A Control Design with Differential Evolution Algorithm for a Screw-actuated Robotic Leg

Nestor Eduardo Nava Rodríguez[1], Luis Moreno Lorente[1], Giuseppe Carbone[2], Marco Ceccarelli[2]
[1]LARM: Laboratory of Robotics and Mechatronics
DiMSAT – University of Cassino, Italy
[2]Robotics Lab. Carlos III University, Spain
[2]carbone@unicas.it, [2]ceccarelli@unicas.it; [1]nnava@ing.uc3m.es, [1]moreno@ing.uc3m.es

***Abstract*—In a recent past, research activities have been carried out for developing a six-legged robot at LARM: Laboratory of Robotics and Mechatronics in Cassino. The proposed designs have been conceived with the aim of achieving suitable mechanism structures and architectures with low-cost and user-friendly features. Simulations and experimental tests have been developed by referring to a novel leg design in order to characterize both the walking operation and design characteristics for a suitable control strategy. In this work, we propose a control architecture that has been set up by using a proper algorithm based on evolution strategies. Successful simulation results have been reported to validate the proposed control architecture.**

Keywords—Control architecture, walking robots, robotic leg, evolution strategy, differential evolution

I. INTRODUCTION

Walking machines have been attempted as technological transportation machinery since the aim to overpass the limits of wheeled systems by looking at legged solutions in nature, [1]. But only in the recent past efficient walking machines have been conceived, designed, and built with performances that are suitable for practical applications, [2-5]. Biped, quadruped and hexapod as well as humanoid robots has been developed in the last part of 20-th century around the world in research centres and universities as machines that can help the human being in dangerous or exhausting tasks, for example, transportation of military staff, [6], mine detection and grass cutting, [7-8], in-pipe inspection, [9-10], or planetary exploration, [11].

Mobile robots can have different architectures that can be equipped either with crawlers/wheels or with biologically inspired legs. This second type of walking machines can be slow and more difficult to design and operate with respect to the first ones. Nevertheless, legged robots are more suitable for rough terrain, where obstacles of any size can appear [12]. In fact, the use of wheels or crawlers limits the size of the obstacle that can be climbed to half the diameter of the wheels. On the contrary, legged machines can overcome obstacles that are comparable with the size of the machine leg. There is also a third type of waking machines that is called hybrid robot since it has legs and wheels at the same time. This type of walking machines may range from wheeled devices to true walking machines with a set of wheels. In one case, the suspensions are arms working like legs to overcome particularly difficult obstacles, and in other case, wheels are used to enhance the speed when moving on flat terrain.

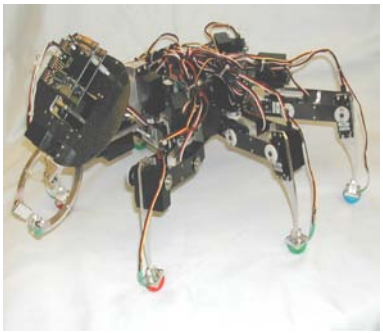
A hexapod robot is a mechanical vehicle that walks by means of six legs. Significant prototypes of hexapod robots can be the multi-legged robot with articulated body "SLAIR2" which has been developed at the Fraunhofer Institute for Factory Operation Ottovon-Guericke University in Magdeburg, Germany [13], Figure 1a). This robot consists of three modular segments that are linked to each other through two DOF joints and 6 legs. Based on anatomy from ants and leg coordination from stick insects, the BILL-Ant-p robot, Figure 1b), was designed by William Lewinger [14]. This is an actively compliant 18-DOF hexapod robot with six force-sensing feet, a 3-DOF neck and head, and actuated mandibles with force-sensing pincers for a total of 28 degrees of freedom. RHEx robot, Figure 1c), has been developed at Boston Dynamics, [5]. It has six legs with only one degree of freedom. The leg has a very simple design that recalls the operation of a wheel at fast rotation speeds. It is capable of achieving a wide variety of dynamically dexterous tasks, such as walking, running, leaping over obstacles, climbing stairs. Two on-board batteries allow a continuous operation time of about 10 min. Additional reference to other prototypes of walking robots from a rich literature can be found for example in [5]. In this paper, suitable control architecture, that includes a PID controller, is developed to achieve a proper hexapod gait by using an algorithm based on an ES, such as Differential Evolution (DE). Design, simulations and experimental tests have been reported by referring to a screw-actuated robotic leg. Simulations of the proposed control algorithm have been elaborated aiming to validate the effectiveness of the proposed control architecture.

II. THE ATTACHED PROBLEM

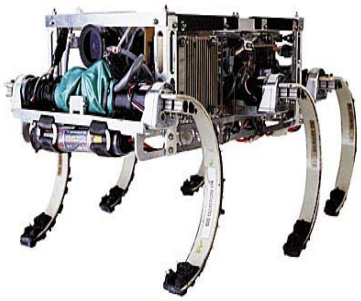
Stable, efficient and fast robots are needed to navigate uneven environments, which can help humans, for example in demining, pipe inspection, inspection and restoration of archaeological sites, and interplanetary exploration, [5, 15]. Since a robot can be statically stable on three or more legs, a hexapod robot has great capabilities during its motion. In fact, it can achieve statically stable walking even if one or two legs are out of order or not properly working. Most often, hexapods are programmed by specific gaits, which allow the robot to move forward, turn, and side-step, [4]. Several methods for hexapod gaits can be considered such as: alternating tripod gait, quadruped gait, one leg at a time. These gaits are typically stable, even in slightly rocky and uneven terrains. However, coordinating a large number of motors for even simple matters, like climbing stairs, can be

very complex/expansive resulting in hexapod robots being unsuitable for practical applications. Therefore, it is advisable to design hexapod robots having easy-operation and low-cost features. A low-cost hexapod robot prototype with legs and wheels has been developed at LARM in Cassino and named as Cassino Hexapod, [16-18].

a)



b)



c)

Figure 1 Examples of hexapod robots: a) SLAIR2, [13]; b) BILL-Ant-p, [14]; c) Rhex, [5].

The proposed design has been conceived with the aim of developing a walking leg by using mainly low-cost industrial components with the following basic requirements

- To have a robust simple mechanical design;

- To have a modular design that can be used for robots with different number of legs;

- To be easily operated with an easy flexible programming;

- To have low-cost both in design and operation.

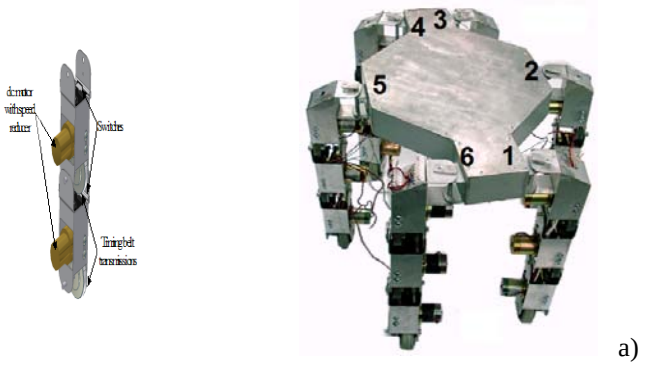
Those requirements can be achieved in a very practical way by using low-cost components from the market into a suitable design for the whole system. A modular mechanical design has been achieved by defining a single link module that can be easily connected with other modules and can have inside all the needed actuators, transmissions and sensors. Figure 2a) shows the proposed design for a single link module. Figure 2b) shows a built prototype of Cassino Hexapod based on the link module that is shown in Figure 2a). The walking machine can fit a cuboid of 60cm x 60 cm x 50 cm without any payload or PLC installation and it weights 18 kg and with the PLC it has an overall weight of 22 kg. The operation of the Cassino Hexapod is achieved by using PLC to control the different motors of the hexapod. A Siemens PLC Simatic S7-200 is used and is installed on board the hexapod. The program for the hexapod operation was written in a windows PC which has STEP-7/Micro WIN 32 installed into it. The program from the PC can be downloaded onto the flash memory of the PLC using a RS 232/PPI Cable. An external DC Power Supply is used to provide power to the hexapod. The feasibility of using the proposed hexapod has been, in particular, investigated for architecture analysis and survey of a historical site in [16] by referring to the Basilica of Montecassino Abbey that was built between 1066 and 1071. Other leg design solutions are still under investigation at LARM with the aim to further reduce complexity/costs and improve overall performance. Adequate movement of a hexapod robot is possible by properly synchronising the motion of all six legs. Thus, planning the movement of one leg is very important for the successful performance of the robot. Suitable walking gaits have been implemented for the hexapod locomotion. For maintaining the stability of the robot, at least three legs must be in contact with the ground simultaneously [2-3].

Figure 3 shows a scheme with movements of limbs in a six legs walking also with the footfall formula representation. The limbs that are in contact with the ground surface are shown as black circles in a table in which the entries represent the possible foot contacts with the ground.

The arrow in Figure 3 represents the moving forward direction of the hexapod walking. In the tripod gait, the front and the rear leg of one side and the middle leg of another side perform their swing movements at the same time. Thus, the swings of right and left tripods have to be synchronised by adding a software-induced delay. Each leg joint of the Cassino Hexapod has been designed as actuated by a DC motor which is activated by PLC, based on the digital logic. Operation of a particular joint will stop as soon as it reaches its extreme position, which is sensed by the limit switches. This has to be done to implement a continuous walk. Thus, the attached problem has considered how to achieve a feasible movement for a hexapod gait through low-cost structure composed of commercial components, mechanical pieces with no complex features and suitable mechanisms.

Several techniques exist to learn and optimize gaits based on objectives such as minimizing energy consumption and maximizing stability margin, speed, and learning rate. Neural Network (NN) [19-20] Reinforcement Learning (RL) [21], imitation-based approaches [19] and Genetic Algorithm GA [22] are the learning and optimization tools used in gait synthesis. NN is a widely used technique for gait generation. Unsupervised and supervised learning methods are adopted in training NN. RL (unsupervised) and human motion capture

data and GA (supervised) are tools useful to train NN for gait generation. In the unsupervised approach the learning process is dependent on the feedbacks from the training environment [19]. Supervised training of NN requires large number of training data for generalization. RL relies on sensory feedback from the environment [21]. The associated learning process should utilize enough training data to enhance the generalization capabilities of the learned gaits, which is a tedious task. It is desirable to utilize the kinematic and dynamic models for faster dynamic walking. Evolution Strategy (ES) is a powerful tool to resolve the issues related to the optimality of gaits. ES is utilized for gait generation by minimizing a weighted cost function of the input energy and ZMP error. ES is useful for path planning since it can produce smooth movements. Therefore, this optimization technique have been using in this work for obtaining an optimized control architecture that manages the Cassino hexapod gait.



a)

b)

Figure 2 Cassino Hexapod: a) a 3D model of a modular leg; b) an Assembled Cassino Hexapod prototype at LARM.

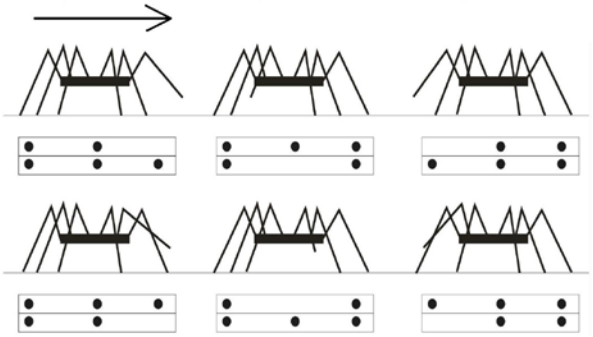
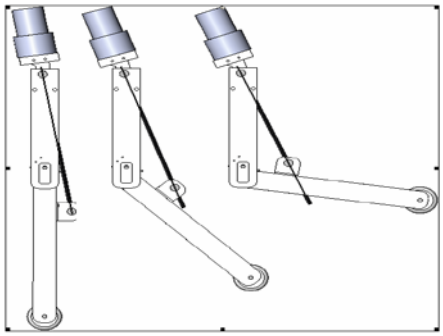
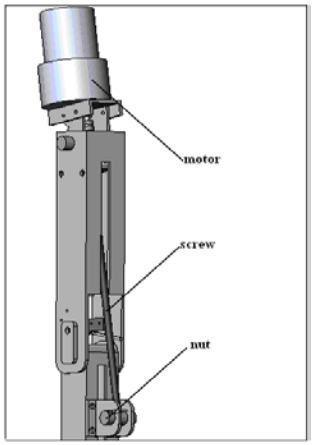


Figure 3 Movements of limbs in a six legs walking with footfall formulas representation (black circles stands for the limbs in contact with the ground surface and arrow indicates moving forward direction).

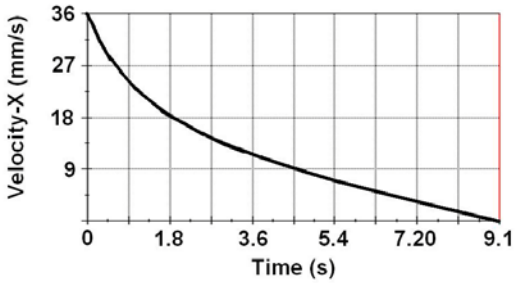


a)

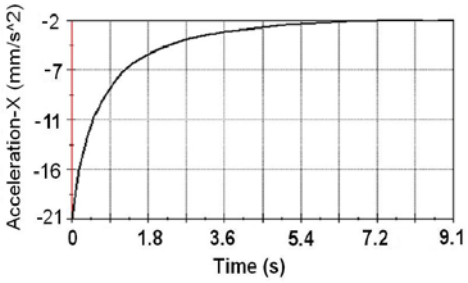


b)

Figure 4 A new leg design: a) a scheme with feasible motion ranges; b) a zoom view with location of motor and screw-nut transmission.



a)



b)

Figure 5 Plots of simulation results for the distal module as function of time: a) velocity; b) acceleration.

III. A NEW ROBOTIC LEG

A new leg has been designed as shown in Figure 4 with a section of the modules 35,5 x 35,5 mm and 29,5 mm x 29,5 mm. The leg is composed of 2 modules and has only one wheel. Simulations have been carried out within Cosmos Motion Software [23]. A first test has consisted in simulating the leg motion that is shown in Figure 4a). Simulation results are reported in Figures 5 and 6. In particular, Figure 5a) shows the plot of velocity of the distal module versus time for the leg motion that is shown in Figure 4a). Similar, Figure 5b) shows the plot of acceleration of the distal module versus time. Figures 6a) and 6b) show the plots of the moments due to friction at proximal and distal module, respectively. Figure 6c) shows the plot of the power consumption for the leg motion that is shown in Figure 3. Other simulations have been carried out in other operation cases. All the simulation results have shown suitable behaviours with good dynamic performance and limited power consumption. Additionally

the use of the screw-nut transmission does not require brakes to keep a desired position, since it does not allow the motion inversion. This aspect allows a significant weight and cost reduction together with a more user-friendly operation. Therefore, a prototype of the new leg has been built at LARM in Cassino. The new leg consists of two modules and the total height of the ground is 540 mm and is has a weight of 850 grams and the maximum step size is 200 mm.

The leg contains one motor, a screw-nut transmission, limit switcher, support for the motor, proximal module, distal module and ball bearings for reduce the friction in the joints as shown in Figure 7. Various experimental tests have been carried out and experimental measures have been obtained for the input torque and accelerations at distal module measurements. In particular, Figure 8 shows the plot of the measured accelerations for the distal module with a maximum acceleration of less than 60 m/s2. Figure 9 shows the plots of the measured motor torque versus time when no load is applied, Figure 9a), when a load of 300 grams is applied, Figure 9b), when a load of 676 grams is applied, Figure 9c). All the experimental tests have shown suitable results with motor torque below the nominal motor torque that is 1200 mNm.

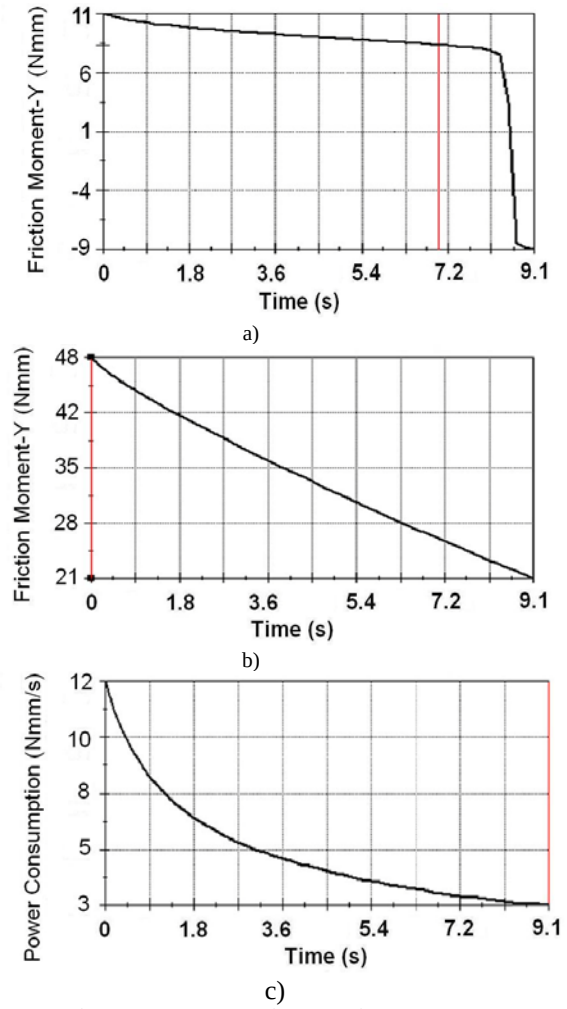


Figure 6 Plots of simulation results as function of time: a) moment due to friction between hinges at proximal module; b) moment due to friction between nut and nut support frame; c) motor power consumption.

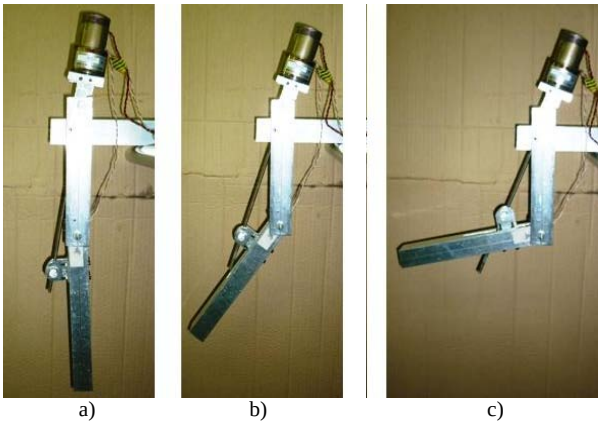


Figure 7 Sequence of images showing the operation of the built robotic leg: a) fully stretched configuration; b) intermediate configuration; c) fully bent configuration.

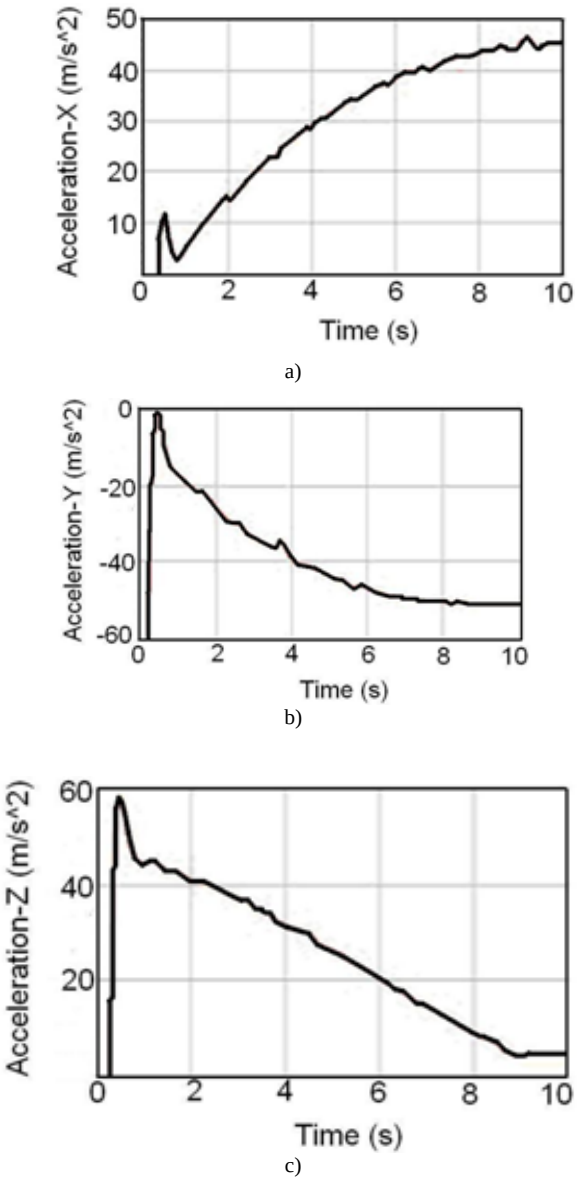


Figure 8 Measured accelerations versus time for the distal module: a) component along X-axis; b) component along Y-axis; c) component along Z-axis.

IV. A CONTROL ARCHITECTURE FOR THE LEG PROTOTYPE

Suitable control architecture has been design by considering the results of operating simulations of the virtual leg and experimental tests on the built prototype. Dynamic analyses have been computed for the operation of Motors 1 and 2. Figure 10 shows the schemes that have been used for the dynamic analyses. In particular, Figure 10a) shows the scheme used for the dynamic analysis of Motor 1. In this case, Motor 1 moves the whole leg structure that can be considered as a mass concentrated in the centre of mass. The vector R is the distance from the global frame to the centre of mass and θR is the angle of R with the vertical axis. Both depend of the angular displacement θ1 and θ2 of Motor 1 and 2, respectively. From the scheme in Figure 10a), the differential equations for the motor 1 can be determined in the form

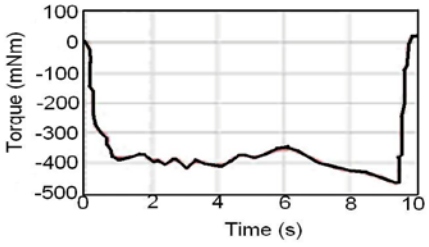
$$P_T \sin(\theta_R)R + J\frac{\partial \dot{\theta}_R}{\partial t} = K_2 i_a \qquad (1)$$

$$L_a \frac{\partial_{ia}}{\partial t} + R_a i_a + K_3 \frac{\partial \theta_R}{\partial T} = K_1 e_V$$

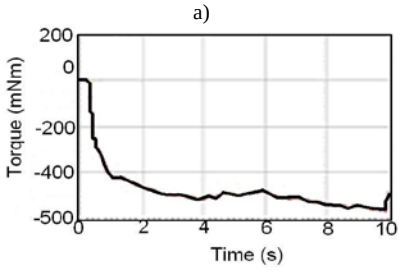
where P_T the whole leg weight, J is the whole leg moment of inertia, K_2 is the motor 1 torque constant, i_a is the motor 1 current, L_a is the motor 1 capacity, R_a is the motor 1 resistance, K_3 is the constant of the electromotive force, K_1 is the motor 1 gain and e_v is the motor 1 input voltage. θ_R represents the system output and e_v represents the system input.

The electromotive force constant K3 has been computed as the division between the angular velocity ωR and the motor 1 input voltage ev and the motor gain K1 will be computed through the control architecture. The motor 1 is a DC motor RS 235-7780 [24] with the parameter

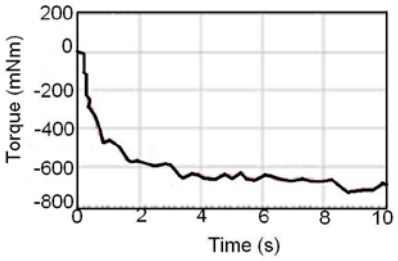
- $P_T = 8.50$ N;
- $J = 1.84e{-}4$ Kg∗m^2;
- $K_2 = 31.10e{-}3$ N∗m/A;
- $i_\alpha = 64.31$ A;
- $L_\alpha = 0.95e{-}3$ H;
- $R_\alpha = 7.40\ \Omega$.



a)

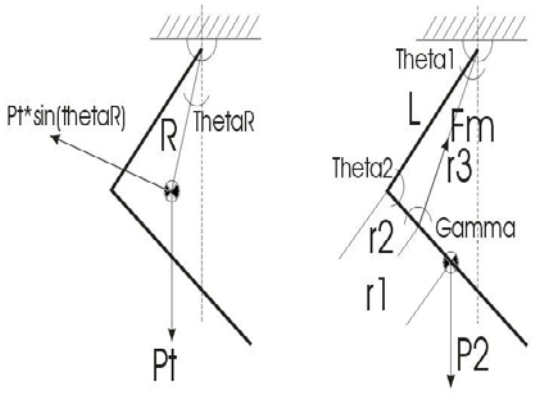


b)



c)

Figure 9 Diagrams of the measured motor torque versus time: a) with no load; b) with a load of 300 grams; c) with a load of 676 grams.



a) b)

Figure 10 Dynamic schemes for the leg actuation: a) scheme for motor 1; b) scheme for motor 2.

Note that the differential equations (1) are not linear. Thus, a linearization is necessary for the proposed analysis. A classic linearization method, named state-space feedback linearization [25], has been used since it yields reliable results with no high computational efforts. For this aim, the state-space equations have been obtained from (1), as first step of this study, in the form

$$\dot{X}_1 = X_2$$

$$\dot{X}_2 = \frac{K_2}{J} X_3 - \frac{P_T R}{J} \sin X_1 \qquad (2)$$

$$\dot{X}_3 = \frac{K_1}{La} u - \frac{Ra}{La} X_3 - \frac{K_3}{La} X_2$$

where X_1 is the angular displacement θ_R, X_2 is the angular velocity ω_R, X_3 is the motor 1 current i_a and u is the system input e_v. The controllability and observability matrixes of (2) are

$$Mc = \begin{bmatrix} 0 & 0 \\ 0 & \frac{K_1 K_2}{JLa} \\ \frac{K_1}{L_a} & -\frac{RaK_1}{La^2} \end{bmatrix}$$

$$Mo = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \qquad (3)$$

Since both matrixes of (3) have ranges equal to 2, the system is controllable and observable to apply the state-space feedback linearization [26]. Subsequently, making a variable change of X1 for Z1 yields the expressions

$$\begin{aligned} \dot{Z}_1 &= \dot{X}_1 = Z_2 \\ \dot{Z}_2 &= \dot{X}_2 = \frac{K_2}{J} X_3 - \frac{P_T R}{J} \sin X_1 = Z_3 \\ \dot{Z}_3 &= \frac{K_2}{J} \dot{X}_3 - \frac{P_T R}{J} \cos X_1 Z_2 \end{aligned} \tag{4}$$

The state-space representation of (4) is

$$\begin{bmatrix} \dot{Z}_1 \\ \dot{Z}_2 \\ \dot{Z}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v \tag{5}$$

$$\theta_R = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} Z_1 & Z_2 & Z_3 \end{bmatrix}^T$$

Where ν is Z_3 and the transfer function is

$$\frac{\theta_R}{v} = \frac{1}{S^3} \tag{6}$$

Therefore, the new system input ν has the form

$$v = \frac{K_2 K_1}{J L_a} u - \frac{R_a}{L_a} Z_3 - \frac{P_T R R_a}{J L_a} \sin X_1 - \frac{K_3 K_2}{J L_a} Z_2 - \frac{P_T R}{J} \cos X_1 Z_2 \tag{7}$$

Figure 11 shows the scheme for the proposed control architecture. In particular, Figure 11a) shows the scheme for the whole control system and Figure 11b) shows the scheme for the motor 1 dynamics. These schemes have been modelled in Simulink environment [27] in order to develop simulations of the control operation; simulation results will be later reported. The Fig. 11 parameters u1, u2, u, R and y are vector R, input of θR, ν of eq. (7), feed-back value of θR and θR output, respectively. Note that the proposed control architecture contains a PID controller, in which the proportional Kp, derivative Kd and integrative Ki gains have been optimized by using an evolutionary method, as will be illustrated later in Section 5. Most of the controllers currently used in industrial applications include control strategies PID or PID modified [26]. Therefore, using a PID controller has been considered as the solution for the proposed architecture since it can be implemented in further practical applications with the built leg prototype of Fig. 7. The used PID structure is that Simulink includes as default [27] with the transference function

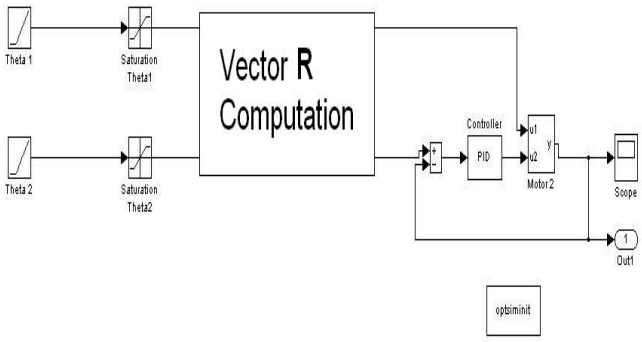
$$\frac{Y(s)}{U(s)} = K_p \left(1 + \frac{K_i}{s} + K_d s \right) \tag{8}$$

Similar to motor 1, motor 2 dynamics has been computed by using the scheme of Figure 10b), in which case motor 2 only moves the leg lower link. In Figure 10b), L is the upper link length and r3 is the length of the screw actuator. The differential equation of motor 2 has the form

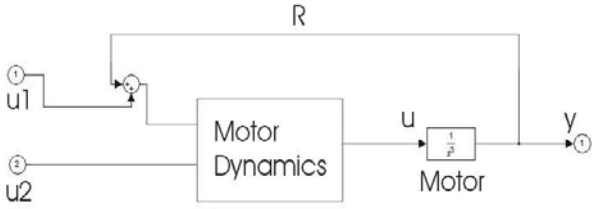
$$\begin{aligned} &\frac{P_2 r_1 D_e \sin(\theta_1 + \theta_2)}{2 r_2 \sin\gamma} + \frac{J_2 D_e}{2 r_2 \sin\gamma} \frac{\partial \dot{\theta}_R}{\partial t} = K_2 i_a \\ &L_a \frac{\partial_{ia}}{\partial t} + R_a i_a + K_3 \frac{\partial \theta_R}{\partial T} = K_1 e_V \end{aligned} \tag{9}$$

where P2 the lower leg weight, r1 is the distance from the knee joint and centre of mass of lower link, De is the diameter of the ball screw actuator, J2 is the moment of inertia of the lower link, r2 is distance from the knee joint and the actuator joint, γ is the angle between the lower link and the screw actuator that depends of θ1 and θ2 displacements, K2 is the motor 2 torque constant, ia is the motor 2 current, La is the motor 2 capacity, Ra is the motor 2 resistance, K3 is the constant of the electromotive force, K1 is the motor 2 gain and ev is the motor 2 input voltage. θ2 represents the motor output and ev represents the motor input. The parameters of motor 2 K2, ia, La and Ra are the same of motor 1 since both are the same commercial DC motor. K3 and K1 will be computed in the same way of motor 1 too. Finally, the others parameters have the values

- $P_2 = 4.30$ N;
- $r_1 = 1.4$oe−1 m;
- $r_2 = 7.00$e−2 m;
- $D_e = 6.00$e−3 m;
- $J_2 = 3.12$e−5 Kg∗m^2.



a)



b)

Figure 11. Control architecture scheme for Motor 1 of the robotic leg: a) whole system scheme b) scheme of motor 1 dynamics.

Similar to (1), the differential equations (2) are not linear and a linearization is also necessary. Thus, by using the same linearization method one can write the dynamic equations of motor 2 as

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= \frac{2 K_2 r_2 \sin\gamma}{J_2 D_e} X_3 - \frac{P_2 r_1}{J_2} \sin(\theta_1 + X_1) \\ \dot{X}_3 &= \frac{K_1}{La} u - \frac{Ra}{La} X_3 - \frac{K_3}{La} X_2 \end{aligned} \tag{10}$$

where X1 is the angular displacement of motor 2 θ2, X2 is the angular velocity of motor 2 ω2, X3 is the motor 2 current ia and u is the motor 2 input ev. Similar to (2), controllability and observability of (10) have been checked in order to apply the state-space feedback linearization. Making a variable change of X1 for Z1, equations (10) have the form

$$\begin{aligned}\dot{Z}_1 &= \dot{X}_1 = Z_2 \\ \dot{Z}_2 &= \dot{X}_2 = \frac{2K_2 r_2 \sin\gamma}{J_2 D_e} X_3 - \frac{P_2 r_1}{J_2}\sin(\theta_1 + X_1) = Z_3 \\ \dot{Z}_3 &= \frac{2K_2 r_2 \sin\gamma}{J_2 D_e}\dot{X}_3 - \frac{P_2 r_1}{J_2}\cos(\theta_1 + X_1) Z_2\end{aligned} \tag{11}$$

The state-space representation of (11) has the same form of (5) and the same transference function of (6). Nevertheless, the input ν change the form for motor 2 as

$$\begin{aligned}\nu = &\frac{2K_2 K_1 r_2 \sin\gamma}{J_2 D_e La} u - \frac{R_a}{L_a} Z_3 - \frac{P_2 R_a r_1}{J_2 L_a}\sin(\theta_1 + X_1) \\ &- \frac{2K_3 K_2 r_2 \sin\gamma}{J_2 D_e L_a} Z_2 - \frac{P_2 r_1}{J_2}\cos(\theta_1 + X_1) Z_2\end{aligned} \tag{12}$$

Figure 12 shows the scheme of the control architecture that has been proposed for motor 2. The scheme for the dynamic representation of motor 2 is the same for motor 1 of Figure 11b) but the parameters u1, u2, u, R and y for motor 2 are input of θ1, input of θ2, ν of equation (12), feed-back value of θ2 and θ2 output. In the same way of motor 1, the control architecture for motor 2 is also controlled by a PID that has also been optimized by using an evolutionary method as well as modelled and operating simulated in Simulink environment.

V. OPTIMIZATION OF PID PARAMETERS

The PID controllers of the control architecture for motor 1 and 2 of Figures 11 and 12, respectively, have been optimized in term of proportional, derivative and integrative gains. The proposed leg architecture has been considered a constant system since it performs appropriate continue and smooth motion when the robot processes on a specified path. The optimization problem has been solved using an evolutionary method called Differential Evolution (DE) [28]. DE is a population-based algorithm of direct search for optimizing multimodal functions. Like most of the evolutionary algorithms found in the literature, DE consists in an iterative process where recombination and mutation operators are applied to build a trial Np-size population of D-dimensional vectors. A selection mechanism based on the objective function evaluation compares the current and trial population vectors to select which individuals survive to the next generation. DE's strength lies in an inherently scheme of adaptive arithmetic mutation that is based on the perturbation with random vectors differentials represented as floating-point variables. The DE algorithm is depicted in the pseudo-code listed in Figure 13, where Pg, Pvg and Pug are the current, mutated, and intermediate population at generation g, respectively. Some modifications over the "DE/rand/1/bin" differential evolution scheme are introduced respect the DE schemes proposed in [28]. Here an initial configuration vector with a learnt PID gains vector Ω0 = [Kp Ki Kd] are supposed to be known. Consider a population Np = 15 of PID gains defined as

$$P_g = \lfloor \Omega_{Np} \rfloor \tag{13}$$

where ΩNp is coded as a floating-point vector array with length D and g = 0, . . . , gmax = 50 is the generation number.

In the DE/rand/1/bin scheme, the initialization process generates a population of Np vector as Ω0 = [0.63 0.0504 1.9688] by following the Ziegler-Nichols method [26]. If previous knowledge on the optimum location is not available, then the use of a uniform distribution is recommended to guarantee population diversity. On the other hand, if the initial Ω1 vector is considered close enough to Ω0, then intuitively the optimal solution can be assumed to be at the proximities of the learnt PID gains. Once the population has been initialized, mutation is used to build an Np trial vectors population Pvg.

The DE mutation mechanism takes three vectors randomly from the current population using a uniform distribution. Then a candidate solution is obtained by adding the weighted difference of two vectors, each called difference vector, to the third, called base vector. This can be expressed as

$$\Omega_{Vi,g} = \Omega_{r1} + F(\Omega_{r2} + \Omega_{r3}) \tag{14}$$

where F = 0.98 ∈ (0, 1+) is a scaled factor and vectors Ωr1, Ωr2 and Ωr3 are randomly chosen so that the condition r≠ r2 ≠ r3 is met.

Besides mutation, DE employs a discrete recombination mechanism, called binomial crossover, which combines two vector parameters: Ωvi,g , taken from the mutated population Pvg, and Ωi,g, of the current population Pg and called target vector. Crossover is triggered by the comparison of a fixed crossover probability value Cr = 0.8 and a random uniform number ∈ (0, 1) generated a new for each parameter. If the random number is less than or equal to Cr then the parameter is inherited from the mutant vector, otherwise the parameter is copied from the target vector. Binomial crossover operator works as follows:

$$\Omega_{Vi,g} = \begin{cases} \upsilon_{k,j}, if\left(rand_j(0,1) \le C_r\right) \\ q_{k,j}, Otherwise \end{cases} \tag{15}$$

With k=1,…….,N and j=1,……..n.

The objective function evaluation allows the assignment of a cost value to each member from the trial and target populations Pvg and Pg. Based on the comparison of this cost value a selection mechanism decides which PID gains from the current population Pg are replaced by trial gains from Pvg in the next generation population Pg+1, as stated next:

$$\Omega_{i,g+1} = \begin{cases} \Omega u_{i,g}, if\left(f(\Omega u_{i,g}) \le f(\Omega_{i,g})\right) \\ \Omega_{i,g}, Otherwise \end{cases} \tag{16}$$

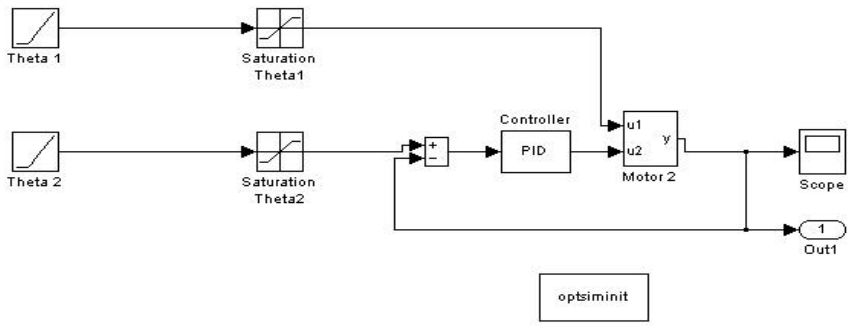


Figure 12 Control architecture scheme for Motor 2 of the robotic leg.

1: **initialization** $P_{g=0}$
2: **evaluation** P_g
3: **while** termination criterion $\neq$ true **do**
4: Pv_g := **differential mutation** P_g
5: Pu_g := **binomial crossover** Pv_g
6: **evaluation** Pu_g
7: P_{g+1} := **selection** $(Pu_g \cup P_g)$
8: $g := g + 1$
9: **end while**

Figure 13 Proceeding for differential evolution strategy.

The objective function evaluates de difference between the system input and output, during the whole operation, in order to reduce the error as much as possible. Since the error in the transient-state is higher than in the steady-state, the objective function gives more weight to the steady-state error as

$$f(\Omega_{i,g}) = \sum_{i=o}^{t_{max}} \left(y_{out_i} - y_{in_i}\right)^2 \times t_i \qquad (17)$$

where yout is the system output, yin is the system input and t is the time instance by instance.

Other objective functions have been studied as possible solutions for our optimization problem. Nevertheless, the equation (16) is that has given better results in terms of velocity of the system response, the lowest maximum error in transitive-state and minimum stable-state error. Finally, a maximum generation number is used as a terminal criterion for the evolutionary algorithm. The obtained optimal PID gains correspond to the best individual of gmax generation, i.e., the gains with the lowest objective function value at the last generation population. DE algorithm has allowed obtaining the optimal gains Kp, Ki and Kd of the proposed PID controller in an effective manner. This strategy has previously demonstrated proper performances for resolving optimization problem, as for example in works like [28]. Therefore, we have decided to include the DE strategy in the design process of the proposed control architecture.

VI. NUMERICAL RESULTS OF THE CONTROL ARCHITECTURE VALIDATION

The proposed scheme has been tested on a simulation environment to solve the problem of obtaining the optimal PID gains for motors of the robotic leg of Figure 7. Three basic modes have been simulated for the leg operation: first, movement of link 1 with link 2 in static position; second, movement of link 2 with link 1 in static position; and third, the combined movement of links 1 and 2 simultaneously. Figure 14 shows the inputs for the first basic mode of the simulated operation. In particular, Figure 14a) shows the input for motor 1 that increases from 0 deg to 1 deg in 0.05 sec and them continues in steady-state at 1 deg. Figure 14b) shows the input for motor 2 that stays steady at 180 deg during the whole operation. Figure 15 shows the θR output for the first mode of the leg simulated operation after 1 iteration, after 25 iterations (middle iteration) and after 50 iterations (last iteration). Note that the output plots of θR show time evolutions that converge at 1 deg (input) for the three cases. At iteration 1 of Figure 15a), the curve evolution presents oscillations around the convergence value (1 deg) that decreases from higher values at the beginning to lower values at the end. In this case, the output does not reach the steady-state with a maximum error of 30 %. At iteration 25 of Figure 15b), the curve oscillates during about 0.35 sec reaching a maximum error of about 50 %. After about 0.35 sec the output at iteration 25 reach the steady-state until the simulation end.

Note that the maximum error obtained at iteration 25 has been higher than the obtained at iteration 1, but after 25 iterations the output reaches the steady-state giving an improvement of the system response. Figure 15c) shows the time evolution of the output during the first operation mode at the final iteration 50 that represents the best member of the population Pgmax. This curve presents a transient-state form 0 sec to about 0.35 sec and a steady-state at 1 deg form about 0.35 sec to the simulation end one. The maximum error has been reduced to 17 % respects the previous iterations and the response velocity has stayed similar form about the iteration 25 to the final. The optimized vector of the PID gains, obtained during the first simulation mode, has been Ω50 = [0.0395 0.3920 1.2179].

The second mode of leg operation has been simulated by using the inputs of Figure 16. From this figure, the motor 1 input can be recognized as constant value of 0 deg during the total simulation time and motor 2 as a ramp, which decreases from 180 deg to 179 deg in 0.05 sec and them continues in steady-state at 179 deg. The optimized vector of the PID gains, obtained during the second simulation mode, has been Ω50 = [1.1958 1.1147 1.2619]. Figure 17 shows the output results of θ2 during the second simulated mode of leg operation. Note

that the curve of Figure 17 shows smoother evolutions, smaller maximum errors and higher response velocities than the first simulation of Figure 15. Since the dynamic of the second motor is less complicated than the first one, the control algorithm can obtain better results in the second simulation mode where only the second motor is actuating. From Figure 17, the maximum error can be recognized as 20 % that occurs in the iteration 1, of Figure 17a), as well as iteration 25, of Figure 17b). In the last iteration, of Figure 17c), the maximum error is reduced to about 5 %. The velocity response has not changed during the whole optimization process of case 2 since the steady-state has been reached from the iteration 1 at about 10 sec. Figure 18 shows the output results of the third mode of leg operating simulation where both motor 1 and 2 are operating.

For the third basic mode of leg operating simulation, the computed value of angle θR has been 0.76 deg, instead of 1 deg of previous simulated modes, as result of the 1-deg-motion of both motors. Figure 18 shows simulation results that present, as expected, characteristics of the previous two simulation results. The steady-state in every output curve of each iteration result has been reached, during the whole optimization process, with different velocities responses. For instance, the velocity responses have been improved for the iteration 1, of Figure 18a), in which the steady-state has been reached at about 80 sec, to last iteration 50, of Figure 18c), in which it has been reached at about 25 sec. The maximum error has been about 31%, as shown in the plots of iterations 1 and 25, of Figures 18a) y b), and then improved to 18% at the final iteration. The optimized vector of the PID gains, obtained during the third simulation mode, has been Ω50 = [0.0105 0.5094 1.5323]. The simulation results illustrate the feasibility of the proposed control architecture, since the velocity responses and maximum errors report suitable values with smooth time evolutions, in the three basic operating simulations, for 150 sec of simulation time.

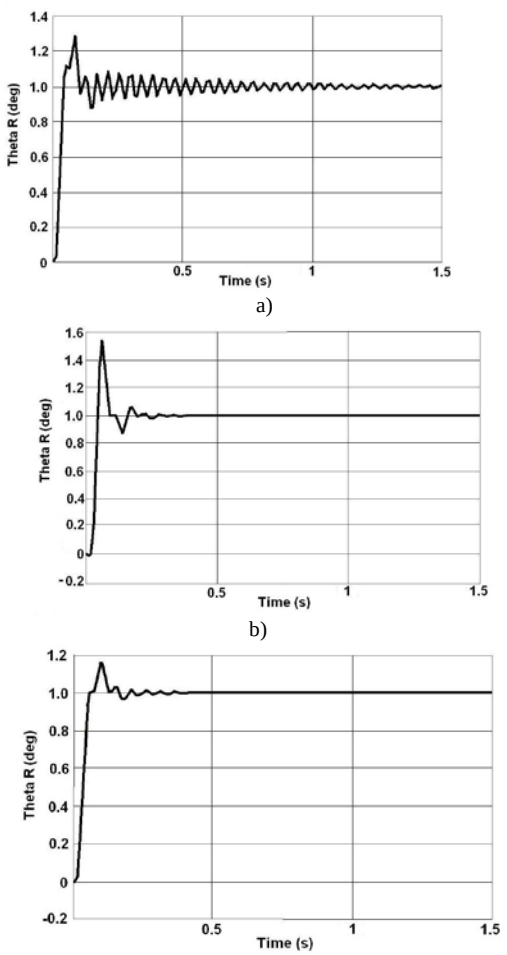


Figure 15 Numeric results of the simulation for the first leg operation mode in Simulink environment: a) after 1 iteration; b) after 25 iterations; c) after 50 iterations.

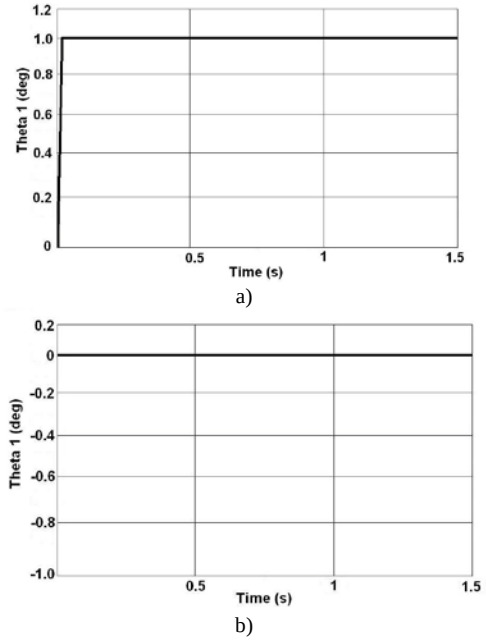


Figure 14 Inputs for the control scheme in the first basic operating simulation: a) position of Motor 1; b) position of Motor 2.

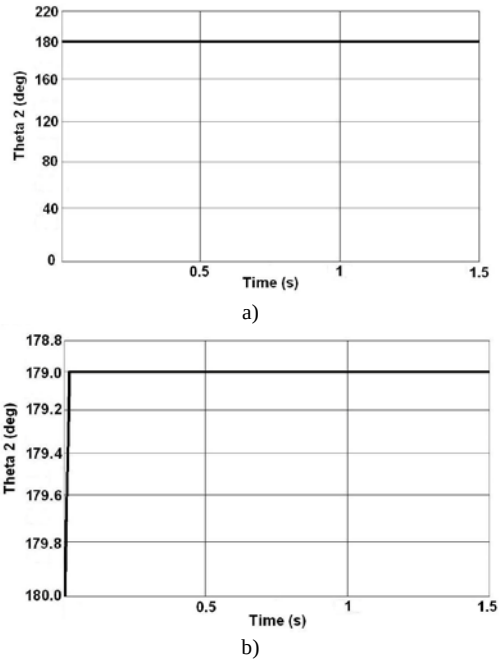


Figure 16 Inputs for the control scheme in the second basic operating simulation: a) position of Motor 1; b) position of Motor 2.

Moreover, the plots of Figures 15, 17 and 18 show time evolutions of the output curves for three representative iterations that allow noting the improvements in the system output according to the proposed evolution strategy.

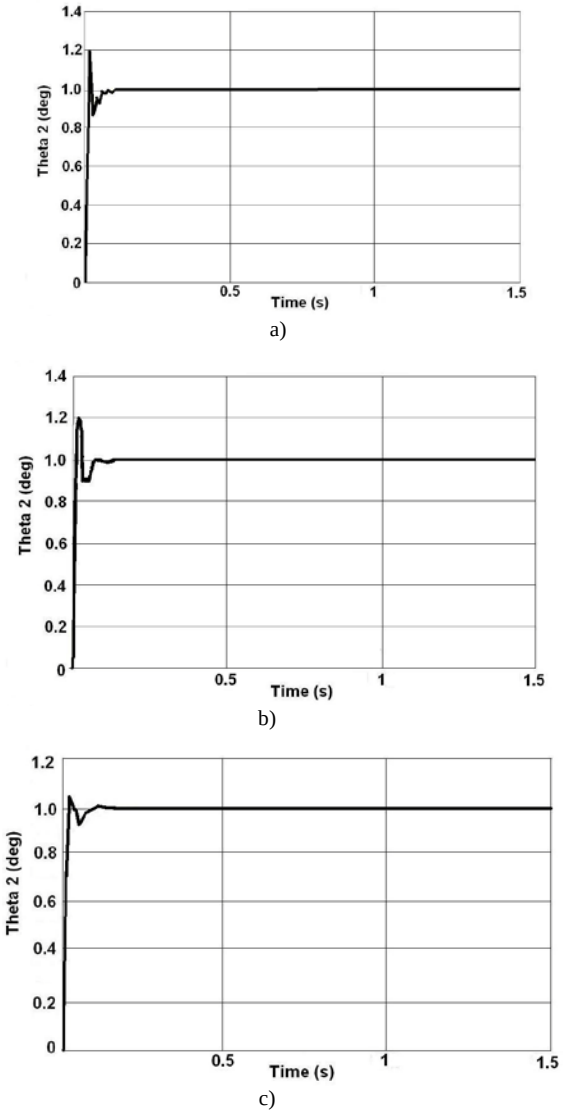
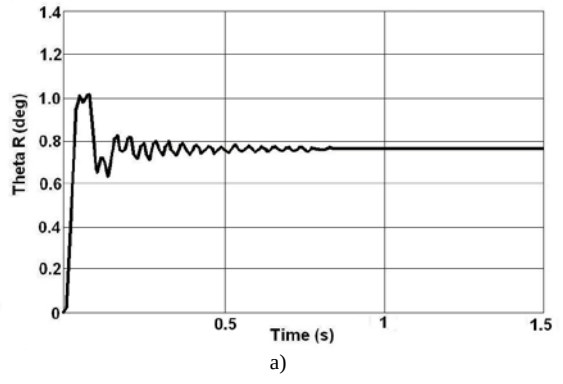


Figure 17 Numeric results of the simulation for the second leg operation mode in Simulink environment: a) after 1 iteration; b) after 25 iterations; c) after 50 iterations.



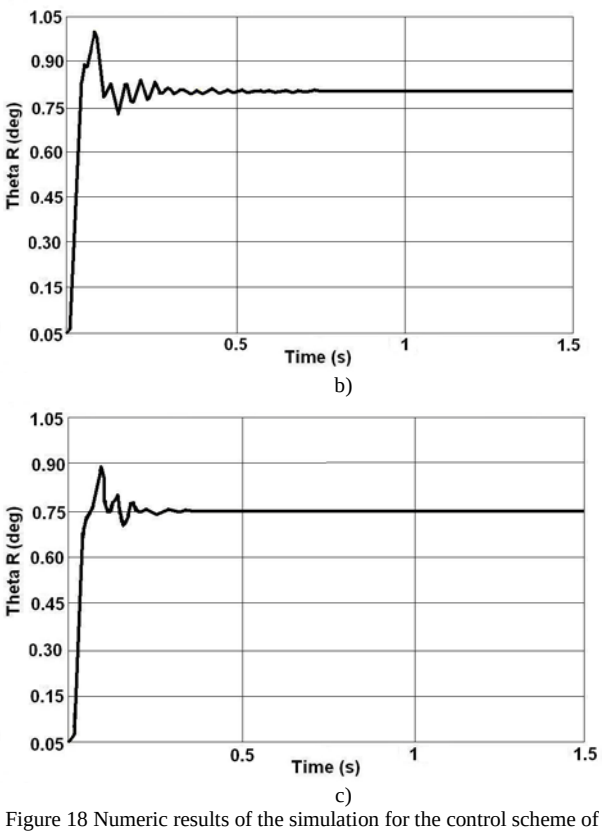


Figure 18 Numeric results of the simulation for the control scheme of during the combined operation of Motors 1 and 2: a) after 1 iteration; b) after 25 iterations; c) after 50 iterations.

VII. CONCLUSIONS

This paper presents a control architecture designed for a new robotic leg of the Cassino Hexapod Robot. A novel design solution has been conceived specifically for the Cassino Hexapod leg in order to achieve better performance of the operation modes than the original structure. In particular, the new leg structure and its operation have been considered both experimentally and numerically through simulations and tests in order to validate the robot design. Validations have shown suitable results for robot operation, limited power consumption, robust and compact design. The mechanical leg formulation has been represented in the state space and linearized by a feedback linearization in order to obtain a proper control architecture. A PID has been proposed as control strategy because its practical application in several robotic fields. The PID gains have been optimized using an evolution strategy called differential evolution. The feasibility of the proposed control strategy has been tested by means of numerical simulations with suitable detailed model architecture. Results have shown efficiency in the system operation in term of response velocity, maximum error in transition-state and steady-state error as well as suitable gait planning.

REFERENCES

[1] Rosheim M. E. Robot evolution, the development of anthrobotics. A Wiley-Interscience Publication, New York, 1994.

[2] Zielinska T., Heng J. Development of a walking machine: mechanical design and control problems. Mechatronics, 12, 5, 737 – 754, 2002

[3] Zhao Y. S., Lu L., Zhao T. S., Du Y. H., Huang Z. Dynamic performance analysis of six-legged walking machines. Mechanism and Machine Theory, 35, 1, 155 – 163, 2000.

[4] Reumoat P., Alezandre P., Ghuya D. Gait analysis and implementation of a six leg walking machine. International Conference Robots in Unstructured Environments, 2, 941- 945, 1991.

[5] Berns K. Walking cachine catalogue. Available on-line at http://www.walking-machines.org/, 2010.

[6] Raibert M., Blankerpoor K., Nelson G., Playter R., BigDog Team. BigDog, the rough-terrain quadruped robot. World Congress the International Federation of Automatic Control, Seoul, 10822-18025, 2008.

[7] Nonami K., Huang Q.J. Humanitarian mine detection six-legged walking robot. International Conference on Climbing and Walking Robots, 861-868, 2000.

[8] Rachkov M., Marques L., Almeida A. Multisensor demining robot. Autonomous Robots, 18, 3, 275-291, 2005.

[9] Kawaguchi Y., Yoshida I., Iwao K., Kikuta T. Development of internal gas pipe inspection robot. Journal of Robotics and Mechatronics, 7, 5, 371-376, 1995.

[10] Horodinca M., Doroftei I., Mignon E., Preumont A. A simple architecture for in-pipe inspection robots. International Colloquium on Mobile and Autonomous Systems, Magdeburd, 2002.

[11] Wilcox B., Matthies L., Genery D. Robotic vehicles for planetary exploration. International Conference on Robotics and automation, Nice, 1992.

[12] Chakraborty N., Ghosal A. Kinematics of wheeled mobile robots on uneven terrain. Mechanism and Machine Theory, 39, 1273 – 1287, 2004.

[13] Fraunhofer Institute for Factory Operation and Automation IFF. SLAIR2 - six legged autonomous intelligent robot 2. http://www.ovgu.de/ieat/robotslab/slair2.php, 2010.

[14] Lewinger W. A. Insect-inspired, actively compliant robotic hexapod. M. S. Thesis, Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH, USA, 2005.

[15] Gonzalez de Santos P., Jimenez M. A., Armada M. A. Dynamic effects in statically stable walking machines. Intelligent and Robotic Systems, 23, 1, 71-85, 1998.

[16] Cigola M., Pelliccio A., Salotto O., Carbone G., Ottaviano E., Ceccarelli M. Application of robots for inspection and restoration of historical sites. International Symposium on Automation and Robotics in Construction, Ferrara, paper 37.

[17] Carbone G., Ceccarelli M. A low-cost easy-operation hexapod walking machine. Advanced Robotic Systems, 5, 2, 161-166, 2008.

[18] Carbone G., Jatsun A., Ceccarelli M., Jatsun S. Design and simulation of cassino hexapod robot. International Conference on Computers, Rhodos, 301-314, 2009.

[19] Manoonpong P., Geng T., Kulvicius T., Porr B., Wrgtter F. Adaptive, fast walking in a biped robot under neuronal control and learning. Computer Biology, 3, 7, 134, 2007.

[20] Capaday C. The special nature of human walking and its neural control. Trends Neuroscience, 25, 370–376, 2002.

[21] Geng T., Porr B., Wrgtter F. Fast biped walking with a sensor-driven neuronal controller and real-time online learning. Robotics Research, 25, 3, 243–259, 2006.

[22] Tang Z., Zhou C., Sun Z. Humanoid walking gait optimization using GA-based neural network. International Conference on Natural Computation (2), 252–261, 2005.

[23] SolidWorks Corporation webpage. http://www.solidworks.com, 2010.

[24] RS webpage. DC Geared Motors. www.rs-components.com, 2010.

[25] Stoline J. J., Li W. “Applied nonlinear control”, First Edition, Prentice Hall, 352, 1990.

[26] Ogata K. “Modern Control Engineering”, Third Edition, Prentice Hall, 998, 1998.

[27] Karris S. T., “Introduction to Simulink with Engineering Applications”, Second Edition, Orchard Publications, 584, 2008.

[28] Storn R., Price K. Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. Global Optimization, 11, 4, 341–359, 1997.