2021 18th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE).
Mexico City, Mexico. November 10-12, 2021

Parameter identification from hybrid model using PSO and penalty functions

1st Ricardo Cortez
UPIITA
Instituto Politecnico Nacional
Mexico City, Mexico
rcortezv@ipn.mx

2nd Yair Lozano
ESIME
Instituto Politecnico Nacional
CDMX, México
ylozanoh@ipn.mx

3rd Ruben Garrido
Departamento de Control Automatico
Cinvestav
Mexico City, Mexico
garrido@ctrl.cinvestav.mx

***Abstract*—This work studies the parameter identification of a hybrid model with Particle Swarm Optimization. A hybrid model is based on selection functions that allow the switching between simple mathematical expressions in order to describe a complex behavior. In this work two performance functions are proposed to perform the identification: The former considers a switching between functions on their structure. The latter implements function penalty functions in order to avoid the evaluation of the selection functions. These functions test for the parameter identification of a Shape Memory Alloy model under a numerical simulation. The quality of the computed estimates is tested using statistical tools to assure repeatability and to verify the influence of the performance functions.**

***Index Terms*—Parameter identification, Hysteresis, Particle Swarm Optimization, Hybrid model, Smart actuator, Shape Memory Alloy**

I. Introduction

The knowledge of the dynamical structure of a system is required to design control laws based on the model. Not only the mathematical expression that describes the system is necessary, the numerical values of the parameters in the model must be known in order to perform the tuning of the control law to assure a high quality tracking process. The process of estimating numerical values of the parameters for a system is called parameter identification. This type of algorithms has been widely studied on the automatic control area. One of the most applied approximations is the Least Square (LS) algorithm [1]. The main idea used on this type of identification consists of designing a regressor vector such that a linear quadratic function is minimized. Other well-known algorithm is the Gradient algorithm [2], which allows the estimation of the parameters for online tasks. It has been widely used for adaptive control problems. These algorithms are usually based on the system output. Even though other systems consider different signals such as the Closed-Loop Input Error (CLIE) [3], this one is similar to the gradient algorithms and compute the estimates using a set of differential equations.

The parameter identification process with meta-heuristic algorithms has been previously performed. Conversely to the standard algorithms mentioned before, these are based on an extensive test of solutions over a set of feasible solutions in order to minimize a performance function. This performance function does not necessary corresponds to a quadratic structure, however, such function is required to be positively defined. Some examples of parameter identification applying meta-heuristic algorithms are found in [4]–[7]. It must be mentioned that this type of algorithms simplify the data acquisition process, since their only require a measurable output in order to perform the task. The classical algorithms like the LS posses a regressor that contains several variables that are not always measurable and must be estimated using observers or numerical differentiators.

A hybrid model contained mathematical expressions that allow to switch between functions based on a conditional expression. Usually, this type of models are used to describe complex behaviors like the hysteresis effect on smart materials. The identification of hybrid models has been performed using classical approximations. On [8] an iterative algorithm has been applied to identify the parameters of a model that posses two cases, each one corresponds to a non-linear function. A similar case is show on the [9] where the model of a Wiener system is identified using an algorithm based on the LS. For this type of systems, more advanced tools has been used as the instrumental variable on [10]. A hydraulic servo-system with switching properties has been identified using the Particle Swarm Optimization (PSO) algorithm [11]. On [12] a parameter identification with the use of a neural network has been performed applied for reluctance motors. The robust nonlinear identification from a Boost DC-DC converter has been reported on [13] with the implementation from an algebraic parameter identification method. In the applied performance function based on the output error is supposed that exists a set of pre-parametrized parameters exists, which allows the inclusion of an additive penalty function based on the difference of the computed and the pre-parametrized parameters. Moreover, there exists a algorithm which has been designed to perform the identification of parameters on hysteresis models based on switching as described on [14], where the parameters of a blade's dynamic stall model are identified using a performance function that considers the square error.

The aim of this work is to make a comparison between the quality of the estimated parameters obtained using a classical performance index where exists a switching between the functions and a performance index that includes penalty functions to emulate the switching process. The proposed performance index allows the evaluation of all the cases that composing the hybrid model at the same time. However, the penalty functions makes the computed value for the functions (which does not correspond to the conditional expression), to quickly decrease until it is negligible.

The outline of this work is defined as follows: Section II includes the preliminaries about the hybrid model, the PSO algorithm and the problem statement. Section III describes the parameter identification problem using the hybrid model and describes the way to implement the penalty functions on the performance index. A study case based on the identification of parameters from a Shape Memory Alloy (SMA) spring model is developed on Section IV. Finally, on the Section V are included the concluding remarks of this work.

II. Preliminaries and problem statement

This section presents the hybrid model, the PSO and useful information used in this work. Also, the problem statement is provided.

A. Hybrid model

The model of phenomenons that possess a complex dynamic could be simplified by using a hybrid model. A hybrid model [15] can be described as,

$$\begin{array}{rcl} \dot{x} & = & f(x,t,u,\theta) \\ y & = & H(x,\dot{x},t,\theta) \end{array} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state vector, with n number of states and $y \in \mathbb{R}$ is the measurable output. The vector $\theta \in \mathbb{R}^m$ corresponds to the m parameters of the model that are supposed invariant over the time. The function $H(x, z, t, \theta)$ is composed by several mathematical functions that are switched between them based on a set of conditional functions such as:

$$H(x,\dot{x},t,\theta) = \begin{cases} h_1(x,t,\theta) & \text{when} \quad p_1(x,\dot{x}) \\ h_2(x,t,\theta) & \text{when} \quad p_2(x,\dot{x}) \\ \vdots & \vdots \qquad \vdots \\ h_s(x,t,\theta) & \text{when} \quad p_s(x,\dot{x}) \end{cases} \tag{2}$$

where $h_1, h_2, \ldots, h_s$ are the s piecewise functions that describes the behavior of the dynamical system output. At the same time, terms $p_1, p_2, \ldots, p_s$ are the s conditional expressions that defines the switching process between the functions h_i in order to emulate a complex dynamic.

B. Particle Swarm Optimization algorithm

A numerical optimization problem consists of the estimation of values from a vector of solutions $\zeta = [\zeta_1, \zeta \in ta_2, \ldots, \zeta_q] \in \Omega$ such as the minimum value of performance index $J(\cdot)$.

The term $\Omega \in \mathbb{R}^q$ is the set of feasible solutions and is defined as,

$$\begin{array}{c} \Omega := \{\zeta \in \mathbb{R}^q | \kappa_1(\zeta) \leq 0, \ldots, \kappa_{c_i}(\zeta) \leq 0, \\ \vartheta_1(\zeta) = 0, \ldots, \vartheta_{c_e}(\zeta) = 0\} \end{array} \tag{3}$$

where the functions $\kappa_i(\zeta)$ correspond to c_i inequality constraints and the functions $\vartheta_i(\zeta)$ are c_e equality constraints. These functions restricts the numerical values of the solutions that are considered as an feasible solution of the problem. Furthermore, the definition of Ω can be extended with the addition of bounds for each component ζ_i of the particle ζ considering upper and/or lower constraints, i.e. $\underline{\zeta}_i \leq x_i \leq \overline{\zeta}_i, i = 1, 2, 3$.

The optimization problem is solved once a optimal value ζ^* is computed. This value is defined as,

$$J(\zeta^*) \leq J(\zeta) \quad \forall \quad z \in \Omega \tag{4}$$

The Particle Swarm Optimization (PSO) algorithm [16] is an very common technique used to solve numerical optimization problems. This algorithm is based on the implementation of an exhaustive search from solutions over the set Ω during k iterations, in order to estimate the value of ζ that provides a minimum value of J. The way the solutions to be tested is selected is based on the behavior of a bird group when they are searching for resources.

The PSO algorithm considers that an vector ζ is the position of a particle on a q dimensional space constrained by Ω, in this space exists a ψ number of particles. The combination of all the particles on this space is named as a swarm. The main idea is that each particle possesses its own dynamical behavior based on its actual position and the position that previously provides a lower value of J. However, at the same time the particles on the swarm share information that modifies their dynamic [17] based on the position of the particle that posses the minimum value of J during the iteration k.

This dynamic is expressed as,

$$\begin{array}{rcl} \omega_i(k+1) & = & \omega_i(k) + \phi_1(k) P\left(\zeta_{i,*}(k) - \zeta_i(k)\right) \\ & & +\phi_2(k) L\left(\zeta_*(k) - \zeta_i(k)\right) \\ \zeta_i(k+1) & = & \zeta_i(k) + \omega_i(k+1) \end{array} \tag{5}$$

where ζ_i and ω_i are the particle position and velocity respectively. The terms P, L are positive constants and ϕ_1, ϕ_2 are random bounded signals on the interval $(0, 1]$. The term $\zeta_{i,*}(k)$ corresponds to the best position that has been obtained by the particle i during the iteration k and is defined as

$$\zeta_{*,i}(k) = \underset{s \in [1,\ldots,k]}{\operatorname{argmin}} \left\{ J\left(\zeta_i(s)\right) \right\} \tag{6}$$

On the other hand, the term ζ_* corresponds to the best solution obtained by the swarm until the iteration k,

$$\zeta_*(k) = \underset{g \in [1,\ldots,n]}{\operatorname{argmin}} \left\{ J\left(\zeta_g(k)\right) \right\} \tag{7}$$

A pair of stop parameters are defined. The first one is a lower bound σ for the performance index, and once that $J(\zeta_i(k)) < \sigma$ is reached, the algorithm stops its execution. This criteria is defined by the used based on the expected quality of the optimized value. The second stop parameter is k_{max} and it defines the maximum number of iterations that the algorithm executes. Once a stop criteria is fulfilled the value of ζ_* is taken as the optimal value ζ^*. The implementation of the PSO algorithm is shown in **Algorithm 1**.

Algorithm 1: Particle Swarm Optimization algorithm

Data: $J(\cdot)$ and Ω
Result: ζ_*
Generate random solutions $\zeta_i(1) \in \Omega \quad i = 1, \ldots, r$;
Evaluate the solutions $\zeta_i(1)$ using $J(\cdot)$;
Obtain the terms (pbest and gbest) $\zeta_{i,*}(1)$ and $\zeta_*(1)$;
Compute the the velocities $\omega_i(2)$ and the positions $\zeta_i(2)$;
Set $k = 2$;
while $k < k_{max}$ **do**
    Evaluate the solutions $\zeta_i(k)$ using $J(\cdot)$;
    Obtain the terms (pbest and gbest) $\zeta_{i,*}(k)$ and $\zeta_*(k)$;
    if $J(\zeta_*(k)) \geq \sigma$ **then**
        Compute the velocities $\omega_i(k)$ and the positions $\zeta_i(k)$;
        while $\zeta_i(k) \notin \Omega$ **do**
            Decrease the velocity as $\omega_i(k) = \frac{\omega_i(k)}{2}$;
            Compute a new position $\zeta_i(k)$ using $\omega_i(k)$;
        end
    else
        break;
    end
    $k = k + 1$;
end

C. Problem statement

The parameter identification problem using a meta-heuristic algorithm requires an extensive test of feasible solutions to verify their fitness on the performance function J. This be implement with use of a virtual model as [18],

$$\begin{aligned} \dot{x}_v &= f(x_v, t, u, \hat{\theta}) \\ y_v &= H(x_v, t, \hat{\theta}) \end{aligned} \tag{8}$$

where x_v is the states of the virtual model, y_v is the virtual output of the model, u is the input applied to the original system (1). The vector $\hat{\theta}$ contents the estimated parameters that are tested as solution by the PSO algorithm. The error between the virtual output and the measured output of the system is defined as $e = y_v - y$ in order to be used on J.

The implementation of this process offline requires the following steps:

- Data acquisition: A set of experiments must be performed in order to save the data on a file for the further comparison. The minimum data required is the input of the system u and the measurable output y.
- PSO optimization process: The PSO algorithm is implemented as illustrated on **Algorithm 1**. It must be noticed that the steps that evaluate the solutions on J includes a numerical simulation of the dynamical system in order to generate the virtual output y_v. The way the PSO is used as parameter identifier is shown in Figure 1.

The way the optimization procedure is performed depends on the performance function J. For this reason the way J is defined modifies the computed estimations of the parameters. Therefore, a feasible J must be proposed in order to enhance the estimated parameters and decrease the computational resources used to solve the identification problem.

III. Parameter identification of a hybrid model

The main procedure used to perform the parameter identification of a system is based on the implementation of a virtual model.

For the classical cases where the system does not possess a function that describes a hybrid behavior, the performance function to be applied could be selected as,

$$J(x_i) = \int_0^t |y(\tau) - y_v(\tau)| d\tau \tag{9}$$

However, when the hybrid function exists on the dynamic model a modified performance function can be defined like,

$$J(x_i) = \int_0^t \begin{cases} |y(\tau) - h_1(x, \tau, \theta)| d\tau & p_1(z) \\ |y(\tau) - h_2(x, \tau, \theta)| d\tau & p_2(z) \\ \vdots & \vdots \\ |y(\tau) - h_n(x, \tau, \theta)| d\tau & p_n(z) \end{cases} \tag{10}$$

It must be noticed that for each sampling time of the dynamical simulation the process of selection between functions must be performed in order to be capable of computing the integral value.

Therefore, there exists another suitable performance function to solve this problem with the addition of penalty functions as a weight for the functions that are included on $H(x_v, t, \hat{\theta})$ such that the performance function can be rewritten as,

$$J(x_i) = \int_0^t \sum_{j=1}^{n} \eta_j |y(\tau) - h_j(x, \tau, \theta)| d\tau \tag{11}$$

where $\eta_j \in \mathbb{R}^+$ is a positive the penalty function that decreases the value computed for the function p_j when the decision condition k_j is not fulfill. In other case, the value of the weight must be at least one or even greater.

IV. Case study: Force hysteresis model from a Shape Memory Alloy spring

In order to evaluate the performance of both functions for the case of a hybrid model, a numerical example using the mathematical model of temperature/force from a Shape Memory Alloy (SMA) [19] spring is performed. A SMA is a special type of material that posses the capability to return to a previously set shape when their temperature increases. However, when the temperature decreases, the alloy losses their stiffness and become malleable [20].

The dynamical model of this material is given by [19],

$$\begin{aligned} \dot{T} &= \alpha u^2 - \beta(T - T_a) \\ H &= \Gamma(T, \dot{T}) \end{aligned} \tag{12}$$

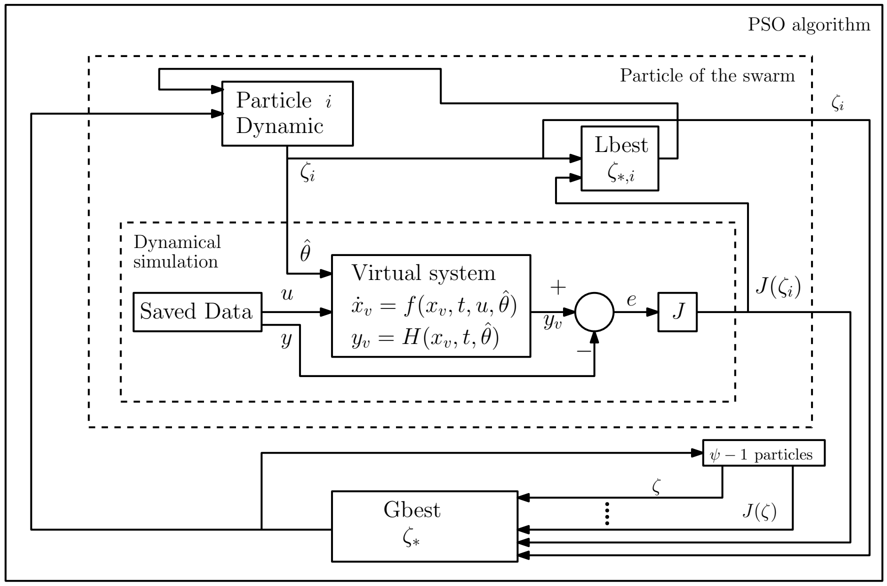

Fig. 1: Scheme of implementation of the PSO as parameter identifier. The dynamic of $\psi - 1$ particles is contained on the corresponding block to simplify the graphic representation.

where T and $\dot{T}$ are the temperature and their first derivative, respectively, T_a the temperature in the operational space. F is the output force generated by the motion of the springs and u is the current applied to the spring in order to increase the temperature. The parameters α and β are positive constants related to the specific heat coefficient and the heat dissipation, respectively. The function Γ corresponds to a description from a hybrid behavior, defined as,

$$\Gamma(T, \dot{T}) = \begin{cases} \dfrac{a_l}{1 + e^{-b_l(T-d_l)}} + c_l & \dot{T} < 0 \\ \dfrac{a_u}{1 + e^{-b_l(T-d_u)}} + c_u & \dot{T} \geq 0 \end{cases} \tag{13}$$

where the terms a_l, a_u defines the lower bound of their functions, b_l, b_u are the slope, c_l, c_u defines the upper bound and d_l, d_u translates the sigmoid over the r axis. The switching between both cases is given by the sign of the temperature derivative.

Such the performance function that switches between cases be defined as,

$$J_1 = \int_0^t \begin{cases} |H(\tau) - \dfrac{a_l}{1 + e^{-b_l(T_v(\tau)-d_l)}} + c_l| d\tau & \dot{T} < 0 \\ |H(\tau) - \dfrac{a_u}{1 + e^{-b_l(T_v(\tau)-d_u)}} + c_u| d\tau & \dot{T} \geq 0 \end{cases} \tag{14}$$

When the performance function uses a penalty function, this one is,

$$\begin{aligned} J_2 = \quad & \int_0^t \eta_1 |H(\tau) - \frac{a_l}{1 + e^{-b_l(T_v(\tau)-d_l)}} + c_l| \\ & + \eta_2 |H(\tau) - \tfrac{a_u}{1+e^{-b_l(T_v(\tau)-d_u)}} + c_u| d\tau \end{aligned} \tag{15}$$

where the term η_i is defined as,

$$\eta_i = \frac{1}{1 + e^{(-1)^i K_i \dot{T}_v}} \tag{16}$$

for $i = 1, 2$.

A. Simulation setup

The implementation has been performed using Matlab 2020b software on a 64-bit AMD Ryzen 5 1600X computer. The simulation SMA spring dynamic model has been performed using Simulink with a sampling time of $0.001s$ and the Runge-Kutta numerical integration method. The tuning of the PSO algorithm (5) has been realized with the IRACE tool [21], which is a package of the RStudio software. The obtained tuned values are $p = 1.2343$ and $l = 1.5846$.

The vector of solutions is given as,

$$x_i = [\alpha, \beta, a_l, a_u, b_l, b_u, c_l, c_u, d_l, d_u] \tag{17}$$

Based on the previous knowledge about the system given on [19], the following inequality constraints must be fulfilled,

$$\begin{aligned} g_1: & \quad \alpha > 0 \\ g_2: & \quad \beta > 0 \\ g_3: & \quad a_l > 0 \\ g_4: & \quad a_u > 0 \\ g_5: & \quad b_l > 0 \\ g_6: & \quad b_u > 0 \\ g_7: & \quad c_l > 0 \\ g_8: & \quad c_u > 0 \\ g_9: & \quad d_l > 0 \\ g_{10}: & \quad d_u > 0 \end{aligned} \tag{18}$$

Moreover, a set of upper bounds for all the parameters has been defined at 100 in order to constraint the search space. The weight terms on the penalty functions on the performance indexes are defined as $K_1, K_2 = 50$. Each case considers 20 particles on the PSO algorithm and has been executed 30 times in order to perform the statistical evaluation of their performance. The terms $k_{max} = 100$ and $\sigma = 0.001$ are set for the stop criteria.

The required time to perform the parameter identification is modified based on the performance function, when the function J_1 is applied on the PSO a total of $7774.82s$ are required to execute the 100 iterations of the algorithm. On the other hand, when J_2 is applied, the time required to execute the algorithm is only of $1630.43s$. It is clearly observed that the performance function J_2 requires less time to compute the parameter identification process than the case that uses J_1. This is at least four times faster to provide results.

The comparison of the median of the estimated parameters is shown in the Table I. The results shown that when the estimation process is performed using the performance function J_2 the parameters are closer to the real values on the simulation.

TABLE I: Median of the estimated parameters of the SMA model using a PSO algorithm

Parameter	Original	PSO with J_1	PSO with J_2
α	2	1.6023	2.1016
β	3	2.8212	3.2961
a_l	2.83	1.4798	2.9440
b_l	8.33	11.4932	8.7648
c_l	2	0.1875	2.0458
d_l	26	28.5339	25.6790
a_u	2.73	1.1697	2.3878
b_u	3.55	2.3459	3.2012
c_u	2.2	0.6175	2.3070
d_u	25.3	25.0295	25.1221

The mean of the performance function computed using the PSO algorithm and the index J_1 is 2.54×10^4, for the PSO when the index is J_2 the mean has the value of 1.76×10^4. The obtained values shown that even if J_2 considers both functions of the model at the same moment, their value is lower than the case J_1. This phenomenon could be explained because the estimated parameters are closer to the real ones making the virtual model, making the error between the original data and the virtual model decrease.

B. Statistical analysis

Considering the stochastic nature of the signals, ϕ_1 and ϕ_2 it is required to perform a statistical analysis of the results to evaluate the quality of the solutions and to verify that the difference between the computed results are based on the performance function and not the stochastic behavior of these signals.

To check the variability of the estimated parameters between executions of the algorithm, the standard deviation of each one is computed and shown in the Table II. On this Table it can be observed that in almost all the cases, the estimates computed using J_2 provides lower variability than the ones computed by J_1. The exception is the parameters c_l and c_u where J_1 provides lower variability than the other case.

In order to prove that difference between results are generated by the change on the performance function and not due the stochastic signals a Bonferroni non-parametrical test has

TABLE II: Standard deviation of the estimated parameters of the SMA model using a PSO algorithm

Parameter	PSO with J_1	PSO with J_2
α	1.1353	0.8154
β	1.4422	1.0853
a_l	1.4014	1.2086
b_l	5.0911	4.3192
c_l	0.1623	0.6866
d_l	11.3044	5.5632
a_u	1.0288	0.7680
b_u	2.1417	1.2045
c_u	0.7080	1.3086
d_u	8.5632	6.2233

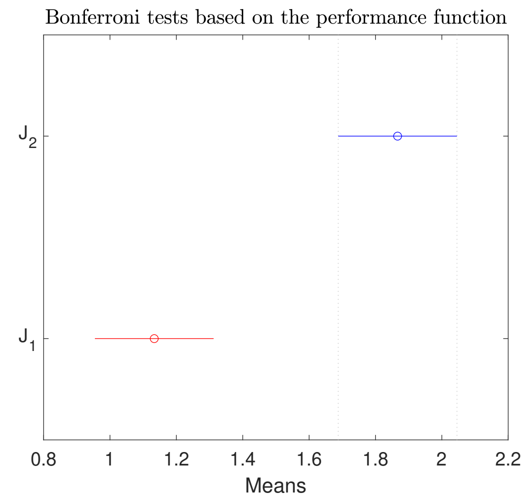

Fig. 2: Bonferroni non-parametrical test performed on the computed performance functions.

been performed [22]. First one, a Kruskall-Wallis test is performed to compute the means and the p-value. The Kruskall-Wallis test makes a comparison between the distribution of the data groups in order to identify if these groups belongs to a same population or not. If they belong to the same population it implies the variation of the data is produced by the stochastic nature of the algorithm and not by the variations proposed in the study. For this particular case it implies that if they belong to the same population the effect of the performance function J could be neglected. The computed p-value corresponds to 5.9035×10^{-5}, this implies that the probability that the differences between the results are significant is over the 99%. Then the Bonferroni test has been performed, this tests allows to compare if the behavior of the data is closer between the groups or it is clearly separable. The results are shown on Figure 2, it is clearly observed that there exists significant differences between the results, and these are not related to the stochastic nature of the signals ϕ_1 and ϕ_2.

To verify that the computed parameters provides a similar behavior to the original signal, a comparison between the original model and the model when the estimated parameters are applied is shown on Figure 3. The behavior of the computed output based on the performance function applied shows that

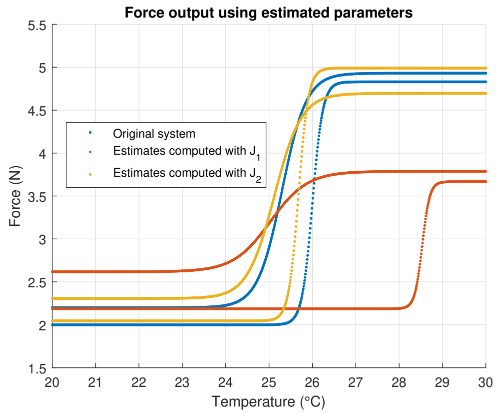

Fig. 3: Behavior of the force output using the estimated parameters computed by the PSO based on the performance function.

the estimates computed using J_2 generate a closer output of the virtual model. For the case of the estimates computed with J_1 it is clearly observed that the difference of values produces changes on the output making the behavior are not closer than the one obtained with J_2. The amplitude differences on this case is generated by the error on the parameters a_l and a_u, and translation of the behavior over the x-axis for the cases where the differences are produced by d_l.

V. Conclusions

This works studies the quality of the estimated parameters computed for a hybrid model using the PSO algorithm for two different performance functions. The goal of the implementation of performance functions is to avoid the necessity of applying a switching during the computation of the performance index. The obtained results shown that the time required to execute the algorithm decreases when the penalty functions are applied instead of the switching between conditions. The repeatability of the estimated parameters has been verified using the standard deviation, where it can be noticed that the function with penalty function decreases the deviation for almost all the parameters. A statistical test of Bonferroni has been performed to assure the results are dependent on the performance function and not from the stochastic nature of the PSO algorithm. The implementation of the estimated parameters on the model allows to verify the use of penalty functions is a feasible option to be implemented on the performance function J in order to improve the quality of the estimated parameters. However, the use of this performance function with a more advanced PSO algorithms could improve these results an remains as a future work to be.

Acknowledgment

The authors acknowledges the support of Secretaría de Investigación y Posgrado of Instituto Politécnico Nacional (SIP-IPN) through the project SIP-20210978 from the *Programa Especial de Consolidación de Investigadores*.

References

[1] Nayak, B. K., A. Mohapatra, and K. B. Mohanty, "Parameters estimation of photovoltaic module using nonlinear least square algorithm: A comparative study." in 2013 Annual IEEE India Conference. 13-15 Dec. 2013

[2] Sastry, Shankar, and M. Bodson. Adaptive control: stability, convergence and robustness. Courier Corporation, USA, 2011.

[3] Garrido, Ruben, and Roger Miranda. "DC servomechanism parameter identification: A closed loop input error approach." ,*ISA transactions*, vol. 51, no.1, pp. 42-49. 2012.

[4] Yang, Bo, et al. "Comprehensive overview of meta-heuristic algorithm applications on PV cell parameter identification." *Energy Conversion and Management* vol.208, no.1, pp. 112595. 2020.

[5] Yousri, D. A., et al. "Parameter identification of fractional-order chaotic systems using different meta-heuristic optimization algorithms." *Nonlinear Dynamics* vol. 95, no. 3, pp. 2491-2542. 2019.

[6] Pelteret, Jean-Paul, Bastian Walter, and Paul Steinmann. "Application of metaheuristic algorithms to the identification of nonlinear magneto-viscoelastic constitutive parameters." *Journal of Magnetism and Magnetic Materials* vol. 464, pp. 116-131. 2018.

[7] Sultan, Hamdy M., et al. "Parameter identification of proton exchange membrane fuel cells using an improved salp swarm algorithm." *Energy Conversion and Management* vol. 224, pp. 113341. 2020.

[8] Voros, Josef. "Iterative algorithm for parameter identification of Hammerstein systems with two-segment nonlinearities." *IEEE transactions on Automatic Control* vol. 44, no.11, pp. 2145-2149. 1999.

[9] Vörös, Jozef. "Parameter identification of Wiener systems with multisegment piecewise-linear nonlinearities." *Systems and Control Letters* vol. 56, no. 2, pp. 99-105. 2007.

[10] Allafi, Walid, et al. "Parameter estimation of the fractional-order Hammerstein–Wiener model using simplified refined instrumental variable fractional-order continuous time." *IET Control Theory Applications* vol. 11, no. 15, pp. 2591-2598. 2017.

[11] Maier, Clemens C., et al. "Modeling and nonlinear parameter identification for hydraulic servo-systems with switching properties." *Mechatronics* vol. 61 pp. 83-95. 2019.

[12] Lu, W., Keyhani, A., Fardoun, A. Neural network-based modeling and parameter identification of switched reluctance motors. *IEEE transactions on energy conversion* vol. 18, no. 2, pp. 284-290. 2003.

[13] Linares-Flores, J., Méndez, A. H., García-Rodríguez, C., Sira-Ramírez, H. Robust nonlinear adaptive control of a "boost" converter via algebraic parameter identification. *IEEE Transactions on Industrial Electronics* vol. 61 no. 8, pp. 4105-4114. 2013.

[14] Zhang, Junhao, and Pinqi Xia. "An improved PSO algorithm for parameter identification of nonlinear dynamic hysteretic models." *Journal of Sound and Vibration* vol. 389, pp. 153-167. 2017.

[15] Hafsi Laabidi, and Moufida Ksouri-Lahmari, "Identification of Wiener-Hammerstein model with multisegment piecewise-linear characteristic.", in 16th IEEE Mediterranean Electrotechnical Conference. 25-28 March 2012.

[16] E. Mezura-Montes and J. I. Flores-Mendoza, "Improved particle swarm optimization in constrained numerical search spaces," in *Nature-Inspired Algorithms for Optimization,* Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, Sec. 3, pp: 299–332.

[17] Freitas, Diogo, Luiz Guerreiro Lopes, and Fernando Morgado-Dias. "Particle swarm optimisation: A historical review up to the current developments." *Entropy* vol. 22, no. 3, pp. 362-367. 2020.

[18] Ljung, Lennart. "System identification." Signal analysis and prediction. Birkhäuser, Boston, MA, 1998. 163-173.

[19] Cortez-Vega, R., et al. "A hybrid dynamic model of shape memory alloy spring actuators." *Measurement* vol. 114, pp. 340-353. 2018.

[20] Simoes, Marlini, and Emilio Martínez-Pañeda. "Phase field modelling of fracture and fatigue in Shape Memory Alloys." *Computer Methods in Applied Mechanics and Engineering* vol. 373 pp. 113504. 2021.

[21] M. López-Ibáñez, et al., "The irace package: Iterated racing for automatic algorithm configuration," *Oper. Res. Perspect.,* vol. 3, no. 1, pp: 43–58. 2016.

[22] Derrac, Joaquín, et al. "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms." *Swarm and Evolutionary Computation* vol. 1, no. 1, pp. 3-18. 2011.