main features of the Freescale development board i.MX6 SoloX [29].

OS	Android 6.0.1 (Marshmallow)
SoC	MCIMX6SX
• CPU	i.MX 6SoloX processor • 1 core ARM Cortex-A9 (1GHz) • 1 core ARM Cortex-M4 (0.2 GHz)
• GPU	3D: Vivante GC400T et 2D: Vivante GC320
• RAM	1 GB
Communication	–
I/O	Touchscreen: HD LCD
Power source	Continuously powered

works on multicore systems in avionics. In their review, the authors pointed out that the sensitivity of systems to faults is constantly increasing due to the reduction in the size of the transistors, and have highlighted the areas in which research is still necessary. [21]. Other research work has been carried out in the field of the failure prognosis of electronic systems, in particular for the estimation of the RUL, where relatively new techniques for its predictions were explored, such as particle filters [22,23], Kalman Filters [24], polynomial models [25], support vector regression [26], iterative nonlinear degradation Auto-Regressive models [27], and Wiener processes [28]. All the previously mentioned works describe methods which focus on a specific aspect or component of the embedded system or SoC (reliability, hardware or software components, etc.), and are mostly applicable offline. This work, on the other hand, deal with the online assessment and modeling of the degradation process in SoCs. In a first step, we investigate the usability of the temperature as a health indicator in the context of failure prognosis by establishing a formal analysis of the physical mechanisms causing the degradation process in SoCs. The link between these mechanisms and temperature drift is highlighted and then experimentally confirmed by temperature drift measurements recorded over three years of use of an SoC under laboratory conditions of use.

However, as the temperature is a dynamic variable that varies in a system, to distinguish temperature drifts caused by degradation from functional variations, a health indicator is constructed by comparing the system's temperature to those generated by a reference model. Once the drift is identified, its trends are modeled using a nonlinear Auto-Regressive model. The main contribution of this work lies in the periodic online updating of the trend model, thus adapting it to take account of changes in the conditions of use of the SoC or changes in the profile of the drift. The update is conditioned by the quadratic error observed in the estimate of the health indicator. The drift is then projected in time to obtain a prediction of the RUL. The value of the latter is reassessed at each model update.

The remainder of this paper is organized into 7 sections. In Section 2, we describe the studied SoC. After this, the usability of the temperature drift as a health indicator is investigated in Section 3. Then, an overview of the approach proposed in this study is detailed in Section 4. Temperature modeling in embedded SoCs is presented in Section 5, followed by the construction of the Health indicator. In the last two sections, The process of introducing the degradation to the system then modeling its trend are detailed, culminating with the prediction of the RUL. Finally, concluding remarks are given at the end of the paper.

2. Case study

2.1. System description

The considered case study is the *Freescale i.XM6 SoloX* development board [29]. This system is generally used for the design of multimedia content execution and display systems, especially for vehicles, as it is certified for use in safety critical systems.

The board works under Linux and is also compatible with Android 6.0.1. It has an ARM Cortex processor with 1 GB of RAM [29]. This system is also connected to a touch screen. The relevant characteristics of the board are summarized in Table 1.

2.2. Measurements and setup

The focus of the present study is to model the effect of the temperature on the RUL of the SoC, which requires both the measurement and the modeling of the said temperature. Modern SoCs are all equipped with temperature sensors to monitor their thermal state and control the frequency of the processors. Indeed, the development board is equipped with a temperature sensing module (called *TempMon*). This module delivers the temperature of the neighboring region in the SoC, as well as the temperature of the surrounding room. However, as it is the case for all electronic temperature sensors, its accuracy is hindered by its greater measurement error relative to more accurate instruments like thermocouples.

Nevertheless, since the module provides the measurement on which the CPU relies to control its thermal regulation, its readings are also going to be used for the validation of the temperature model and the prediction of the RUL, thus requiring no external sensors or intrusion. Additionally, by comparing the readings from the temperature sensing module and measurements from an external thermocouple, we found a mean deviation between the two equal to 0.6 °C (*TempMon* measures in increments of 1 °C whereas the thermocouple has a sensitivity of 0.2 °C), and standard deviation $\sigma \approx 1$ °C comparable to the results reported by the manufacturer. Hence the accuracy of the sensor module is sufficient for the use case in hand.

In addition to temperature values, several other readings are necessary to model the thermal behavior of the SoC. These variables are the frequencies of the CPU cores, the occupied RAM, and the power

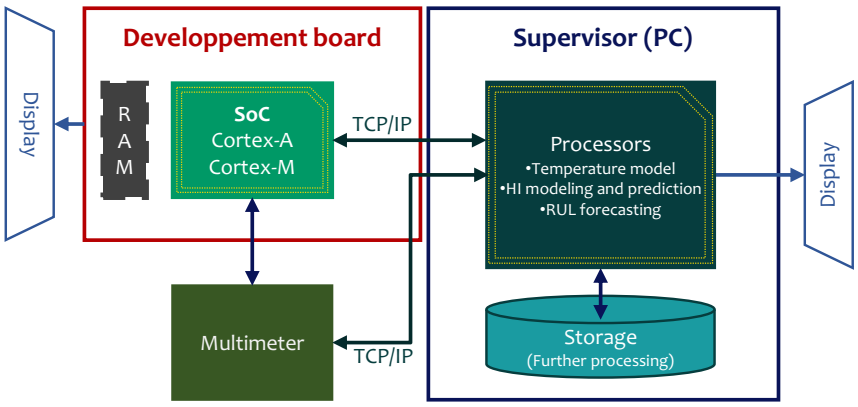

Fig. 1. The development board alongside the multimeter and the monitoring PC.

consumption. The correlation between the thermal output of the SoC and these variables has been thoroughly studied in the literature. Furthermore, in the previous works that paved the way for this study [30,31], this relationship was exploited to monitor the operating state of the SoC and its software workload, and detect faults caused by the environment or over-solicitation of the system.

The values of the frequencies and the RAM are obtained through system traces and are read every 20 ms, the minimum period at which the frequencies are reevaluated by the Frequency Governor. Since the frequency is the fastest-changing variable, it also dictates the sampling time of the readings and the model. Thus, T_s=20 ms. Finally, the power consumption is measured by an external multimeter at the same rate as the other variables.

All the measurements are transferred via a TCP/IP protocol to a supervising PC where they are processed. Fig. 1 is a schematic representation of the experimental setup used in this study. A more detailed description of this setup can be found in previous works [30,31].

3. Wear-out mechanisms and temperature drift

Under continuous usage, SoC in embedded systems suffer from diverse effects—both internal and external—that causes their reliability to be compromised and causes them to fail. In this section, the formal relationships between the degradation mechanisms and the temperature drift are given and discussed [32,33].

3.1. Backend time-dependent dielectric breakdown (BTDDB)

The characteristic lifetime η, which is the time at which 63% of the population have failed [34], is clearly a function of temperature. For the dielectric segment of microprocessors associated with a linespace S_i, η is equal to:

$$\eta = AL_i^{-1/\beta_i} e^{\left(-\gamma E^m - \frac{E_a}{kT}\right)} \tag{1}$$

where A is a constant depending on the properties dielectric material, L_i and β_i are the vulnerable length and the shape parameter associated with the linespace S_i. γ is the field acceleration factor. E which equals to:

$$E = \frac{V}{S_i} \tag{2}$$

represents the electric field. m is a constant that depends on the model. E_a is the activation energy. k is the Boltzmann constant, and T is the temperature [33]. Hence, for a working processor where the geometry is fixed, η is characterized by the voltage and the temperature.

3.2. Electromigration (EM)

In microelectronic chips, EM causes an increase in the resistance of the material. Once the resistance reaches a certain threshold, the system fails. Eq. (3) describes the characteristic lifetime under EM for a line [34].

$$\eta = \frac{h}{\delta_s} \frac{kTL_{via}}{eZ^* \rho j D_s} \tag{3}$$

h is the thickness of the line. δ_s represents the thickness of the surface. k is the Boltzmann constant. L_{via} is the size of the via. e is the electron charge. Z^* is the effective charge. j is is the current density. ρ is the resistivity of the metal. This equation clearly shows how EM is temperature-dependent [33].

3.3. Stress induced voiding (SIV)

As it was seen with the previous two mechanisms, SIV is also temperature-dependent [35,36]. Under SIV, the characteristic lifetime for an interconnect is written as:

$$\eta = AW^{-M}(T_0 - T)^{-N} e^{\left(\frac{E_a}{kT}\right)} \tag{4}$$

In this equation, A is a constant, W is the line width, M depends on the stress component, T_0 is the stress-free temperature, and N is the thermal stress component.

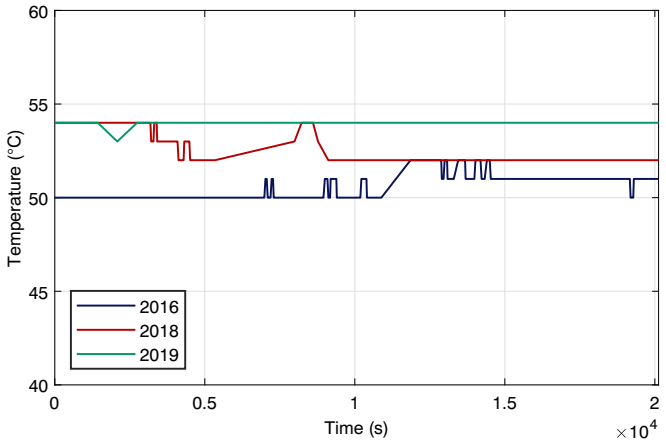

Fig. 2. Average temperature increase over a period of three years for the same workload and operating conditions (f=996 MHz, T_{amb}=25 °C, 180 s after power on).

3.4. Bias temperature instability (BTI) and hot carrier injection (HCI)

Negative BTI affects PMOS transistors, while PBTI affects NMOS. These mechanisms lead into an irrecoverable increase in the value of the threshold voltage of the transistor $|V_{th}|$ [37]. The value of $|V_{th}|$ is a direct function of both the supply voltage V_{dd} and the temperature [36]. The threshold voltage temperature-dependence is described by the Arrhenius relationship $\left(e^{\frac{-E_a}{kT}}\right)$ [32].

In a similar fashion to NBTI and PBTI, this mechanism leads to an increase in $|V_{th}|$. HCI depends on the frequency f, V_{dd}, and the temperature T which is also described by the Arrhenius relationship [38].

3.5. Thermal cycling

As the name suggests, this mechanism is the is directly temperature dependent. The number of cycles of thermal fatigues to cause a failure is described by the Coffin-Manson equation [39].

$$N_{cyc} = \sum \frac{(\Delta T_{ref})^{-m}}{(\Delta T)^{-m}} \tag{5}$$

In Eq. (5), ΔT is the is the thermal swing that the die goes through. ΔT_{ref} is the thermal swing that it is designed to withstand, whereas m is the Coffine-Manson exponent.

The degradation mechanisms mentioned above highlight the phenomenon of the SoC temperature increasing with and wear out. This deduction is reinforced by an experimental result obtained on the board used in this work, where after three years of experimentation, temperature drift has been recorded as shown in Fig. 2, under the same conditions of use. This increase in temperature appears more clearly on the average of the signal which further evidences that it is a progressive drift.

4. Overview of the proposed approach

RUL prediction is a problem that has been approached with several methods. Most of them are data-driven, such as run-to-failure [40],

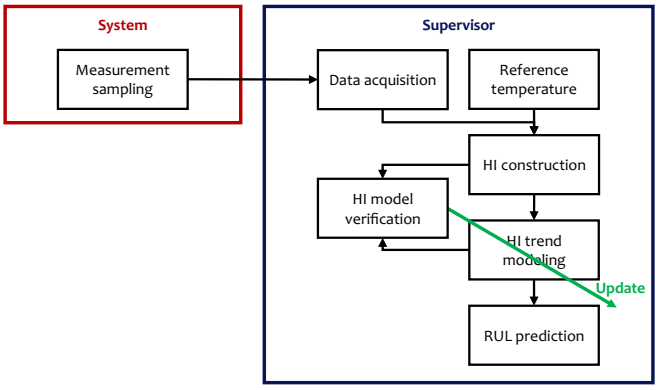

Fig. 3. General overview of the RUL prediction approach. *HI*: Health Index.

lifetime data [41,42], and the method we use in this work—Threshold data [43]. This approach relies on generating and studying a Health Indicator (*HI*). As the system ages or is stressed, this *HI* evolves until it reaches a point of no return—a threshold where it fails or can no longer function properly or be operated safely [43].

Accordingly, the effectiveness of the threshold data approach depends on the choice of *HI* and the models used to forecast its future values. Nevertheless, it offers two main advantages that have favored its use in this particular work. Firstly, it allows for the exploitation of data verified and supplied by the manufacturer to choose the value of the threshold and the initial value of the RUL. Secondly, the prediction of the RUL using the threshold data is a function of the actual operating state and condition of the system, contrary to how the Mean Time To Failure and the run-to-failure are determined. Both of these methods, the SoC runs at the highest settings [44], hence focusing only on one operating scenario. Furthermore, in this work, the predictions are further improved by the online updating of the *HI* model according to current operating settings, and the latest incoming data.

In this work, *HI* is constructed by comparing the real state of the system—temperature measured online—to a reference one estimated by a model of normal operation. Reference values are indicative of the state of the board without faults and before wear. The values measured online, on the other hand, reflect the state of the system under actual working conditions and during the process of degradation. Hence, as the board is subjected to stress, the measured temperature of the SoC would continuously drift from its reference values, thus raising the values of *HI*. Fig. 3 shows a diagram of the proposed approach, in which the SoC sends temperature measurements to a supervisor. In the latter, reference temperature values are generated online, allowing for the construction of HI, and the building of a prediction model. Finally, the *HI* model is verified and updated by comparing its prediction with the constructed values.

This approach accounts for the two cases that can arise in practice:

- If the data describing the degradation profile are available beforehand, the model can be trained offline and then updated online to the specific use case of the SoC, and generate RUL estimations accordingly.
- If no data is available, the proposed model can be trained online and updated to follow the real degradation drift, and generate RUL prediction online.

Fig. 4 details the process of the construction of the health indicator, the building for its trend model—and its subsequent updates, and finally the prediction of the RUL.

There are mainly two types of health indicators; physical and virtual ones [45]. *HI* in this study is a physical one. It is the result of the calculation of the residuals. Firstly, the estimation residuals—hereafter called raw HI (HI_r). As information about the drifts is contained in the average value of the signal [2], HI_r is then filtered using the moving mean to obtain the final *HI*. This approach of using residuals as health indicators have been previously used to detect drifts in characteristics of embedded systems, and the identification and isolation of both internal and external faults in these systems with great results [30], hence encouraging their further use in the diagnosis and prognosis of SoCs.

The RUL is predicted by the temporal projection of the degradation evolution (*HI*) until it reaches the predefined End Of Life (EOL) threshold, which represents the maximum drift of the SoC temperature from its nominal value. In this work, it is derived from the certified maximum operating temperature set by the system's constructor [44]. The temporal projection, on the other hand, is done through the building of a trend model and using it to forecast $\widehat{HI}$.

To further fit the predictions of the RUL to the use case of SoC, the trend model is updated online when the predicted values of $\widehat{HI}$ deviates from the measured one (*HI*), this update criterion is determined according to the values of the tolerated error th_{HI}.

5. Temperature modeling and estimation

5.1. Temperature modeling

Temperature dynamics in SoC depend on a wide variety of factors such as the impedance of the circuit [46], the power leakage [47], and the workload [30,48]. Thus, a theoretical temperature model would have to account for these factors. Nevertheless, since the objective of this

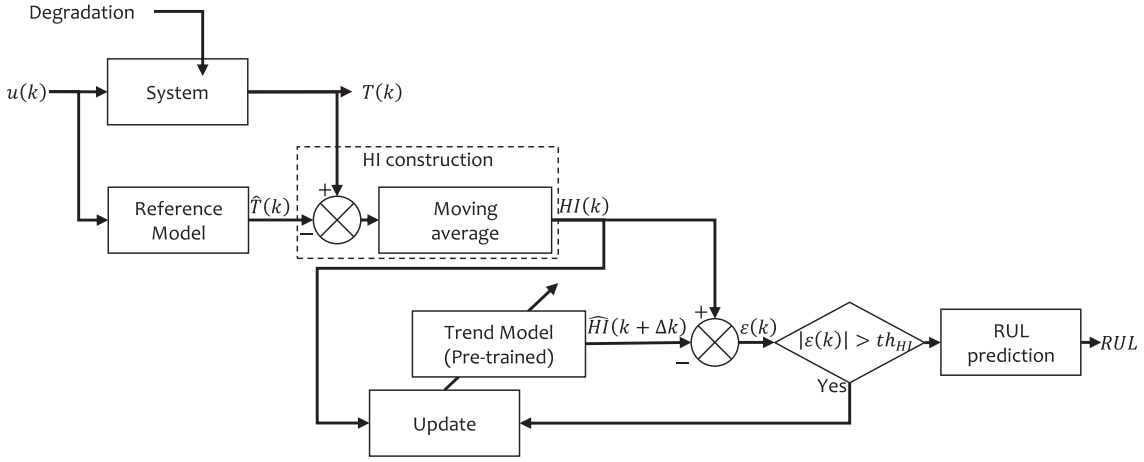

Fig. 4. Overview of the RUL prediction approach.

Table 2
The accuracy of the ARMAX model according to its orders.

Orders of the model			MAPE (%)
n_a	n_b	n_c	
2	2	2	11.1853
3	3	3	7.9672
4	4	2	1.1658
4	4	4	1.3757
5	5	2	0.7215
6	6	2	1.3667

Table 3
A comparison between the accuracy in terms of the Mean Absolute Error (MAE) and the Mean Absolute Percentage Error (MAPE) and time needed to generate estimations during the online and the offline tests of the ARMAX model.

Test	MAE (°C)	MAPE (%)	Average sampling time (s)
Offline test	0.52	1.1658	$\sim 1 \times 10^{-3}$
Online test	0.56	1.3757	18×10^{-3}

paper is to observe temperature evolution through the lifetime of the board, a model that is capable of generating temperature estimations according to operating conditions is largely sufficient. Hence, we opted for an Auto-Regressive-Moving-Average with eXogenous Input model (ARMAX).

ARMAX models are linear models that estimate or predict the output of a system according to its inputs and the history of its outputs and inputs. Eq. (6) is the difference equation that represents a discrete multiple inputs single output ARMAX model.

$$\begin{aligned} y(k) = a_1 y(k-1) + \cdots + a_{n_a} y(k-n_a) \\ + b_{1,1} u_1(k-d_u) + \cdots + b_{m,n_b} u_m(k-d_u-n_b) \\ + e(k) + c_1 e(k-1) + \cdots + c_{n_c} e(t-n_c) \end{aligned} \tag{6}$$

whereas $y(k)$ is the output, $u(k)$ is the input, and $e(k)$ is the moving average. The parameters: $[a_1, \cdots, a_{na}]$, $[b_{1,\ 1}, \cdots, b_{1,\ nb}]$, $\cdots$, $[b_{m,\ 1}, \cdots, b_{m,\ nb}]$ and $[c_1, \cdots, c_{nc}]$ are the regression, the input, and the moving average coefficients, respectively. n_a, n_b and n_c are the orders of the model, τ is the input delay and m is the number of inputs in the input vector $u(k)$.

In our case, the output is the temperature of the SoC, and the vector input is:

$$u(k) = [f_1(k), \ldots, f_n(k), f_{GPU}(k), MOR(k), P_{SoC}(k)] \tag{7}$$

5.2. Parameters identification and model validation

For data gathering, the device is set to run through a scenario of various workloads to simulate different types of usages [49]. At average, each run results in a set of 2×10^5 samples. Once the data is gathered, it is divided into two sets; a training set (70%) and a validation set (30%). The training set is used to compute the regression, input and mean average parameters. While the validation set is used to measure the accuracy of the model.

The identification of the set of orders that generate the best accuracy in the offline tests is done iteratively with sets varying from [2,2,2] to [9,9,9]. Table 2 shows the accuracy of the model in terms of Mean Absolute Percentage Error (MAPE) for each configuration. It also shows the time needed by the supervising equipment to gather the data and generate estimations. Amongst these results, the set [n_a=4,n_b=4,n_c=2] has one of lowest MAPE (0.8377%), with an average sampling time (Table 3) that allows for sampling the frequency without any loss of information (T_s= 20 ms) [2,50]. Hence, it is the chosen set.

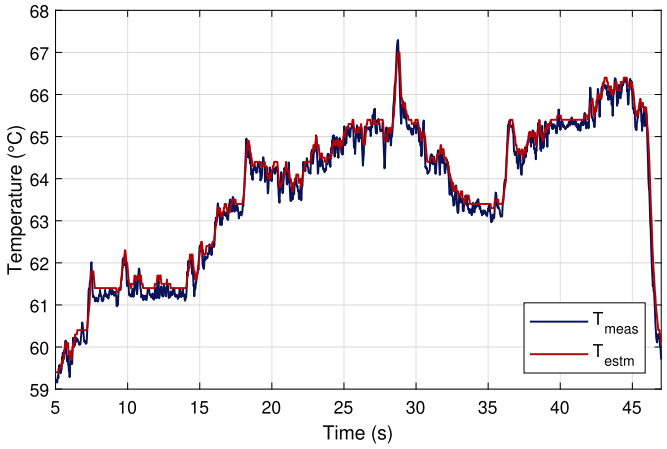

Fig. 5. The ARMAX model temperature estimation against device measurements.

Then, we proceeded to the test of the model online by directly comparing its estimations to the measurement from the board. Table 2 highlights the accuracy of the model during an online test. Fig. 5 show how the estimations generated by the model fit the measurements from the system, during both heating and cooling phases whereas Fig. 6 show how low the estimation errors are on average. The ARMAX model is thus validated.

6. Health Indicator construction

To construct a health indicator sensitive only to degradation, we use analytical redundancy, which consists of comparing the measured behavior of a system with a reference one describing its normal operation. This reference behavior can be accomplished by a model; either physical or data-driven. Accordingly, in this study, the *HI* is constructed

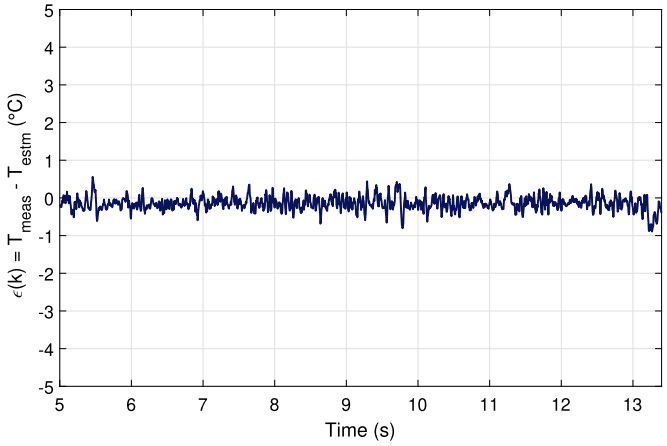

Fig. 6. The estimation errors of the ARMAX temperature model: $\varepsilon(k) = T_{meas}(k) - T_{estm}(k)$.

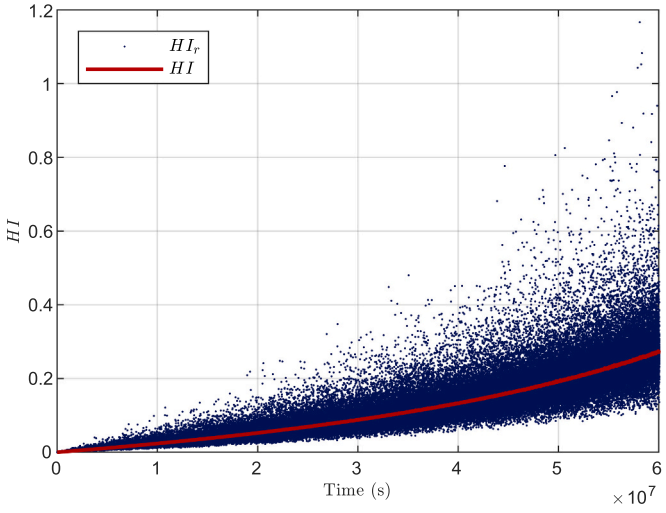

Fig. 7. Progression of the noisy raw health indicator (HI_r), and the subsequent health indicator (HI) over the life on the SoC.

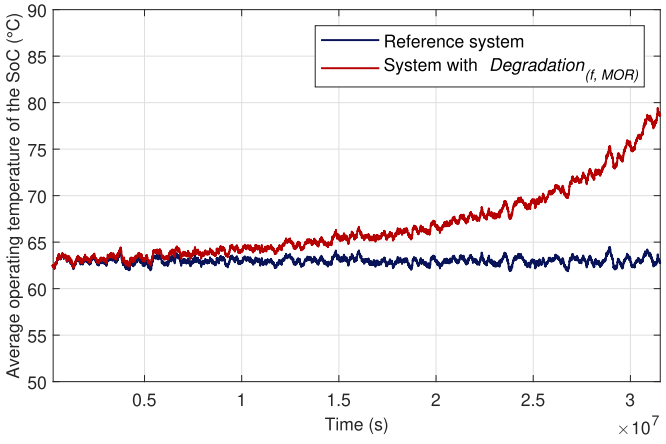

Fig. 8. Measured temperature compared to the estimated one.

by comparing the measured temperature with the reference ones estimated by the model of normal operation presented in the previous section. In the first step, the raw health indicator (HI_r) is computed by the following equation:

$$HI_r\left(k\right) = \frac{T_{meas}(k) - T_{estm}(k)}{T_{estm}(k)} \tag{8}$$

However, the raw form of HI contains also information related to the noise of the signal and estimation errors. Since the drift trend information is contained in the health indicator average and in order to reduce the noise from the estimation errors, the raw HI_r is processed by extracting its moving average yielding HI, and is expressed as follows:

$$HI(k) = \frac{1}{n}\sum_{i=k-(n-1)}^{k} HI_r(i) \tag{9}$$

whereas n is the length of the observation window used for the calculation of the moving average.

Fig. 7 shows how HI filters the noise from the raw HI_r. It also highlights the suitability of HI to detect a progressive deviation of the temperature drift from its reference value, thus supporting our choice of this physical health indicator.

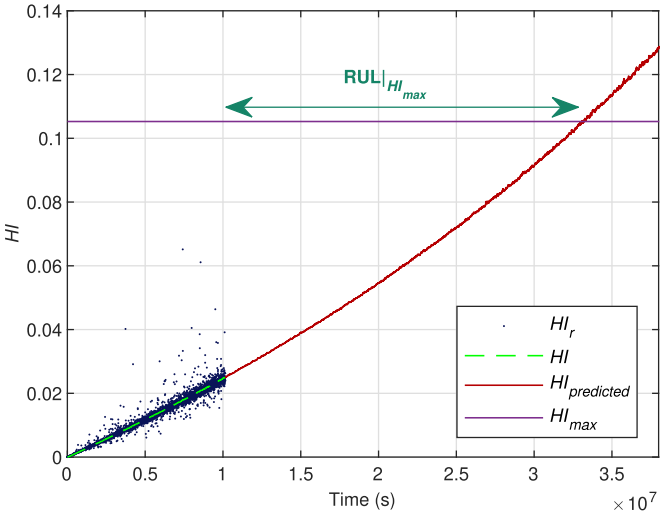

Fig. 9. Prediction of the $RUL|_{HI_{max}}$ by the temporal projection of HI.

7. Introduction of degradation process

In addition to the experimental dataset including a degradation process of the considered SoC in the case study, a simulator was validated experimentally and detailed in [49]. It was used to simulate degradation processes in the SoC and generate representative profiles of the latter [30], and utilized in this work in the learning step. Since the simulator is open and extensible, it offers the possibility of introducing degradation in its subsystems. Thus, a progressive degradation is introduced in the system for a period of 1.576×10^6h (equivalent to 3 years of continuous functioning) with a sampling time of 20 ms, resulting in 4.730×10^9 samples. The workload used during these simulations—as described by the frequency and the MOR—corresponds to stress loads inspired by the use of benchmarks such as AnTuTu Benchmark [51], 3DMark [52], to simulate an accelerated wear out scenario.

Fig. 8 shows the how averaged temperatures drifts slowly from the reference values due to the degradation. These temperature are computed using the moving average on a duration of a simulated week ($n \approx 3 \times 10^7$ samples). The temperature drift continues its increase until it reaches and surpasses the maximum temperature at which the device can no longer function according to its manufacturer, in this case the Maximum Tolerated Temperature (MTT) equals 105 °C [53]. In the considered application, the Maximum Reference Temperature (MRT) under stress loads is equal to 95 °C. Hence, the failure threshold can be set to:

$$HI_{\max} = \frac{MTT - MRT}{MTT} = 0.1052 \tag{10}$$

8. Trend modeling and RUL prediction

After defining the failure threshold, the prediction of the RUL is carried out by modeling the trend of the degradation process, and then by the temporal projection of the HI trend up to the failure threshold (HI_{max}), which corresponds to the duration that HI take to reach HI_{max}, as illustrated in Fig. 9. We define this duration as the $RUL|_{HI_{max}}$.

In the rest of this section, we first describe the HI trend model. Then, this model is validated offline using heavy use scenarios representative of accelerated wear. Finally, it is used and validated online against a scenario representative of all different types of usages to showcase how it is adapted.

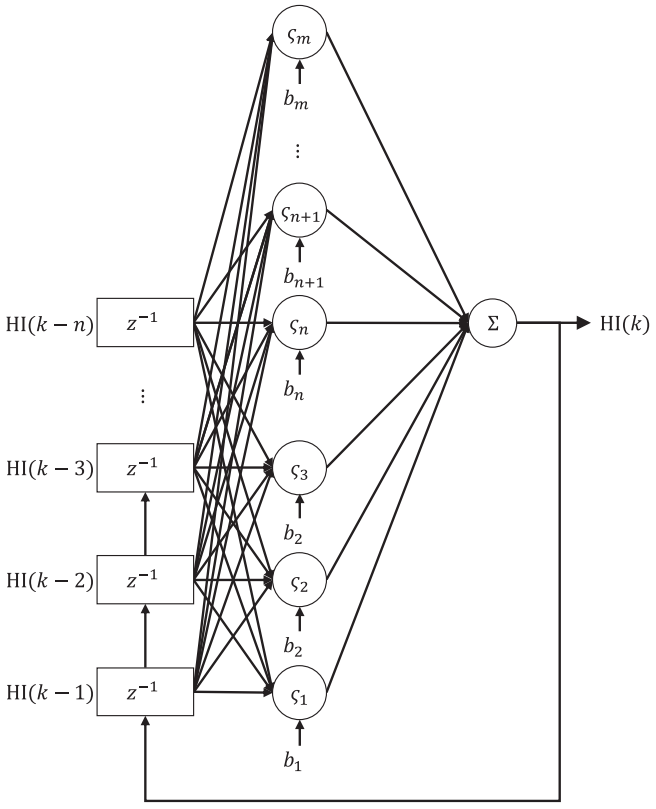

Fig. 10. General structure of the NAR *HI* trend model.

8.1. Description of the HI trend model

At time t=0, the *RUL* is considered equal to the MTTF announced by the manufacturer of the SoC and obtained thanks to reliability tests in laboratory conditions. Then, the *RUL* value is predicted and updated according to the evolution of the degradation state. In this work, we rely on machine learning-based times series prediction to achieve this task, by the use of a Nonlinear Auto-Regressive (NAR) neural network. These networks are commonly used to predict the evolution of timestamped variables making them an optimal choice for our use case. NAR neural networks are recurrent networks of which the output is a function of the history of the outputs:

$$y(k) = \varphi[y(k-1), \cdots, y(k-n)] \tag{11}$$

where $y(k)$ is the output to be predicted at the time step k, φ is the characteristic function of the network, and n is the order of time delays for the output (also called input and output memory). The NAR model hereby used, is composed of a hidden layer with sigmoids (ς) as activation functions and an output layer containing a single neuron with a linear transfer function (Fig. 10).

The data including degradation processes generated using the simulator are obtained using heavy workloads scenario. These extreme workloads are used to train and validate the NAR model offline. Then, these pre-trained models will be used to predict *HI* values on a real degradation profile online. During this process, the models will also be adapted online according to the profile of the degradation to generate accurate predictions. The prediction model is updated each time the absolute value of the *HI* estimation error exceeds a predefined threshold th_{HI}. This process is described in the flowchart in Fig. 11.

Table 4
$\widehat{HI}(k)$ NAR model performance.

Set	Results	$\widehat{HI}(k)$
Test set	MAE	2.3993×10^{-4}
	μ_ε	-1.6744×10^{-6}
	σ_ε	4.2155×10^{-4}
	max($_\varepsilon$)	0.0051
	MSE	1.7599×10^{-7}
	SSE	0.0250
	R	1
Prediction	MAE	9.3164×10^{-4}
	μ_ε	2.7813×10^{-4}
	σ_ε	$.15178 \times 10^{-4}$
	max($_\varepsilon$)	0.01605
	MSE	0.0023
	SSE	1.8031
	R	0.9994

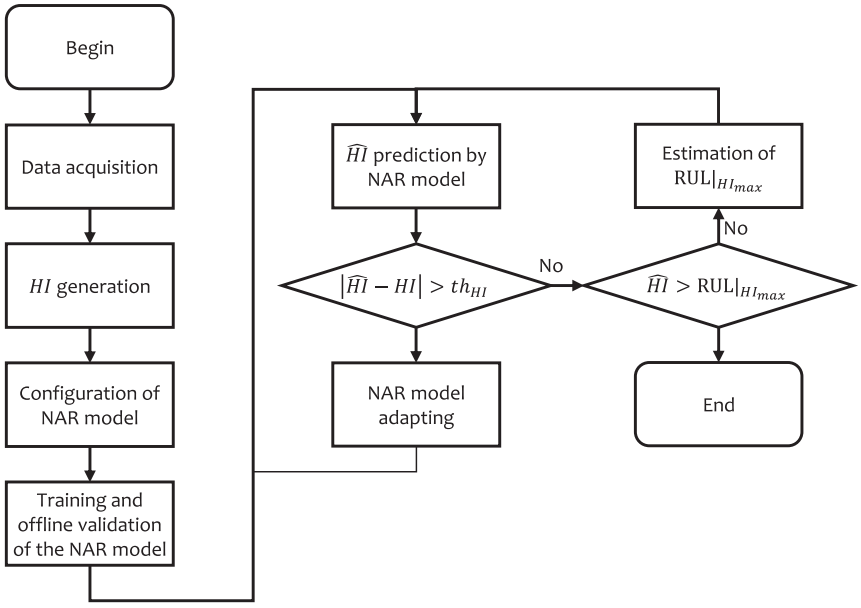

Fig. 11. Process of the forecasting and updating of *HI*, and the prediction of $\text{RUL}|_{HI_{max}}$.

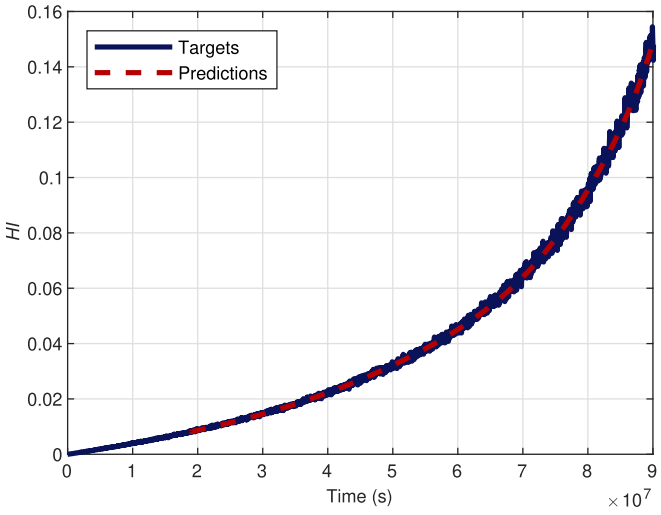

Fig. 12. NAR model predictions ($\widehat{HI}(k)$) against constructed values $HI(k)$.

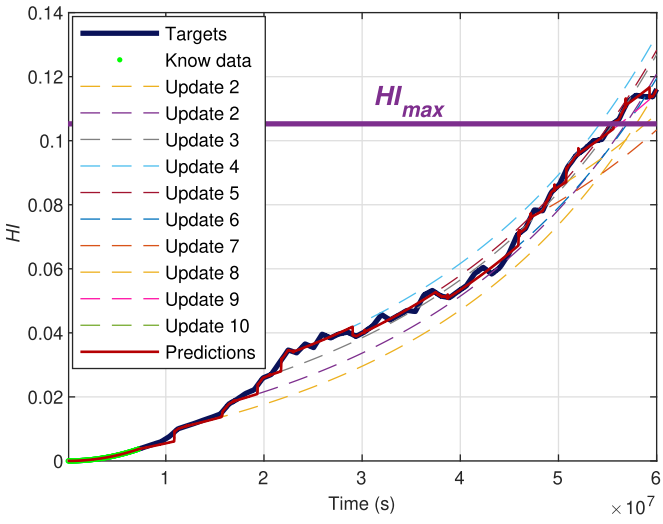

Fig. 13. Evolution of the *HI* trend through time, along with the predictions of the model and its updates.

8.2. NAR model offline validation

After subjecting the system to the accelerated wear process described in Section 7, the collected values of *HI* are then used as targets for the training and the validation of the NAR model. $HI(k)$ is then randomly divided into three sets: a training set (50%), a validation set (25%), and test set (25%). The models is built with 5 output delays (n=5), and 12 neurons in the hidden layer (m=12), and is trained with the Mean Squared Errors (MSE) as performance criterion.

Once the model is trained and validated, its performances are firstly evaluated with the test set. Then, to evaluate its prediction capabilities, it is fed with 3 months of *known horizon* data, and then left to predict $\widehat{HI}(k)$ until its values reach HI_{max}. The obtained results are summarized in Table 4.

The results of the online *HI* estimation are given in Fig. 12, which shows that the values of *HI* estimated by the NAR Model follow the values measured with a low estimation error with a maximum prediction error of 1.5% and a mean absolute error *MAE*=0.0931.

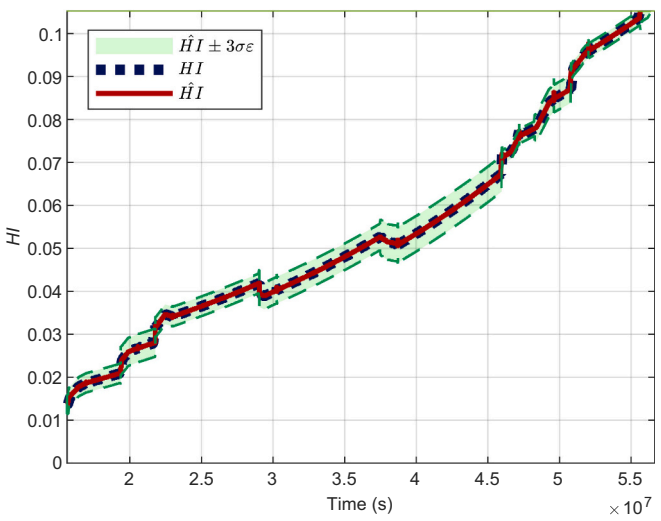

Fig. 14. Predictions of the Health Indicator ($\widehat{HI}$) enveloped by $\pm 3\sigma\varepsilon$, with $\varepsilon = HI - \widehat{HI}$.

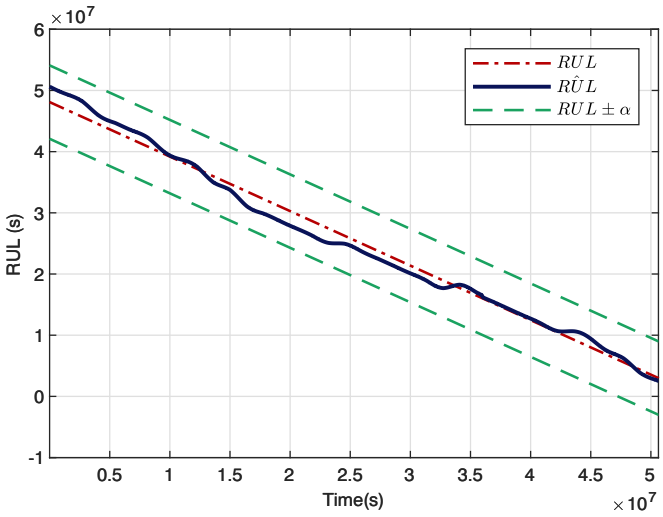

Fig. 15. Result of the PH metric applied to the case study of SoC, with a confidence interval α=0.1.

8.3. Online adapting and validation of the trend model

After validating the trend model in the scenario use by the manufacturer to compute the MTTF. In this paragraph, the NAR model is validated online on a scenario corresponding to actual use case of the SoC with frequency scaling.

The long-term prediction of the *HI* evolution and the updating of the prediction are shown in Fig. 13. This figure highlights the effect of the update on the correction of the trajectory of $\widehat{HI}$ which makes it possible to update the RUL prediction online. In fact, the degradation process is a function of changes in the usage of the system, commonly called Condition Monitoring, and the RUL changes accordingly. The result of Fig. 13, shows the capacity of the proposed method to adapt online the different trends of the degradation process.

The final results of the online *HI* prediction are shown in Fig. 14, where the predicted $\widehat{HI}$ is put in an envelope of $\pm 3\sigma$ of the prediction error $\varepsilon = HI - \widehat{HI}$. The narrowness of this envelope highlights the accuracy of the $\widehat{HI}$ predictions online, as it contains 99.54% of the errors, which is also confirmed by a *MAPE*=2.67%.

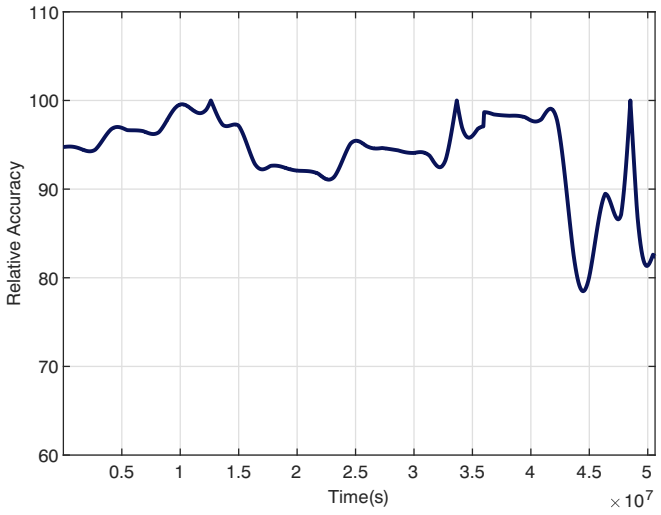

Fig. 16. The relative accuracy (RA) of the prediction of the studied SoC *RUL*.

8.4. RUL prediction and performance evaluation

In order to evaluate the accuracy and the performance of the proposed method, the Prognosis Horizon (*PH*) and the *Relative Accuracy* metrics are used [54,55]. The PH determines if the prediction performance meets the desired specifications, and it is calculated using the following as shown in Eq. (12).

$$PH(i) = EOP - i \tag{12}$$

It represents the difference between the actual time index (i) and the end of prediction time index (*EOP*). The latter is obtained when the prediction crosses the failure threshold.

To give the user an easily interpretable measurement tool of the confidence that can be given to the PH metric, another metric is proposed in Saxena et al. [54], where the accuracy is quantified according to the real RUL. This metric is called Relative Accuracy (*RA*) and expressed as follows:

$$RA(k) = 1 - \frac{\left|RUL(k) - \widehat{R}UL(k)\right|}{RUL(k)} \tag{13}$$

whereas $\widehat{R}UL$ is the predicted RUL. The value of RA ranges between [0,1], with the best score being closest to 1 (100%).

The results of evaluation the RUL of the SoC using these metrics, are given in the Figs. 15 and 16. Fig. 15 displays both the measured *RUL* and the predicted $\widehat{R}UL$. It also shows that by setting α=0.1, the *PH* is maximum since it is equal to the whole interval between the start of the prediction and the failure time. This high level of performance is confirmed by the calculation of the *RA* metric, which is close to 100% on most of the *PH*. The decrease in *RA* performance at the end of the prediction can be explained by the analysis of *RA* expression given in Eq. (13), which shows that when we approach the total failure time, the values of *RUL* and $\widehat{R}UL$ become small. So, the division by *RUL* increases the sensitivity of the metric to uncertainties.

9. Conclusion

In this paper, a method for predicting the failure of onboard SoCs is proposed to meet the practical need linked to the reliability of these systems, which are increasingly used in safety-critical equipment.

After a formal demonstration of the cause and effect link between the degradation process of SoC and their heating, the temperature drift is examined in this work as a health indicator. The trend of the drift of SoC temperature is then modeled by a NAR model updated when the estimation error of the real drift exceeds a predefined threshold. This online update makes it possible to adapt the prediction of the RUL to changes in the conditions of use of the SoC, and therefore guarantees a low prediction error.

The results obtained thanks to learning on simulation data and validation of the approach on an experimental profile, show the effectiveness of the proposed method. Hence, the use of temperature drift as a health indicator allows the deployment of this method on a wide range of SoC.

CRediT authorship contribution statement

Oussama Djedidi: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - Original Draft, Resources, Data Curation.

Mohand Djeziri: Conceptualization, Methodology, Formal analysis, Validation, Resources, Writing - Review & Editing, Supervision.

Samir Benmoussa: Methodology, Formal analysis, Validation.

Declaration of competing interest

The authors certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

References

[1] J. Srinivasan, S. V. Adve, P. Bose, J. Rivers, C.-K. Hu, RAMP: A Model for Reliability Aware MicroProcessor Design, IBM Research Report 23048.
[2] O. Djedidi, M.A. Djeziri, N.K. M'Sirdi, Data-driven approach for feature drift detection in embedded electronic devices, IFAC-PapersOnLine 51 (24) (2018) 1024–1029. ISSN 24058963, https://doi.org/10.1016/j.ifacol.2018.09.714.
[3] M. Djeziri, S. Benmoussa, E. Zio, Review of health indices extraction and trend modeling methods for remaining useful life estimation, in: M. Sayed-Mouchaweh (Ed.), Artificial Intelligence Techniques for a Scalable Energy Transition, Springer, 1 edn., ISBN 978-3-030-42725-2, VIII, 292, 2020.
[4] Z. Gao, C. Cecati, S.X. Ding, A survey of fault diagnosis and fault-tolerant techniques-part II: fault diagnosis with knowledge-based and hybrid/active approaches, IEEE Trans. Ind. Electron. 62 (6) (2015) 3768–3774. ISSN 02780046, https://doi.org/10.1109/TIE.2015.2419013.
[5] P. Baraldi, G. Bonfanti, E. Zio, Differential evolution-based multi-objective optimization for the definition of a health indicator for fault diagnostics and prognostics, Mech. Syst. Signal Process. 102 (2018) 382–400.
[6] F.D. Maio, F. Antonello, E. Zio, Condition-based probabilistic safety assessment of a spontaneous steam generator tube rupture accident scenario, Nucl. Eng. Des. 326 (2018) 41–54.
[7] M. Djeziri, T. B. L. Nguyen, S. Benmoussa, N. M'Sirdi, Fault prognosis based on physical and stochastic models, in: 2016 European Control Conference, ECC 2016, IEEE, ISBN 9781509025916, 2269–2274, doi:https://doi.org/10.1109/ECC.2016.7810629, URL http://ieeexplore.ieee.org/document/7810629/, 2017.
[8] M.A. Djeziri, S. Benmoussa, M.S. Mouchaweh, E. Lughoffer, Fault diagnosis and prognosis based on physical knowledge and reliability data: application to MOS field effect transistor, Microelectron. Reliab. 110 (2020) 113682.
[9] A. Steininger, Testing and built-in self-test – a survey, J. Syst. Archit. 46 (9) (2000) 721–747. ISSN 13837621, https://doi.org/10.1016/S1383-7621(99)00041-7.
[10] S. Deb, K.R. Pattipati, V. Raghavan, M. Shakeri, R. Shrestha, Multi-signal flow graphs: a novel approach for system testability analysis and fault diagnosis, IEEE Aerosp. Electron. Syst. Mag. 10 (5) (1995) 14–25. ISSN 08858985, https://doi.org/10.1109/62.373993.
[11] J. Sheppard, Maintaining diagnostic truth with information flow models, in: Conference Record. AUTOTESTCON '96, IEEE, ISBN 0-7803-3379-9, 447–454, doi: https://doi.org/10.1109/AUTEST.1996.547773, 1996.
[12] G. Zhang, Optimum Sensor Localization/Selection in a Diagnostic/Prognostic Architecture, University System of Georgia, Phd dissertation, 2005.
[13] Y. Cui, J. Shi, Z. Wang, An analytical model of electronic fault diagnosis on extension of the dependency theory, Reliability Engineering I& System Safety 133 (2015) 192–202. ISSN 09518320, https://doi.org/10.1016/j.ress.2014.09.015.
[14] D. Gizopoulos, M. Psarakis, S.V. Adve, P. Ramachandran, S.K.S. Hari, D. Sorin, A. Meixner, A. Biswas, X. Vera, Architectures for online error detection and recovery in multicore processors, 2011 Design, Automation I& Test in Europe (c) (2011) 1–6. ISSN 1530-1591, https://doi.org/10.1109/DATE.2011.5763096.

[15] N. Aggarwal, P. Ranganathan, Configurable isolation: building high availability systems with commodity multi-core processors, in: Acm Sigarch …, vol. 35, ACM Press, New York, New York, USA, ISBN 9781595937063, ISSN 0163-5964, 470–481, doi:https://doi.org/10.1145/1250662.1250720, 2007.

[16] C. LaFrieda, E. Ipek, J.F. Martínez, R. Manohar, Utilizing dynamically coupled cores to form a resilient chip multiprocessor, in: Proceedings of the International Conference on Dependable Systems and Networks, IEEE, ISBN 0769528554 (2007) 317–326, https://doi.org/10.1109/DSN.2007.100.

[17] S. S. Mukherjee, M. Kontz, S. K. Reinhardt, Detailed design and evaluation of redundant multi-threading alternatives, in: 29th Annual International Symposium on Computer Architecture, 2002. Proceedings, IEEE, ISBN 076951605X, ISSN 0163-5964, 99–110, doi:https://doi.org/10.1109/ISCA.2002.1003566, 2002.

[18] S. Shyam, K. Constantinides, S. Phadke, V. Bertacco, T. Austin, Ultra low-cost defect protection for microprocessor pipelines, in: Proceedings of the 12th international conference on Architectural support for programming languages and operating systems - ASPLOS-XII, vol. 41, ACM Press, New York, New York, USA, ISBN 1595934510, ISSN 01635980, 73, doi:https://doi.org/10.1145/1168857.1168868, 2006.

[19] A. Meixner, M.E. Bauer, D. Sorin, Argus: low-cost, comprehensive error detection in simple cores, in: 40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 2007), IEEE, ISBN 0-7695-3047-8, 210–222, 2007, https://doi.org/10.1109/MICRO.2007.18.

[20] M.-L. Li, P. Ramachandran, S.K. Sahoo, S.V. Adve, V.S. Adve, Y. Zhou, Understanding the propagation of hard errors to software and implications for resilient system design, SIGOPS Oper. Syst. Rev. 42 (2) (2008) 265–276. ISSN 0163-5980, https://doi.org/10.1145/1353535.1346315.

[21] A. Löfwenmark, S. Nadjm-Tehrani, Fault and timing analysis in critical multi-core systems: a survey with an avionics perspective, J. Syst. Archit. 87 (2018) 1–11. ISSN 13837621, https://doi.org/10.1016/j.sysarc.2018.04.001.

[22] Y. Jiang, Y. Wang, Y. Wu, Q. Sun, Fault prognostic of electronics based on optimal multi-order particle filter, Microelectronics Reliability 62 (2016) 167–177, ISSN 0026-2714, doi:https://doi.org/10.1016/j.microrel.2016.03.030, URL https://www.sciencedirect.com/science/article/pii/S0026271416300658.

[23] H. Zhang, Q. Miao, X. Zhang, Z. Liu, An improved unscented particle filter approach for lithium-ion battery remaining useful life prediction, Microelectronics Reliability 81 (2018) 288–298, ISSN 0026-2714, doi:https://doi.org/10.1016/j.microrel.2017.12.036, URL https://www.sciencedirect.com/science/article/pii/S0026271417306005.

[24] A. Al-Mohamad, G. Hoblos, V. Puig, A hybrid system-level prognostics approach with online RUL forecasting for electronics-rich systems with unknown degradation behaviors, Microelectron. Reliab. 111 (2020) 113676.

[25] Y. Sun, X. Hao, M. Pecht, Y. Zhou, Remaining useful life prediction for lithium-ion batteries based on an integrated health indicator, Microelectronics Reliability 88–90 (2018) 1189–1194, ISSN 0026-2714, doi:https://doi.org/10.1016/j.microrel.2018.07.047, URL https://www.sciencedirect.com/science/article/pii/S002627141830595X, 29th European Symposium on Reliability of Electron Devices, Failure Physics and Analysis (ESREF 2018).

[26] Q. Zhao, X. Qin, H. Zhao, W. Feng, A novel prediction method based on the support vector regression for the remaining useful life of lithium-ion batteries, Microelectronics Reliability 85 (2018) 99–108, ISSN 0026-2714, doi:https://doi.org/10.1016/j.microrel.2018.04.007, URL https://www.sciencedirect.com/science/article/pii/S0026271418301690.

[27] Y. Song, D. Liu, C. Yang, Y. Peng, Data-driven hybrid remaining useful life estimation approach for spacecraft lithium-ion battery, Microelectron. Reliab. 75 (2017) 142–153.

[28] M. Djeziri, S. Benmoussa, M. Benbouzid, Data-driven approach augmented in simulation for robust fault prognosis, Engineering Applications of Artificial Intelligence 86 (2019) 154–164, ISSN 0952-1976, doi:https://doi.org/10.1016/j.engappai.2019.09.002, URL https://www.sciencedirect.com/science/article/pii/S0952197619302180.

[29] NXP Inc., i.MX 6SoloX Applications Processors | Arm® Cortex®-A9, Cortex-M4 | NXP, URL https://www.nxp.com/products/processors-and-microcontrollers/applications-processors/i.mx-applications-processors/i.mx-6-processors/i.mx-6solox-processors-heterogeneous-processing-with-arm-cortex-a9-and-cortex-m4-cores:i.MX6SX, 2019.

[30] O. Djedidi, M. Djeziri, Incremental modeling and monitoring of embedded CPU-GPU chips, Processes 8 (6) (2020) 678, ISSN 2227-9717, doi:https://doi.org/10.3390/pr8060678, URL doi:https://doi.org/10.3390/pr8060678.

[31] O. Djedidi, Modélisation Incrémentale des Processeurs Embarqués pour l'Estimation des Caractéristiques et le Diagnostic, Theses, Aix-Marseille Université (AMU), URL https://tel.archives-ouvertes.fr/tel-02472080, 2019.

[32] C. C. Chen, S. Cha, T. Liu, L. Milor, System-level modeling of microprocessor reliability degradation due to BTI and HCI, in: IEEE International Reliability Physics Symposium Proceedings, IEEE, ISBN 9781479933167, ISSN 15417026, CA.8.1–CA.8.9, doi:https://doi.org/10.1109/IRPS.2014.6861125, 2014.

[33] C.C. Chen, L. Milor, Microprocessor aging analysis and reliability modeling due to back-end wearout mechanisms, IEEE Transactions on Very Large Scale Integration (VLSI) Systems 23 (10) (2015) 2065–2076. ISSN 10638210, https://doi.org/10.1109/TVLSI.2014.2357756.

[34] C.-C. Chen, L. Milor, System-level modeling and microprocessor reliability analysis for backend wearout mechanisms, in: Design, Automation I& Test in Europe ConferenceI & Exhibition (DATE), 2013, IEEE Conference Publications, New Jersey, 2013, pp. 1615–1620, https://doi.org/10.7873/DATE.2013.328 (ISBN 9781467350716).

[35] M. Altieri, S. Lesecq, D. Puschini, O. Heron, E. Beigne, J. Rodas, Evaluation and mitigation of aging effects on a digital on-chip voltage and temperature sensor, in: Proceedings - 2015 25th International Workshop on Power and Timing Modeling, Optimization and Simulation, PATMOS 2015, IEEE, 2015, pp. 111–117, https://doi.org/10.1109/PATMOS.2015.7347595 (ISBN 9781467394192).

[36] Temperature-dependent electromigration reliability, in: Y.-K. Cheng, C.-H. Tsai, C.-C. Teng, S.-M. S. Kang (Eds.), Electrothermal Analysis of VLSI Systems, Springer US, Boston, MA, ISBN 978-0-306-47024-0, 121–155, doi:https://doi.org/10.1007/0-306-47024-1_6, 2005.

[37] F. Oboril, M.B. Tahoori, Reducing wearout in embedded processors using proactive fine-grain dynamic runtime adaptation, in: Proceedings - 2012 17th IEEE European Test Symposium, ETS 2012, IEEE, 2012, pp. 1–6, https://doi.org/10.1109/ETS.2012.6233012 (ISBN 9781467306973).

[38] X. Chen, Y. Chen, M. Dong, C. Zhang, Demystifying Energy Usage in Smartphones, Design Automation Conference (DAC) (2014) 1–5ISSN 0738100X, doi:https://doi.org/10.1109/DAC.2014.6881397.

[39] E. Karl, D. Blaauw, D. Sylvester, T. Mudge, Reliability modeling and management in dynamic microprocessor-based systems, in: 2006 43rd ACM/IEEE Design Automation Conference, ISBN 0738-100X VO -, 1057–1060, 2006, https://doi.org/10.1145/1146909.1147174.

[40] K. Keller, K. Swearingen, J. Sheahan, M. Bailey, J. Dunsdon, K. W. Przytula, B. Jordan, Aircraft electrical power systems prognostics and health management, in: 2006 IEEE Aerospace Conference, 12 pp.–, 2006.

[41] K. Sharma, H. Krishna, Reliability bounds for some static system models using lifetime data on their components, Microelectronics Reliability 33 (8) (1993) 1081–1083, ISSN 0026-2714, doi:https://doi.org/10.1016/0026-2714(93)90334-U, URL http://www.sciencedirect.com/science/article/pii/002627149390334U.

[42] J. Lopes, J. Martins, A. Tavares, S. Pinto, DIHyper: providing lifetime hypervisor data integrity, in: 2018 IEEE 27th International Symposium on Industrial Electronics (ISIE), 2018, pp. 645–650, https://doi.org/10.1109/ISIE.2018.8433832 (ISSN 2163-5145).

[43] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, J. Lin, Machinery health prognostics: a systematic review from data acquisition to RUL prediction, Mechanical Systems and Signal Processing 104 (2018) 799–834, ISSN 0888-3270, doi:https://doi.org/10.1016/j.ymssp.2017.11.016, URL http://www.sciencedirect.com/science/article/pii/S0888327017305988.

[44] Freescale Semiconductor, i. MX 6SoloX Product Lifetime Usage Estimates, Tech. Rep., 2016.

[45] P. Wen, S. Zhao, S. Chen, Y. Li, A generalized remaining useful life prediction method for complex systems based on composite health indicator, Reliability Engineering & System Safety 205 (2021) 107241, ISSN 0951-8320, doi:https://doi.org/10.1016/j.ress.2020.107241, URL http://www.sciencedirect.com/science/article/pii/S0951832020307419.

[46] J. Feng, P. Kvam, Y. Tang, Remaining useful lifetime prediction based on the damage-marker bivariate degradation model: a case study on lithium-ion batteries used in electric vehicles, Eng. Fail. Anal. 70 (2016) 323–342. ISSN 1350-6307, https://doi.org/10.1016/J.ENGFAILANAL.2016.04.014.

[47] D. Brooks, M. Martonosi, Dynamic thermal management for high-performance microprocessors, in: HPCA: SEVENTH INTERNATIONAL SYMPOSIUM ON HIGH-PERFORMANCE COMPUTING ARCHITECTURE, PROCEEDINGS, IEEE COMPUTER SOC, NUEVO LEON, MEXICO, 171–182, 2001, https://doi.org/10.1109/HPCA.2001.903261.

[48] T. Li, G. Yu, J. Song, Minimizing energy by thermal-aware task assignment and speed scaling in heterogeneous MPSoC systems, J. Syst. Archit. 89 (2018) 118–130. ISSN 13837621, https://doi.org/10.1016/j.sysarc.2018.08.003.

[49] O. Djedidi, M. A. Djeziri, N. K. M'Sirdi, A. Naamane, Modular modelling of an embedded mobile CPU-GPU chip for feature estimation, in: Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics, vol. 1, SciTePress, Mardrid, Spain, ISBN 978-989-758-263-9, 338–345, doi:https://doi.org/10.5220/0006470803380345, 2017.

[50] Samsung, Samsung Opensource Release Center, URL http://opensource.samsung.com/, 2019.

[51] AnTuTu, AnTuTu Benchmark - Android Apps on Google Play, URL https://play.google.com/store/apps/details?id=com.antutu.ABenchMark, 2019.

[52] Futuremark Oy, 3DMark - The Gamer's Benchmark - Android Apps on Google Play, URL https://play.google.com/store/apps/details?id=com.futuremark.dmandroid.application, 2019.

[53] Freescale Semiconductor Inc, i.MX 6 Series Thermal Management Guidelines, URL https://www.nxp.com/docs/en/application-note/AN4579.pdf, 2012.

[54] A. Saxena, J. Celaya, B. Saha, S. Saha, K. Goebel, On applying the prognostic performance metrics, in: Annual Conference of the Prognostics and Health Management Society, 2009.

[55] S. Benmoussa, M. A. Djeziri, R. Sanshez, Support vector machine classification of current data for fault diagnosis and prognosis in wind turbine systems., Book Chapter Springer Nature Switzerland. Doi:https://doi.org/10.1007/978-3-030-42726-9-7 (2020) 157–182.