Comparison of Different Control Techniques on a Bipedal Robot of 6 Degrees of Freedom

Michael F. Cifuentes-Molano, Byron S. Hernandez, Eduardo Giraldo

***Abstract*—In this paper, a mathematical model of a bipedal robot with six degrees of freedom (6 DoF) is presented and tested for some control strategies. For simplicity, one of the extremities is modeled, and it is assumed that the second one is similar. Some widely used tools, like Denavit-Hattender parametrization and Euler-Lagrange approaches, are applied to obtain the movement equations. A set of control laws are designed, applied to the system model, and compared among them. The study is carried out to evaluate deviation errors in the extremities, the proposed bipedal model's performance, and control strategies. The controllers' performances are evaluated in terms of the deviation errors, which are computed as the root mean square (RMS) of differences between desired and actual extremity-joint positions.**

***Index Terms*—Control, mathematical modeling, bipedal robot**

I. Introduction

Since the beginning of robotics, and especially in recent years, humans have desired to create machines that are similar to them. That is why they have created various types of robots that try to accomplish tasks using any human being's motion principles. The above can be noticed to a great extent with manipulators pretending to be a human arm. However, the spectrum is much broader and covers many types of machines similar to human capabilities; An example is bipedal locomotion [1].

The problem of bipedal locomotion in robots can be seen in different ways depending on the techniques, the desired level of precision, the degrees of freedom of the robot under analysis, and if it is a real platform case or just a concept for model development. The main objective is to devise a mechanism that achieves the stabilization of a body on two extremities. The matter is to find a mathematical model that can be implemented in simulation, as close as possible to the dynamic model of the real human legs, taking into account all their degrees of freedom and their intrinsic characteristics. Many approaches have been proposed to develop bipedal robot models, but they are all very different. Approaches, like proposed by Chevallereau et al. in "*Bipedal Robots: Modeling, Design and Walking Synthesis*" [2] decompose the walking cycle into several phases depending on the feet shape. They derive an analysis based mostly on geometric viewpoints from the contact between the extremity and the floor. Other approaches like "*Modelling, simulation, and control of a bipedal robot AAU-BOLT1*" [3] by B. Jensen and M. Niss, and "*Bipedal walking for a full-size humanoid robot utilizing sinusoidal feet trajectories and its energy consumption*" [4] by J. Han, show dynamical models based on the energy functions acquired from the Lagrange dynamics method. Finally, in "*An optimal control method for a biped robot with stable walking gait*" [5] by N. Phuong et al., the authors propose a model based on a triple inverted pendulum using the Lagrange approach.

Furthermore, it is essential to find a more straightforward approach to find the dynamic model and then carry out its simulation applying different control strategies in order to analyze their effects on that particular system [6]. Several control techniques as sliding modes control [7] or bio-inspired control techniques based on cerebellum behaviour [8] are also used for multivariable systems where the dynamical model of the plant is estimated from data. Finding a model that allows performing the proposed analyzes requires several essential considerations. Since bio-mechanics are very complex, we need to simplify some system features, given that if the model includes all the restrictions and rigorous behaviors, the mathematical expression complexity would be unsuitable. Naturally, knee behavior is label-like, or in mechanical terms, it is like a system of four-bar cross-linkages [9].

In this work, a bipedal robot's mathematical model with six degrees of freedom (6 DoF) is presented and tested for some control strategies. It is worth noting that the performance of the model and its controllers are evaluated in terms of the RMS deviation errors. This paper is organized as follows: Section II introduces the concepts of the geometrical and fundamental dynamic required for this modeling and introduces the proposed approach for devising the bipedal system's mathematical model. Section III describes the design of benchmarking control strategies for robotic systems applied to the proposed model. Section IV describes the performance of the proposed controllers, and finally, Section V concludes the article.

II. Mathematical model

To perform a mathematical model for any dynamic system, we must know the system's behavior features; in other cases, the found expressions would be inaccurate. The robot model presented has two aspects of high relevance that will define the mathematical derivations. The first one is that it will be modeled as a triple pendulum for kinematics as in [5] and [10]. The second one is that the friction in the model will not be considered for a matter of simplicity. This

Manuscript received July 16, 2020; revised January 20, 2021. This work is carried out under the funding of the Universidad Tecnológica de Pereira, Vicerrectoría de Investigación, Innovación y Extensión. Research project: 6-20-7 "Estimación Dinámica de estados en sistemas multivariables acoplados a gran escala".

M.F. Cifuentes-Molano is a researcher at the Research group in Automatic Control, Universidad Tecnológica de Pereira, Pereira, Colombia, e-mail: mcifuentes@utp.edu.co

B.S. Hernandez is a researcher at the Research group in Automatic Control, Universidad Tecnológica de Pereira, Pereira, Colombia, e-mail: bshernandez@utp.edu.co

E. Giraldo is a full professor at the Department of Electrical Engineering, Universidad Tecnológica de Pereira, Pereira, Colombia.

Research Group in Automatic Control, e-mail: egiraldos@utp.edu.co.

simplification will not have significant repercussions on the system's behavior since the operational points tracked by the controller (to be designed in a subsequent stage) have minimum friction.

For devising the model, the first step is to perform the geometric analysis to obtain the Denavit-Hartenberg parameters [11]; these parameters allow us to find the reference frames for each joint between the limbs. As shown in Fig. 1, the very first frame (base) is allocated in the hip, and the last frame (end effector) is allocated in the ankle. We need to describe each frame pose in terms of the base frame pose.

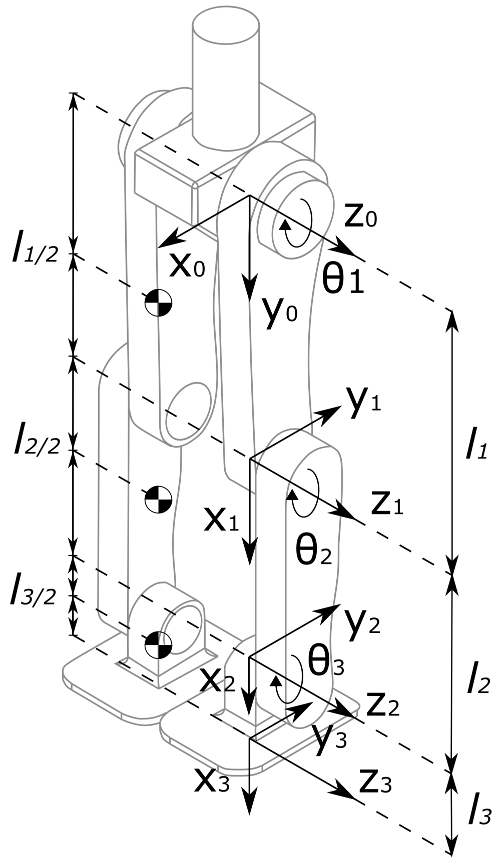

Fig. 1. Bipedal Robot of 6 Degrees of Freedom.

$$x_1 = l_{1/2}\cos(\theta_1)$$
$$y_1 = -l_{1/2}\sin(\theta_1)$$

$$x_2 = l_1\cos(\theta_1) + l_{2/2}\cos(\theta_1+\theta_2)$$
$$y_2 = -l_1\cos(\theta_1) - l_{2/2}\sin(\theta_1+\theta_2)$$

$$x_3 = l_1\cos(\theta_1) + l_2\cos(\theta_1+\theta_2) + l_{3/2}\cos(\theta_1+\theta_2+\theta_3)$$
$$y_3 = -l_1\cos(\theta_1) - l_2\sin(\theta_1+\theta_2) - l_{3/2}\cos(\theta_1+\theta_2+\theta_3)$$

A. Dynamics

Starting from the coordinate frames for each center of mass shown in Fig. 1, we can find the kinetic energy for each element, see (1). Recall that the sum of them will be the total kinetic energy of the system. Also, generalized coordinates q are introduced for each point.

$$\begin{aligned}
K_1 &= \frac{1}{2}m_1(l_{1/2}^2\dot{q}_1^2) + \frac{1}{2}I_1\omega_1 \\
K_2 &= \frac{1}{2}m_2((l_1C_1\dot{q}_1 + l_{2/2}C_{1+2}(\dot{q}_1+\dot{q}_2))^2 \\
&\quad ... + (l_1S_1\dot{q}_1 + l_{2/2}S_{1+2}(\dot{q}_1+\dot{q}_2))^2) + \frac{1}{2}I_2\omega_2 \\
K_3 &= \frac{1}{2}m_3(l_1C_1\dot{q}_1 + l_2C_{1+2}(\dot{q}_1+\dot{q}_2)^2 \\
&\quad ... + (l_{3/2}S_{1+2+3}(\dot{q}_1+\dot{q}_2+\dot{q}_3))^2 + \frac{1}{2}I_3\omega_3
\end{aligned} \tag{1}$$

where $C_{m+...+n}$ represents $\cos(\theta_m + ... + \theta_n)$.

To find the dynamic equations of the system, one must also take into account the potential energy of each body. For this, potential energies are individually defined as shown in (2). Taking the sum of all of them will be the total potential energy of the system.

$$\begin{aligned}
U_1 &= -m_1g(l_{1/2}C_1) \\
U_2 &= -m_2g(l_1C_1) - m_2g(l_{2/2}C_{1+2}) \\
U_3 &= -m_3g(l_1C_1) - m_3g(l_2C_{1+2}) - m_3g(l_{3/2}C_{1+2+3})
\end{aligned} \tag{2}$$

Then, the general Lagrangian in (3) can be expressed using the kinetic and potential energies in (1) and (2).

$$L(q(t),\dot{q}(t)) = K(q(t),\dot{q}(t)) - U(q(t)) \tag{3}$$

With the general description of the Lagrangian, we can describe the equation for the torque of each junction.

$$\frac{d}{dt}\left(\frac{\partial L(q,\dot{q})}{\partial \dot{q}_i}\right) - \frac{\partial L(q,\dot{q})}{\partial q_i} = \tau_i \tag{4}$$

The general model for the torque in each junction may be rewritten as follows:

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_i}\right) - \frac{\partial K}{\partial q_i} + \frac{\partial U}{\partial q_i} = \tau_i \tag{5}$$

The form of the expression in (5) make computations more easy to perform, here, i represents the index for each link of the mechanism. In this case $i = \{1,2,3\}$. It must be considered also the temporal derivative written in (6).

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_i}\right) = \sum_{i=1}^{n}\left(\frac{\partial}{\partial q_i}\left(\frac{\partial K}{\partial \dot{q}_i}\right)\frac{dq_i}{dt} + \frac{\partial}{\partial \dot{q}_i}\left(\frac{\partial K}{\partial q_i}\right)\frac{dq_i}{dt}\right) \tag{6}$$

It becomes clear that the solution to this system of nonlinear equations requires computations demanding numerical methods or iterative algorithms, which are better performed by computers. Furthermore, the modeling behavior is similar to a triple pendulum with no controller working; thus, it will show chaotic behavior.

It is crucial to mention computation power to describe the method used when deriving the robot's movement Equations. Computers become an essential tool if desired to reach a good representation of the system's real dynamics. It is also necessary to know about symbolic computation since it substantially decreases the computing cost. For the particular case of the robot studied in this paper, it is required to give the algorithm parameters such as masses, lengths, and other constants of the robotic mechanism. Then, we need to define θ_i, $\dot{\theta}_i$ y $\ddot{\theta}_i$ symbolically, and x and y position for the mass center of each link of the mechanism. In particular, let each mass center match with its respective geometric center for every link. Then, we perform the derivatives for each position to obtain the velocity. These masses, positions, and velocities are used to compute the system's potential and kinetic energies.

Energies are used to build a Lagrangian equation, which is later used to obtain the movement equations. This process is carried out using (5) and (6). This process results from the well-known inverse dynamics of the system, from which we can solve for $\ddot{q}$ and find the direct dynamics this way. All this process is carried out using symbolic variables to be changed to numeric variables for initializing and performing the simulation.

III. Control Techniques

In robotics, many control strategies can be used to achieve the desired behavior for any system. Choosing a technique depends on the proper system and its domain nature (continuous or discrete). Some continuous stable systems could become unstable when they are transformed to the discrete domain, and it could cause the misbehavior of some digital control strategies. It is important to analyze different control techniques because it allows defining which approaches are better to reach the desired behavior based on any performance index. One of the most critical indexes is the computational cost that a technique involves. Some advanced strategies require more computational power, which implies an increase in costs if the controller is implemented over a real platform.

A. Calculated torque control

The calculated torque control strategy is commonly used in robotic manipulators. It essentially computes the control signal that commands the system to achieve the desired position using the inverse robot dynamics. The above is a crucial factor we must keep in mind; the model must be as loyal as possible to the real system in order to reach an effective control law [12]. Given the manipulator's desired paths, the calculated torque controller uses the inverse dynamics to compute the required torque for achieving these paths. It passes through a sum with two gains to the desired path and the derivative of the errors. We can see a scheme of this in Fig. 2.

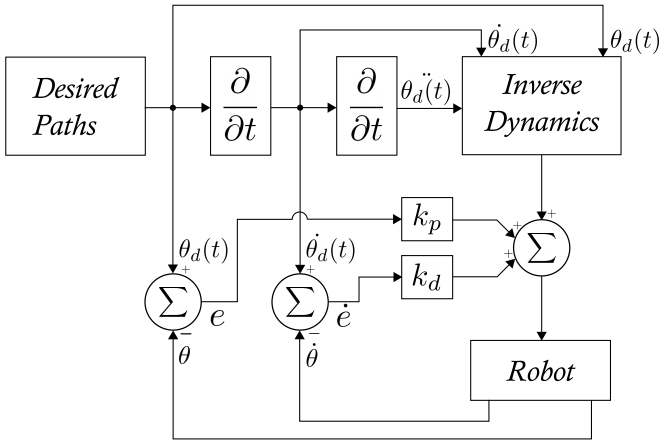

Fig. 2. Calculated torque block diagram.

The calculated torque control must make use of the robot's precise dynamics to be effective. So, the uncertainties in the model affect the accuracy of the control [13]. The control law is then computed based on the parameter estimation using the model, as in (7).

$$\tau = \widehat{M}(\theta_d)\ddot{\theta}_d + \widehat{C}(\theta_d, \dot{\theta}_d) + \widehat{G}(\theta_d) + \widehat{F}(\theta_d, \dot{\theta}_d) + \\ ...K_d\dot{e} + K_p e \tag{7}$$

where $\widehat{M}$ is the estimated inertia matrix, $\widehat{C}$ is the estimated matrix containing the Coriolis accelerations, $\widehat{G}$ is the gravity matrix, and $\widehat{F}$ is the friction in each of the joints. Many parameters influence the model's behavior, and taking them all into account would not be easy. Their implementation in a simulation would be an impossible task. A significant factor that must be taken into account is the friction in each joint of the mechanism to have a model close to reality and achieve a good performance of the calculated torque control. The calculation of friction is shown in (8).

$$\tau_f = \beta\dot{\theta} + \mu \cdot \text{sgn}(\dot{\theta}) \tag{8}$$

where β is the material's viscous friction coefficient and μ is the dry or Coulomb friction coefficient.

B. PID control

PID Control is the most used control strategy in many areas where it is required to carry out the system to the desired state. It is widely used because of its simplicity and effectiveness, given that it requires low computational power in its computations. The PID controller is theoretically straightforward, and it is widely studied in the literature. Since this paper is not tackling an in-depth description for this controller, readers can always refer to the bibliography [14], [15], [16].

PID controllers for bipedal robots are performed by designing a PID control law for each extremity. Equation (9) shows the general form of this controller for the extremities [17].

$$\tau_i = K_p e(\theta_i) + K_D e(\theta_i) + K_I \int e(\theta_i) \tag{9}$$

This control law is applied directly in the direct kinematics robot model, and its error is computed as:

$$e(\theta_i) = \theta_{id} - \theta_{iR} \tag{10}$$

where θ_{id} represents the extremity desired angle, and θ_{iR} is the real angle measured by the acquisition system.

C. Hyperbolic Sine-Cosine control

In [18] is presented as an algorithm for robot manipulators control. This control law could be extended to our robot mechanism since the model is obtained using common manipulator modeling approaches. Equation (11) describes the controller and considers two constants: one of them is for the proportional component, based on hyperbolic sine and cosine, and the other constant is for the derivative component based on a hyperbolic sine. The third term is gravity compensation.

$$\tau = K_p \sinh(e)\cosh(e) - K_d \sinh(\dot{q}) + G(q) \tag{11}$$

D. Control tuning

Each of the described control architectures has gains that must be tuned, so we need a methodology or algorithm that establishes them to reach each controller's correct operation. A genetic algorithm is proposed as a tuning method that provides the appropriate values for the gains. So far, a genetic algorithm is a meta-heuristic optimization technique whose goal is to optimize (minimize o maximize) a loss function that describes a performance index. In this case, we chose the impulse or step response of the closed-loop system shown in (12) and (13), respectively.

$$J = \int_0^{t_f} q^2(t) \tag{12}$$

$$J = \int_0^{t_f} (1 - q(t))^2 \tag{13}$$

where $q(t)$ is the system's output. The genetic algorithm starts an initial population of possible gains and iteratively improves them by some heuristic updates like crossing or different mutation types. Finally, at the elitism step, the best individuals of the population are preserved. For details on genetic algorithm implementations see [19], [20], [21].

IV. Results

As a case study, we developed the model applying (6) that uses the kinetic and potential energies given by (1) and (2). Then, we designed each of the controllers described in the previous section and tune-up them using the genetic algorithm. The parameters for the simulation are shown in Table I.

TABLE I
SIMULATION PARAMETERS

link	length (m)	mass (kg)
1	0.4	1
2	0.45	1
3	0.2	0.5

All joints have an initial position of $\frac{\pi}{6}$. The testing references are chosen to be 0. This way, all controllers must carry out the system to the equilibrium point, starting from the given initial conditions.

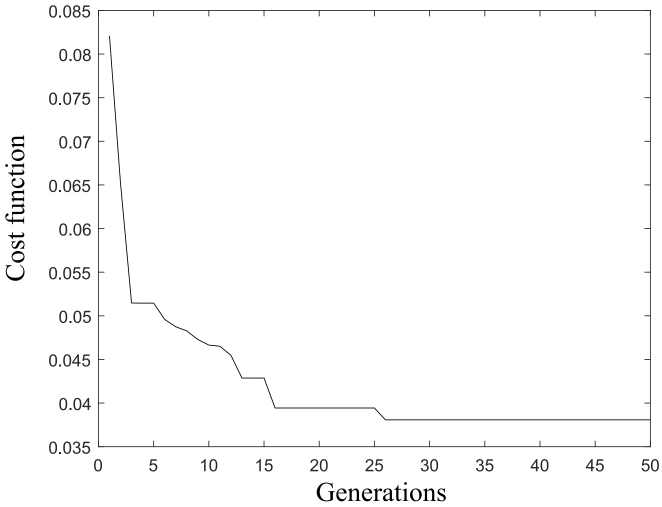

Fig. 3. Cost function evolution of the computed torque parameters tuning.

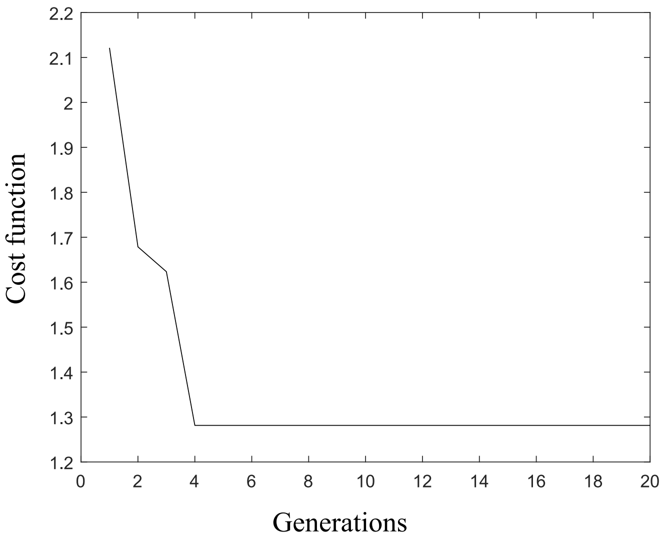

Fig. 4. Cost function evolution of the PID parameters tuning.

It is essential to clarify that each of the joints requires an individual controller. Thus, for each strategy, there is a set of three controller gains for the articulation hip, knee, and ankle.

By applying the genetic algorithm to tune the controller, the evolution of the computed torque's cost function is presented in Fig. 3.

Table II shows the gains found by the genetic algorithm for the calculated torque controller set.

TABLE II
CALCULATED TORQUE CONTROL PARAMETERS

CTC	k_p	k_d
Hip	205.1099	47.1080
Knee	181.3973	32.6045
Ankle	159.5147	1.9213

By applying the genetic algorithm to tune the controller, the evolution of the PID controller's cost function is presented in Fig. 4.

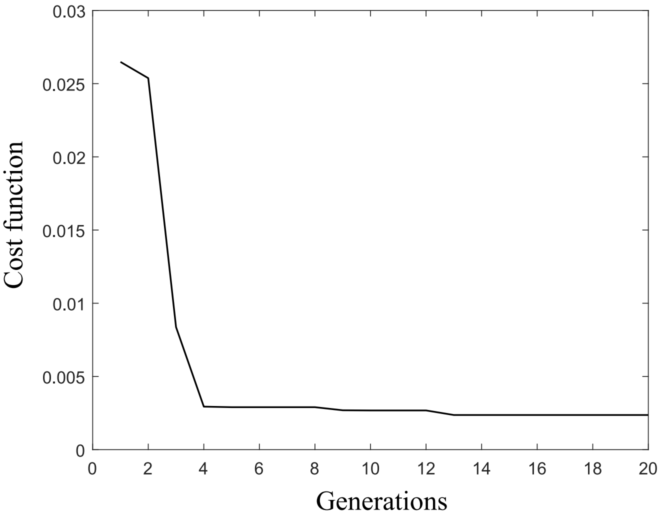

Fig. 5. Cost function evolution of the hyperbolic sine cosine parameters tuning.

Similarly, the gains for the PID controller are shown in Table III.

TABLE III
PID CONTROL PARAMETERS

PID	k_p	k_i	k_d
PID_1	6.9163	0.3638	7.5522
PID_2	8.2239	2.3298	0.8590
PID_3	9.3359	0.6616	4.5187

Also, by applying the genetic algorithm to tune the controller, the evolution of the hyperbolic sine cosine controller's cost function is presented in Fig. 5.

Finally, Table IV shows the hyperbolic sine cosine controller gains found by the genetic algorithm.

TABLE IV
HYPERBOLIC SINE-COSINE CONTROL PARAMETERS

HSC	k_p	k_d	G
Hip	0.5359	1.0270	0.9231
Knee	0.3033	0.1666	0.9737
Ankle	0.0331	0.4297	0.4955

A. Behavior Comparison

Figure 6 shows the simulation results for the output angles of each controller using the parameter gains obtained in Tables II - IV. The setpoint is defined as 0 to provide a slightly more straightforward tool for comparing behaviors.

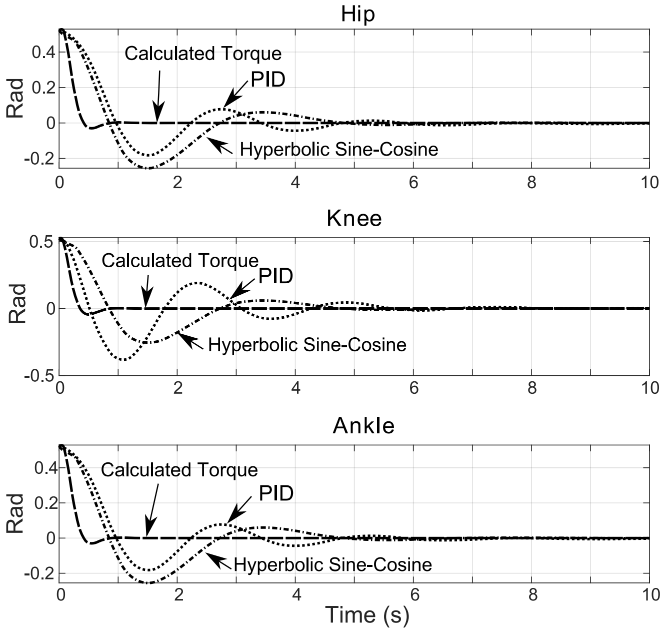

Fig. 6. Output angles obtained for each controller. **Upper**: hip output angle, **middle**: knee output angle, and **lower**: ankle output angle.

We can also perform a simulation using synthetic trajectories for each joint, as in [22].

In robotics, it is also important to take care of the control signal magnitude, since in practice, it is not possible to provide large control signals. As aforementioned, the analysis points are rotating systems, so a torque must be applied to them. Figure 7 shows the torques for each of the controllers, which provides a tool to analyze the controllers' efficiency.

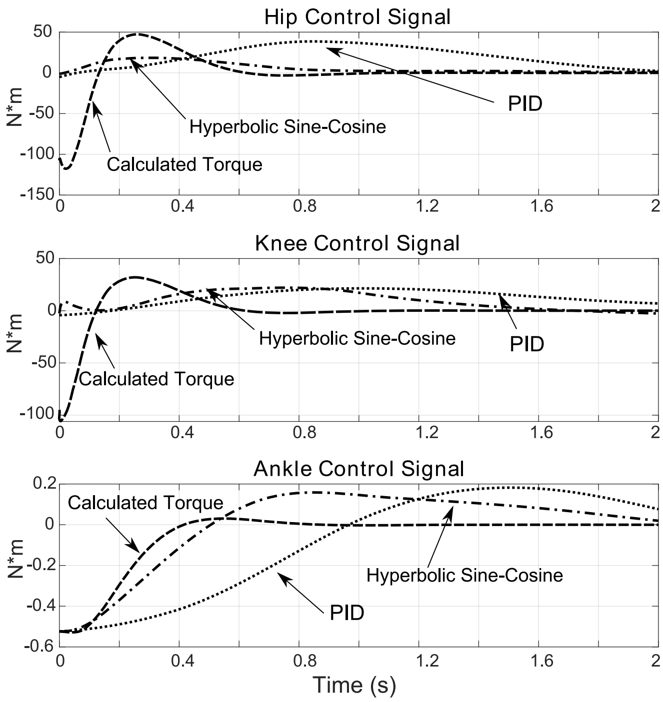

Fig. 7. Control signal for each joint.

B. Errors Comparison

For comparison, we use the mean square error metric (e_{RMS}) as in [23], where the authors compare the behavior of a redundant robot. In our case, we compute the e_{RMS}, as shown in (14) for all the outputs in each controller.

$$e_{RMS} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} \tag{14}$$

where e_i is the deviation error defined in (10).

The performed simulations provide deviation errors. This information is used to draw a comparative graph between the studied controllers. Intuitively, the plot with the least errors would have the best performance. Figure 8 show the obtained mean square errors.

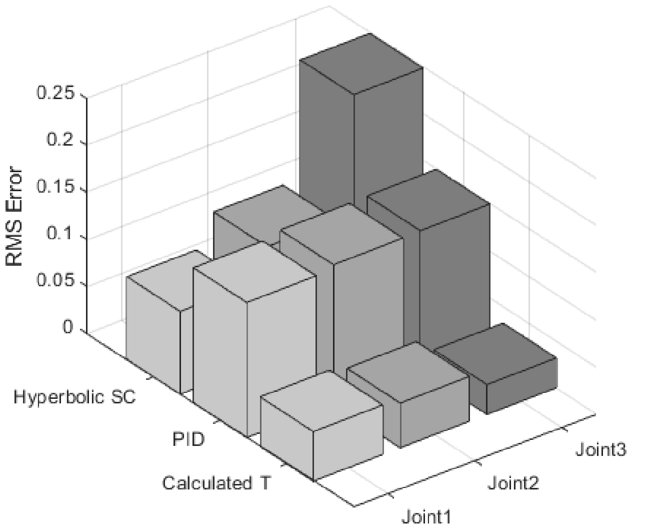

Fig. 8. RMS errors of each Controller for each Joint.

In order to have another perspective to compare the effectiveness of the controllers driving the robot. Figure 9 presents the deviation errors along time for each controller, which allows doing another behavior assessment of the limbs.

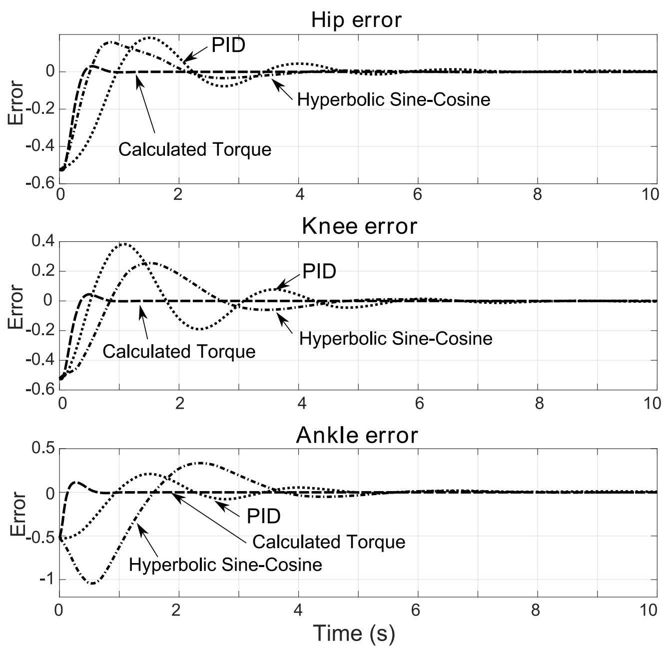

Fig. 9. Errors of each for each Joint.

V. Conclusions

In this paper, a bipedal robot's dynamic model with 6 degrees of freedom (3 for a leg) was developed, and some control strategies were applied for assessment. The model is based on a simple triple pendulum without friction. It was obtained through the Denavit-Hatenberg parameters, and an Euler-Lagrange framework was used to derive the movement equations. The approach shows to give a good representation of the real system.

We show the system's behavior driven by three different controllers: The calculated torque, the PID controller, and the hyperbolic sine-cosine strategy. Those are three SOTA approaches in robotic systems control. The results are shown in Fig. 6, Fig. 7, and Fig. 9. Figure 6 has the angular positions of the articulations. The calculated torque controller has an excellent settling time. In contrast, the hyperbolic sine-cosine and PID controllers show similar behavior with a little bit longer settling time. It is also supported by the deviation errors in Fig. 9. Figure 7 shows the control signal. We can see good control torques for the hyperbolic sine-cosine and PID controllers and a bigger but still possible control signal for the calculated torque controller.

The root mean square error, is presented as a bar graph in Fig. 8, is used for a final comparison of the implemented control techniques. We can quickly identify that the calculated torque controller reaches the least error, and, in cumulative average, the PID controller gets the maximum error.

This type of "low-level" controllers are advantageous because the desired trajectories can be computed off-line, decreasing the time consumption and the computational costs compared to other more advanced techniques.

Finally, we evidenced that using computational simulators to develop control strategies on robots is very useful since it offers a numerical and visual representation of the real system's behavior. The above allows for formulating and evaluating new robotic models and control laws, even if the mechanisms are not physically available.

References

[1] S. Schütz, A. Nezhadfard, K. Mianowski, P. Vonwirth, and K. Berns, "Carl — a compliant robotic leg featuring mono- and biarticular actuation," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics*, 11 2017, pp. 289–296.

[2] C. Chevallereau, G. Bessonnet, G. Abba, and Y. Aoustin, *Bipedal robots: modeling, design and walking synthesis*. John Wiley & Sons, 2013.

[3] B. T. Jensen and M. Niss, "Modeling, simulation, and control of biped robot aau-bot1," Ph.D. dissertation, Master's thesis, Aalborg University, 2009.

[4] J.-K. Han, "Bipedal walking for a full size humanoid robot utilizing sinusoidal feet trajectories and its energy consumption," Ph.D. dissertation, Virginia Tech, 2012.

[5] N. T. Phuong, D. W. Kim, H. K. Kim, and S. B. Kim, "An optimal control method for biped robot with stable walking gait," in *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2008, pp. 211–218.

[6] F. Osorio-Arteaga, J. J. Marulanda-Durango, and E. Giraldo, "Robust multivariable adaptive control of time-varying systems," *IAENG International Journal of Computer Science*, vol. 47, no. 4, pp. 605–612, 2020.

[7] F. Osorio-Arteaga, D. Giraldo-Buitrago, and E. Giraldo, "Sliding mode control applied to mimo systems," *Engineering Letters*, vol. 27, no. 4, pp. 802–806, 2019.

[8] C. D. Molina-Machado and E. Giraldo, "Bio-inspired control based on a cerebellum model applied to a multivariable nonlinear system," *Engineering Letters*, vol. 28, no. 2, pp. 464–469, 2020.

[9] R. Sanjuan Cerveró, P. Jiménez Honrado, E. Gil Monzó, R. Sánchez Rodríguez, and J. Fenollosa Gómez, "Biomecánica de las prótesis de rodilla," *Patol. apar. locomot. Fund. Mapfre Med*, pp. 242–259, 2005.

[10] A. Elhasairi and A. Pechev, "Humanoid robot balance control using the spherical inverted pendulum mode," *Frontiers in Robotics and AI*, vol. 2, p. 21, 2015.

[11] T. P. Valayil, V. Selladurai, and N. R. Ramaswamy, "Kinematic modeling of a serial robot using denavit-hartenberg method in matlab," *(TAGA) Journal of grahic technology*, vol. 14, pp. 2347–2445, 2018.

[12] F. Pérez Menéndez, "Desarrollo de sistema de control para manipulador de seis grados de libertad," Ph.D. dissertation, Universidad de Oviedo, 2014.

[13] C. Chen, C. Zhang, T. Hu, H. Ni, and W. Luo, "Model-assisted extended state observer-based computed torque control for trajectory tracking of uncertain robotic manipulator systems," *International Journal of Advanced Robotic Systems*, vol. 15, no. 5, p. 1729881418801738, 2018.

[14] K. Ogata and Y. Yang, *Modern control engineering*. Prentice-Hall, 2002, vol. 4.

[15] K. J. Åström and T. Hägglund, *PID controllers: theory, design, and tuning*. Instrument society of America Research Triangle Park, NC, 1995, vol. 2.

[16] ——, *Control PID avanzado*. Pearson, Madrid, 2009.

[17] R. Gautam and A. T. Patil, "Modeling and control of joint angles of a biped robot leg using pid controllers," in *2015 IEEE international conference on engineering and technology (ICETECH)*. IEEE, 2015, pp. 1–5.

[18] L. Espinosa, J. Barahona, and F. Reyes, "Nuevos controladores de posición para robot manipuladores basados en funciones hiperbólicas," in *Segundo Congreso Nacional De Electrónica: Ciencia y Tecnología Con Sentido Humano. Presented at the Robótica y Control, Centro de Convenciones William O. Jenkins, Puebla, Mexico*, 2002.

[19] S. Mirjalili, J. S. Dong, and A. Lewis, *Nature-inspired optimizers: theories, literature reviews and applications*. Springer, 2019, vol. 811.

[20] Z. Wang and A. Sobey, "A comparative review between genetic algorithm use in composite optimisation and the state-of-the-art in evolutionary computation," *Composite Structures*, vol. 233, p. 111739, 2020.

[21] N. Saini, "Review of selection methods in genetic algorithms," *International Journal of Engineering and Computer Science*, vol. 6, no. 12, pp. 22 261–22 263, 2017.

[22] S. Cuatlaxahue-Formacio and P. Sánchez-Sánchez, "Simulador basado en un sistema bípedo simple," *Pistas Educativas*, vol. 35, no. 108, 2018.

[23] C. Urrea and J. Kern, "Design, simulation and comparison of controllers for a redundant robot," *Case Studies in Mechanical Systems and Signal Processing*, vol. 3, pp. 9–21, 2016.