Citation information: DOI 10.1109/TIE.2016.2605622

A New Run-to-Run Approach for Reducing Contact Bounce in Electromagnetic Switches

Edgar Ramirez-Laboreo, *Student Member, IEEE,* Carlos Sagues, *Senior Member, IEEE,* and Sergio Llorente

***Abstract*—Contact bounce is probably the most undesirable phenomenon of electromagnetic switches. It reduces the performance of relays and contactors and is directly related to some of the processes that result in the destruction of the device. In this paper, a complete formulation of the problem is provided and a new strategy inspired by Run-to-Run control is presented for reducing contact bounce. The method, which makes use of the repetitive functioning of these systems, is highly versatile and may be applied to different switches under diverse operating conditions. In addition, it is able to deal with changes during the service life of the device, such as plastic deformations or the erosion of the contacts. Several experimental results are included to prove the effectiveness of the method.**

***Index Terms*—Contactors, contact bounce, electromechanical devices, pattern search, relays, run-to-run control, run-to-run optimization, switches.**

I. Introduction

ELECTROMAGNETIC switches, i.e., relays and contactors, offer some advantages with respect to solid state switches: they can conduct and block current in both directions, have very low energy losses, are generally cheaper and their activation mode is simpler. Additionally, while semiconductor devices can only provide single-pole single-throw (SPST) arrangements, relays and contactors with multiple-pole and multiple-throw designs (Fig. 1) are common and really useful in many applications. These advantageous features cause electromagnetic switches to be used in several present-day applications, e.g., battery chargers for electric vehicles [1], multiphase electric machines [2] or wireless power transfer devices [3]. Nevertheless, electromagnetic switches have also some drawbacks related to their operation mode. In particular, contact bounce, which is produced when the contacts hit together with an excess of energy, has been studied during the last 50 years [4], [5] and is probably the most known and undesirable. This phenomenon intensifies the mechanical wear of the contacts, prolongs unnecessarily the making process and increases the duration of electric arc and the probability of contact welding. In short, the apparition of contact bounce leads to an unavoidable reduction in the service life of these devices and the equipment in which they are embedded. Besides, additional problems of electromagnetic switches include, e.g., the acoustic noise generated in the impact of the contacts or the switching duration, which is four or five orders of magnitude longer than in solid state switches.

Many previous works have analyzed the behavior of these electromechanical devices and the origin of their related problems [6]–[8]. Much research has also been focused on eliminating or mitigating the aforementioned problems. One of the first works concerning the control of contact bounce [9] was based on the idea of reducing the kinetic energy of the moving contact by timing the coil energization process. Although this early approach was clearly rigid and non self-adaptive to changing conditions, it inspired several subsequent works. For instance, dos Santos Dias de Moraes and Perin [10] proposed a method which detects the start of the closing process and accordingly modifies the coil energization, and Lin *et al.* [11] presented an intelligent strategy that adjusts the coil voltage to reduce the magnetic force just before the impact. Other works have adopted a more control-oriented approach. The underlying idea is that a position controller of the moving contact together with a well-designed reference path would allow the device to switch faster, without bounces and with lower noise generation. In this regard, Carse *et al.* [12] already showed in simulation that a simple fuzzy controller could reduce contact bounce; nevertheless, the position and velocity of the mechanism are hard to be measured in practice. As a solution, Rahman *et al.* [13] used electrical variables to estimate the position of a solenoid actuator. Similar approaches have been used to control hydraulic valves [14] and, more recently, electromechanical contactors [15], [16].

In spite of the extensive literature on the field, the problem of contact bounce in electromagnetic switches is still not successfully solved. The already presented strategies use expert rules [10], intelligent algorithms [11] or nonlinear parameter estimators [15], all of which are strongly dependent on having a good knowledge of the system and its parameters. Consequently, if the device is subject to wear or drift, works under changing conditions or some of the parameters have not been properly identified, the effectiveness of any of these algorithms is considerably reduced. Besides, the large amount of different devices which are present in the market encourages the search of a more flexible or adaptive approach.

This paper explores the so-called Run-to-Run (R2R) algorithms [17], a type of techniques which have been mostly used

Manuscript received March 29, 2016; revised June 6, 2016; accepted July 6, 2016. This work was supported in part by the Ministerio de Economía y Competitividad, Gobierno de España - European Union, under project RTC-2014-1847-6 of subprogram Retos-Colaboración, and in part by the Ministerio de Educación, Cultura y Deporte, Gobierno de España, under Grant FPU14/04171.

E. Ramirez-Laboreo and C. Sagues are with the Departamento de Informatica e Ingenieria de Sistemas (DIIS) and the Instituto de Investigacion en Ingenieria de Aragon (I3A), Universidad de Zaragoza, Zaragoza 50018, Spain (e-mail: ramirlab@unizar.es, csagues@unizar.es).

S. Llorente is with the Research and Development Department, Induction Technology, Product Division Cookers, BSH Home Appliances Group, Zaragoza 50016, Spain (e-mail: sergio.llorente@bshg.com).

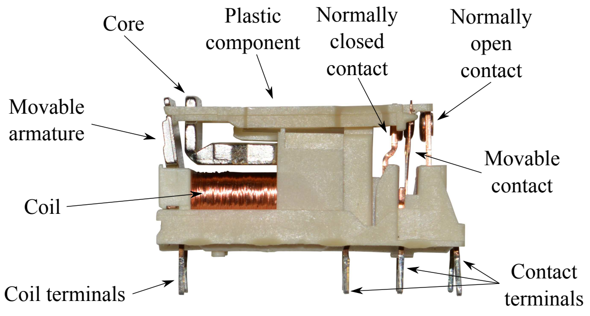


Fig. 1. Single-pole double-throw relay.

in the semiconductor industry for controlling the silicon wafer manufacturing process [18]. Nevertheless, they have been also utilized in the chemical industry [19] and, more recently, with medical purposes [20], obtaining satisfactory results in both fields. R2R techniques are especially indicated for systems that perform a process in a repetitive mode and only permit off-line measurements. As Srinivasan *et al.* explain in [21], off-line measurements include not only measurements taken at the end of the process but also variables obtained by post-processing experimental data recorded during the operation. This latter is the case of electromechanical switches, since some of their interest variables, e.g., the bounce duration, cannot be known until the making process has completely finished.

As a result, a new strategy inspired by R2R control is proposed in this paper for reducing contact bounce. The method does not require previous information about the system; therefore, it is highly versatile and can be applied to different electromagnetic switches in several operating conditions. Additionally, it is able to deal with changes in the system during its service life, such as those caused by wear or plastic deformation, by making use of the repetitive nature of these devices. With slight modifications, it may be used to minimize other variables besides contact bounce, such as the making duration or the acoustic noise generated during the switchings, and even be applied to other devices or processes characterized by a repetitive operation.

II. Problem formulation

A. Dynamic optimization of electromagnetic switches

The problem has been analyzed from the point of view of dynamic optimization. In this regard, note that electromagnetic switches, i.e., relays and contactors, are dynamic systems directed by a voltage input. Different voltage waveforms lead to different making processes with different results. The aim of the problem is to minimize a variable, namely the bounce duration, the noise level or the switching time, which can be valued by a scalar function that depends on the final state and/or the path followed by the system during its operation. Note also that electromagnetic switches work in a repetitive manner, i.e., they always perform the same tasks with the same objectives: closing and/or opening electrical connections. This means that the problem must be solved for each and every operation of the process. Besides, these devices usually allow for measuring some variables during the operation, so the problem falls into the field of measurement-based optimization. Assuming that measurements from previous operations can be stored and used to take decisions in the current one, the problem may be formulated as

$$\begin{aligned}
\underset{t_f{}^i,\ \boldsymbol{u}^i(t)}{\text{minimize}} \quad & J^i = \phi\big(\boldsymbol{x}^i(t_f{}^i)\big), && (1)\\
\text{subject to} \quad & \dot{\boldsymbol{x}}^i(t) = \boldsymbol{F}\big(\boldsymbol{x}^i(t), \boldsymbol{u}^i(t), \boldsymbol{d}^i(t)\big), && (2)\\
& \boldsymbol{x}^i(0) = \boldsymbol{x_0}, && (3)\\
& \boldsymbol{S}\big(\boldsymbol{u}^i(t), \boldsymbol{x}^i(t)\big) \leq \boldsymbol{0}, && (4)\\
& \boldsymbol{T}\big(\boldsymbol{x}^i(t_f{}^i)\big) \leq \boldsymbol{0}, && (5)\\
& \boldsymbol{y}^i(t) = \boldsymbol{H}\big(\boldsymbol{x}^i(t)\big) + \boldsymbol{v}^i(t), && (6)\\
\text{given} \quad & \boldsymbol{y}^j(kT_s),\ k = \big\{1, ..., N^j\big\},\ \forall j < i, && (7)
\end{aligned}$$

where $t_f{}^i$ is the final time, $\boldsymbol{u}^i(t)$ the input vector, J^i the scalar variable to minimize, $\boldsymbol{x}^i(t)$ the state vector, $\boldsymbol{d}^i(t)$ the disturbance vector, $\boldsymbol{y}^i(t)$ the output vector and $\boldsymbol{v}^i(t)$ the measurement additive noise vector of the ith operation, ϕ is the cost function, $\boldsymbol{F}$ the vector function that describes the system dynamics, $\boldsymbol{x_0}$ the initial state, $\boldsymbol{S}$ and $\boldsymbol{T}$ functions to define, respectively, path and final state constraints, $\boldsymbol{H}$ the output function, T_s the sampling interval and N^j the total number of samples taken during the jth operation. Note that (1) defines a cost which only depends on the final state. Nevertheless, if the scalar variable to minimize depends also on the path followed, this formulation can still be used by applying a simple transformation [22].

B. Looking for a solution. The Run-to-Run approach

Several strategies may have been considered to solve the problem. However, the repetitive operating nature of electromagnetic switches, which has never been exploited, encouraged the search of a new approach. This feature allows the operation of these devices to be analyzed as a batch process [21] and, together with the possibility of measuring some variables of the system, permits the use of certain *learning-type* control techniques [23].

Although switches always perform the same tasks, their behavior may differ between operations due to external disturbances or changes in the inputs. According to R2R fundamentals, all the controllable elements for each operation of the switch must be able to be determined by a finite set of decision variables. Thus, a finite-dimensional decision vector for the ith operation, $\boldsymbol{\nu}^i$, can be built. This vector can be modified between operations but cannot change during each one. Though all the decision variables are determined prior to the next operation, this does not strictly require that the inputs for the ith operation, $\boldsymbol{u}^i(t)$, be fully determined, since closed-loop controllers may exist to assure reference tracking and prevent from disturbances. Such implementation may benefit from the R2R algorithm by including the controller parameters in the decision vector. That way, references and control strategies would be optimized at once.

R2R methods evaluate the process at the end of the ith operation by means of a finite set of evaluation variables which may be grouped in a finite-dimensional evaluation vector, $\boldsymbol{\psi}^i$. Assume that $\boldsymbol{\psi}^i$ can be fully assessed from the value of the output vector, $\boldsymbol{y}^i$, at the final time of the ith operation, ${t_f}^i$. Then, the formulation including the R2R strategy becomes

$$\begin{aligned} \underset{\boldsymbol{\nu}^i}{\text{minimize}} \quad & J^i = \phi\big(\boldsymbol{x}^i({t_f}^i)\big), && (8)\\ \text{subject to} \quad & \text{(2)-(6)}, \\ & \boldsymbol{r}^i(t) = \boldsymbol{V}\big(t, \boldsymbol{\nu}^i\big), && (9)\\ & \boldsymbol{u}^i(t) = \boldsymbol{C}\big(\boldsymbol{\nu}^i, \boldsymbol{r}^i(t), \boldsymbol{y}^i(t)\big), && (10)\\ & \boldsymbol{\psi}^i = \boldsymbol{W}\big(\boldsymbol{y}^i({t_f}^i)\big), && (11)\\ \text{given} \quad & (7), \end{aligned}$$

where $\boldsymbol{\nu}^i$ is the decision vector, $\boldsymbol{\psi}^i$ the evaluation vector and $\boldsymbol{r}^i(t)$ the reference vector for the ith operation and $\boldsymbol{V}$, $\boldsymbol{C}$ and $\boldsymbol{W}$ are, respectively, the reference, control and evaluation functions. Like the previous formulation, this can be easily transformed so that the cost depends not only on the final state but also on the path followed by the system. This can be achieved, e.g., by including an integral term in $\boldsymbol{W}$,

$$\boldsymbol{\psi}^i = \boldsymbol{W}\left(\boldsymbol{y}^i({t_f}^i), \int_0^{{t_f}^i} \boldsymbol{L}\left(\boldsymbol{y}^i(t)\right) \mathrm{d}t\right), \qquad (12)$$

where $\boldsymbol{L}$ may be any vector function. As a complete overview, Fig. 2 shows a system controlled by a R2R method.

Despite the temporal dynamics, R2R methods analyze the process as a non-dynamical black-box system whose input and output are, respectively, the decision and evaluation vectors. Assume that the cost J^i can be calculated from $\boldsymbol{\psi}^i$, and assume also that $\boldsymbol{S}$ and $\boldsymbol{T}$ can be replaced by equivalent functions in terms of the decision and evaluation vectors. Then, from the point of view of the R2R algorithm, the problem is faced as

$$\begin{aligned} \underset{\boldsymbol{\nu}^i}{\text{minimize}} \quad & J^i = \eta\big(\boldsymbol{\psi}^i\big), && (13)\\ \text{subject to} \quad & \boldsymbol{\psi}^i = \boldsymbol{\mathcal{F}}\big(\boldsymbol{\nu}^i\big) + \boldsymbol{\mu}^i, && (14)\\ & \boldsymbol{\mathcal{S}}\big(\boldsymbol{\nu}^i, \boldsymbol{\psi}^i\big) \leq \boldsymbol{0}, && (15)\\ & \boldsymbol{\mathcal{T}}\big(\boldsymbol{\psi}^i\big) \leq \boldsymbol{0}, && (16)\\ \text{given} \quad & \boldsymbol{\psi}^j, \quad \forall j < i, && (17) \end{aligned}$$

where η is the new cost function, $\boldsymbol{\mathcal{F}}$ is the function that directly relates the decision and evaluation vectors, $\boldsymbol{\mu}^i$ includes the effects of disturbances and measurement noise during the ith operation, and $\boldsymbol{\mathcal{S}}$ and $\boldsymbol{\mathcal{T}}$ are constraint functions equivalent to $\boldsymbol{S}$ and $\boldsymbol{T}$. Although the problem (13)-(17) is raised from the perspective of optimization theory, the R2R approach is also suitable for control purposes only by selecting a proper cost function. Hence, if the goal is for the evaluation vector to reach a reference value $\boldsymbol{\psi_r}$, costs of the type $J^i = \left\|\boldsymbol{\psi}^i - \boldsymbol{\psi_r}\right\|_p$, where $\|\cdot\|_p$ means p-norm, may then be defined.

C. R2R methods on electromechanical switches

In the literature, R2R strategies are classified as explicit or implicit depending on whether they use or not a model of the system [21]. Therefore, selecting either an explicit or an implicit method depends greatly on the possibility of finding an accurate and invertible model for function $\boldsymbol{\mathcal{F}}$. Regarding the problem of electromagnetic switches, it is easy to see the difficulty of finding such a model which, in addition, should be general enough and adaptable to the great variety of devices in the market. In this sense, using a model-based method would probably require a specific model for any single relay or contactor, thus leading to the same problems that face the state-of-the-art algorithms. For these reasons, explicit R2R methods have been discarded and, consequently, the algorithm proposed in this paper is an implicit or model-free method.

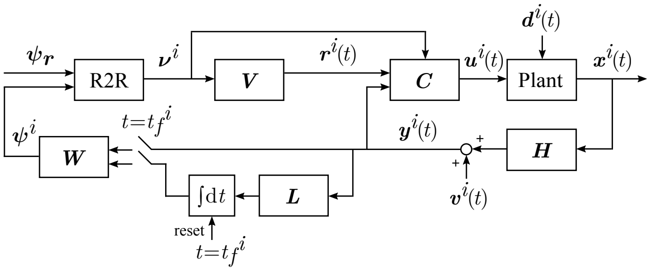


Fig. 2. General R2R control loop. The input of the plant for the ith operation, $\boldsymbol{u}^i$, is calculated by the closed-loop controller, $\boldsymbol{C}$, while the reference, $\boldsymbol{r}^i$, and the controller itself are set by the R2R algorithm.

In addition, there are some particular features of electromagnetic switches that should be taken into account when applying a R2R method. The main difference with respect to other processes where R2R control has been applied is that these devices usually do not perform a single repetitive process but two: the *making* or *closing* process and the *breaking* or *opening* process. Excluding special devices and regardless of the number of poles, every electromagnetic switch performs these two tasks: making when the coil is energized and breaking when it is de-energized. Specifically, double throw switches, in which every pole can be connected to either a normally open or a normally closed circuit, are of particular interest because both processes, i.e., making and breaking, may suffer from the problems related to contact bounce. The operation of electromagnetic switches may then be controlled by a single R2R algorithm or, more interestingly, by two separated but coordinated algorithms that optimize the two processes independently. However, note that making and breaking follow one another and are not completely independent because the initial conditions of the one match exactly the final state of the other. Given that the initial state should be constant (3), in this particular case it should also be guaranteed that the final state of both processes be constant or at least nearly constant. In this way both operations are made independent and the performance of the algorithm is likely to be improved.

III. Algorithm

The algorithm proposed in this paper may be defined as a direct-search R2R optimization method that considers the special features of electromagnetic switches (section II-C). Nevertheless, it is not limited to these devices and may be applied with slight modifications to any other dynamic system which performs a repetitive task. The method is influenced by the *Evolutionary Operation* described by G. Box in the 1950s [24], which consisted in introducing little deviations in the operation of a repetitive process to get more information of the system and, if possible, to improve its performance. The presented algorithm combines this concept with other features from present pattern-search methods [25], [26] to obtain a better performance. As already discussed, it is actually composed of two R2R subalgorithms which proceed alternatively, one for the making process and another one for the breaking process. The subscripts m and b are used in this and the following sections to differentiate the variables and functions specific to the making and breaking subalgorithms. The variables with no m or b are used equally by the two parts.

Although a closed-loop controller may be included in the process (see $\boldsymbol{C}$ in Fig. 2) and its parameters added to the decision vector, in this paper only the simpler case where no such type of controller exists is considered, i.e., $\boldsymbol{u}^i(t)=\boldsymbol{C}\big(\boldsymbol{\nu}^i,\boldsymbol{r}^i(t),\boldsymbol{y}^i(t)\big)=\boldsymbol{r}^i(t)=\boldsymbol{V}\big(t,\boldsymbol{\nu}^i\big)$. This configuration implies that the input of the ith operation, $\boldsymbol{u}^i(t)$, be fully defined by the decision vector $\boldsymbol{\nu}^i$. The switch has consequently no feedback during each operation but only between them, hence not being able to reject random disturbances. However, since the usual disturbances of electromechanical switches, i.e., changes in ambient temperature or mechanical wear, have very low dynamics compared to the one of the device, the algorithm will deal with them by modifying the input profile from one iteration to the next. The main advantage of this implementation is that few real-time calculations are needed, thus permitting the use of low-cost microcontrollers.

The method requires the following functions:

- $\boldsymbol{V_m}$ and $\boldsymbol{V_b}$, which define the input vectors of the ith making and the ith breaking processes, $\boldsymbol{u_m}^i(t)$ and $\boldsymbol{u_b}^i(t)$, as functions of the corresponding decision vectors, $\boldsymbol{\nu_m}^i\in\mathbb{R}^{n_m}$ and $\boldsymbol{\nu_b}^i\in\mathbb{R}^{n_b}$.
$$\boldsymbol{u_m}^i(t)=\boldsymbol{V_m}\big(t,\boldsymbol{\nu_m}^i\big)\qquad \boldsymbol{u_b}^i(t)=\boldsymbol{V_b}\big(t,\boldsymbol{\nu_b}^i\big)\tag{18}$$
As explained before, the algorithm needs the initial and final states of both making and breaking to be constant. For this reason, it is recommended that each input vector be composed of two temporal stages. First, a transient stage which depends on the decision vector and changes between operations, thus permitting the system to follow different paths and resulting in different evaluation vectors. And second, a constant stage not depending on the decision vector, i.e., always of the same value, and being long enough so that the system can reach the steady state.
- $\boldsymbol{W_m}$ and $\boldsymbol{W_b}$, which define the evaluation vectors of the ith making and the ith breaking processes, $\boldsymbol{\psi_m}^i$ and $\boldsymbol{\psi_b}^i$, as functions of the corresponding output vectors at the final time, $\boldsymbol{y_m}^i(t_{f_m}{}^i)$ and $\boldsymbol{y_b}^i(t_{f_b}{}^i)$.
$$\boldsymbol{\psi_m}^i=\boldsymbol{W_m}\big(\boldsymbol{y_m}^i(t_{f_m}{}^i)\big)\qquad \boldsymbol{\psi_b}^i=\boldsymbol{W_b}\big(\boldsymbol{y_b}^i(t_{f_b}{}^i)\big)\tag{19}$$
Remember that these functions may depend not only on the final state but also on the path (12). In this case, the functions $\boldsymbol{L_m}$ and $\boldsymbol{L_b}$ have also to be defined.
- η_m and η_b, that define, respectively, the costs $J_m{}^i$ and $J_b{}^i$ of the ith making and the ith breaking of the system.
$$J_m{}^i=\eta_m\big(\boldsymbol{\psi_m}^i\big),\qquad J_b{}^i=\eta_b\big(\boldsymbol{\psi_b}^i\big).\tag{20}$$

In addition to these functions, several parameters of the optimization algorithm have also to be determined:

- $\boldsymbol{M_m}\in\mathbb{R}^{n_m\times p_m}$ and $\boldsymbol{M_b}\in\mathbb{R}^{n_b\times p_b}$, meshes for the making and breaking algorithms. Their columns define the search directions around the current solutions. Hence, if $\boldsymbol{C_m}^i\in\mathbb{R}^{n_m\times p}$ and $\boldsymbol{C_b}^i\in\mathbb{R}^{n_b\times p}$ are the matrices whose columns define the candidate solutions for the ith making and the ith breaking, these are calculated as
$$\boldsymbol{C_m}^i=\boldsymbol{1_{p_m}}\otimes\boldsymbol{\nu_m}^i+\alpha_m{}^i\boldsymbol{M_m},\tag{21}$$
$$\boldsymbol{C_b}^i=\boldsymbol{1_{p_b}}\otimes\boldsymbol{\nu_b}^i+\alpha_b{}^i\boldsymbol{M_b},\tag{22}$$
where $\boldsymbol{1_{p_m}}\in\mathbb{R}^{p_m}$ and $\boldsymbol{1_{p_b}}\in\mathbb{R}^{p_b}$ are row vectors with all components equal to 1, $\otimes$ denotes the Kronecker product and $\alpha_m{}^i$ and $\alpha_b{}^i$ are the mesh size factors of the ith making and the ith breaking. Since each making process is followed by a breaking, the number of candidate points per iteration has to be equal, i.e., $p_m{=}p_b{=}p$. Meshes with a different number of vectors may also be used, e.g., by replicating some of the points of the smaller mesh. However, since the decision variables may have different dimensions and magnitudes, the use of scaled meshes is highly recommended. Given $\boldsymbol{\lambda_m}\in\mathbb{R}^{n_m}$ and $\boldsymbol{\lambda_b}\in\mathbb{R}^{n_b}$, vectors whose elements define characteristic lengths of the decision variables, several types of scaled meshes may be used, e.g., scaled-$2n$-meshes, $\boldsymbol{M_m}=\big[\mathrm{diag}(\boldsymbol{\lambda_m}),\ -\mathrm{diag}(\boldsymbol{\lambda_m})\big]$, $\boldsymbol{M_b}=\big[\mathrm{diag}(\boldsymbol{\lambda_b}),\ -\mathrm{diag}(\boldsymbol{\lambda_b})\big]$, or scaled-$n$+1-meshes, $\boldsymbol{M_m}=\big[\mathrm{diag}(\boldsymbol{\lambda_m}),\ -\boldsymbol{\lambda_m}\big]$, $\boldsymbol{M_b}=\big[\mathrm{diag}(\boldsymbol{\lambda_b}),\ -\boldsymbol{\lambda_b}\big]$, where $\mathrm{diag}(\boldsymbol{\lambda})$ is the square diagonal matrix with the elements of $\boldsymbol{\lambda}$ on the main diagonal. Since the two meshes must have the same number of vectors, the previous types of mesh lead to $n_m{=}n_b{=}n$, i.e., the decision vectors for the making and breaking algorithms must have also the same size. Otherwise some operations may not provide new information and the method would be underused.
- $\alpha_m{}^0$, $\alpha_b{}^0$, α_{mmin}, α_{bmin}, α_{mmax} and α_{bmax}, initial, minimum and maximum mesh size factors.
- ϵ_m and ϵ_b, mesh expansion factors, which multiply the mesh size factors when a new point is found. In the case that no better solution is found in the iteration, the meshes are contracted by multiplying by $1/\epsilon_m$ or $1/\epsilon_b$.
- $\boldsymbol{\nu_m}^0$ and $\boldsymbol{\nu_b}^0$, initial decision vectors.
- $\boldsymbol{\nu_{mmin}}$, $\boldsymbol{\nu_{bmin}}$, $\boldsymbol{\nu_{mmax}}$ and $\boldsymbol{\nu_{bmax}}$, lower and upper bounds for the decision vectors.

The algorithm is presented in three parts. Algorithm 1 is the function that is executed when a making order arises. It receives the decision vector for the making, executes the closing of the switch and computes the corresponding cost. Similarly, Algorithm 2 presents the analogous function for the breaking process. Finally, Algorithm 3 includes the set of instructions of the continuous R2R optimization. In this pseudocode, the operator $\mathrm{Col}_j(\cdot)$ refers to the jth column of a matrix. Note that each iteration evaluates the set of points defined by the meshes. Then, instead of moving directly to the best point found, the present point is re-evaluated so as to prevent from changes in the system between iterations. The total number of making-breaking cycles per iteration is therefore $p+1$. Note also that the algorithm has not been designed as an optimization process that ends when a good point is found, but to be executed during the normal operation of the device and continuously looking for better points.

Algorithm 1 Making Function

```
function MAKE($\boldsymbol{\nu_m}$)
    $\boldsymbol{u_m}(t) \leftarrow \boldsymbol{V_m}(t, \boldsymbol{\nu_m})$;    ▷ Prepare input
    Apply $\boldsymbol{u_m}(t)$ on the system and register $\boldsymbol{y_m}(t)$;
    $\boldsymbol{\psi_m} \leftarrow \boldsymbol{W_m}(\boldsymbol{y_m}(t_{f_m}))$;    ▷ Evaluate output
    $J_m \leftarrow \eta_m(\boldsymbol{\psi_m})$;    ▷ Compute cost
    return $J_m$
end function
```

Algorithm 2 Breaking Function

```
function BREAK($\boldsymbol{\nu_b}$)
    $\boldsymbol{u_b}(t) \leftarrow \boldsymbol{V_b}(t, \boldsymbol{\nu_b})$;    ▷ Prepare input
    Apply $\boldsymbol{u_b}(t)$ on the system and register $\boldsymbol{y_b}(t)$;
    $\boldsymbol{\psi_b} \leftarrow \boldsymbol{W_b}(\boldsymbol{y_b}(t_{f_b}))$;    ▷ Evaluate output
    $J_b \leftarrow \eta_b(\boldsymbol{\psi_b})$;    ▷ Compute cost
    return $J_b$
end function
```

Algorithm 3 Continuous R2R Optimization.

```
Require: $\boldsymbol{M_m}$, $\boldsymbol{M_b}$, $\boldsymbol{\nu_m}^0$, $\boldsymbol{\nu_b}^0$, $\boldsymbol{\nu_{mmin}}$, $\boldsymbol{\nu_{mmax}}$, $\boldsymbol{\nu_{bmin}}$,
    $\boldsymbol{\nu_{bmax}}$, $\alpha_m{}^0$, $\alpha_b{}^0$, $\alpha_{mmin}$, $\alpha_{mmax}$, $\alpha_{bmin}$, $\alpha_{bmax}$, $\epsilon_m$, $\epsilon_b$
loop
    Compute $\boldsymbol{C_m}{}^i$ and restrict cols. to $[\boldsymbol{\nu_{mmin}}, \boldsymbol{\nu_{mmax}}]$;
    Compute $\boldsymbol{C_b}{}^i$ and restrict cols. to $[\boldsymbol{\nu_{bmin}}, \boldsymbol{\nu_{bmax}}]$;
    for $j \leftarrow 1$ to $p$ do
        Wait for making order;
        $J_m{}^{i,j} \leftarrow$ MAKE($\mathrm{Col}_j(\boldsymbol{C_m}{}^i)$);
        Wait for breaking order;
        $J_b{}^{i,j} \leftarrow$ BREAK($\mathrm{Col}_j(\boldsymbol{C_b}{}^i)$);
    end for
    Wait for making order;
    $J_m{}^i \leftarrow$ MAKE($\boldsymbol{\nu_m}{}^i$);    ▷ Reevaluate current making
    Wait for breaking order;
    $J_b{}^i \leftarrow$ BREAK($\boldsymbol{\nu_b}{}^i$);    ▷ Reevaluate current breaking
    if $\exists q \ni J_m{}^{i,q} < J_m{}^i$ then    ▷ Making improved
        $\boldsymbol{\nu_m}{}^{i+1} \leftarrow \mathrm{Col}_q(\boldsymbol{C_m}{}^i)$;    ▷ Update decision vector
        $\alpha_m{}^{i+1} \leftarrow \epsilon_m \cdot \alpha_m{}^i$;    ▷ Expand mesh
    else    ▷ Making not improved
        $\boldsymbol{\nu_m}{}^{i+1} \leftarrow \boldsymbol{\nu_m}{}^i$;    ▷ Update decision vector
        $\alpha_m{}^{i+1} \leftarrow (1/\epsilon_m) \cdot \alpha_m{}^i$;    ▷ Contract mesh
    end if
    if $\exists q \ni J_b{}^{i,q} < J_b{}^i$ then    ▷ Breaking improved
        $\boldsymbol{\nu_b}{}^{i+1} \leftarrow \mathrm{Col}_q(\boldsymbol{C_b}{}^i)$;    ▷ Update decision vector
        $\alpha_b{}^{i+1} \leftarrow \epsilon_b \cdot \alpha_b{}^i$;    ▷ Expand mesh
    else    ▷ Breaking not improved
        $\boldsymbol{\nu_b}{}^{i+1} \leftarrow \boldsymbol{\nu_b}{}^i$;    ▷ Update decision vector
        $\alpha_b{}^{i+1} \leftarrow (1/\epsilon_b) \cdot \alpha_b{}^i$;    ▷ Contract mesh
    end if
    Constrain $\alpha_m{}^{i+1}$ to $[\alpha_{mmin},\ \alpha_{mmax}]$;
    Constrain $\alpha_b{}^{i+1}$ to $[\alpha_{bmin},\ \alpha_{bmax}]$;
    $i \leftarrow i + 1$;
end loop
```

IV. Experimental evaluation

A. Experimental setup. Inputs and outputs

The electromagnetic switch used in this work is a general purpose dc power relay that features a high electrical insulation between the activation and the power circuit (Fig. 1). It is single-pole double-throw (SPDT), i.e., the mechanism controls a single electrical switch having two paths for the current, a normally-closed one and a normally-open one. This particular model has been selected because it has a significant play that causes the motion of the armature and the movable contact to be decoupled in some stages. Since both components have partially independent motion dynamics, the estimation and control strategies for the position of the armature [15] are useless or hardly applicable in this relay.

The device is activated by an electronic circuit (Fig. 3). Two bipolar junction transistors and a resistor network permit the relay to be controlled by a digital activation signal, $a(t)$. A coil suppression circuit is included in the network to prevent from voltage spikes. A switch permits this circuit to be configured as a single free-wheeling general purpose diode (Configuration 1) or as a diode in series with a 47 V zener diode (Configuration 2). Besides the voltage across the coil, $v(t)$, which can be directly measured, a current-sensing resistor R_s permits also the measuring of the coil current $i(t)$. The power circuit, which is on the other side of the relay, is represented also in Fig. 3 as a power supply and two impedances, each connected to one of the power terminals. By measuring and processing the voltage across the contacts, two digital signals are obtained, $NC(t)$ for the normally closed connection and $NO(t)$ for the normally open connection, indicating whether they are open (0) or closed (1). Although the power circuit may be in practice any other type of electrical or electronic circuit, note that these latter variables are usually easily obtainable. Hence, according to the system described, the input and output vectors are defined as

$$\boldsymbol{u}(t) = a(t), \tag{23}$$

$$\boldsymbol{y}(t) = \begin{pmatrix} v(t) & i(t) & NC(t) & NO(t) \end{pmatrix}^{\mathrm{T}}. \tag{24}$$

An experimental setup (Fig. 4) has been designed and implemented looking for flexibility as primary criterion. While the activation and power circuits have been implemented on a printed circuit board, the relay is placed on a protoboard, thus permitting the test of different models with different footprints. The dc supply voltage is provided by a power supply unit, adjustable from 0 up to 60 V, covering the usual rated voltages of dc relays. Experimental measurements, i.e., voltages across the coil, the sensing resistor and the contacts, are taken by an eight-channel PicoScope 4824 USB oscilloscope and sent to a personal computer. The PC processes the data, calculates the activation signal and sends the command to the built-in waveform generator also included in the oscilloscope.

B. Decision and evaluation variables

As explained in section III, the algorithm presented in this paper does not include a usual closed-loop controller, i.e., the inputs of the system for the ith making and the subsequent breaking, $\boldsymbol{u_m}{}^i(t)$ and $\boldsymbol{u_b}{}^i(t)$, respectively, have to be completely described by the functions $\boldsymbol{V_m}$ and $\boldsymbol{V_b}$ according to the

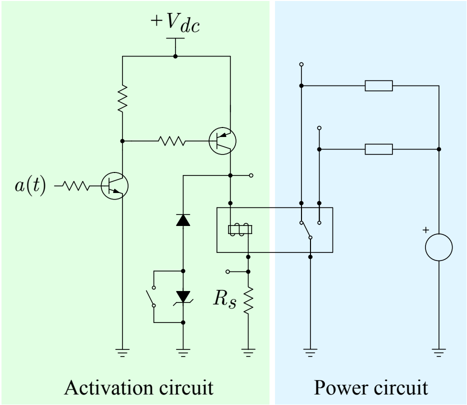


Fig. 3. Activation and power circuits diagram. The electromagnetic switch links both circuits with no electrical path between them.

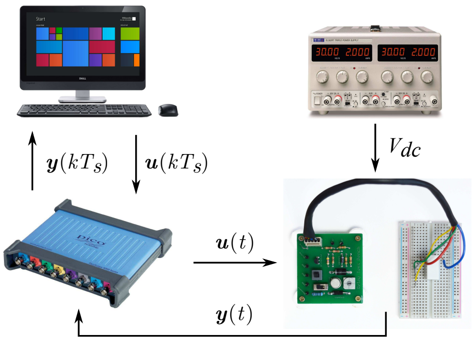
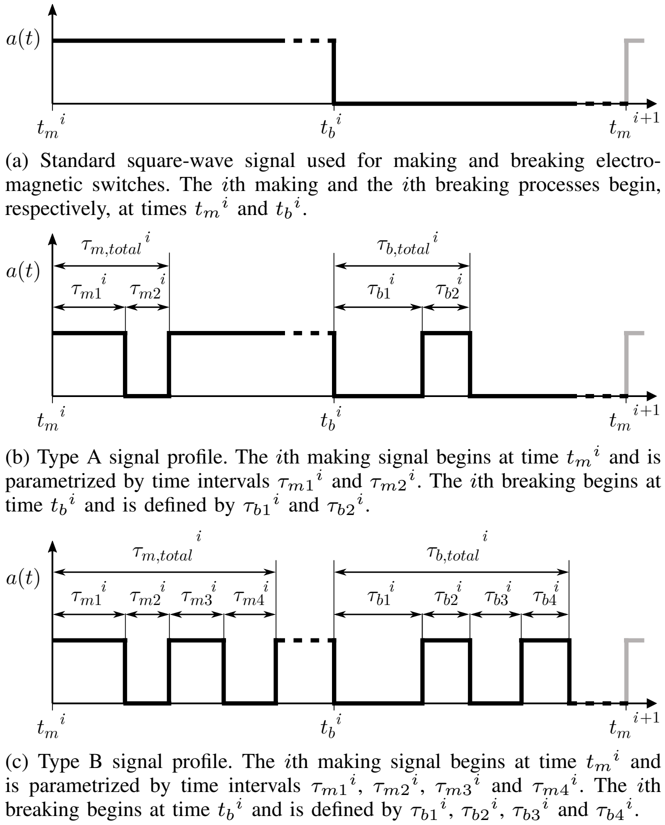


Fig. 4. Experimental setup.



(a) Standard square-wave signal used for making and breaking electromagnetic switches. The ith making and the ith breaking processes begin, respectively, at times $t_m{}^i$ and $t_b{}^i$.

(b) Type A signal profile. The ith making signal begins at time $t_m{}^i$ and is parametrized by time intervals $\tau_{m1}{}^i$ and $\tau_{m2}{}^i$. The ith breaking begins at time $t_b{}^i$ and is defined by $\tau_{b1}{}^i$ and $\tau_{b2}{}^i$.

(c) Type B signal profile. The ith making signal begins at time $t_m{}^i$ and is parametrized by time intervals $\tau_{m1}{}^i$, $\tau_{m2}{}^i$, $\tau_{m3}{}^i$ and $\tau_{m4}{}^i$. The ith breaking begins at time $t_b{}^i$ and is defined by $\tau_{b1}{}^i$, $\tau_{b2}{}^i$, $\tau_{b3}{}^i$ and $\tau_{b4}{}^i$.

Fig. 5. Activation signals.

decision vectors $\boldsymbol{\nu_m}^i$ and $\boldsymbol{\nu_b}^i$. Considering the system under study (Fig. 3), this means that it is necessary to define the time-dependent making and breaking profiles for signal $a(t)$, which is the only input, as well as the parametrization of these profiles according to two sets of decision variables. Since the performance of the whole control strategy depends greatly on the selection of these profiles, two different alternatives have been considered for the validation of the algorithm. Both are based on the square signal standardly used for making and breaking electromagnetic switches (Fig. 5a). The signal profile of type A (Fig. 5b), which is the simplest variant of the standard signal, depends on two decision variables for the making and another two for the breaking. Hence, the decision vectors when using this type of signal are defined as $\boldsymbol{\nu_m}^i = \left(\tau_{m1}{}^i\ \tau_{m2}{}^i\right)^{\mathrm{T}}$ and $\boldsymbol{\nu_b}^i = \left(\tau_{b1}{}^i\ \tau_{b2}{}^i\right)^{\mathrm{T}}$. On the other hand, the type B signal profile (Fig. 5c), which is more flexible, is described by four decision variables for the making and another four for the breaking. In this case, the four-dimensional decision vectors are defined as $\boldsymbol{\nu_m}^i = \left(\tau_{m1}{}^i\ \tau_{m2}{}^i\ \tau_{m3}{}^i\ \tau_{m4}{}^i\right)^{\mathrm{T}}$, and $\boldsymbol{\nu_b}^i = \left(\tau_{b1}{}^i\ \tau_{b2}{}^i\ \tau_{b3}{}^i\ \tau_{b4}{}^i\right)^{\mathrm{T}}$.

Note that both types of signals, either for making or for breaking, are characterized by two temporal stages which correspond to those described in section III. Firstly, an adjustable and transient stage which lasts $\tau_{m,total}{}^i$ for the ith making and $\tau_{b,total}{}^i$ for the complementary breaking. And secondly, a constant stage which lasts indefinitely until the following process is started. Recall that these constant stages ensure that the final state of each making or breaking and, consequently, the initial state of the following process, has always a similar value. Furthermore, given the system under study, they are also necessary to guarantee that the switch reaches and maintains the desired final position, either it is open or closed.

Regarding the outputs, only the measurements of the two contacts are used by the algorithm. In particular, the signal from the normally open connection, $NO(t)$, is used by $\boldsymbol{W_m}$ to evaluate the duration of bounces during the ith making process, $\beta_m{}^i$, while the signal from the normally closed connection, $NC(t)$, is used by $\boldsymbol{W_b}$ for calculating the duration of bounces during the ith breaking process, $\beta_b{}^i$. Note that functions $\boldsymbol{W_m}$ and $\boldsymbol{W_b}$ only have to compute the time between the first and the last rising edge of their respective input signals (see Fig. 6). The evaluation vectors for the two processes are therefore one-dimensional, $\boldsymbol{\psi_m}^i = \beta_m{}^i$ and $\boldsymbol{\psi_b}^i = \beta_b{}^i$. Finally, the costs have been directly defined as $J_m{}^i = \beta_m{}^i$ and $J_b{}^i = \beta_b{}^i$, i.e., the purpose of the algorithm is to reduce the duration of bounces in both making and breaking.

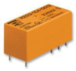
C. Parameters of the algorithm

The values of the parameters related to the decision vectors, which are different depending on which signal is used, are presented in Table I. The upper and lower bounds have been selected considering the characteristic times of a standard switching process (Fig. 6) and providing sufficiently wide margins. Then, for simplicity, the initial decision vectors have been defined as the midpoint between these bounds, and the characteristic length vector has been selected as the distance between this point and each of the limits. Scaled-$2n$-meshes

TABLE I
PARAMETERS USED IN THE ALGORITHM. SPECIFIC.

Parameter	Value (Signal A)	Value (Signal B)
$\boldsymbol{\nu_{mmin}}$	$(1\ \ 0)^{\mathrm{T}}$ ms	$(1\ \ 0\ \ 0\ \ 0)^{\mathrm{T}}$ ms
$\boldsymbol{\nu_{mmax}}$	$(7.5\ \ 2.5)^{\mathrm{T}}$ ms	$(7.5\ \ 2.5\ \ 2.5\ \ 2.5)^{\mathrm{T}}$ ms
$\boldsymbol{\nu_m}^0$	$(4.25\ \ 1.25)^{\mathrm{T}}$ ms	$(4.25\ \ 1.25\ \ 1.25\ \ 1.25)^{\mathrm{T}}$ ms
$\boldsymbol{\lambda_m}$	$(3.25\ \ 1.25)^{\mathrm{T}}$ ms	$(3.25\ \ 1.25\ \ 1.25\ \ 1.25)^{\mathrm{T}}$ ms
$\boldsymbol{\nu_{bmin}}$	$(2\ \ 0)^{\mathrm{T}}$ ms	$(2\ \ 0\ \ 0\ \ 0)^{\mathrm{T}}$ ms
$\boldsymbol{\nu_{bmax}}$	$(15\ \ 5)^{\mathrm{T}}$ ms	$(15\ \ 5\ \ 5\ \ 5)^{\mathrm{T}}$ ms
$\boldsymbol{\nu_b}^0$	$(8.5\ \ 2.5)^{\mathrm{T}}$ ms	$(8.5\ \ 2.5\ \ 2.5\ \ 2.5)^{\mathrm{T}}$ ms
$\boldsymbol{\lambda_b}$	$(6.5\ \ 2.5)^{\mathrm{T}}$ ms	$(6.5\ \ 2.5\ \ 2.5\ \ 2.5)^{\mathrm{T}}$ ms

TABLE II
CONTACT BOUNCE DURATION. AVERAGE RESULTS OF 50 OPERATIONS.

	**Configuration 1**		**Configuration 2**	
	Making	**Breaking**	**Making**	**Breaking**
	β_m (ms)	β_b (ms)	β_m (ms)	β_b (ms)
Square signal	1.791	3.228	1.792	4.576
Type A signal*	0.144	0.324	0.293	0.909
Reduction	*91.96%*	*89.97%*	*83.64%*	*80.13%*
Type B signal*	0.281	0.291	0.516	0.452
Reduction	*84.32%*	*90.97%*	*71.19%*	*90.12%*

*Best signal found, for each circuit configuration, in the first 25 iterations.

have been used in this evaluation and, consequently, the making and breaking meshes when using the Type A and the Type B signals have, respectively $p = 4$ and $p = 8$ vectors.

On the other hand, the parameters related to the meshes are independent of the signal used and equal for the making and breaking subalgorithms. Both meshes are initialized at the maximum size, i.e., ${\alpha_m}^0 = {\alpha_b}^0 = \alpha_{mmax} = \alpha_{bmax} = 1$, and can be halved in successive iterations, $\epsilon_m = \epsilon_b = 2$, to minimum mesh size factors of $\alpha_{mmin} = \alpha_{bmin} = 2^{-10}$, approximately a thousandth of the original size.

D. Results and discussion

Given the two configurations of the circuit (section IV-A), and the two activation signals proposed (section IV-B), a total of four evaluations of the algorithm have been performed. In all of them, the supply voltage V_{dc} has been set to the nominal value of the relay, which is 24 V. Table II presents, for the two circuit configurations, the average contact bounce duration when applying the standard square signal and the best Type A and Type B signals found after 25 iterations. Although the results may be improved if further iterations are performed, they show that the presented strategy is highly effective. The contact bounce reduction with respect to the standard square-wave activation, for both the making and breaking operations, is in all the cases above the 70%. In some of them, it even reaches the 90%, which is far superior to the results obtained by the already presented strategies [10]. Figs. 7 and 8 show, for the circuit configuration 1, a making and a breaking operation when applying, respectively, the best Type A and B signals. The contact bounce reduction with respect to the standard activation (Fig. 6) can be clearly seen. In these figures, the change in the values of the parameters can also be noted. For instance, for the optimization of signal A, the initial parameters ${\tau_{m1}}^0 = 4.25$ ms, ${\tau_{m2}}^0 = 1.25$ ms, ${\tau_{b1}}^0 = 8.5$ ms, and ${\tau_{b2}}^0 = 2.5$ ms, take at the end of the 25th iteration the values ${\tau_{m1}}^{25} = 3.69$ ms, ${\tau_{m2}}^{25} = 0.48$ ms, ${\tau_{b1}}^{25} = 7.85$ ms, and ${\tau_{b2}}^{25} = 0.44$ ms. As an additional remark, the designer must consider that contact bounce might be reduced at the expense of a longer switching time, as it happens with the Type B making signal in Fig. 8.

The performance of the algorithm along the iterations has also been analyzed. Figs. 9 and 10 show the contact bounce duration of all the making and breaking operations carried out by the algorithm when using, respectively, the Type A and the Type B signals. Note that, as previously stated, the algorithm performs a total of $p + 1$ evaluations per iteration, i.e., 5 when using the Type A signal and 9 when using the Type B signal. As expected, the contact bounce duration for the best points (green lines) has a decreasing trend along the iterations, i.e., the algorithm works properly from the optimization perspective. More interesting is the fact that the average and the worst points (yellow and red lines, respectively) have also an improving trend. This means that the algorithm not only is able to find better points from one iteration to another, but also to move the complete set of points to better regions which, in addition, are increasingly closer to the optimum. In this respect, note that, although some of the first operations have a long contact bounce duration, there exists an iteration from which all the making and breaking operations are better than the standard square-wave activation (black horizontal line). Besides, this initial stage may be reduced, or even eliminated, if the decision vectors are properly initialized, e.g., by means of an analytical energy-based analysis [27] or using results from a previous laboratory execution of the algorithm.

A final issue is the choice of the best type of signal for reducing contact bounce. Although no one can be considered completely superior, each performs better in a particular aspect. On the one hand, the Type A signal, which is simpler and depends on less parameters, moves under the level of the square signal in less experimental evaluations. On the other hand, the Type B signal, which is more flexible but depends on more parameters, needs more evaluations but may potentially reach higher bounce reductions. Hence, there is a trade-off between potential bounce reduction and speed of convergence, and the selection should be made according to the application.

V. CONCLUSION

In this paper, a novel strategy for reducing contact bounce in electromagnetic switches has been proposed. The algorithm, which is inspired by R2R control theory, makes use of the repetitive operating mode of these devices and does not need any model of the system. Besides the algorithm, a complete formulation of the problem and some particularities regarding the application of R2R methods on electromagnetic switches have been presented. The results obtained are very good and show that the contact bounce duration can be reduced by more than 90%, which is far superior to what had been achieved by past approaches. In this regard, it is expected a great reduction in the erosion of the contacts and, consequently, an improvement in the reliability of the devices.

Although similar to an online optimization, the algorithm does not stop when a good solution is found, but it always re-

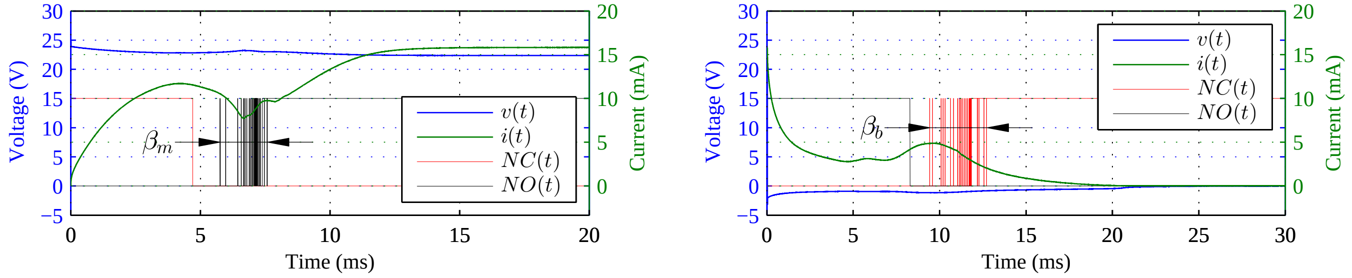


Fig. 6. Standard square-wave operations. Making (left) and breaking (right). Circuit configuration 1.

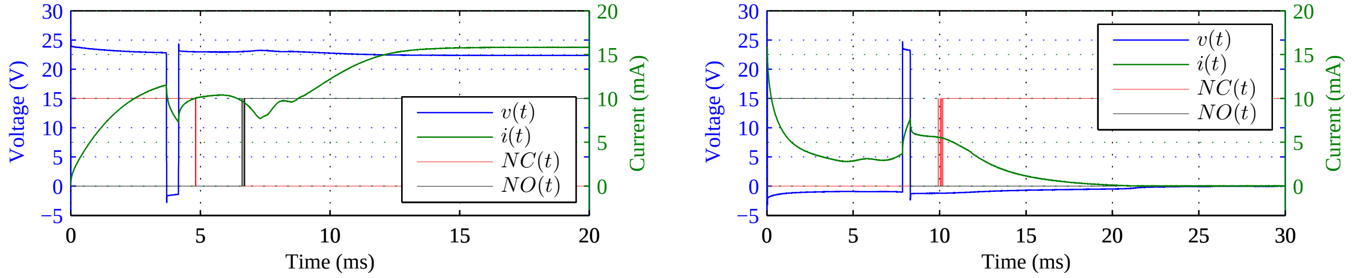


Fig. 7. Type A activation signal. Best making (left) and breaking (right) operations found in 25 iterations of the algorithm. Circuit configuration 1.

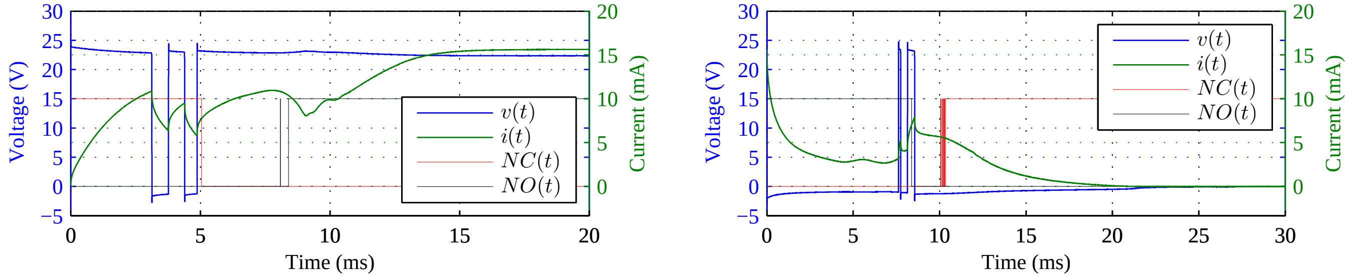
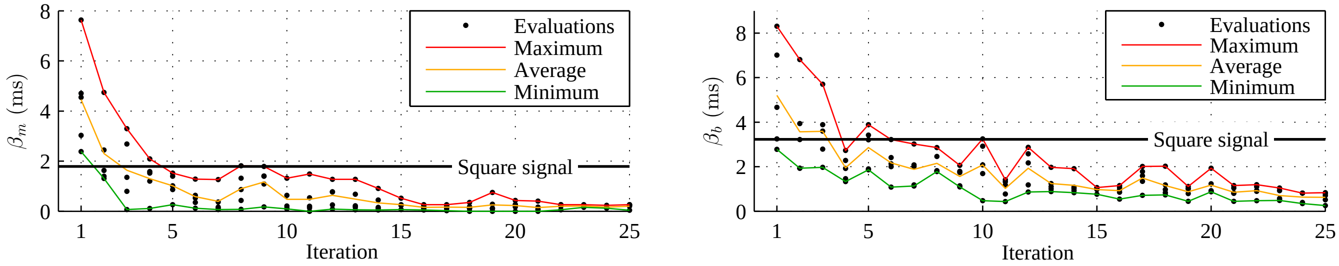


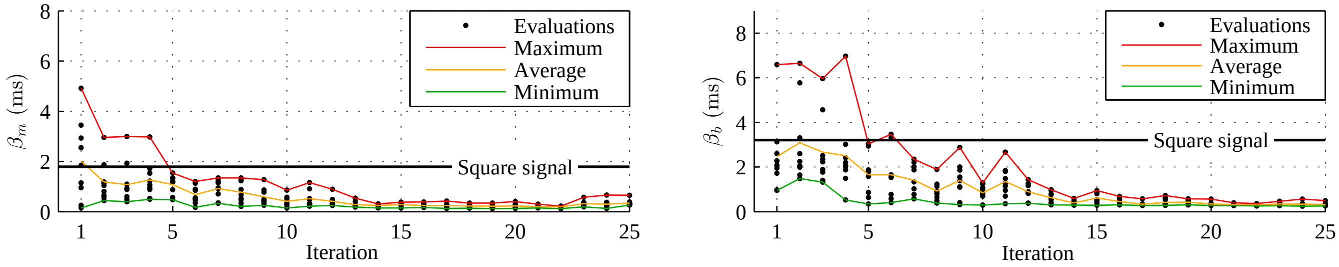
Fig. 8. Type B activation signal. Best making (left) and breaking (right) operations found in 25 iterations of the algorithm. Circuit configuration 1.



(a) Making optimization.

(b) Breaking optimization.

Fig. 9. Type A activation signal. Contact bounce duration in all the evaluations performed by the algorithm during the first 25 iterations. Circuit configuration 1.



(a) Making optimization.

(b) Breaking optimization.

Fig. 10. Type B activation signal. Contact bounce duration in all the evaluations performed by the algorithm during the first 25 iterations. Circuit configuration 1.

evaluates the current point and looks for better ones. Thus, it is able to adapt the activation signal when the system changes due, e.g., to wear, plastic deformations or contact erosion. In addition, since it does not require any model of the system, it obtains good results when using different activation signals or circuit configurations. Furthermore, it may even be applied to different types of switches including high-voltage ones, which are particularly affected by contact bounce due to the presence of high-energy electric arcs. As a result, the presented algorithm is more versatile and robust than past model-based approaches. Besides, it is computationally simple and requires few real-time calculations, so it is particularly well suited to be implemented in low-cost microcontrollers.

References

[1] S. Haghbin, S. Lundmark, M. Alaküla, and O. Carlson, "Grid-connected integrated battery chargers in vehicle applications: Review and new solution," *IEEE Trans. Ind. Electron.*, vol. 60, no. 2, pp. 459–473, Feb. 2013.

[2] S. Sadeghi, L. Guo, H. A. Toliyat, and L. Parsa, "Wide operational speed range of five-phase permanent magnet machines by using different stator winding configurations," *IEEE Trans. Ind. Electron.*, vol. 59, no. 6, pp. 2621–2631, Jun. 2012.

[3] T. C. Beh, M. Kato, T. Imura, S. Oh, and Y. Hori, "Automated impedance matching system for robust wireless power transfer via magnetic resonance coupling," *IEEE Trans. Ind. Electron.*, vol. 60, no. 9, pp. 3689–3698, Sep. 2013.

[4] P. Barkan, "A study of the contact bounce phenomenon," *IEEE Trans. Power App. Syst.*, no. 2, pp. 231–240, Feb. 1967.

[5] R. C. Tung, A. Fruehling, D. Peroulis, and A. Raman, "Multiple timescales and modeling of dynamic bounce phenomena in rf mems switches," *J. Microelectromech. Syst.*, vol. 23, no. 1, pp. 137–146, Feb. 2014.

[6] Z. Guofu, W. Qiya, Y. Wenying, and L. Huimin, "Permanent-magnet equivalent model of calculating relay's static attractive torque characteristics by finite element method," *IEEE Trans. Magn.*, vol. 48, no. 9, pp. 2467–2471, Sep. 2012.

[7] D. Wattiaux and O. Verlinden, "Modelling of the dynamic behaviour of electromechanical relays for the analysis of sensitivity to shocks and vibrations," *Exp. Mech.*, vol. 51, no. 9, pp. 1459–1472, Nov. 2011.

[8] E. Ramirez-Laboreo, C. Sagues, and S. Llorente, "A new model of electromechanical relays for predicting the motion and electromagnetic dynamics," *IEEE Trans. Ind. Appl.*, vol. 52, no. 3, pp. 2545–2553, May/Jun. 2016.

[9] T. S. Davies, H. Nouri, and F. W. Britton, "Towards the control of contact bounce," *IEEE Trans. Compon. Packag. Manuf. Technol.*, vol. 19, no. 3, pp. 353–359, Oct. 1996.

[10] P. M. dos Santos Dias de Moraes and A. J. Perin, "An electronic control unit for reducing contact bounce in electromagnetic contactors," *IEEE Trans. Ind. Electron.*, vol. 55, no. 2, pp. 861–870, Feb. 2008.

[11] H. Lin, X. Wang, S. Fang, P. Jin, and S. Ho, "Design, optimization, and intelligent control of permanent-magnet contactor," *IEEE Trans. Ind. Electron.*, vol. 60, no. 11, pp. 5148–5159, Nov. 2013.

[12] B. Carse, N. Larsen, H. Nouri, and T. Davies, "An approach to the reduction of contact bounce using fuzzy control," in *Proc. IEEE Int. Symp. Ind. Electron.*, vol. 3. IEEE, Jul. 1999, pp. 1025–1029.

[13] M. F. Rahman, N. C. Cheung, and K. W. Lim, "Position estimation in solenoid actuators," *IEEE Trans. Ind. Appl.*, vol. 32, no. 3, pp. 552–559, May/Jun. 1996.

[14] F. Malaguti, "Proportional control of on/off solenoid operated hydraulic valve by nonlinear robust controller," in *Proc. IEEE Int. Symp. Ind. Electron.*, vol. 2. IEEE, Jul. 2002, pp. 415–419.

[15] A. García Espinosa, J. R. Riba Ruiz, J. Cusidó, and X. Alabern Morera, "Sensorless control and fault diagnosis of electromechanical contactors," *IEEE Trans. Ind. Electron.*, vol. 55, no. 10, pp. 3742–3750, Oct. 2008.

[16] X. Wang, H. Lin, S. Ho, S. Fang, and P. Jin, "Analysis of dynamic characteristics of permanent magnet contactor with sensorless displacement profile control," *IEEE Trans. Magn.*, vol. 46, no. 6, pp. 1633–1636, Jun. 2010.

[17] E. Del Castillo and A. M. Hurwitz, "Run-to-run process control: literature review and extensions," *J. of Quality Technology*, vol. 29, no. 2, p. 184, Apr. 1997.

[18] S. W. Butler and J. A. Stefani, "Supervisory run-to-run control of polysilicon gate etch using in situ ellipsometry," *IEEE Trans. Semicond. Manuf.*, vol. 7, no. 2, pp. 193–201, May. 1994.

[19] T. L. Clarke-Pringle and J. F. MacGregor, "Optimization of molecular-weight distribution using batch-to-batch adjustments," *Ind. Eng. Chem. Res.*, vol. 37, no. 9, pp. 3660–3669, Jul. 1998.

[20] C. Owens, H. Zisser, L. Jovanovic, B. Srinivasan, D. Bonvin, and F. J. Doyle III, "Run-to-run control of blood glucose concentrations for people with type 1 diabetes mellitus," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 6, pp. 996–1005, Jun. 2006.

[21] B. Srinivasan, D. Bonvin, E. Visser, and S. Palanki, "Dynamic optimization of batch processes: Ii. role of measurements in handling uncertainty," *Comput. Chem. Eng.*, vol. 27, no. 1, pp. 27–44, Jan. 2003.

[22] B. Srinivasan, S. Palanki, and D. Bonvin, "Dynamic optimization of batch processes: I. characterization of the nominal solution," *Comput. Chem. Eng.*, vol. 27, no. 1, pp. 1–26, Jan. 2003.

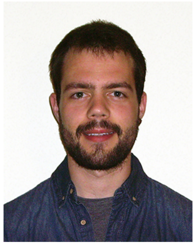
[23] Y. Wang, F. Gao, and F. J. Doyle, "Survey on iterative learning control, repetitive control, and run-to-run control," *J. Process Control*, vol. 19, no. 10, pp. 1589–1600, Dec. 2009.

[24] G. E. Box, "Evolutionary operation: A method for increasing industrial productivity," *J. R. Stat. Soc. Ser. C*, pp. 81–101, Jun. 1957.

[25] C. Audet and J. E. Dennis Jr, "Mesh adaptive direct search algorithms for constrained optimization," *SIAM J. Optimization*, vol. 17, no. 1, pp. 188–217, Jul. 2006.

[26] L. M. Rios and N. V. Sahinidis, "Derivative-free optimization: a review of algorithms and comparison of software implementations," *J. Global Optimization*, vol. 56, no. 3, pp. 1247–1293, Jul. 2013.

[27] C. Do, M. Lishchynska, M. Cychowski, K. Delaney, and M. Hill, "Energy-based approach to adaptive pulse shaping for control of rf-mems dc-contact switches," *J. Microelectromech. Syst.*, vol. 21, no. 6, pp. 1382–1391, Dec. 2012.



Edgar Ramirez-Laboreo (S'15) received the M.S. degrees in industrial engineering and in electronics engineering from the University of Zaragoza, Zaragoza, Spain, in 2014 and 2015, respectively, where he is currently working toward the Ph.D. degree in systems engineering and computer science.

Since 2013, he has been with the Departamento de Informatica e Ingenieria de Sistemas (DIIS) and with the Instituto de Investigacion en Ingenieria de Aragon (I3A), University of Zaragoza. His research interests include modeling and control of electromechanical devices, optimization, and Run-to-Run control.

Carlos Sagues (M'00–SM'11) received the M.S. degree in computer science and systems engineering and the Ph.D. degree in industrial engineering from the University of Zaragoza, Zaragoza, Spain, in 1989 and 1992, respectively.

In 1994 he joined, as an Associate Professor, the Departamento de Informatica e Ingenieria de Sistemas, University of Zaragoza, where he became a Full Professor in 2009, and was also the Head Teacher. He was engaged in research on force and infrared sensors for robots. His current research interests include control systems and industry applications, computer vision, visual control, and multivehicle cooperative control.

Sergio Llorente received the M.S. and Ph.D. degrees in electronics engineering from the University of Zaragoza, Zaragoza, Spain, in 2001 and 2016, respectively.

In 2001, he joined BSH Home Appliances Group in Zaragoza, Spain, where he has held different positions in the Research and Development Department of Induction Cooktops. He is currently in charge of several research lines and is an Inventor in more than 150 patents. He has also been an Assistant Professor with the University of Zaragoza since 2004. His research interests include power electronics, simulation and control algorithms for power electronics, and temperature control.