

DESIGN OF A ROBOTIC DEVICE ACTUATED BY CABLES FOR HUMAN LOWER LIMB REHABILITATION USING SELF-ADAPTIVE DIFFERENTIAL EVOLUTION AND ROBUST OPTIMIZATION

DISPOSITIVO ROBÓTICO ATUADO POR CABOS PARA REABILITAÇÃO DO MEMBRO INFERIOR HUMANO UTILIZANDO EVOLUÇÃO DIFERENCIAL AUTO-ADAPTÁVEL E OTIMIZAÇÃO ROBUSTA

Rogério Sales GONÇALVES¹; João Carlos Mendes CARVALHO²; Fran Sérgio LOBATO³

1. School of Mechanical Engineering (FEMEC), Federal University of Uberlândia, Uberlândia-MG – Brazil, rsgoncalves@ufu.br. 2. School of Mechanical Engineering (FEMEC), Federal University of Uberlândia, Brazil; 3. NUCOP – Laboratory of Modeling, Simulation, Control and Optimization, Federal University of Uberlândia, School of Chemical Engineering, Brazil.

ABSTRACT: In engineering designed systems it is commonly considered that mathematical models, variables, and parameters are sufficiently reliable, i.e., there are no errors in modeling and estimation. However, the systems to be optimized can be sensitive to small changes in the designed variables causing significant changes in the objective function. Robust optimization is an approach for modeling optimization problems under uncertainty in which the modeler aims to find decisions that are optimal for the worst-case realization of the uncertainties within a given set of values. In this contribution, a self-adaptive heuristic optimization method, namely the Self-Adaptive Differential Evolution (SADE), is evaluated. Differently from the canonical Differential Evolution algorithm (DE), the SADE strategy is able to update the required parameters such as population size, crossover parameter, and perturbation rate, dynamically. This is done by considering a defined convergence rate on the evolution process of the algorithm in order to reduce the number of evaluations of the objective function. For illustration purposes, the SADE strategy is associated with the Mean Effective Concept (MEC) for insertion robustness, is applied to minimize forces applied in cables used for the rehabilitation of the human lower limbs by determining the positioning of motors. The results show that the methodology that was proposed (SADE+MEC) appears as an interesting strategy for the treatment of robust optimization problems.

KEYWORDS: Robust Optimization. Self-Adaptive Differential Evolution. Mean Effective Concept. Rehabilitation. Robotic Device.

INTRODUCTION

Traditionally, during the engineering systems design it is considered that the result is not subject to the influence of small perturbations of design variables and/or parameters involved in the process. However, the global optimum solution can be sensitive to small perturbations of design variables vector. In this case, an optimal local solution may be less sensitive, which from a real point of view, is configured as a feasible and can be implemented in industry. The concept of robust optimization should be used in order to minimize this effect in engineering systems design. This approach is applied for modeling optimization problems under uncertainty, in which the modeler aims to find decisions that are optimal for the worst-case realization of the uncertainties within a given set of values.

Robust optimization is defined as an approach that produces a solution which is not very sensitive to small changes in design variables (TAGUCHI, 1984). It is emphasized that a robust solution may not coincide with the nominal solution,

i.e., a solution without robustness, as observed in Figure 1. In this context, the robustness characterizes an important tool to help getting a not very sensitive solution under certain conditions, when exposed to given conditions of uncertainty.

In the literature, some studies considering the introduction of robustness in the mono and multi-objective optimization context can be found (DEB; GUPTA, 2006; SOUZA et al., 2015). Several of these studies require the introduction of new restrictions and/or new objectives, i.e., relations between the mean and the standard deviation of objective functions vector, and probability distribution functions for the design variables and/or objectives. As an alternative to these classical formulations, Deb and Gupta (2006) extended the Mean Effective Concept (MEC), originally proposed for mono-objective problems, to the multi-objective optimization context. In this approach, no additional restriction is inserted into the original problem. Thus, the problem is rewritten as a mean vector of original objectives.

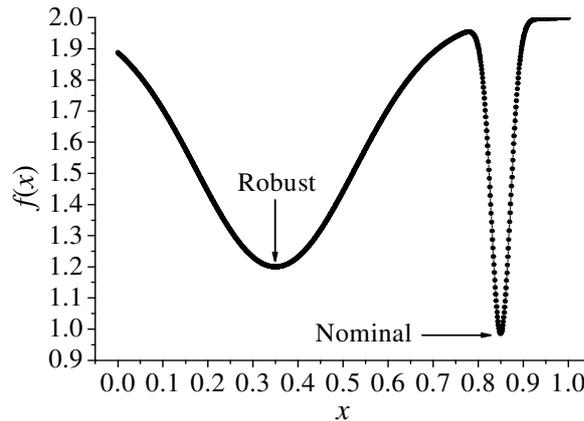


Figure 1. Nominal solution versus robust solution (Taguchi, 1984; Deb and Gupta, 2006).

Traditionally, the engineering systems design has been obtained by using Deterministic Optimization Methods. In the last years, Non-Deterministic Optimization Methods have been used to solve this kind of problem. During the solution of optimization problems, the input parameters of any evolutionary algorithm are kept constant (i.e., population size, crossover parameter, and perturbation rate, among others). In this case, the computational implementation of the algorithms is simplified. However, the variation of the population size inherent to real biological systems (important aspect of biological evolution) is disregarded.

Intuitively, it is interesting to expand the population size during the first generations (i.e., assume its maximum value) due to the high diversity faced at this stage. This aspect offers the opportunity to the individuals to explore the design space accordingly. On the other hand, from the optimization point of view, in the end of the evolutionary process, the natural tendency of the population is to become homogeneous, which implies unnecessary evaluations of the objective function and, consequently, the increase of computational cost, when the population size is kept constant.

To overcome this disadvantage, in this work the Self-Adaptive Differential Evolution algorithm (SADE), proposed by Cavalini et al. (2015), is considered with an optimization strategy. In this evolutionary approach, the population size, crossover parameter, and perturbation rate are dynamically updated during the convergence process in the Differential Evolution algorithm (DE). Basically, the SADE strategy reduces the number of objective function evaluations based on the definition of a convergence rate in order to evaluate the homogeneity of the population in the evolutionary process (CAVALINI et al., 2015).

In the engineering systems design context, the development of robotic devices to be applied in the rehabilitation process of human lower limbs is configured as an important application due to the large number of people with lower limb problems caused by stroke and/or accidents. In general, human beings have always tried to perform tasks making less effort as possible. In the last years, machineries and equipments were developed to simplify the execution of multiple tasks, reducing the time needed to fulfill them, improving the patient's quality of life and safety. With the advancement of technology, it was possible to apply robots also in health, using them for surgeries, prosthetics and structures to assist in rehabilitation.

In this contribution, the SADE algorithm associated with the MEC is used to optimize the motors positioning in a parallel robotic device actuated by cables for human lower limb rehabilitation used to assist patients and health professionals during rehabilitation sessions. The results obtained by proposed methodology (SADE+MEC) are compared with those obtained by using the canonical DE algorithm.

MATERIAL AND METHODS

Robust optimization

Mean effective concept (Deb and Gupta, 2006): For the minimization of an objective function ($f(x)$), a solution x^* is called a robust solution if it is the global minimum of the mean effective function $f_{eff}(x)$, defined with respect to a δ -neighborhood as follows:

$$f_{eff}(x) = \min \frac{1}{|\Upsilon_{\delta}(x)|} \int_{y \in \Upsilon_{\delta}(x)} f dy \quad (1)$$

where Υ_{δ} is the δ -neighborhood of the solution x and $|\Upsilon_{\delta}|$ is the hypervolume of the neighborhood.

Basically, a finite set of solutions H can be generated randomly using the latin hypercube for the evaluation of integral given by Eq. (1). In this case, defining the δ -neighborhood with respect to design variables vector, N solutions are generated employing the latin hypercube, with the integral evaluated numerically. It should be mentioned that the aspect increases the computational cost due to the number of integral evaluations necessary to evaluate the objective function (DEB; GUPTA, 2006).

Robotic devices actuated by cables for lower limb human rehabilitation

The parallel cable-driven manipulator consists on a base and a moving platform which are connected by multiple cables that can extend or retract (see Figure 2). Then, a cable-based manipulator can move the end-effector by changing the cables' lengths while preventing any cables to become slack. Therefore, feasible tasks are limited due to main static or dynamic characteristics of the cables because they can only pull the end-effector but cannot push it (CANNELLA et al., 2008; HILLER et al., 2009).

These structures have characteristics that make them suitable for rehabilitation purposes. They have large workspace, which may be adapted to different patients and different training. The mechanical structure is easy to assembly and disassembly, which makes it easier to transport, and can be reconfigured in order to perform different therapies. In the clinical point of view, the use of cables instead of rigid links makes patients feel less constrained, which is important to help them accept the technology. These characteristics make the cable-based parallel manipulators ideal for rehabilitation. The drawbacks related to the use of a cable-driven parallel structure are the physical nature of cables that can only pull and not push and the fact that the workspace evaluation turns into dependent forces and can have a complex and irregular shape (HILLER et al., 2009).

The cable-driven parallel manipulator, proposed in this paper, can be assembled from two to six cables arranged in a rigid structure (fixed platform) having a moving platform (splint), Figure 2(a). Figure 2(b) shows the prototype built at the Laboratory of Robotics and Automation at the Federal University of Uberlândia. Figure 2(a) shows the elements of the cable-based parallel manipulator, consisting on sets formed by 24 Volts x 45 Nm DC motor, encoder with 500 pulses per revolution and pulley. In this first step towards implementation of graphic simulations and

experimental tests, a 1.80 m anthropometric wooden puppet was used to simulate a human body, Figure 2(b).

The equipment works by using the "teaching by showing", repeating movements predetermined by the therapist to the patient. Therefore, it is necessary to perform the control of actuators in two distinct steps: a step called "teaching", in which the therapist "teaches" the movements to be performed by the machine, and another step of "playing", in which the machine runs the predetermined movements. The control of the actuators was conducted using three PIC18F4550 controllers that communicate with each other by the I²C interface. One is used as master, so it can command the others as slaves, and promote communication with a PC using a USB port.

In the "teaching" step, in which the movements will be taught to the machine, the acquisition of position data and speed of each motor shaft is done through digital encoders. Each movement performed by the therapist in the splint in which the patient's lower limb is positioned, Figure 2(a), should cause a shift in the axis of the motors. However, just as cables are used to attach the splint to the axis of the actuators, the movement of the therapist could just loosen the cables, without any movement on the shaft of the engine was triggered. Thus, there must be a control loop that maintains the tension of the cables at this stage so as to cause the movement of the actuator when the therapist moves the splint. The signal from load cells attached to the cables will be used as a control variable, and the PWM signal of PIC microcontrollers promote the rotation of the actuators. Position data and angular velocity of each actuator are obtained by digital encoders, and saved to be replayed during the stage of playing (GONCALVES et al., 2015).

The "playing" step, in which the structure that was proposed repeats the movements "taught" by the therapist, is done through the closed-loop control of speed and position of the actuators shafts, using PIC microcontrollers. This occurs via PWM control in order to guarantee the reproduction of movements executed in the "teaching" step (GONCALVES et al., 2013).

To ensure the safe operation, emergency buttons are installed and the maximum allowable forces acting on the cables are set to prevent injuries involving patients. A graphical interface for PC was developed to control in which step of the operation the structure should run.

The kinematic model of cable-driven parallel robots is obtained similarly to the model obtained from traditional parallel structures (CÔTÉ,

2003). The inverse kinematic problem consists in finding the cables lengths (ρ_i) as function of the end-effector pose. The forward kinematic problem consists of finding the end-effector poses for a given set of cables lengths. For the kinematic model, the parameters used are shown in Fig. 2(c). The kinematic variables are the cables' lengths.

The equilibrium equations for forces and moments acting on each cable can be given by

$$F = J^{-1}W \tag{2}$$

where vector F represents the cable tension, which consists on forces that must be done by actuators, W is the vector of external forces and moments applied to the system, which are the limb and the splint weight and, J is the Jacobian matrix of the structure. More details about the Jacobian calculus can be found in (CARVALHO et al., 2013; BARBOSA, 2013).

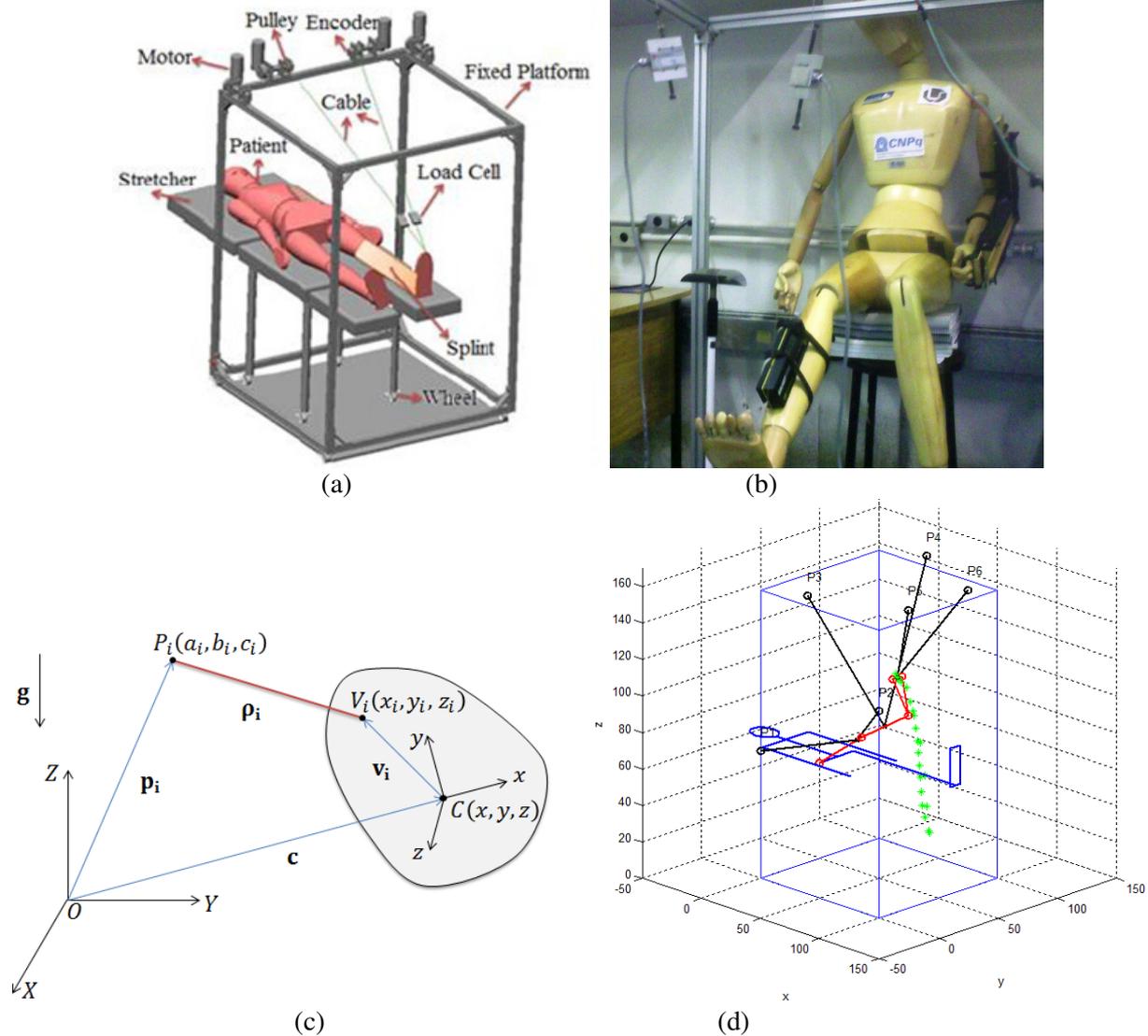


Figure 2. (a) Scheme of the parallel structure proposed; (b) Prototype built; (c) Kinematic parameters; (d) gait simulation with the proposed structure.

During the rehabilitation movements it is possible that at least one cable has no traction, leading to a region of non-movement control in the structure, since cables cannot push the lower limb but only pull it. To avoid this, it is necessary that positive forces exist in the cables throughout the movement, otherwise it is not possible to move the limb in all positions required for the rehabilitation

session. Thus, in this paper, the optimization approach was used to determine the best points to fix the motor in a fixed platform, Figure 2(a), according to the type of movement performed, through the following objective function (OF):

$$\min_{i \in \Gamma} OF = \sum_{i \in \Gamma} (\max(0, -(F_i - 0.5)))^2 \quad (3)$$

where F_i is the compression forces set. Intuitively, this function seeks to minimize the number of negative or very low force on cables throughout the movement (less than 0.5 N). For this case, the design variables are the positions of the motors in the structure (x_i , $i = 1, \dots, 6$), as presented in Figure 2 (a).

Self-adaptive differential evolution

The DE algorithm is an optimization technique that belongs to the family of evolutionary computation, which differs from other evolutionary algorithms in the mutation and recombination schemes. Basically, DE executes its mutation operation by adding a weighted difference vector

between two individuals to a third individual. Then, the mutated individuals will perform discrete crossover and greedy selection with the corresponding individuals from the last generation to produce the offspring. The key control parameters for DE are the population size (NP), the crossover constant (CR), and the so-called weight (F_w). The canonical pseudo-code of DE algorithm is presented in Figure 3, in which P is the population of the current generation, and P' is the population to be constructed for the next generation, $C_{[i]}$ is the candidate solution with population index i , $C_{[i][j]}$ is the j -th entry in the solution vector of $C_{[i]}$, and r is a random number between 0 and 1.

```

Differential Evolution
Initialize and evaluate population P
while (not done) {
  for (i = 0; i < N; i++) {
    Create candidate C[i]
    Evaluate C[i]
    if (C[i] is better than P[i])
      P0[i] = C[i]
    else
      P0[i] = P[i] } and P = P0 }
Create candidate C[i]
Randomly select parents P[i1], P[i2], and P[i3]
where i, i1, i2, and i3 are different.
Create initial candidate
C'[i] = P[i1] + Fw × (P[i2] - P[i3]).
Create final candidate C[i] by crossing over the genes of P[i] and C'[i] as follows:
  for (j = 0; j < N; j++) {
    if (r < CR)
      C[i][j] = C'[i][j]
    else
      C[i][j] = P[i][j]
  }

```

Figure 3. Canonical DE algorithm (Storn and Price, 1995).

Storn and Price (1995) have given some simple rules for choosing the key parameters of DE for general applications. Normally, NP should be about 5 to 10 times the dimension of the problem (i.e., number of design variables). As for F_w , it lies in the range between 0.4 and 1.0. Initially, $F_w = 0.5$ can be tried, then F_w and/or NP can be increased if the population converges prematurely. Storn et al.

(2005) proposed various mutation schemes for the generation of new candidate solutions by combining the vectors that are randomly chosen from the current population, as shown by Eq. (4).

$$\begin{aligned}
 DE / best / 1 &\rightarrow x = x_{best} + F_w (x_{r_1} - x_{r_2}) \\
 DE / rand / 1 &\rightarrow x = x_{r_1} + F_w (x_{r_2} - x_{r_3}) \\
 DE / best / 2 &\rightarrow x = x_{best} + F_w (x_{r_1} - x_{r_2}) + F_w (x_{r_3} - x_{r_4}) \\
 DE / rand / 2 &\rightarrow x = x_{r_1} + F_w (x_{r_2} - x_{r_3}) + F_w (x_{r_4} - x_{r_5})
 \end{aligned} \tag{4}$$

where x_{r_1} , x_{r_2} , x_{r_3} , x_{r_4} , and x_{r_5} are candidate solutions randomly chosen, and x_{best} is the candidate solution associated to the best fitness value, all of them appear in the population of the current generation (denoted by C in Figure 3). The vector x is denoted by C' in Figure 3.

The DE algorithm has been applied with success both to mono and multi-objective problems (STORN; PRICE, 1995) in various research fields. As examples, we can mention the solution of multi-objective optimal control problems with index fluctuation applied to fermentation process (LOBATO et al., 2007), digital filter design (STORN; PRICE, 1995), synthesis and optimization of heat integrated distillation system (BABU; SINGH, 2000), multi-objective optimization of mechanical structures (LOBATO; STEFFEN, 2007), solution of inverse radiative transfer problems in two-layer participating media (LOBATO et al., 2008a), estimation of drying parameters in rotary dryers (LOBATO et al., 2008b), apparent thermal diffusivity estimation of the drying of fruits (MARIANI et al., 2008), Gibbs free energy minimization in a real system (LOBATO et al., 2009), estimation of space-dependent single scattering albedo in radiative transfer problems (LOBATO et al., 2010a; 2010b; 2010c), design of fractional order PID controllers (BISWAS et al., 2009), other interesting applications (STORN et al., 2005; DAS; SUGANTHAN, 2011), and in the identification problem associated to rotating machines (CAVALINI Jr et al., 2012).

The DE algorithm uses a fixed population size during the evolutionary process. According to Velleev (2008), this characteristic can affect the robustness and increase the computational cost of the algorithms. A small population size may result in local convergence, while large population size will increase the computational effort and may lead to slow convergence. Thus, an appropriate population size can assure the effectiveness of the algorithm. It is worth mentioning that in the end of the evolutionary process the population tends to become homogeneous leading to unnecessary evaluations of objective function and, consequently, increasing the computational cost.

The main problem found in the so-called non-deterministic optimization approach is the high number of evaluations of the objective function needed to solve the optimization problem. According to Coelho and Mariani (2006), even for an extending number of objective function evaluations, there is no guarantee that premature convergence will be avoided. In addition, the DE algorithm is sensitive to control parameters (STORN et al., 2005) and it is highly problem-dependent (ZAHARIE, 2002, 2003; QIN; SUGANTHAN, 2005), claiming for ad-hoc configurations.

According to Nobakhti and Wang (2006), the special mutation mechanism used in DE leads to the stop of the search. If, for any reason, the DE population loses diversity, such as an incorrect choice of the perturbation rate F_w , the mutation becomes zero. In order to overcome this problem, various methodologies have been proposed. Zaharie (2003) proposes a feedback update rule for F_w that is designed to maintain the diversity of the population at a given level. This procedure is able to avoid premature convergence. Recently, chaotic search models have been used for the adaptation of parameters in non-deterministic approach due to their ability in avoiding premature convergence (COELHO; MARIANI, 2006). Coelho and Mariani (2007) have used the Ant Colony algorithm with logistic maps in engineering problems. Additionally, the number of objective function evaluations can decrease by using special strategies to update the population size.

In this context, the Self-Adaptive Differential Evolution (SADE) algorithm, proposed by Cavalini et al. (2015), is used to update the following parameters: perturbation rate, crossover parameter and population size. Thus, the DE drawbacks mentioned can be minimized.

Updating of the population size

The canonical non-deterministic optimization approaches keep the population size fixed during the evolutionary process. This aspect simplifies the algorithms, but it represents a restriction that does not follow the biological evolution. The natural phenomenon includes

continuous variation in the number of individuals, which increases when there are highly fitted individuals and abundant resources, and decreasing otherwise. As mentioned, it may be beneficial to expand the population in early generations when the phenotype diversity is high. However, the population can be contracted when the unification of the individuals in terms of structure and fitness no

$$NP = (NP_{max} - NP_{min}) \left(\frac{Iter_{max} - Iter}{Iter_{max}} \right) + NP_{min} \tag{5}$$

$$NP = \max \left(\frac{NP_{max}}{2} \sin \left(\frac{Iter}{A} \right) + \frac{NP_{max}}{2}, NP_{min} \right) \tag{6}$$

where $Iter_{max}$ is the limit of generations, $Iter$ is the current generation, while NP_{min} and NP_{max} are the minimal and maximum number of individuals in the population, respectively. A is the periodicity of the updating scheme. In these equations, the population size is updated considering only the evaluation of these mathematical expressions, i.e., no information about the evolutionary process is considered in this analysis. This characteristic represents the main disadvantage of this purely mathematical approach.

In order to overcome this disadvantage, an adaptive population size strategy with partial increasing or decreasing number of individuals according to diversities in the end of each generation is adopted in this work. Therefore, a convergence rate (λ) is defined as (CAVALINI et al., 2015):

$$\lambda = \frac{f_{average}}{f_{worst}} \tag{7}$$

where $f_{average}$ and f_{worst} are the average and worst values of the objective function (i.e., fitness values), respectively.

The defined convergence rate evaluates the homogeneity of the population in the evolutionary process. If λ is close to zero, the worst value of the objective function is different from the average value. If λ is close to one, the population is homogeneous. Thus, a simple equation for dynamic updating of the population size can be proposed by Cavalini et al. (2015):

$$NP = \text{round} \left(NP_{min} + (1 - \lambda) NP_{max} \right) \tag{8}$$

where the operator $\text{round}(\square)$ indicates the rounding to the nearest integer.

It should be emphasized that the equation Eq. (8) updates the population size based on the convergence rate. Differently, any information regarding the evolution of the process is considered in Eq. (5) and Eq. (6).

longer justifies the maintenance of a large population and the higher computational costs associated (VELLEV, 2008).

Sun et al. (2007) evaluated two strategies for the dynamic updating of the population size during the evolutionary process, as shown by Eq. (5) and Eq. (6).

F_w and CR updating

The procedure adopted in this work for the updating of F_w and CR is presented by Zaharie (2003). The methodology is based on the evolution of the population variance (i.e., a measure of the population diversity), which is given by:

$$Var(x) = x^2 - \left(\sum_{i=1}^{NP} \frac{x_i}{NP} \right)^2 \tag{9}$$

According to Zaharie (2002, 2003), if the best element of the population is not taken into account, the expected value of the variance after recombination is given by (determined from the obtained population):

$$E(Var(x)) = \left(1 + 2F_w^2 CR - \frac{2CR}{NP} + \frac{CR^2}{NP} \right) Var(x) \tag{10}$$

Consider that $x(g-1)$ is the population obtained at generation $g-1$ (the previous population). During the g -th generation, the vector x is transformed into x' (recombination) and then in x'' (selection). The vector x'' represents the starting population for the next generation $x(g+1)$.

The information about the variance tendency can be provided by Eq. (11). If $\gamma < 1$, the variance is increased and the convergence is accelerated. However, premature convergence can be induced. On the other hand, if $\gamma > 1$ the variance is decreased and premature convergence situations can be avoided.

$$\gamma = \frac{Var(x(g+1))}{Var(x(g))} \tag{11}$$

The controlling idea is based on the parameter F_w , such that the recombination applied in the generation g compensates the effect of the previous application of recombination and selection.

For this aim, Eq. (12) and Eq. (13) have to be solved.

$$1 + 2F_w^2 CR - \frac{2CR}{NP} + \frac{CR^2}{NP} = c \quad (12)$$

where c is defined as

$$c \equiv \gamma \frac{Var(x(g+1))}{Var(x(g))} \quad (13)$$

Equation (12) can be solved with respect to F_w . Thus,

$$F_w \equiv \begin{cases} \sqrt{\frac{1}{NP}} \sqrt{\frac{\eta}{2CR}} & \text{if } \eta \geq 0 \\ F_{wmin} & \text{otherwise} \end{cases} \quad (14)$$

where $\eta \equiv NP(c-1) + CR(2-CR)$ and F_{wmin} is the minimal value of F_w .

A sufficient condition for increasing the population variance by recombination is that $F_w > (1/NP)^{0.5}$. Thus, $F_{wmin} = (1/NP)^{0.5}$ should be used. An upper bound for F_w can also be imposed as suggested by Storn et al. (2005). In this case, $F_{wmax} = 2$. By solving Eq. (14) with respect to CR , one obtains the following adaptation rule for CR :

$$CR \equiv \begin{cases} -\frac{(NP \times F_w^2 - 1) + \sqrt{(NP \times F_w^2 - 1)^2 - NP(1-c)}}{2} & \text{if } c \geq 0 \\ CR_{min} & \text{otherwise} \end{cases} \quad (15)$$

with $CR_{min} = 0.01 < CR < 1$.

RESULTS AND DISCUSSION

A gait motion simulation is shown in Fig. 2(d) using the same prototype dimensions, Fig. 1(b), emphasizing the points achieved by the tip of the foot during the movement. More details can be found in (CARVALHO et al., 2013; BARBOSA, 2013).

For evaluating the methodology proposed (SADE+MEC), some practical points regarding the application of the procedure should be emphasized: Motor coordinates in [cm] (BARBOSA, 2013): Motor 01 – [0 0 70 (x_1)], Motor 02 – [0 100 70 (x_2)], Motor 03 – [40 (x_3) 0 163.5], Motor 04 – [40 (x_4) 100 163.5], Motor 05 – [100 25 (x_5) 163.5] and Motor 06 – [100 75 (x_6) 163.5];

For motors P1 and P2, the allowable range is between 50 cm and 150 cm. For motors P3, P4, P5, and P6, the allowable range is between 10 cm and 90 cm in relation to the reference axes. It was given a limit of 10 cm after the start and before the end of the bar to ensure the assembling of the motor; It is important to emphasize that these limits considered to define the domain of design variables are due to design constraint;

The parameters used by DE algorithm (STORN; PRICE, 1995) are the following: $NP=25$, $F_w=0.8$, $CR=0.8$, 100 generations, and DE/rand/1/exp strategy (see STORN; PRICE (1995) to details) for the generation of potential candidates; The parameters used by SADE are the following: NP defined by Eq. (8) (minimal and maximum numbers of individuals in the population equals to 5 and 25, respectively), F_w and CR defined by Eq. (14) and Eq. (15), respectively, and DE/rand/1 strategy (STORN; PRICE, 1995; STORN et al., 2005) for the generation of potential candidates. The stopping criterion for all the algorithms is associated to the difference between the best and the worst values of the objective function; this difference should be smaller than 10^{-6} . All case studies were run 10 times to obtain the upcoming average values, considering in each simulation the following seeds: 0, 1, 2, ..., and 9. In this test case were considered the following values for the δ parameter: [0.01 0.025 0.05]. In addition, 20 points were used to generate samples in Monte Carlo Method (necessary for insertion robustness).

Nominal optimization

Table 1 present the results obtained with simulation (first line), i.e., considering the design variables proposed by Barbosa (2013) to simulate this mechanism and with application of the DE and SADE algorithms for nominal optimization considering different seeds to initialize the evolutionary process.

In this table it is important to observe that both DE and SADE were able to estimate the parameters satisfactorily as shown by the values obtained for the objective function. However, the SADE algorithm leads to a smaller number of objective function evaluations as compared with the original DE algorithm (a reduction (average) of 29.57% in the number of objective function evaluations with respect to canonical DE). In addition, the values obtained for the variables x_i ($i=3, \dots, 6$) are the inferior or superior limits due to design constraint.

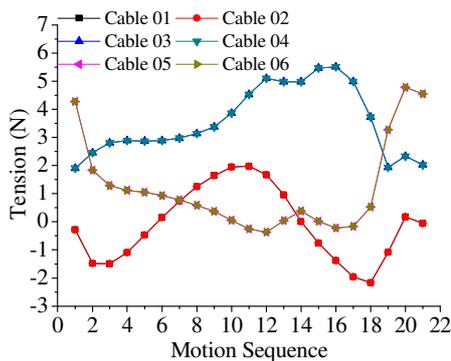
Figure 4 presents the results of the forces and the cable lengths obtained with simulation and application of the SADE algorithm. In this figure, for simulation, there is an overlapping of cables due to the symmetry of the positioning of the motors' structure and to the fact that the puppet is situated in the middle of it, and so that the cable length 1 is equal to cable 2, cable length 3 = 4, cable length 5 = 6 (the coordinates of the motors' attachment are presented in first line of Table 1).

Table 1. Results obtained with nominal optimization by using the DE and SADE algorithms (N_{eval} - Number of objective function evaluations).

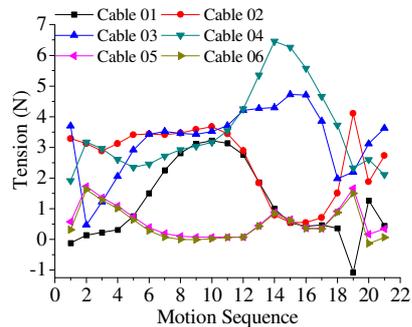
Seed	Method	x_1 (cm)	x_2 (cm)	x_3 (cm)	x_4 (cm)	x_5 (cm)	x_6 (cm)	OF	N_{eval}
-	-	70.00	70.00	40.00	40.00	25.00	75.00	46.00	-
0	DE	95.9149	95.9149	10.00	10.00	10.00	9.00	0.6687	2200
	SADE	95.9149	95.9149	10.00	10.00	10.00	9.00	0.6687	1200
1	DE	95.9149	95.9149	10.00	10.00	10.00	9.00	0.6687	2500
	SADE	95.9149	95.9149	10.00	10.00	10.00	9.00	0.6687	2025
2	DE	95.9149	95.9149	10.00	10.00	9.00	10.00	0.6687	2400
	SADE	95.9149	95.9149	10.00	10.00	9.00	10.00	0.6687	1825
3	DE	95.9149	95.9149	10.00	10.00	9.00	10.00	0.6687	3700
	SADE	95.9149	95.9149	10.00	10.00	9.00	10.00	0.6687	2125
4	DE	95.9149	95.9149	10.00	10.00	10.00	9.00	0.6687	2450
	SADE	95.9149	95.9149	10.00	10.00	10.00	9.00	0.6687	1925
5	DE	95.9149	95.9149	10.00	10.00	10.00	9.00	0.6687	2225
	SADE	95.9149	95.9149	10.00	10.00	10.00	9.00	0.6687	1500
6	DE	95.9149	95.9149	10.00	10.00	10.00	9.00	0.6687	2525
	SADE	95.9149	95.9149	10.00	10.00	10.00	9.00	0.6687	1925
7	DE	95.9149	95.9149	10.00	10.00	9.00	10.00	0.6687	2425
	SADE	95.9149	95.9149	10.00	10.00	9.00	10.00	0.6687	1950
8	DE	95.9149	95.9149	10.00	10.00	10.00	9.00	0.6687	4025
	SADE	95.9149	95.9149	10.00	10.00	10.00	9.00	0.6687	2425
9	DE	95.9149	95.9149	10.00	10.00	10.00	9.00	0.6687	2775
	SADE	95.9149	95.9149	10.00	10.00	10.00	9.00	0.6687	2275

In relation to the forces' values found, it may be noticed that after the optimization, the forces are distributed in more cables, and the amount of negative force points is decreased. In addition, the cable's lengths increased after optimization, which does not mean loss of quality (this increase is due to repositioning of the motors in the structure). Thus, the optimization program provided the optimal positions of the motors in

order to reduce the regions in which the forces on the cables were negative to perform the movement. It should be noted that for practical applications in the prototype there are points where the forces are negative. As prototype application, the solutions can limit the range of movement of a human leg or decrease the number of cables used in the structure to enable the same type of movement desired.



(a) Cable tension (Simulation).



(b) Cable tension (SADE).

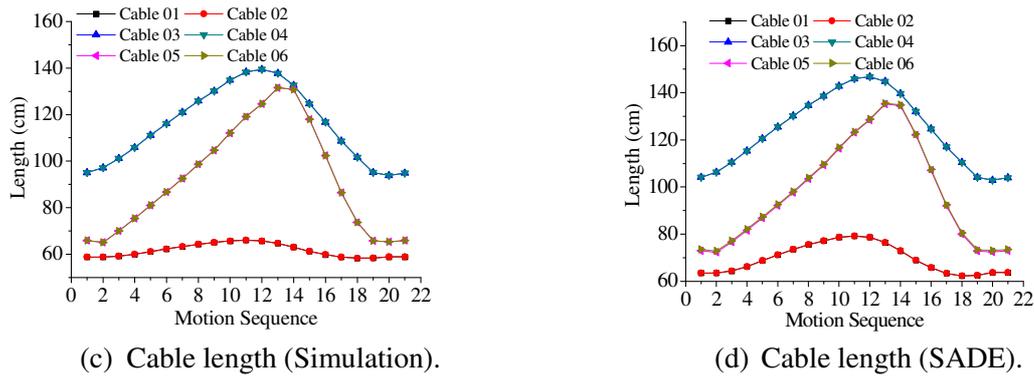


Figure 4. Comparison between results (tension and cable length) considering simulation and optimization by using the SADE algorithm (seed equal to zero).

Robust optimization

Table 2 presents the results (values' average considering 10 runs with different seeds to initialize

the evolutionary process) obtained with robust optimization using of the DE and SADE algorithms considering different values for the δ parameter.

Table 2. Results obtained with robust optimization by using the DE and SADE algorithms (N_{eval} - Number of objective function evaluations).

δ	Method	x_1 (cm)	x_2 (cm)	x_3 (cm)	x_4 (cm)	x_5 (cm)	x_6 (cm)	OF	N_{eval}^*
-	DE	95.9149	95.9149	10.00	10.00	10.00	9.00	0.6687	2722.5
	SADE	95.9149	95.9149	10.00	10.00	10.00	9.00	0.6687	1917.5
0.01	DE	95.8999** (0.1048)***	95.9350 (0.1235)	10.00 (0.0006)	10.00 (0.0003)	90.00 (32.1565)	10.00 (32.6833)	0.6737 (0.0001)	56000
	SADE	95.8999 (0.00948)	95.9350 (0.0257)	10.00 (0.0002)	10.00 (0.0012)	90.00 (30.1098)	10.00 (31.9898)	0.6736 (0.0001)	32000
0.025	DE	95.4760 (0.4201)	96.5190 (0.4333)	10.00 (0.0079)	10.00 (0.0015)	90.00 (1.0443)	10.1844 (0.0544)	0.6871 (0.0008)	70000
	SADE	95.4765 (0.02501)	96.5193 (0.3333)	10.00 (0.0007)	10.00 (0.0012)	90.00 (1.0309)	10.1845 (0.2153)	0.6870 (0.0002)	45400
0.05	DE	97.5360 (1.0988)	95.1330 (1.0063)	10.00 (0.0072)	10.00 (0.0640)	90.00 (1.2232)	13.7979 (0.0994)	0.7363 (0.0024)	68200
	SADE	97.5362 (0.0838)	95.1345 (0.0340)	10.00 (0.0020)	10.00 (0.0040)	90.00 (0.9032)	13.7980 (0.0988)	0.7362 (0.2388)	48700

* Average, **Best result, *** Standard deviation.

In this table, it is possible to observe that the robust optimal is worse than the nominal optimal. In addition, the increase of δ parameter deteriorates the value of nominal solution. As observed in Table 1, the SADE algorithm leads to a smaller number of objective function evaluations as compared with the original DE algorithm (a reduction (average) of 35.1% in the number of objective function evaluations with respect to canonical DE). Figure 5

presents the comparison between tension and length cables obtained by the using SADE (nominal and robust with δ equal to 0.05). In relation to tensions, it can be observed that the forces are more distributed in cables, which does not mean loss of quality (this increase is due to repositioning of the motors in structure).

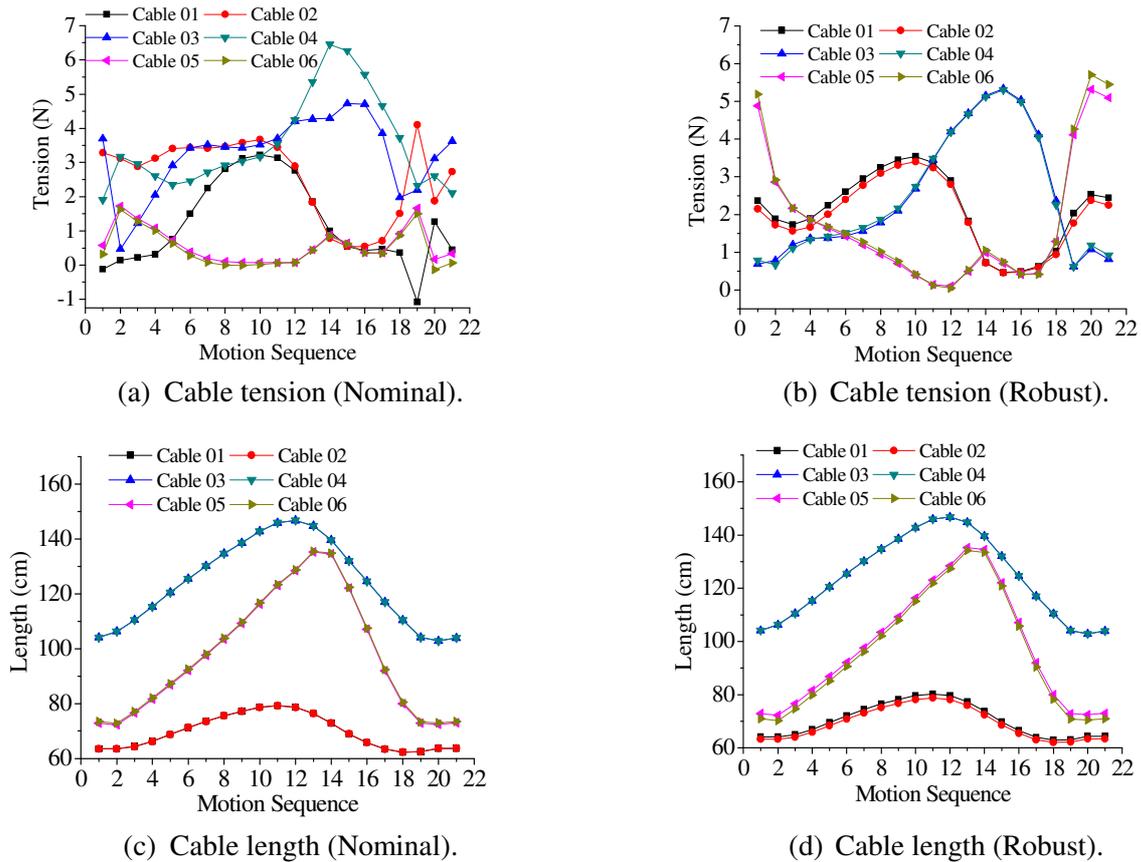


Figure 5. Comparison between results (tension and cable length) by using the SADE algorithm (nominal and robust with δ equal to 0.05).

CONCLUSIONS

In this work a Self-Adaptive Differential Evolution – SADE - algorithm, associated with the Mean Effective Concept - MEC - (for insertion robustness), was used to design a robotic device actuated by cables by human lower limb rehabilitation for determining the positioning motors through the minimization of forces in cables. The SADE optimizer is based on the population’s diversity and convergence rate concepts (the control parameters (CR , F_w and NP) are dynamically updated based on the diversity of the population, which avoids premature convergence of the evolutionary process). The SADE strategy proves to be beneficial for the investigated physical problem as compared to the canonical DE algorithm, with fixed parameters and other existing self-adaptive algorithms.

Finally, the results showed that the methodology (SADE+MEC) represents a promising

alternative for dealing with optimization problems. The main disadvantage of this approach is the increase of the number of objective function evaluations necessary to evaluate the integral considered in the mean effective concept and independent of the optimization strategy considered.

Further research work will be focused on the influence of the parameter values required by SADE in the solution of other interest case studies.

ACKNOWLEDGMENT

Financial support from Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Pró-reitoria de Pesquisa e Pós-Graduação (PROPP/UFU) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) is gratefully acknowledged.

RESUMO: No projeto de sistemas de engenharia é comum considerar que os modelos, as variáveis e os parâmetros são confiáveis, isto é, não apresentam erros de modelagem e de estimação. Entretanto, os sistemas a serem

otimizados podem ser sensíveis a pequenas alterações nas variáveis de projeto causando significativas modificações no vetor de objetivos. Otimização robusta é uma abordagem para modelagem de problemas de otimização sob incerteza em que o modelador tem como objetivo encontrar decisões que são ideais para o pior caso de realização das incertezas dentro de um determinado conjunto de valores. Neste trabalho, um método de otimização heurística auto-adaptável, nomeada Self-Adaptive Differential Evolution (SADE), é avaliada. Diferentemente do algoritmo de Evolução Diferencial, a estratégia SADE é capaz de atualizar os parâmetros necessários, tais como o tamanho da população, o parâmetro de passagem e taxa de perturbação, de forma dinâmica. Isto é feito considerando uma taxa de convergência definido no processo de evolução do algoritmo, a fim de reduzir o número de avaliações da função objetivo. Para fins de ilustração, a estratégia SADE associado ao conceito de média efetiva, para inserção da robustez, é aplicada para minimizar as forças aplicadas nos cabos da estrutura robótica utilizada para a reabilitação dos membros inferiores humanos, determinando o posicionamento dos atuadores. Os resultados mostram que o método proposto neste trabalho configura-se como uma estratégia interessante para o tratamento de problemas de otimização robustos.

PALAVRAS-CHAVE: Otimização Robusta. Evolução Diferencial. Média Efetiva. Reabilitação. Dispositivo Robótico.

REFERENCES

- BABU, B. V.; SINGH, R. P. **Synthesis and optimization of heat integrated distillation systems using differential evolution.** In: PROCEEDINGS OF ALL-INDIA SEMINAR ON CHEMICAL ENGINEERING PROGRESS ON RESOURCE DEVELOPMENT, 2000.
- BARBOSA, A. M. **Desenvolvimento de um dispositivo robótico atuado por cabos para reabilitação do membro inferior humano**, 2013. 1118 f. Dissertação (Mestrado em Engenharia Mecânica) – Curso de Pós-Graduação em Engenharia Mecânica, Universidade Federal de Uberlândia, 2013.
- BISWAS, A. D., ABRAHAM, S. A., DASGUPTA, S. Design of fractional order PID controllers with an improved differential evolution, *Engineering Applications of Artificial Intelligence*, Elsevier Science, v. 22, n. 2, p. 343-350, 2009. <http://dx.doi.org/10.1016/j.engappai.2008.06.003>
- CARVALHO, J. C. M.; BARBOSA, A. M.; GONÇALVES, R. S. **Workspace and Tension Analysis of a Cable-Based Parallel Manipulator for Lower Limb Rehabilitation.** In: VICTOR PETUYA; CHARLES PINTO; ERWIN-CHRISTIAN LOVASZ. (Ed.). *New Advances in Mechanisms, Transmissions and Applications*. 1ed. Netherlands: Springer, 2013, v. 17, p. 165-172.
- CAVALINI JR, A. A.; KOROISHI, E. H.; STEFFEN JR, V. ; LOBATO, F. Sergio. **Evaluation of heuristic optimization methods used for rotor model updating**, In: XV ENCONTRO DE MODELAGEM COMPUTACIONAL, 2012.
- CAVALINI JR, A. A., KOROISHI, E. H., STEFFEN JR, V., LOBATO, F. S. **Model Updating of a Rotating Machine using the Self-Adaptive Differential Evolution Algorithm. Inverse Problems in Science and Engineering**, 1, p. 1-12, 2015.
- CANNELLA, G.; OTTAVIANO, E.; CASTELLI, G. **A cable-based system for aiding elderly people in sit to stand transfer.** In: THE INT. SYMP. ON MULTIBODY SYSTEMS AND MECHATRONICS, San Juan, 2008. p. 8-12.
- COELHO, L. S., MARIANI, V. C. Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. *IEEE Transactions on Power Systems*, v. 21, n. 2, p. 989-996, 2006. <http://dx.doi.org/10.1109/TPWRS.2006.873410>
- COELHO, L. S., MARIANI, V. C. **Improved differential evolution algorithms for handling economic dispatch optimization with generator constraints**, *Energy Conversion and Management*, 48, p. 1631-1639, 2007. <http://dx.doi.org/10.1016/j.enconman.2006.11.007>

CÔTÉ, G. **Analyse et conception de mécanismes parallèles actonnés par câbles**. M. Sc. dissertation, Université Laval, Quebec, 2003.

DAS, S., SUGANTHAN, P. N. Differential evolution: a survey of the state-of-the-art, **IEEE Trans. on Evolutionary Computation**, p. 4-31, 2011. <http://dx.doi.org/10.1109/TEVC.2010.2059031>

DEB, K., GUPTA, H. **Introducing robustness in multi-Objective optimization**, **Evolutionary Computation**, 14, p. 463-494, 2006. <http://dx.doi.org/10.1162/evco.2006.14.4.463>

GONCALVES, R., S., CARVALHO, J. C. M., Ribeiro, J. F.; Salim, V. V. Cable-Driven Robot for Upper and Lower Limbs Rehabilitation. Handbook of Research on Advancements in Robotics and Mechatronics. 1ed.: IGI Global, 2015, v. 1, p. 284-315.

GONCALVES, R. S., CARVALHO, J. C. M., RODRIGUES, L. A. O.; BARBOSA, A. M. Cable-Driven Parallel Manipulator for Lower Limb Rehabilitation. **Applied Mechanics and Materials**, v. 459, p. 535-542, 2013. <http://dx.doi.org/10.4028/www.scientific.net/AMM.459.535>

HILLER, M., HIRSCH, K., BRUCKMANN, T., BRANDT, T., SCHRAMM, D. **Common aspects in methods for the design of mechatronic systems - applications in automotive and robotic**, In: Symposium on Dynamic Problems of Mechanics, 2009.

LOBATO, F. S.; OLIVEIRA-LOPES, L. C.; MURATA, V. V.; STEFFEN Jr., V. **Solution of multi-objective optimal control problems with index fluctuation using differential evolution**. In: 6TH BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND APPLICATIONS - DINCON, 2007.

LOBATO, F. S.; STEFFEN JR., V. **Engineering system design with multi-objective differential evolution**. In: 19TH INTERNATIONAL CONGRESS OF MECHANICAL ENGINEERING, 2007.

LOBATO, F. S.; STEFFEN JR., V.; SILVA NETO, A. **Solution of inverse radiative transfer problems in two-layer participating media with differential evolution**. In: INTERNATIONAL CONFERENCE ON ENGINEERING OPTIMIZATION, Rio de Janeiro – Brasil, 2008a.

LOBATO, F. S., ARRUDA, E. B., BARROZO, M. A. S., STEFFEN JR, V. Estimation of drying parameters in rotary dryers using differential evolution. **Journal of Physics**. Conference Series, 135, p. 1-8, 2008b. <http://dx.doi.org/10.1088/1742-6596/135/1/012063>

LOBATO, F. S., FIGUEIRA, C. E., SOARES, R. R., STEFFEN JR., V. A comparative study of gibbs free energy minimization in a real system using heuristic methods. **Computer-Aided Chemical Engineering**, 27, p. 1059-1064, 2009. [http://dx.doi.org/10.1016/S1570-7946\(09\)70397-9](http://dx.doi.org/10.1016/S1570-7946(09)70397-9)

LOBATO, F. S.; STEFFEN JR., V.; SILVA NETO, A. **Estimation of space-dependent single scattering albedo**. In: RADIATIVE TRANSFER PROBLEMS, INVERSE PROBLEMS, DESIGN AND OPTIMIZATION SYMPOSIUM, João Pessoa, Brazil, August 25-27, 2010a.

LOBATO, F. S., STEFFEN JR., V., SILVA NETO, A. J. A comparative study of the application of differential evolution and simulated annealing in inverse radiative transfer problems. **Journal of the Brazilian Society of Mechanical Sciences and Engineering**, XXXII, p. 518-526, 2010b. <http://dx.doi.org/10.1590/S1678-58782010000500012>

LOBATO, F. S., STEFFEN JR., V., SILVA NETO, A. J. Self-adaptive differential evolution based on the concept of population diversity applied to simultaneous estimation of anisotropic scattering phase function, albedo and optical thickness. **Computer Modeling in Engineering & Sciences**, 1, p. 1-17, 2010c.

MARIANI, V. C., LIMA, A. G. B., COELHO, L. S. Apparent thermal diffusivity estimation of the banana during drying using inverse method. **Journal of Food Engineering**, 85, p. 569-579, 2008.

<http://dx.doi.org/10.1016/j.jfoodeng.2007.08.018>

NOBAKHTI, A.; WANG, H. **A self-adaptive differential evolution with application on the ALSTOM gasifer**. In: PROCEEDINGS OF THE 2006 AMERICAN CONTROL CONFERENCE MINNEAPOLIS, Minnesota, USA, 2006.

QIN, A. K., SUGANTHAN, P. N. **Self-adaptive differential evolution algorithm for numerical optimization**. In: PROCEEDINGS OF THE 2005 CONGRESS ON EVOLUTIONARY COMPUTATION, 2, p. 2005, 1785-179. <http://dx.doi.org/10.1109/cec.2005.1554904>

SOUZA, D. L., LOBATO, F. S., GEDRAITE, R. Multiobjective Optimization applied to optimal control problems using differential evolution. **Chemical Engineering and Technology**, 1, p. 1-8, 2015.

<http://dx.doi.org/10.1002/ceat.201400571>

STORN, R., PRICE, K. **Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces**. International Computer Science Institute, 12, p. 1-16, 1995.

STORN, R. M.; PRICE, K.; LAMPINEN, J. A. **Differential evolution - a practical approach to global optimization**. Springer – Natural Computing Series, 2005.

SUN, S. Y., QUIANG, YC., LIANG, Y. LIU, Y., PAN, Q. **Dynamic population size based on particle swarm optimization**, ISICA 2007, Spring-Verlag Berlin, p. 382-392, 2007.

TAGUCHI, G. **Quality engineering through design optimization**. Kraus International Publications. New York, 1984.

VELLEV, S. **An adaptive genetic algorithm with dynamic population size for optimizing join queries**, International Conference: In: INTELLIGENT INFORMATION AND ENGINEERING SYSTEMS, INFOS 2008, Varna, Bulgaria, June-July, 2008.

ZAHARIE, D. **Critical values for the control parameters of differential evolution algorithms**. In: PROCEEDINGS OF THE 8TH INTERNATIONAL CONFERENCE ON SOFT COMPUTING, 2002, p. 62-67.

ZAHARIE, D. **Control of population diversity and adaptation in differential evolution algorithms**, R. Matouek, P. Omera (eds.). In: PROC. OF MENDEL 2003, 9TH INTERNATIONAL CONFERENCE ON SOFT COMPUTING, 2003, p. 41-46.