Optimization of Walking Control of a Biped Robot using Differential Evolution

Emmanuel Estrada, Héctor M. Becerra and Carlos Segura
Centro de Investigación en Matemáticas (CIMAT), A.C., Computer Science Department,
Jalisco S/N, Col. Valenciana, Guanajuato, Gto., Mexico.
E-mails:{jesus.estrada, hector.becerra, carlos.segura}@cimat.mx

***Abstract*—This paper presents a solution to the problem of optimizing parameters (gains) of a walking control scheme of a biped robot with feet. The problem is addressed using Differential Evolution, a generic optimization method not dependent on mathematical properties of the fitness function. For the optimization of control parameters, we propose and compare three different mono-objective fitness functions and two different encodings of optimization variables that were evaluated in simulation. We have introduced an explicit management of important constraints like joint limits, center of pressure and energy consumption, in a mono-objective formulation. The explicit management of these constraints has allowed to obtain improved results than those obtained by an expert.**

***Index Terms*—biped robot, walking control optimization, Differential Evolution.**

I. Introduction

Within the field of robotics, there is a wide variety of ways to make a robot move on a surface. Common ways are by means of wheels, chains and legs. The locomotion of systems that use legs to move is, in general, very complex [1]. We will focus on the study of biped robots, which are characterized by their mobility from two legs. One of the challenges of this kind of robots, due to the complexity of mathematical model; is to generate a controller that is efficient and robust [2]. An issue with biped robots control is the inherent instability of the two legged motion. The biped robot walking control consists in enforcing a cycle of two phases: a single-support phase where one leg is fixed in contact with the ground and the other leg swings from behind to the front until an instantaneous double-support phase occurs where both legs touch the ground. The stability of this walking pattern depends on the appropriate convergence to desired values of some control variables, for instance, the swing foot position must describe a parabolic motion [3]. The convergence in turn depends on the control strategy and the good tuning of control parameters (gains).

In the literature, several models of mechanisms assumed to move in a plane (planar bipeds) have been considered with feet [4], and without feet [5]. The first models are more realistic since important effects of the reaction force on the ground can be considered when the feet are included. Through the years several control techniques have been used to control the biped gait. Among them, one of the most helpful is *sliding mode control* [6]. This control technique stabilize electromechanical systems subject to external perturbations. Classical sliding mode control has been used in [5] but with the issue of yielding discontinuous control signals. In the literature the called second-order sliding mode control [7] has been proposed to solve this issue. These control techniques require a tunning of their parameters to achieve their control objectives. Usually, this tunning is performed by an expert by trial-and-error based on the performance of the convergence of some important variables. This task is complicated and time consuming; a bad tunning can yield that the robot is not able to walk [3]. Thus, an automatic strategy to determine appropriate control parameters is desirable in walking control.

In the field of optimization, in the last fifty years, several different problem solving methods belonging to the group of metaheuristics techniques have appeared. Metaheuristics techniques are search procedures that are based on relatively simple rules inspired in nature [8], that have presented great performances in solving complex problems (e.g. multi-modal or non-linear problems). These techniques try to leave from local optima, orienting the search at every moment depending on the evolution of the search process. In general, the use of metaheuristics algorithms in humanoid robotics seems well motivated, since these methods can be implemented even in cases where a complete dynamic model of the system under study is either not available or too complex to be useful [9]. Comparing metaheuristics with classical optimization processes in robotics [10], metaheuristics allow great freedom when modeling a fitness function. One of the most widely applied method in the field of continuous optimization is Differential Evolution (DE), this is a simple yet efficient population-based metaheuristic. Since its inception, it has yielded remarkable results in several optimization competitions [11]. In addition, it has been successfully applied to demanding practical optimization problems [12]. DE has been used successfully in the field of robotics. For example, it was used to optimize the movement of a small bipedal robot in locomotion on flat terrain [13], and in more complex routes, e.g., crossing obstacles [14].

This paper addresses the problem of optimizing control parameters of a given walking control scheme for a planar biped with feet. The optimization parameters correspond to control gains of a torque control with auxiliary input given by a continuous second order sliding-mode control. We propose to use DE in order to have a generic optimization method

The first two authors have been partially supported by Conacyt [grant 220796].

978-1-5386-4626-7/18/$31.00 ©2018 IEEE

not dependent on mathematical characteristics of the fitness function, like differentiability. We have compare the results for three different mono-objective fitness functions and two different encodings of optimization variables. We have found that our automatic selection of control gains have improved the performance of the walking task with respect to a solution given by an expert and reported in [3]. A contribution with respect to other works on optimization of walking control is the explicit management of important constraints like joint limits, center of pressure and energy consumption in a mono-objective formulation. This management of physical constraints has allowed to obtain better results than those obtained by an expert without an arbitrary increment of the energy consumption.

The paper is organized as follows. Section 2 summarizes the robot model, physical constrains for walking, and the feedback control used in this paper. Section 3 details the optimization problem, defining both the fitness function and the proposed optimization method. Section 4 shows results and statistics of the optimized parameters and simulations of the best cases of robot walking. Finally, in Section 5, we give conclusions.

II. Robot Model and control

This paper is based on the previous results of [3], where a complete model and walking control of a biped robot is detailed. In this section, we summarize the important points of that work to later specify the optimization aspects.

The biped robot considered in this work has the following physical features and is depicted in Fig. 1:

- It consists of 7 links (a torso, 2 tibias, femurs, and feet)
- Its motion is constrained to the sagittal plane, which is identified with a vertical xy-plane.
- It has 6 joints (2 ankles, 2 knees, and 2 hips), which are one-degree-of-freedom rotational actuated joints.

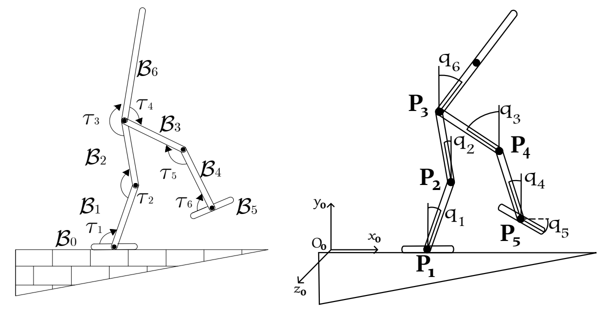
This structure implies that we have a fully actuated robot, meaning that each one of the joints has associated a torque τ_i (control input) as shown in Fig. 1(a) and a configuration angle q_i (see Fig. 1(b)). The simulation of the robot model requires to solve 6 second-order nonlinear differential equations.

A. Constraints for Walking

To guarantee the correct walking in all instants of time, the following important constraints must be considered.

Joints Constraints: These constraints refers to the configuration angles q_i, which must be limited and accomplish relationships between them to generate feasible robot configurations during walking. In [3] some joints constraints are given for the swing foot, however it is also necessary to add constraints for the angles of the support foot. Fig. 2 shows feasible and non-feasible configurations for the support foot, as well as the angles that are formed with these configurations. Therefore, adding these constraints to those described in [3], we obtain the following joint constraints that must be satisfied to obtain feasible walking configurations:

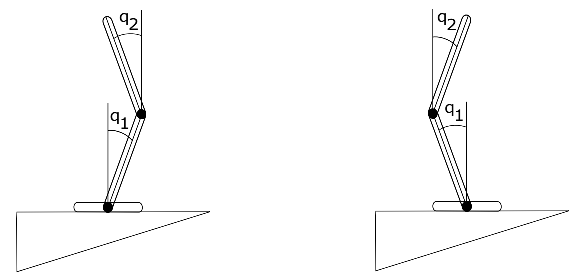
$$| q_i | < \frac{\pi}{2}, \quad q_4 < q_3, \quad q_1 < 0, \quad q_2 > 0, \quad q_2 \geq q_1 \qquad (1)$$



(a) Links $(\mathcal{B}_0, \cdots, \mathcal{B}_6)$, and torques $(\tau_1, \cdots, \tau_6)$.

(b) Joint's positions $(\mathbf{P}_1, \cdots, \mathbf{P}_5)$, angles of configuration $(q_1, \cdots, q_6)$ and fixed reference frame $\mathcal{F}_0$ with origin O_0.

Fig. 1: Scheme of the 7-link biped robot. Taken from [3].



(a) $q_2 > 0$, $q_1 < 0$, $q_2 \geq q_1$.

(b) $q_2 < 0$, $q_1 > 0$, $q_2 < q_1$.

Fig. 2: Different support foot configurations: (a) feasible and (b) non-feasible.

Dynamic Constraint: The model detailed in [3] can be validated by verifying the condition of the center of pressure (CoP). The CoP represents the point at which the distributed foot-ground reaction acts and must be bounded as follows:

$$\mathbf{CoP} := \frac{\tau_1}{\mathbf{R}_s^y} \in [-l_0, l_0] \qquad (2)$$

where l_0 is the mean length of the support foot's link, $\mathbf{R}_s^y$ is the vertical component of the resultant ground reaction force, and τ_1 is the torque applied in the support foot's ankle [3].

Energy Constraint: Additionally, we include an energy constraint, because it is an important factor to take into account. The energy can be calculated with the torque vector τ, as:

$$\text{energy} := \tau^T \cdot \tau \qquad (3)$$

where $\tau := \{\tau_1, \cdots, \tau_6\}$. We are interested in ensuring that the energy does not exceed an upper limit.

B. Walking Control

A set of virtual constraints (VC) on the joint's positions are added to guaranty a stable gait [3]. A VC is considered as the relationship between the joint's variables that are added through a feedback control [1]. The VC coordinate the evolution of the joints/links throughout a step, i.e., reduce the degrees of freedom to generate a closed-loop mechanism resulting in a desired periodic motion. The VC are imposed in our case by means of state-feedback input-output linearization [2]. These constraints have been designed in [3] in such a way that the torso is maintained nearly upright, the hips remain slightly in front of the midpoint between both feet, and the swing foot's ankle traces a parabolic trajectory. The VC are encoded in an output function $\mathbf{Y} \in \mathbb{R}^6$, defined as:

$$\mathbf{Y} := \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{bmatrix} := \begin{bmatrix} \mathbf{P}_5^x - (p^x + s) \\ \mathbf{P}_5^y - \rho(\mathbf{P}_5^x) \\ \mathbf{P}_3^x - (p^x + \mu s) \\ q_2 \\ q_5 \\ q_6 \end{bmatrix} \quad (4)$$

where $\mathbf{P}_3$ and $\mathbf{P}_5$ are the Cartesian coordinates of the hips and the swing foot's ankle, respectively. p^x represents the horizontal coordinate of the support foot's ankle, s and μ are constants such that $0 < s < 2l_1$ and $0 < \mu < 1$, and the function $\rho : \mathbb{R} \to \mathbb{R}$ defines a parabolic trajectory as follows:

$$\rho(u) := \frac{d^2}{s^2}(u - p^x)^2 + d \quad (5)$$

with d a constant such that $0 < d < 2l_1$, and l_1 is the mean distance of the support tibia. The parameters s, d, μ define the walking pattern and they are the step size, the maximum step height and a symmetry factor when both feet are on the ground, respectively.

1) Control Law: The biped robot considered in this work has a torque actuator at each joint. Computed torque control has been widely used for biped robots walking control, for instance in [15]. This kind of control allows canceling the non-linear terms of the robot model, keeping a linear relation between auxiliary control entries $\mathbf{v}$ and the outputs $\mathbf{Y}$. In our case, the computed torque linearizing control has the following form:

$$\tau := \beta^{-1}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{v} + \gamma(\mathbf{q}, \dot{\mathbf{q}}) \quad (6)$$

where the auxiliary control $\mathbf{v}$ must be adequately assigned to achieve convergence of the output vector $\mathbf{Y}$ and the nonlinear functions β and γ cancel the nonlinearities of the model (see [3] for more details). Applying this torque control, 6 decoupled linear systems of second-order with respect to the output vector $\mathbf{Y}$ are obtained. Thus, the auxiliary control must be design to yield convergence to zero of each element of the output vector $\mathbf{Y}$.

2) Auxiliary Control: We can use different options to define the auxiliary control, however, second-order sliding mode control [16] has the good property of achieving convergence in finite-time (unlike the asymptotic convergence of a proportional-derivative control) through continuous control signals. The auxiliary control used in this work, called *continuous twisting control* in [3], is given by:

$$\mathbf{v} = -\mathbf{K_1} \mid \dot{\mathbf{Y}} \mid^{\sigma} sign(\dot{\mathbf{Y}}) - \mathbf{K_2} \mid \mathbf{Y} \mid^{\frac{\sigma}{(2-\sigma)}} sign(\mathbf{Y}) \quad (7)$$

where $\mathbf{K_1}$ and $\mathbf{K_2} \in \mathbb{R}^{6\times6}$ are diagonal matrices formed by gains associated to every joint and $\sigma \in \mathbb{R}$. To guarantee stability in finite-time, $\mathbf{K_2} > \mathbf{K_1}$ (meaning that every entry of $\mathbf{K_2}$ must be greater than its corresponding entry of $\mathbf{K_1}$) and $0 < \sigma < 1$ must be accomplished according to [17]. Then, the control gains encoded in $\mathbf{K_1}, \mathbf{K_2}$ correspond to 12 unknown parameters that must be defined in some optimum sense to generate a feasible walking pattern.

III. Optimization Method for Control Parameters

Setting the control parameters to obtain a feasible robot walking is not an easy task, even for an expert. Thus, we focus on the automatic solution of the following problem: Given some physical parameters that define the geometric structure of the biped robot described in the previous section (link's lengths $l_0, ..., l_6$) and parameters that define the walking pattern (s, d, μ), the problem is to find the control parameters encoded in matrices $\mathbf{K_1}, \mathbf{K_2}$ that yield the *best convergence* of the output vector $\mathbf{Y}$ to zero, subject to fulfill constraints on joint's angles, CoP and energy. It is understood by best convergence a fast and uniform convergence of the outputs in finite time. This is because in the case of biped robots, the convergence of the outputs must be guaranteed in each step.

A. Encoding Optimization Parameters and Fitness Function

The optimization based on metaheuristics needs two elements, the first one is the encoding in real numbers of the parameters to optimize, the second one is a fitness function to know the performance of the solution obtained. Typically, in unconstrained optimization problems such a function is just the objective function. However, when dealing with constrained optimization problems, the objective function is usually modified to take into account the constraints. Additionally, in some cases the fitness function is altered to avoid some drawbacks such as the appearance of many local optima.

As described above, we have 12 variables to optimize, 6 elements for each gain matrix. To decide the best encoding, the solution proposed in [3] was analyzed. We noted that the matrices $\mathbf{K_1}$ and $\mathbf{K_2}$ have a similar order of magnitude. To accomplish the constraint $\mathbf{K_2} > \mathbf{K_1}$, we proposed to write $\mathbf{K_2} = (1 + \alpha)\mathbf{K_1}$ with $0 < \alpha < 1$. By doing this, two possibilities can be considered. The first is to handle a single α for the entire profit matrix $\mathbf{K_1}$. In this case, the problem becomes simpler because the number of parameters to be optimized is reduced. Then, the vector to optimize with a single α can be written as:

$$\mathbf{K} = \{k_1, k_2, k_3, k_4, k_5, k_6, \alpha\} \quad (8)$$

with $k_1 \cdots k_6$ the elements of the diagonal of matrix $\mathbf{K_1}$. The second option is to write each element of $\mathbf{K_2}$ as $k_{2_i} = (1 + \alpha_i)k_{1_i}$ with $0 < \alpha_i < 1$. The vector to optimize with multiple α's can be written as:

$$\mathbf{K} = \{k_1, k_2, k_3, k_4, k_5, k_6, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6\} \quad (9)$$

To verify if there is a significant difference between the encodings, both were tested. Regarding the fitness function, it was modeled as the following constrained minimization problem:

$$\begin{aligned}
&\min \ f(\mathbf{K}) = \frac{1}{N}\sum_{i}^{N} \parallel \mathbf{Y}(i) \parallel_n^n \\
&\text{subject to:} \\
&\quad -l_0 \leq CoP \leq l_0 \\
&\quad |\ q_i\ | < \frac{\pi}{2},\ q_4 < q_3,\ q_1 < 0,\ q_2 > 0,\ q_2 \geq q_1 \\
&\quad \text{energy} \leq \text{upper limit}
\end{aligned} \tag{10}$$

where N is the number of times the measurement was made throughout the walking, n refers to the degree of the norm, and to the power of the error, and *upper limit* is the maximum energy permitted. The objective function measures the norm of the convergence of the outputs.

To work in a simple way the problem of Eq. 10 the constraints were incorporated into the fitness function. We can rewrite the fitness function as:

$$\begin{aligned}
\min \ f(\mathbf{K}) =& \frac{1}{N}\sum_{i}^{N} \parallel \mathbf{Y}(i) \parallel_n^n + g(\mathbf{K})_{CoP} \\
&+ g(\mathbf{K})_{joint} + g(\mathbf{K})_{energy} \\
where:& \\
&g(\mathbf{K})_{CoP} = C_1 * \max(CoP_{ind}, 0) \\
&g(\mathbf{K})_{joint} = C_2 * V_{ind} \\
&g(\mathbf{K})_{energy} = C_3 * \max(e_{ind}, 0)
\end{aligned} \tag{11}$$

with:

$$\begin{aligned}
&CoP_{ind} = \max(\max(CoP), |\min(CoP)|) - l_0. \\
&V_{ind} = \begin{cases} w & \text{is the number of joint violations} \\ 0 & \text{otherwise} \end{cases} \\
&e_{ind} = \text{energy} - \text{upper limit} \\
&C_1, C_2, C_3 = \text{positive scalars}
\end{aligned}$$

Three different variants of the fitness function were considered. The first was using the Euclidean norm ($n = 2$). Following the hypothesis that the outputs of the system converge in finite time, a fourth power was implemented ($n = 4$), looking for the outputs to converge faster. Finally, in the last one, with the aim of trying to make the outputs converge as uniformly as possible, the infinite norm was implemented (∞-norm, $n = 1$). We refer to uniform convergence to the action of the outputs converging to the reference value at the same time, regardless of their initial conditions.

B. Differential Evolution (DE)

DE [18] is a quick and simple technique that works well in a wide variety of problems. Algorithm 1 describes the procedure of DE. The different stages of DE are described below:

1) Initialization: A population $P_{x,0}$ of NP D-dimensional parameter vectors $\mathbf{x}_{i,0} = \left[x_{i,0}^1, \cdots, x_{i,0}^D\right]$, $i = 1, \cdots, NP$ is randomly generated between a lower and upper bound $\mathbf{b}_L = \left[b_L^1, \cdots, b_L^D\right]$ and $\mathbf{b}_U = \left[b_U^1, \cdots, b_U^D\right]$.

2) Mutation: For each target vector $\mathbf{x}_{i,g}$ in the g^{th} generation, a mutant vector $\mathbf{v}_{i,g} = \left[v_{i,g}^1, \cdots, v_{i,g}^D\right]$ is created. Several different ways of creating mutant vectors have been devised. In this paper the *rand/1* method, described in Eq. 12 is used. In this process $F \in (0, 1)$ is a positive scalar related to the velocity of convergence and to the strength of perturbation, whereas $r0$, $r1$, $r2$ are randomly chosen integers in $[1, NP]$ such that $r0 \neq r1 \neq r2 \neq i$.

$$\mathbf{v}_{i,g} = \mathbf{x}_{r0,g} + F \cdot (\mathbf{x}_{r2,g} - \mathbf{x}_{r1,g}) \tag{12}$$

3) Cross: After the mutation phase, a crossing operation is applied between each target vector $\mathbf{x}_{i,g}$ and its corresponding mutant vector $\mathbf{v}_{i,g}$ to generate a trial vector $\mathbf{u}_{i,g} = \left[u_{i,g}^1, \cdots, u_{i,G}^D\right]$. Eq. 13 shows the crossing process where $CR \in (0, 1)$ controls the fraction of the values of the parameters copied from the mutant vector and j_{rand} is a randomly chosen integer in the interval $[1, D]$ that ensures that at least one variable is taken from the mutant vector.

$$u_{i,g}^j = \begin{cases} v_{i,g}^j & \text{if } (rand(0,1) \leq CR) \text{ or } (j = j_{rand}) \\ x_{i,g}^j & \text{otherwise} \end{cases} \tag{13}$$

4) Selection: The performance of the vectors $\mathbf{u}_{i,g}$ and $\mathbf{x}_{i,g}$ are compared with the purpose of knowing the individual that will pass to the next generation by using the fitness function. Eq. 14 shows this process.

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g}, & \text{if } (f(\mathbf{u}_{i,g}) \leq f(\mathbf{x}_{i,g})) \\ \mathbf{x}_{i,g}, & \text{otherwise} \end{cases} \tag{14}$$

IV. Results

In this section we detail the experiments carried out to validate the performance of our proposals. Results are obtained from an implementation of the biped model and the DE algorithm in Python. The robot's parameters are the same as those in [3]. Due to the time required for evaluation of the fitness function (near to 2 minutes), we decided to carry out executions with a small population. The stop condition was established by number of generations, doing 30 generations in each execution, and it was carried out 30 times. For the fitness function described in Eq. 11, we used the following values $C_1 = 1e3$, $C_2 = 1$, $C_3 = 10$. As it was mentioned in Section III, 2 encodings were implemented, and 3 fitness functions, and thus, 6 configurations are obtained. The different configurations are denoted by α_{quad}, α_{max}, α_{pow4}, α_{quad}^s, α_{max}^s, α_{pow4}^s. Being α_{***} the configurations with a single α (Eq. 8), and α_{***}^s the configurations with multiple α's (Eq. 9). "quad" the configuration of quadratic error ($n = 2$), "max" the configuration with infinity norm, and "pow4" the configuration of the fourth power ($n = 4$).

In [3] a solution obtained by a human expert is presented for the same control scheme. The control gains reported in [3] for $\sigma = \mathbf{0.85}$ are $\mathbf{K_1} = \mathbf{diag}(\mathbf{2}, \mathbf{2}, \mathbf{2.5}, \mathbf{3.5}, \mathbf{2}, \mathbf{2})$ and $\mathbf{K_2} = \mathbf{diag}(\mathbf{2.01}, \mathbf{2.1}, \mathbf{2.51}, \mathbf{3.6}, \mathbf{2.5}, \mathbf{2.5})$. Evaluating these gains with our fitness function with $n = 2$, we obtain the value $\mathbf{0.06506546}$ for the fitness function, with an energy

Algorithm 1: Differential Evolution

Set $g \leftarrow 0$
Generate an initial population $P_{\mathbf{x},0}$
repeat
  Mutation Stage.
  for *i=1 to NP* **do**
    Generate a mutant vector $\mathbf{v}_{i,g}$
  end
  Crossing Stage.
  for *i=1 to NP* **do**
    j_{rand}=**rand**$(1, D)$
    for *j=1 to D* **do**

$$\mathbf{u}_{i,g}^{j} = \begin{cases} \mathbf{v}_{i,g}^{j} & \textbf{if}(\textbf{rand}(0,1) \leq CR) \\ & \textbf{or } (j = j_{rand}) \\ \mathbf{x}_{i,g}^{j} & \textbf{otherwise} \end{cases}$$

    end
  end
  Selection Stage.
  for *i=1 to NP* **do**
    if $f(\mathbf{u}_{i,g}) \leq f(\mathbf{x}_{i,g})$ **then**
      $\mathbf{x}_{i,g+1} = \mathbf{u}_{i,g}$
      $f(\mathbf{x}_{i,g+1}) = f(\mathbf{u}_{i,g})$
      if $f(\mathbf{u}_{i,g}) \leq f(\mathbf{x}_{best,g})$ **then**
        $\mathbf{x}_{best,g} = \mathbf{u}_{i,G}$
        $f(\mathbf{x}_{best,g}) = f(\mathbf{u}_{i,g})$
      end
    end
    else
      $\mathbf{x}_{i,g+1} = \mathbf{x}_{i,g}$
    end
  end
  g=g+1
until *to stop criteria*;
return $\mathbf{x}_{best,G}$

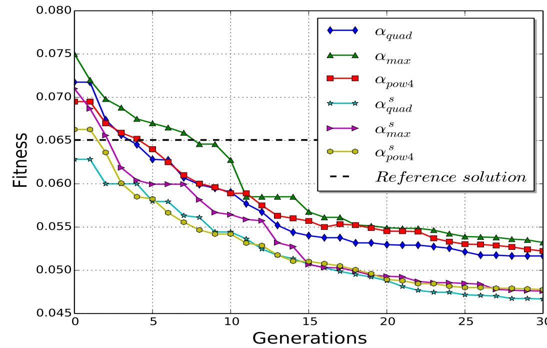


Fig. 3: Median of the fitness of all configurations.

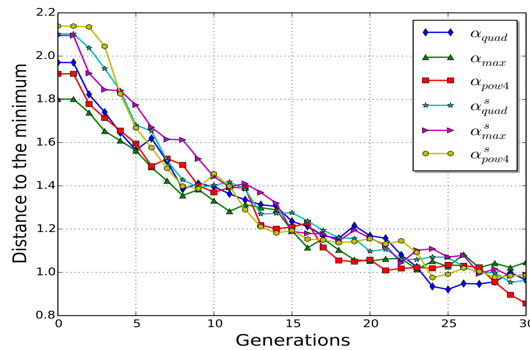


Fig. 4: Average diversity of all configurations.

consumption of **328899**. This energy value was used as *upper limit* for the variants of the fitness function in Eq. 11. This solution was obtained by trial and error by the expert, and it was used as reference solution. As aforementioned, a complete execution of the optimization algorithm has a very high cost, which led us to leave as future work a study on the control parameters of the DE algorithm. Therefore, we decided to use the values NP (population size)= 30, $CR = 0.8$ and $F = 0.5$. The parameters CR, and F have an important effect on the solution quality and the speed of convergence. Parameters were chosen looking for a *fast* convergence without losing quality of the optimization process. A standard value of CR was chosen, and the value of F is a little low compared to the usual ones.

Model	Min	Max	Mean	Median	S.D
α_{quad}	4.89e-2	5.53e-2	5.18e-2	5.16e-2	1.46e-3
α_{max}	4.98e-2	2.19	1.29e-1	5.22e-2	3.96e-1
α_{pot4}	4.99e-2	5.74e-2	5.18e-2	5.16e-2	1.34e-3
α^{s}_{quad}	4.53e-2	4.86e-2	4.66e-2	4.66e-2	8.94e-4
α^{s}_{max}	4.57e-2	5.16e-2	4.78e-2	4.75e-2	1.69e-3
α^{s}_{pow4}	4.59e-2	4.93e-2	4.75e-2	4.77e-2	8.56e-4

TABLE I: Results obtained for all configurations.

Despite the implementation and use of three different fitness functions in our algorithm, all the results are presented under the $n = 2$ metric, so that they can be properly compared. Table I shows a summary of the fitness obtained by the six configurations. It is appreciated that despite the few generations, the results obtained have good quality, with values in median lower than the reference value of [3] in all the evaluated configurations of encoding and fitness function. The best individuals for each configuration are presented in Table II. The evolution of the fitness for the six configurations can be seen in Fig. 3. For each generation, the median of the fitness of the best individual is shown, taking into account 30 independent executions. It can be seen that the 2 encodings are separated in the figure and that the configurations with several α's reach a fitness lower than those using a single α. We can say that although the configurations of one α present better results than in [3], the use of more parameters leads to much better results. The diversity of the population is shown in Fig. 4. It was measured as the mean distance to the closest individual in each generation [19]. As expected, diversity decreases globally throughout generations. However, the number of generations are not enough to reach convergence, i.e., reaching a degree of diversity close to zero. This means that probably, by altering the DE parameters to induce a faster convergence or by performing larger executions, higher-quality solutions might be obtained. However, this is left as future work. To have a clearer notion of which configuration is better, a statistical comparison was performed among them. A

Model	k_1	k_2	k_3	k_4	k_5	k_6	α_1	α_2	α_3	α_4	α_5	α_6	Fitness	Rank
α_{quad}	9.28e-1	2.95	3.02	4.99	6.05	1.7	3.36e-1						4.89e-2	4
α_{max}	1.16	3.80	3.65	5.75	3.08	7.47	7.85e-2						4.98e-2	5
α_{pot4}	1.02	3.37	3.18	8.22	3.96	3.31	2.40e-1						4.99e-2	6
α^s_{quad}	1.07	1.53	2.32	4.14	3.67	3.09	6.94e-2	8.89e-1	9.63e-1	4.03e-1	3.77e-1	1.15e-1	4.53e-2	1
α^s_{max}	1.23	3.53	1.96	1.43	3.82	7.64e-1	3.07e-2	9.35e-1	9.19e-1	4.08e-1	4.23e-1	4.26e-1	4.57e-2	2
α^s_{pow4}	9.25e-1	1.71	1.85	5.68	1.35e-1	4.98	1.91e-1	8.44e-1	4.45e-1	4.32e-1	3.04e-1	1.92e-1	4.59e-2	3

TABLE II: Best individual of each configuration.

	α_{quad}	α_{max}	α_{pot4}	α^s_{quad}	α^s_{max}	α^s_{pow4}	SR
α_{quad}	-	↑	↔	↓	↓	↓	1.0
α_{max}	↓	-	↓	↓	↓	↓	0.96
α_{pot4}	↔	↑	-	↓	↓	↓	1.0
α^s_{quad}	↑	↑	↑	-	↑	↑	1.0
α^s_{max}	↑	↑	↑	↓	-	↔	1.0
α^s_{pow4}	↑	↑	↑	↓	↔	-	1.0

TABLE III: Statistical Comparison of the different configurations.

similar guideline as the one applied in [20] was used, where, for this case, the non-parametric *Kruskal-Wallis* test is used to compare the medians of the results obtained. Table III shows the comparison of the different configurations with a significance level of 5%. The success rate (SR) shown in the last column is measured with respect to [3]. It can be seen that configurations with several α's have a statistical superiority to the rest. Also being α^s_{quad} superior to all.

We considered important to verify qualitatively the behavior of the outputs of the optimization, as well as the evolution of the CoP and the trajectory of the steps that the robot took along the simulation with the results shown in the Table II. Figs. 5, 6 and 7 show these 3 aspects for the configurations α^s_{quad}, α^s_{max} and α^s_{pow4} respectively. In the graphs (a) of each figure, the behavior of the outputs of the system over time is appreciated. In general, there is good convergence to zero in all these graphs, however the outputs in Fig. 5(a) are better than the rest. The outputs in that figure converge closer to zero in a smoother way (without overshooting) than with the other configurations. Meanwhile in Fig. 6(a), the outputs that must keep at zero oscillate a little, and can be appreciated that the outputs do not converge completely. In addition, the CoP in the Fig. 5(b) has a smoother behavior compared to the Fig. 6(b) and 7(b). Finally, the graphs (c) show the desired trajectory in dotted line and the trajectory described by each foot in continuous line. A closer tracking to the parabolic reference in the movement of the swing foot can be seen in Fig. 5(c) compared to the other subfigures (c).

V. Conclusions

In this paper, a metaheuristic (differential evolution) has been implemented to automatically tune the control parameters to allow a bipedal robot to walk with a second-order sliding mode control. To do this, we propose and compare three different mono-objective fitness functions and two different encodings of optimization variables We can see from the results that the proposed optimization models are good enough and they are not very different from each other. Configurations with several α's present better fitness values than those with only one α. This is because by increasing the number of parameters to be optimized (several α's), the search space is bigger, allowing a greater number of feasible configurations with better fitness values. Besides, better fitness values were obtained using the quadratic fitness function and qualitatively the behavior of the robot walking is better with that optimization model. We consider that the explicit management of constraints as joint limits, center of pressure and energy consumption has allowed to obtain much better results than those obtained by an expert. We are currently working on addressing the same problem in a multi-objective fashion, where energy is not more a constraint but a second objective.

References

[1] B. Siciliano and O. Khatib, *Springer handbook of robotics*. Springer Science & Business Media, 2008.

[2] E. R. Westervelt, C. Chevallereau, J. H. Choi, B. Morris, and J. W. Grizzle, *Feedback control of dynamic bipedal robot locomotion*. CRC press, 2007.

[3] J. E. Machado, H. M. Becerra, and M. Moreno Rocha, "Modeling and finite-time walking control of a biped robot with feet," *Mathematical Problems in Engineering*, vol. 2015, no. Article ID 963496, p. 17, 2015.

[4] Q. Lu and J. Tian, "Research on walking gait of biped robot based on a modified cpg model," *Mathematical Problems in Engineering*, no. Article ID 793208, p. 9, 2015.

[5] M. Nikkhah, H. Ashrafiuon, and F. Fahimi, "Robust control of underactuated bipeds using sliding modes," *Robotica*, vol. 25, no. 3, pp. 367–374, 2007.

[6] V. Utkin, J. Guldner, and J. Shi, *Sliding mode control in electromechanical systems*. CRC press, 2009.

[7] Y. Orlov, Y. Aoustin, and C. Chevallereau, "Finite time stabilization of a perturbed double integratorpart i: Continuous sliding mode-based output feedback synthesis," *IEEE Transactions on Automatic Control*, vol. 56, no. 3, pp. 614–618, 2011.

[8] E. Hopper and B. C. Turton, "An empirical investigation of metaheuristic and heuristic algorithms for a 2d packing problem," *European Journal of Operational Research*, vol. 128, no. 1, pp. 34–57, 2001.

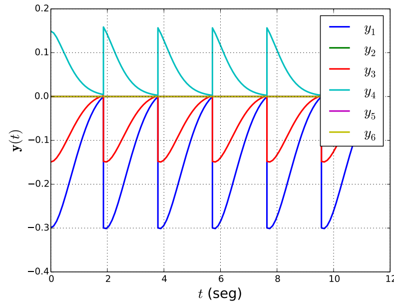
[9] K. Wolff and P. Nordin, "Evolution of efficient gait with humanoids using visual feedback," in *IEEE-RAS International Conference on Humanoid Robots*, pp. 99–106, 2001.

[10] T. Narukawa, M. Takahashi, and K. Yoshida, "Efficient walking with optimization for a planar biped walker with a torso by hip actuators and springs," *Robotica*, vol. 29, no. 4, pp. 641–648, 2011.

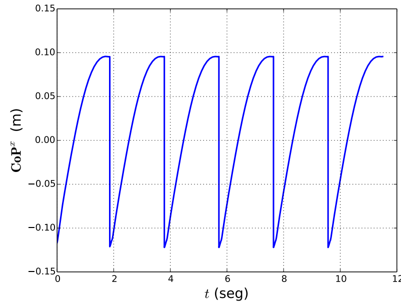
[11] S. Das and P. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, pp. 4–31, Feb 2011.

[12] J. Zhao, Y. Xu, F. Luo, Z. Dong, and Y. Peng, "Power system fault diagnosis based on history driven differential evolution and stochastic time domain simulation," *Information Sciences*, vol. 275, pp. 13 – 29, 2014.

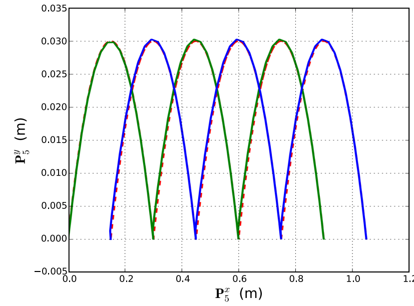
[13] T. Nguyen, L. Tao, and H. Hasegawa, "Gait generation for a small biped robot using approximated optimization method," in *IOP Conference Series: Materials Science and Engineering*, vol. 157, p. 012009, IOP Publishing, 2016.



(a) Evolution of system outputs.

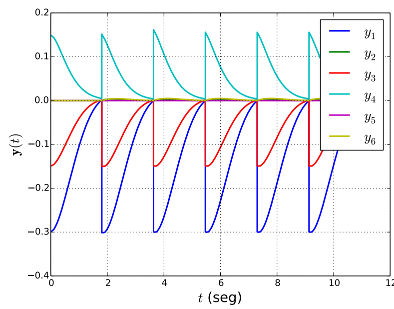


(b) Evolution of the CoP.

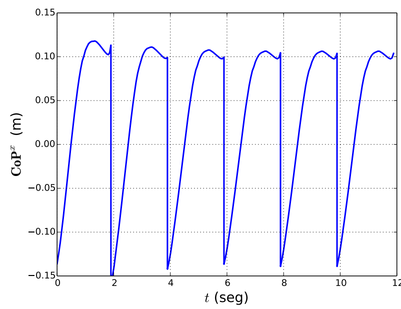


(c) Comparison of the trajectories described by the feet against the reference trajectory.

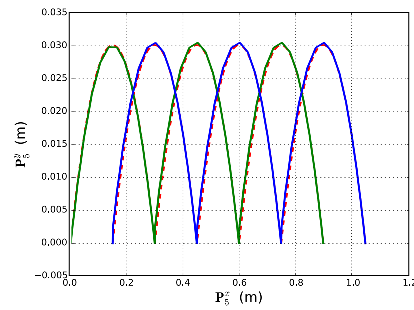
Fig. 5: Simulation results of α^s_{quad} configuration.



(a) Evolution of system outputs.

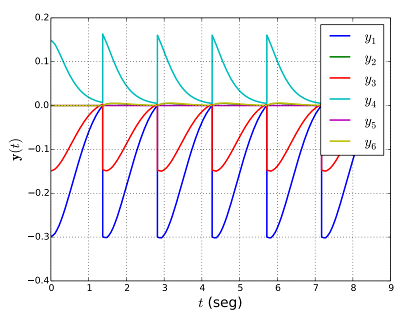


(b) Evolution of the CoP.

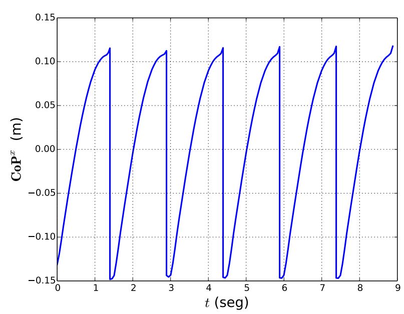


(c) Comparison of the trajectories described by the feet against the reference trajectory.

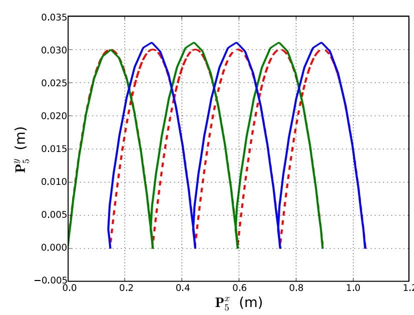
Fig. 6: Simulation results of α^s_{max} configuration.



(a) Evolution of system outputs.



(b) Evolution of the CoP.



(c) Comparison of the trajectories described by the feet against the reference trajectory.

Fig. 7: Simulation results of α^s_{pow4} configuration.

[14] P. Vundavilli, "Obstacle crossing gait generation of a two-legged robot using differential evolution trained neural networks," *International Conference on Production and Mechanical Engineering*, pp. 46–51, 2014.

[15] C. M. C. O. Valle, R. Tanscheit, and L. A. F. Mendoza, "Computed-torque control of a simulated bipedal robot with locomotion by reinforcement learning," in *IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pp. 1–6, 2016.

[16] W. Perruquetti and J.-P. Barbot, *Sliding mode control in engineering*. CRC Press, 2002.

[17] Y. Orlov, "Finite time stability and robust control synthesis of uncertain switched systems," *SIAM Journal on Control and Optimization*, vol. 43, no. 4, pp. 1253–1271, 2004.

[18] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of global optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[19] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, p. 35, 2013.

[20] C. Segura, C. A. C. Coello, E. Segredo, G. Miranda, and C. León, "Improving the diversity preservation of multi-objective approaches used for single-objective optimization," in *IEEE Congress on Evolutionary Computation (CEC)*, pp. 3198–3205, 2013.