Modeling of an 18 DoF Humanoid Robot Using a Recursive Analytical Method

Miguel Angel Ortega-Palacios, Amparo Palomino-Merino, and Fernando Reyes-Cortés

***Abstract*— Humanoid robots have been widely studied by the scientific community to understand and reproduce human motion in a variety of tasks. Over the past decades, numerous modeling strategies have been proposed for describing humanoid dynamics, including analytical formulations, geometric modeling, and the Denavit–Hartenberg convention. In this work, a recursive formulation based on Euler's analytical mechanics is presented, and leverages the properties of the skew-symmetric matrix (S) to address key modeling challenges, including forward kinematics, differential kinematics, linear and angular velocity computation, kinetic and potential energy calculation, and the Euler–Lagrange equations of an 18-degree-of-freedom (DoF) humanoid robot. Furthermore, three position controllers—Proportional-Derivative (PD), Hyperbolic Tangent, and Saturation-based control—are implemented and evaluated using the virtual model of robot. Their performance is validated in simulation using a virtual model of the humanoid robot developed in Simscape Multibody, where a dynamic walking gait is executed. A comparative analysis is presented to assess tracking performance for each controller. Finally, the simulation results in MATLAB and Simscape demonstrate the applicability and computational advantages of the proposed modeling and control framework.**

Link to graphical and video abstracts, and to code: https://latamt.ieeer9.org/index.php/transactions/article/view/10058

***Index Terms*— Dynamics, Euler, humanoid robot, recursive method, skew-symmetric matrix.**

I. Introducción

THE study of the dynamics of biped robots has been extensively investigated in the scientific community. However, several limitations have been identified in the modeling approaches currently available in the literature, highlighting the need for alternative formulations. In this regard, we propose the use of Euler's analytical mechanics, which allows the kinematic and dynamic modeling to be addressed through a recursive approach. This is particularly relevant from a computational standpoint, as the development of a recursive algorithm capable of obtaining the dynamic model of any biped robot is envisioned for future work.

The modeling of humanoid robots has been approached through dynamic analysis [1–3]. Most of the reviewed works apply the Euler-Lagrange methodology [1, 4–9] to derive the dynamic model, while others rely on the Newton-Euler formulation [10–13]. All of them aim to implement stable gait patterns using control techniques that ensure robot balance during locomotion, such as PD control strategies [14–16].

It is important to note that most authors focus primarily on modeling the dynamics of the legs, occasionally including the torso, with the main goal of achieving stable walking trajectories.

Kajita's work [17] extensively studied the HRP humanoid robot series from 2000 to the present. The objective is to deploy practical humanoid robots in large-scale assembly industries such as construction, aerospace, and shipbuilding. Nevertheless, practical deployment in real environments remains unachieved, largely due to the insufficient physical capabilities of current HRP robots.

The development of biped gait models continues to be an active research area. The challenge of humanoid control is often simplified using an inverted pendulum approach. This model not only predicts the humanoid robot's ZMP trajectory without requiring full multibody dynamics but also generates the CoM trajectory for ZMP tracking. This approach has been implemented in the HRP-4C humanoid robot [18].

Despite the extensive research on humanoid robot dynamics, finding studies that explicitly develop the complete dynamic model for humanoid robots with 18 degrees of freedom (DoF) remains challenging. Most recent works focus on general whole-body dynamics or simplified models with fewer joints (e.g., 6-DoF or 12-DoF systems), while others address higher-DoF structures without publishing the full derivation of their dynamic equations. Recursive formulations, such as the Newton–Euler or Euler–Lagrange approaches, are often applied to subsystems or simplified kinematic chains, leaving few references that detail the full 18-DoF configuration. The limited availability of such studies is mainly due to the computational complexity and the numerical instability that arise when modeling and solving the coupled dynamic equations of multi-body humanoid systems. One of the few known works addressing an 18-DoF humanoid robot [19], which focuses on gait generation rather than on the analytical derivation of the complete dynamic model. Therefore, the development of an explicit and computationally efficient dynamic model for a humanoid robot with 18 DoF — particularly through recursive and matrix-based methods such as the skew-symmetric formulation — represents a relevant and still underexplored

The associate editor coordinating the review of this manuscript and approving it for publication was Javier Moreno-Valenzuela (*Corresponding author: Miguel Angel Ortega Palacios*).

Miguel Angel Ortega Palacios, A. Palomino-Merino, and F. Reyes-Cortés are with the Benemérita Universidad Autónoma de Puebla, Puebla, México (e-mails: miguel.ortegap@alumno.buap.mx, amparo.palomino@correo.buap.mx, and fernando.reyes@correo.buap.mx).

research contribution.

The Bioloid robot has been widely used by the scientific community for various humanoid research topics. For example, it has been employed in gait control studies based on genetic algorithms [20–23] and fuzzy logic [24]. Other works utilize the kinematic model of the Bioloid robot for gait generation [25].

Most works derive the kinematic model of the legs using a single foot as the base reference frame [26], while some use the torso [2, 16, 27], and others model both arms and legs using the pelvis and torso as reference frames [28, 29]. In [30], a recursive algorithm based on Euler angles is used to solve the forward kinematics of the CHARMIE anthropomorphic robot.

Based on this review, we propose to study the Bioloid Premium robot with 18 degrees of freedom (DOF). The primary motivation of this article is to develop a methodology based on Euler's analytical mechanics. We propose a solution to the problems of forward kinematics, differential kinematics, linear velocity, and kinetic energy for the Bioloid Premium robot. The methodology considers a recursive model of its four limbs using the pelvis as the initial reference frame. This recursive approach is computationally attractive and paves the way for developing efficient recursive algorithms.

The document is organized as follows. Section II presents the development of Euler's analytical methodology to obtain the dynamic model of the robot. Quantitative analysis of the recursive method is presented in Section III. Implementation of the control algorithms are presented in Section IV. Finally, conclusions and future work are discussed in Section V.

II. Methodology

A. Recursive Forward Kinematics

We propose the generation of four open kinematic chains to describe the position and orientation of each link of the Bioloid Premium robot, using the robot's pelvis as the initial reference frame $\Sigma_0(x_0, y_0, z_0)$. Fig. 1 shows the reference frames assigned to each joint of the robot.

The dimensions of the robot are as follows: $l_0 = 33\,mm, d_1 = 118\,mm, d_2 = 73\,mm, l_1 = l_2 = 76\,mm, l_3 = 77\,mm, l_4 = l_6 = 108\,mm, l_5 = 66\,mm$.

Using the Denavit-Hartenberg convention and the corresponding coordinate transformation [31]. In order to compute the cartesian position of each link of the humanoid robot, a recursive formulation of the forward kinematics is adopted. Let $\boldsymbol{p}_i \in \mathbb{R}^3$ denote the position of link i with respect to the base reference frame. Each link is connected through a rotation about the z-axis, followed by a rotation about the x-axis, and a displacement vector representing the geometric parameters of the robot.

The orientation of link i with respect to the base frame is defined as the accumulated product of the elementary rotations of the preceding joints, as shown in (1):

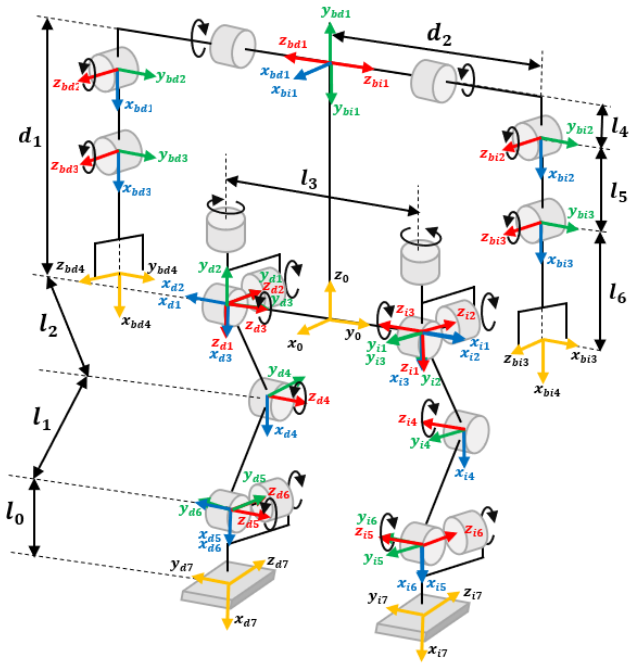

Fig. 1. Reference frames assigned to the joints of the Bioloid robot

$$R_0^i = \prod_{k=1}^{i} [R_{z_k} R_{x_k}] \tag{1}$$

where R_{z_k} is the rotation matrix about the z-axis of link k, R_{x_k} is the rotation matrix about the x-axis of link k.

For computational clarity, the cumulative rotation can be expressed recursively, as given in (2):

$$R_0^i = R_0^{i-1} R_{z_i} R_{x_i} \tag{2}$$

where $R_0^0 = I$, and I denotes the identity matrix. This recurrence propagates the orientation from the base to the end-effector using only matrix multiplications.

Let $d_i \in \mathbb{R}^3$ denote the displacement vector from frame $i-1$ to frame i, expressed in the local coordinate frame of link i. The position of link i in the global frame is computed recursively as in (5):

$$\boldsymbol{p_i} = \boldsymbol{p_{i-1}} + R_0^i \, R_{z_i} \boldsymbol{d_i} \tag{3}$$

with the initial condition defined in (4):

$$\boldsymbol{p_1} = R_{z_1} \boldsymbol{d_1} \tag{4}$$

where $\boldsymbol{p}_i$ is the cartesian position of link i un the base reference frame, $\boldsymbol{p}_{i-1}$ is the position of the previous link, from which the current pose is propagated, R_0^i is the total rotation of link i with respect to the base frame, and $\boldsymbol{d}_i$ is the local displacement from joint $i-1$ to joint i, containing link geometry (lengths and offsets).

This formulation builds the Cartesian position of each link incrementally, progressing forward along the kinematic chain.

Using the parameters shown in Tables I and II as an example, the forward kinematics of each link of the robot can be obtained.

Depending on the robot configuration, it is important to consider the center of mass in the parameter d_i, when necessary.

It is also important to note that the parameters of p_0 correspond to the coordinate transformation from the pelvis to

the first link of each extremity of the robot.

TABLE I
PARAMETERS TO COMPUTE THE FORWARD KINEMATICS OF EACH LINK OF THE RIGHT LEG

Position p_{i-1}	Matrix $R_{xi-1}(\phi)$	Matrix $R_{zi-1}(\theta)$	Displacement d_{i-1}
p_0	$R_{x0} = R_{x0}(\pi)$	$R_{z0} = R_{z0}(-\pi/2)$	$d_0 = [l_3/2 \quad 0 \quad 0]^T$
p_1	$R_{x1} = R_{x_1}(-\pi/2)$	$R_{z1} = R_{z1}(q_{d1})$	$d_1 = [0 \quad 0 \quad 0]^T$
p_2	$R_{x2} = R_{x_2}(\pi/2)$	$R_{z2} = R_{z2}(q_{d2})$	$d_2 = [0 \quad 0 \quad 0]^T$
p_3	$R_{x3} = R_{x_3}(0)$	$R_{z3} = R_{z3}(q_{d3})$	$d_3 = [l_2 \quad 0 \quad 0]^T$
p_4	$R_{x4} = R_{x_4}(0)$	$R_{z4} = R_{z4}(q_{d4})$	$d_4 = [l_1 \quad 0 \quad 0]^T$
p_5	$R_{x5} = R_{x_5}(-\pi/2)$	$R_{z5} = R_{z5}(q_{d5})$	$d_5 = [0 \quad 0 \quad 0]^T$
p_6	$R_{x6} = R_{x_6}(0)$	$R_{z6} = R_{z6}(q_{d6})$	$d_6 = [l_0 \quad 0 \quad 0]^T$

TABLE II
PARAMETERS TO COMPUTE THE FORWARD KINEMATICS OF EACH LINK OF THE RIGHT ARM

Position p_{i-1}	Matrix $R_{xi-1}(\phi)$	Matrix $R_{zi-1}(\theta)$	Displacement d_{i-1}
p_0	$R_{x0} = R_{x0}(\pi/2)$	$R_{z0} = R_{z0}(0)$	$d_0 = [0 \quad 0 \quad d_1]^T$
p_1	$R_{x1} = R_{x_1}(-\pi/2)$	$R_{z1} = R_{z1}(q_{d7})$	$d_1 = [l_4 \quad 0 \quad d_2]^T$
p_2	$R_{x2} = R_{x_2}(0)$	$R_{z2} = R_{z2}(q_{d8})$	$d_2 = [l_5 \quad 0 \quad 0]^T$
p_3	$R_{x3} = R_{x_3}(0)$	$R_{z3} = R_{z3}(q_{d9})$	$d_3 = [l_6 \quad 0 \quad 0]^T$

To compute the forward kinematics of the left leg, the signs of parameters R_{z0}, R_{x1}, R_{x2} and R_{x5} are changed. In the case of the left arm, only the sign of parameters R_{x0} and R_{x1} are changed.

B. Linear Component of Differential Kinematics

The robot's linear differential kinematics is obtained by differentiating the position vector $\boldsymbol{p}_i$. For this, the skew-symmetric matrix $S \in \mathbb{R}^{3x3}$ in expression (5) is used [31, 32].

$$S = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{5}$$

Using the properties of the matrix S, the linear velocity of each link can be computed using Equation (6):

$$\boldsymbol{v}_i = \boldsymbol{v}_{i-1} + \prod_{k=1}^{i} R_{z_{k-1}} R_{x_{k-1}} S \boldsymbol{p}_i \dot{q}_i \tag{6}$$

where $\boldsymbol{v}_i = \dot{\boldsymbol{p}}_i$ is the linear velocity of link i, $\boldsymbol{v}_{i-1}$ is the linear velocity of link $i-1$, S is the Skew-symmetric matrix, $\dot{q}_i$ is the time derivative of q_i, which corresponds to each joint angle θ_i with the initial condition: $\boldsymbol{v}_0 = [0 \quad 0 \quad 0]^T$. The angular component of the differential kinematics is developed in the subsequent expressions, and together both linear and angular terms complete the differential kinematic model of the robot

C. Recursive Linear Velocity

To determine the linear velocity of each link, the transpose of Equation (6) is considered. The linear velocity is computed by squaring each velocity component $\|v_i\|^2 = (v_i)^T(v_i)$, resulting in expression (7):

$$\boldsymbol{v}_i{}^2 = (\boldsymbol{v}_{i-1})^2 + \boldsymbol{p}_i{}^T S^T S \boldsymbol{p}_i \dot{q}_i{}^2 + 2\sum_{j=1}^{i-1}\left[\boldsymbol{p}_j{}^T S^T \left(\prod_{m=j}^{i-1}[R_{z_m} R_{x_m}]\right) S \boldsymbol{p}_i \dot{q}_i \dot{q}_j\right] \tag{7}$$

D. Recursive Inertia Matrix

To compute the inertia matrix $\boldsymbol{M}(\boldsymbol{q})$ of the robot, the Jacobians ${J_{v_0}}^i \in \mathbb{R}^{3xn}$ must first be determined, where q denotes the joint position of each angle θ, n is the number of robot links and i ranges from 1 to n. A general expression is proposed using the rotation matrices $R_{z_i}(\theta)$ and $R_{x_i}(\phi)$, the displacement vector d_i, and the skew-symmetric matrix S. Each ${J_{v_0}}^i$ can be expressed as in Equation (8).

$${J_{v_0}}^i = {J_{v_0}}^{i-1} + T_i \tag{8}$$

Considering ${J_{v_0}}^1 = T_1$, where $T_i \in \mathbb{R}^{3xn}$ is a matrix that adds terms to each column of ${J_{v_0}}^{i-1}$, with column j of T_i denoted by (9):

$$T_{i_{(1:3,j)}} = \prod_{k=0}^{j-1}\left(R_{z_k} R_{x_k}\right) S \prod_{k=j}^{i-1}\left(R_{z_k} R_{x_k}\right) p_i \tag{9}$$

Additionally, if $k \geq i-1$, then term $\prod_{k=j}^{i-1}\left(R_{z_k} R_{x_k}\right)$ becomes the identity matrix.

By multiplying the transposed of linear velocity jacobian by the same Jacobian $\left({J_{v_0}}^i\right)^T\left({J_{v_0}}^i\right) \in \mathbb{R}^{nxn}$, we obtain Equation (10):

$$\left({J_{v_0}}^i\right)^T\left({J_{v_0}}^i\right) = \left({J_{v_0}}^{i-1}\right)^T {J_{v_0}}^{i-1} + \left({J_{v_0}}^{i-1}\right)^T T_i + T_i{}^T {J_{v_0}}^{i-1} + T_i{}^T T_i \tag{10}$$

Subsequently, the angular velocity jacobian of each robot link is obtained. First, the rotation matrices R_0^i that contain the angular velocity components are determined using Equation (11):

$$R_0^i = \prod_{j=0}^{i}\left[R_{z_j}(\theta) R_{x_j}(\phi)\right] \text{ para } i = 0,1,\ldots,n \tag{11}$$

where $R_0^i \in \mathbb{R}^{3x3}$ and n denote the number of robot links. The angular velocities $\boldsymbol{\omega_z}{}_0^i$ about the $\boldsymbol{z}$ axis are extracted by multiplying the resulting matrix R_0^i by the vector $\boldsymbol{z}(\boldsymbol{k}) = [0 \quad 0 \quad 1]^T$. This yields the angular velocity vectors of each link about the $\boldsymbol{z}$ axis, as shown in (12).

$$\boldsymbol{\omega_z}{}_0^i = R_0^i \boldsymbol{z}(\boldsymbol{k}) \tag{12}$$

where $\boldsymbol{\omega_z}{}_0^i \in \mathbb{R}^{3x1}$. The vectors $\boldsymbol{\omega_z}{}_0^i$ are used to construct the angular velocity jacobians $J_\omega{}_0^i$ for each link. Equation (13) defines the angular velocity Jacobian matrix:

$$J_\omega{}_0^i = [\boldsymbol{\omega_z}{}_0^1 \quad \boldsymbol{\omega_z}{}_0^2 \quad \ldots \quad \boldsymbol{\omega_z}{}_0^i] \tag{13}$$

where $J_\omega{}_0^i \in \mathbb{R}^{3xn}$. Using Equations (8) and (13), expression (14) defines the robot's inertia matrix:

$$M(q) = \sum_{i=1}^{n}\left[m_i \left({J_{v_0}}^i\right)^T\left({J_{v_0}}^i\right) + \left({J_{\omega_0}}^i\right)^T I_i \left({J_{\omega_0}}^i\right)\right] \tag{14}$$

where m_i is the mass of inertia of i^{th} link; ${J_{v_0}}^i \in \mathbb{R}^{3xn}$ is the linear velocity Jacobian; ${J_{\omega_0}}^i \in \mathbb{R}^{3xn}$ is the angular velocity Jacobian; and I_i is the inertia tensor. The matrix $M(q) \in \mathbb{R}^{nxn}$ is symmetric and positive definite. Recursively:

$$\boldsymbol{M_i}(\boldsymbol{q}) = \boldsymbol{M_{i-1}}(\boldsymbol{q}) + m_i \left({J_{v_0}}^i\right)^T\left({J_{v_0}}^i\right) + \left({J_{\omega_0}}^i\right)^T I_i \left({J_{\omega_0}}^i\right) \tag{15}$$

for $i = 2$ to n, where:

$$\boldsymbol{M_1}(\boldsymbol{q}) = m_1 \left({J_{v_0}}^1\right)^T\left({J_{v_0}}^1\right) + \left({J_{\omega_0}}^1\right)^T I_1 \left({J_{\omega_0}}^1\right) \tag{16}$$

E. Potential Energy

To compute the robot's potential energy, the heights of the center of mass (CoM) of each link must first be determined relative to the feet of the robot. This requires the use of the forward kinematics parameters listed in Tables 1 and 2. Once obtained, these vectors are multiplied by transposed vector $\boldsymbol{z(k)^T} = [0 \quad 0 \quad 1]$. Accordingly, Equation (17) defines how to compute the vertical positions of the CoM of each link of the robot:

$$h_i(q) = -\boldsymbol{z(k)^T p_n} \tag{17}$$

where $\boldsymbol{p_n}$ is the forward kinematics vector of the n link.

F. Lagrangian of the Robot

The Lagrangian of the robot is obtained as the difference between the kinetic energy and the potential energy, as presented in Equation (18).

$$\boldsymbol{\mathcal{L}(q,\dot{q}) = \mathcal{K}(q,\dot{q}) - \mathcal{U}(q)} \tag{18}$$

where $\dot{q}_i$ denotes the time derivative of q_i, which corresponds to each joint angle θ_i.

By expanding the above expression, the resulting formulation is given by equation (19):

$$\boldsymbol{\mathcal{L}(q,\dot{q})} = \sum_{i=1}^{n} \left(\frac{1}{2} [m_i \, (\boldsymbol{v_i}^T \boldsymbol{v_i}) + \boldsymbol{\omega_i}^T I_i \, \boldsymbol{\omega_i}] - m_i g h_i(q) \right) \tag{19}$$

where $\boldsymbol{\omega_i} \in \mathbb{R}^{3x1}$ represents the angular velocity of the center of mass of the i^{th} link; which is computed using each of the matrices $J_{\omega_0}^{\ i}$, as defined in Equation (13). Thus, the angular velocity $\boldsymbol{\omega}_i$ is obtained through expression (20):

$$\boldsymbol{\omega}_i = \left(J_{\omega_0}^{\ i}\right)\dot{\boldsymbol{q}} \tag{20}$$

G. Euler-Lagrange Equations

The equations of motion of the robot, based on the Euler-Lagrange formulation, are given by the expression (21):

$$\boldsymbol{\tau_i} = \frac{d}{dt}\left[\frac{\partial \boldsymbol{\mathcal{L}}(q,\dot{q})}{\partial \dot{\boldsymbol{q}}_i}\right] - \frac{\partial \boldsymbol{\mathcal{L}}(q,\dot{q})}{\partial \boldsymbol{q}_i} + f_f(\dot{\boldsymbol{q}}_i) \tag{21}$$

for $i = 1,2,\ldots,n$. Where $\boldsymbol{\tau_i}$ denotes the input torque applied to the i^{th} joint. Additionally, equation (18) includes friction denoted as $\boldsymbol{f_f}(\dot{\boldsymbol{q}}_i) \in \mathbb{R}^n$ for each joint of the robot.

The friction present in the robot is modeled as a linear combination of as shown in equation (22).

$$f_f(\dot{\boldsymbol{q}}_i) = b_i \dot{q}_i + f_{ci} sign(\dot{q}_i) \tag{22}$$

where the terms b and f_c represent the viscous and Coulomb friction respectively.

III. Quantitative Analysis of the Recursive Method

A quantitative comparison between the proposed recursive method and the conventional Euler–Lagrange formulation is proposed, both implemented using symbolic programming in MATLAB.

The tests were conducted on a standard laptop computer, executing the codes that compute the forward kinematics, Jacobian matrices, and the inertia matrix of the humanoid robot. Each program was executed 20 times to obtain the average execution time and its corresponding standard deviation.

The analysis considers only the symbolic computation up to the generation of the inertia matrix, excluding numerical evaluation or dynamic simulation stages.

Performance was evaluated for robot configurations with 3 DoF, 6 DoF, 9 DoF, and 18 DoF, corresponding to different subsets of the humanoid robot model.

For each configuration, the execution time of both methods was recorded, and the speed-up factor of the recursive method with respect to the conventional approach was calculated.

Table III indicates that the recursive method consistently achieves a computational acceleration of approximately 2× compared to the conventional symbolic derivation.

Furthermore, the recursive algorithm exhibits a near-linear growth in computation time as the number of degrees of freedom increases, while the conventional approach shows a quadratic trend, confirming the better scalability and computational efficiency of the recursive formulation.

TABLE III
Quantitative Comparison between Recursive and Conventional Methods

DoF	Recursive method Time ± Std. Deviation (s)	Conventional method Time ± Std. Deviation (s)	Speed-up
3	0.0320 ± 0.1430	0.0663 ± 0.2965	2.07
6	0.0781 ± 0.3492	0.1560 ± 0.6976	2.00
9	0.0920 ± 0.4117	0.1943 ± 0.8692	2.11
18	0.1714 ± 0.7665	0.3662 ± 1.6377	2.14

Figs. 2 and 3 compare the computational performance of the proposed recursive formulation and the conventional Euler–Lagrange approach.

The recursive method achieves roughly twice the computational efficiency, with execution time increasing linearly $O(n)$ as the number of DoF increases, while the conventional symbolic derivation shows a near-quadratic trend $O(n^2)$.

These results demonstrate the superior scalability and efficiency of the recursive algorithm for symbolic dynamic modeling of humanoid robots.

Fig. 2 illustrates the average symbolic computation time for both methods, including error bars that represent the standard deviation across 20 executions. The recursive method consistently exhibits lower execution times than the conventional formulation for all robot configurations (3, 6, 9, and 18 DoF).

Fig. 3 shows the speed-up ratio (acceleration factor) of the recursive method relative to the conventional one. The recursive algorithm maintains an approximately constant acceleration of about 2× faster across all configurations, confirming its linear computational growth with respect to the number of degrees of freedom.

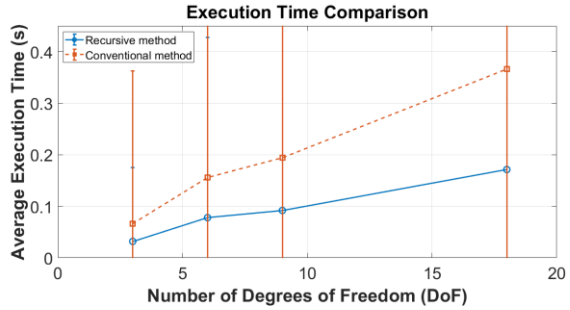

Fig. 2. Average execution time of the recursive and conventional methods for different numbers of degrees of freedom (DoF).

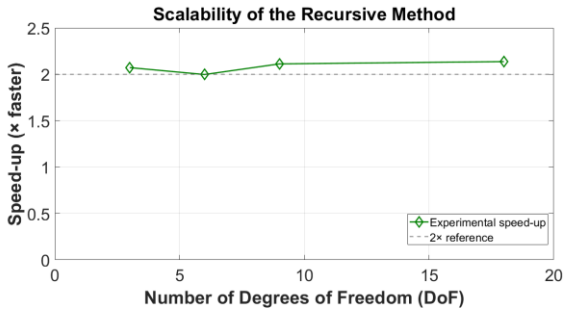

Fig. 3. Computational speed-up of the recursive method relative to the conventional approach.

IV. Position Control Implementation

The following section presents the simulation results obtained in Simulink for the position control of the 18-degree-of-freedom biped robot using three different controllers.

The control algorithms could not be implemented on the physical Bioloid Premium robot because its joints are equipped with standard servomotors rather than DC motors with encoders. These servomotors do not feature an open architecture that would allow direct implementation of custom control algorithms for torque output. Therefore, a virtual model of the Bioloid Premium robot was developed in Simscape Multibody, enabling the testing and validation of the proposed control strategies in a simulated environment before any potential real-world application.

The results of the position control implemented within the recursive method for obtaining the dynamic model of the robot are presented. To summarize the findings, the response of the joints corresponding to the right leg and right arm of the Bioloid Premium robot is shown. This selection is justified because these representative joints allow illustrating the performance of the control strategies without redundancy, while providing a clear view of the system's dynamic behavior.

Saturation occurs when the control law requires a torque beyond the limits of the electronic servo amplifier. In other words, since the amplifier cannot exceed its physical limits, it simply provides the maximum torque τ_{max}.

To address this, a bounded control strategy is proposed such that the position error $\tilde{q}$ and velocity $\dot{q}$ converge asymptotically to zero, regardless of the robot's initial conditions. Three types of controllers were implemented for comparison: a standard PD controller, a saturated control, and a hyperbolic tangent controller. The PD controller is described by equation (20):

$$\boldsymbol{\tau} = K_p\tilde{q} - K_v\dot{q} + g(q) \tag{23}$$

where $\boldsymbol{\tau}$ denotes the control input in the model of the equation (18). Additionally, equation (18) K_p is the proportional gain, and K_v is the derivative gain. The gains were selected as follows: $K_p = 15, K_v = 1.5$.

The second proposed controller uses a saturated controller, described by equation (25):

$$\boldsymbol{\tau} = K_p\left[\frac{arctan(\alpha\tilde{q})}{\sqrt{1+tanh^2(\alpha\tilde{q})}}\right] - K_v\left[\frac{arctan(\alpha\dot{q})}{\sqrt{1+tanh^2(\alpha\dot{q})}}\right] + g(q) \tag{24}$$

where α is a positive constant. The values were selected as follows: $K_p = K_v = 1.5, \alpha = 50$ for the legs, and $\alpha = 4$ for the arms.

The third proposed controller uses a hyperbolic tangent function, described by equation (25):

$$\boldsymbol{\tau} = K_p\tanh(\Lambda\tilde{q}) - K_v\tanh(\Gamma\dot{q}) + g(q) \tag{25}$$

where $\Lambda, \Gamma \in \mathbb{R}^{nxn}$ are positive definite diagonal gain matrices. The gains were selected as follows: $K_p = 1.6, K_v = 1, \Lambda = 10, \Gamma = 1$.

The tuning of the gains depends on the maximum torque that the motors can deliver. To verify the performance of the proposed position control strategy, a simulation was conducted in MATLAB.

Fig. 4 illustrates the virtual model of the Bioloid Premium robot in its home position, which was first assembled in SolidWorks and then imported into Simscape Multibody within Simulink to create a virtual representation of the robot. This virtual model enables the testing of kinematic and dynamic behaviors in a controlled environment, ensuring accuracy and facilitating the development of control algorithms.

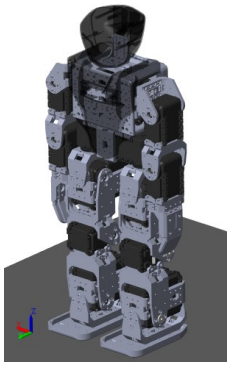

Fig. 4. Virtual model of the Bioloid Premium robot imported in Simscape Multibody.

To validate the proposed dynamic model of the robot, the three previously described joint position controllers were implemented and evaluated. A dynamic walking motion was generated by defining a parametric sinusoidal trajectory for each joint, with the objective of producing forward locomotion. No balance or high-level gait stabilization strategies (such as ZMP-based control, inverted pendulum models, or MPC) were incorporated. Therefore, robot motion is governed solely by the

commanded joint-space trajectories and the performance of the position controllers, allowing the validation to focus strictly on the accuracy of the dynamic model and the controller behavior without the influence of additional stabilization algorithms. The reference trajectory for each joint $q_i(t)$ is defined in (26):

$$q_i(t) = A_i \sin(\omega_i t + \phi_i) + q_{o,i} \quad (26)$$

where A_i is the amplitude of oscillation, ω_i is the angular frequency, ϕ_i is the phase offset used to coordinate symmetric joints (e.g., left and right legs), and $q_{0,i}$ is the nominal central position of the joint.

This function was selected for three main reasons. First, it provides smooth, continuous, and differentiable joint reference, which is essential for evaluating tracking performance under a well-defined dynamic motion. Second, sinusoidal parameterization allows gait-like periodic movements to be generated without the need for an explicit balance controller, reducing modeling and implementation complexity. Third, this approach is widely used in early-stage humanoid robot development when the intention is to analyze controller performance rather than full-body dynamic stability.

Using these trajectories, the robot performs a continuous forward walking motion for six seconds. During this interval, the 18 joint controllers operate in closed-loop position mode. The three control strategies—conventional PD, saturated control, and hyperbolic tangent control—are executed under identical motion conditions. Their performance is compared by analyzing joint tracking errors obtained from simulations of the full robot model in Simscape Multibody. This validation setup allows for a quantitative evaluation of controller efficiency under a dynamic task while keeping the system structure focused on joint-level control rather than whole-body stability control.

Fig. 5 illustrates the implementation of the proposed dynamic walking trajectory in the virtual humanoid model developed in Simscape Multibody. The generated joint references were applied to the simulated robot, and the resulting motion demonstrates the complete execution of the gait cycle under the dynamic conditions defined in the control framework. This visualization confirms that the computed joint coordinates, velocities, and torques from the recursive dynamic model can successfully drive the multibody simulation, validating the feasibility of the proposed trajectory generation and control approach in a realistic physics-based environment.

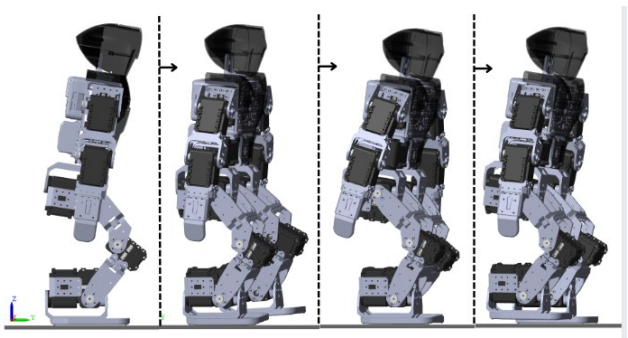

Fig. 5. Desired dynamic gait proposed for the robot.

To evaluate the accuracy of the proposed dynamic model, a comparative analysis was performed between the joint torques generated by the analytical model and those applied to the virtual robot implemented in Simscape. For this purpose, a walking sequence of six seconds was simulated, in which the proposed reference trajectory was applied to both systems. In both cases, the same saturated control law, defined in the manuscript, was used to track the desired motion. The joint torques were collected from the dynamic model and from the virtual robot, and the comparison was carried out only during the actual walking interval, intentionally excluding the initial and final posture transitions.

Fig. 6 to Fig. 11 show the torque error curves obtained by comparing the joint torques computed from the analytical dynamic model with the torques generated by the Simscape Multibody simulation. Only the leg joints are included, as they play the most critical role in trajectory generation and joint tracking control during dynamic walking. The figure highlights how closely the applied torques follow the model-based predictions throughout the motion. Additionally, Table IV reports the RMSE values for each joint, providing a quantitative measure of the torque tracking performance and confirming the consistency between the analytical model and the simulated robot.

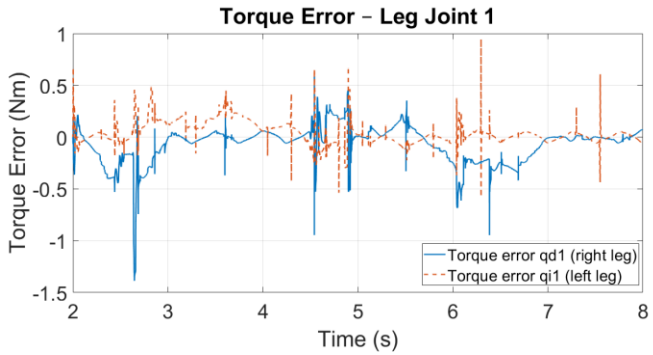

Fig. 6. Torque errors between the analytical dynamic model and the Simscape Multibody simulation for joints q_{d1} and q_{i1}.

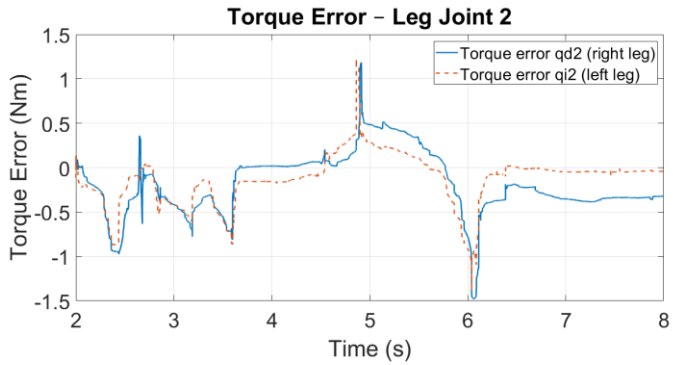

Fig. 7. Torque errors between the analytical dynamic model and the Simscape Multibody simulation for joints q_{d2} and q_{i2}.

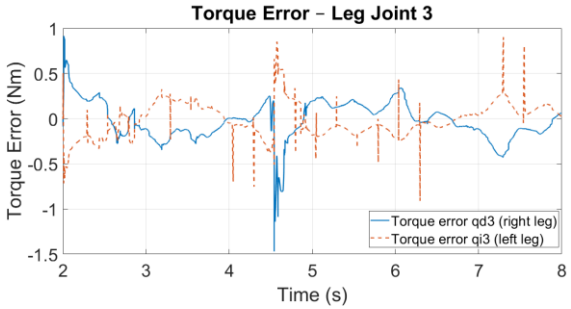

Fig. 8. Torque errors between the analytical dynamic model and the Simscape Multibody simulation for joints q_{d3} and q_{i3}.

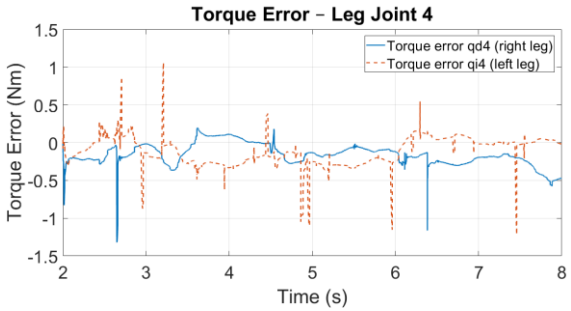

Fig. 9. Torque errors between the analytical dynamic model and the Simscape Multibody simulation for joints q_{d4} and q_{i4}.

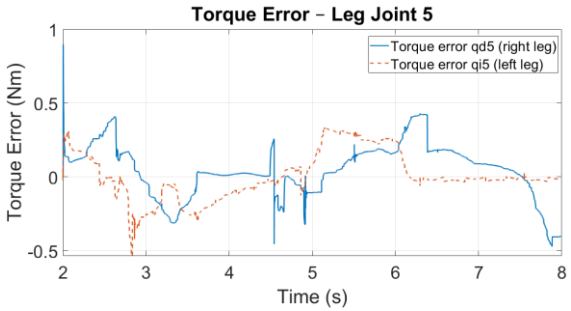

Fig. 10. Torque errors between the analytical dynamic model and the Simscape Multibody simulation for joints q_{d5} and q_{i5}.

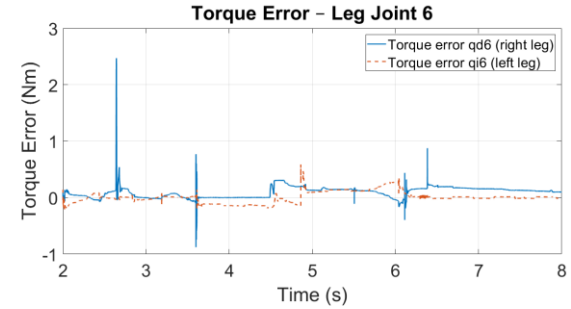

Fig. 11. Torque errors between the analytical dynamic model and the Simscape Multibody simulation for joints q_{d6} and q_{i6}.

The performance was quantified using the Root Mean Square Error (RMSE) for each of the 18 joints (9 right and 9 left), obtaining values such as 0.1906 for the right hip roll (Joint 1) and 0.1330 for its counterpart on the left side, among others. These results demonstrate that the torques predicted by the dynamic model closely match those obtained from the virtual robot, validating the correctness of the modeling and control framework used in this study.

TABLE IV
RMSE OF JOINT TORQUE TRACKING BETWEEN THE DYNAMIC MODEL AND THE SIMSCAPE VIRTUAL ROBOT

Joint	RMSE Right (Nm)	RMSE Lefft (Nm)
1	0.19062	0.13304
2	0.38839	0.31689
3	0.195664	0.18962
4	0.22268	0.21168
5	0.19307	0.16367
6	0.15134	0.10375
7	0.14327	0.12724
8	0.03932	0.04034
9	0.03091	0.03082

The performance evaluation of the three controllers was quantitatively evaluated using the L_2 norm criterion, which is widely used tool by scientific community in robotics to measure the control performance. It is given by:

$$L_2 = \sqrt{\frac{1}{T}\int_0^T \|\tilde{q}(\sigma)\|\, d\sigma} \tag{27}$$

where $T \in \mathbb{R}_+$ represents the simulation time, in this case 6 seconds. The smaller norm means smaller position error and it is the best evaluated scheme performance.

Based on this analysis, the saturated controller exhibited the best overall performance compared to the hyperbolic tangent and PD controllers.

At the global level, the saturated controller achieves the best performance, with a normalized value of approximately 32%, which corresponds to a 68% reduction in tracking error relative to the PD controller. This confirms that bounding the control command through a saturation function significantly enhances closed-loop tracking accuracy, particularly in the presence of fast cyclic motion and large torque demands.

The hyperbolic tangent controller also improves the trajectory tracking compared to PD, with a normalized performance of approximately 90%, representing a 10% reduction in global tracking error. While the tanh function introduces smooth nonlinear damping that moderates aggressive control actions, the improvement is more modest than that achieved by the saturated controller. The tangent hyperbolic controller still allows high-magnitude control signals when required, which explains why it does not provide the same level of error reduction.

In contrast, the PD controller exhibits the highest tracking error (set as 100%), confirming the limitations of a purely linear control law for highly dynamic joint motion. Without any nonlinearity or amplitude bounding, the PD controller generates larger overshoot and control effort peaks during the gait cycle, resulting in inferior tracking precision. Fig. 12 presents the global performance comparison of the three controllers during the dynamic walking test, evaluated using the L_2 norm metric.

By construction, values below 100% indicate better overall tracking than the PD baseline (smaller accumulated error), while values above 100% indicate worse performance.

The per-joint performance results based on the L_2 norm are shown in Fig. 13, where each joint includes three bars corresponding to the PD controller (baseline at 100%), the

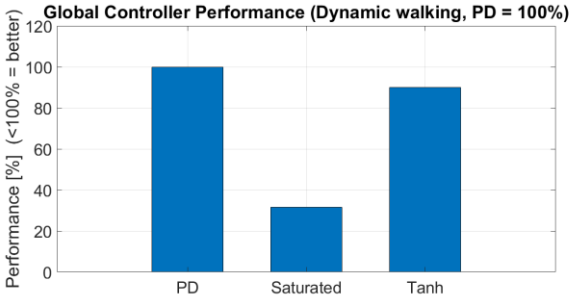

Fig. 12. Global performance indexes of controllers.

Saturated controller, and the Hyperbolic Tangent controller. Since values below 100% indicate lower accumulated tracking error, the results demonstrate that the Saturated controller achieves the best performance in most joints, particularly in the legs. The consistent reductions across key degrees of freedom such as $q_{d1} - q_{d6}$ and $q_{i1} - q_{i6}$ confirm that bounded actuation improves tracking accuracy and stability during dynamic walking.

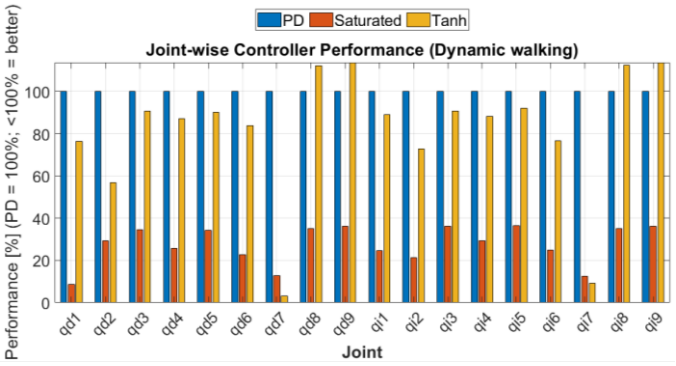

Fig. 13. Performance indexes for each joint.

The Hyperbolic Tangent controller shows mixed performance, outperforming PD in several joints but exceeding the 100% baseline in others, reflecting sensitivity to the dynamics of individual degrees of freedom. The PD controller remains at the reference level and performs competitively in specific joints; however, overall it is surpassed by the Saturated controller across the majority of the articulated chain. These results can be observed in Figure 7.

V. Conclusions and Future Work

The Euler analytical method proved effective for solving the dynamic modeling problem of the Bioloid Premium robot using a recursive approach. This methodology manages the computational complexity more efficiently, resulting in faster symbolic computation without altering the inherent complexity of the dynamic model. The novelty of this work lies not in proposing a new recursive algorithm, but in applying an existing recursive formulation to fully derive, implement, and validate the complete dynamic model of an 18-DoF Bioloid humanoid robot, something not documented in the literature

Consequently, it is not necessary to compute analytical derivatives of the equations, as the properties of the skew-symmetric matrix S facilitate the recursive computation of the resulting matrices or vectors. This enables the implementation of a recursive framework for modeling the robot's kinematics, velocities, jacobians, and lagrangian.

It is worth noting that this methodology can be extended to other bipedal robots. Moreover, the resulting model has significant relevance in computational applications, as the recursive equations reduce the computational cost of modeling humanoid robots. This work thus represents a meaningful contribution and lays the groundwork for the future development of efficient recursive algorithms for dynamic modeling.

References

[1] X. Leng, S. Piao, L. Chang, Z. He and Z. Zhu, "Universal walking control framework of biped robot based on dynamic model and quadratic programming," *Complexity,* vol. 1, 2020, doi: 10.1155/2020/2789039.

[2] D. A. Bravo M and C. F. Rengifo Rodas, "Design of a dynamic simulator for a biped robot," *Modelling and Simulation in Engineering,* vol. 1, 2021, doi: 10.1155/2021/5539123.

[3] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi and B. Morris, Feedback control of dynamic bipedal robot locomotion, CRC press, 2018, doi: 10.1201/9781420053739.

[4] S. A. Chander, A. Mukherjee, V. D. Shivling and A. Singla, "Enhanced Euler–Lagrange Formulation for Analyzing Human Gait With Moving Base Reference," *Journal of Mechanisms and Robotics,* vol. 17, no. 1, 2025, doi: 10.1115/1.4065520.

[5] R. M. Carnier and Y. Fujimoto, "Precise Optimization of Robotic Bipedal Walking Using Hamiltonian Dynamics," *IEEE 29th International Symposium on Industrial Electronics (ISIE),* pp. 567-572, 2020, doi: 10.1109/ISIE45063.2020.9152262.

[6] S. Gupta and A. Kumar, "A brief review of dynamics and control of underactuated biped robots," *Advanced Robotics,* vol. 31, no. 12, pp. 607-623, 2017, doi: 10.1080/01691864.2017.1308270.

[7] M. Rameez and L. A. Khan, "Modeling and dynamic analysis of the biped robot," *15th International Conference on Control, Automation and Systems (ICCAS),* pp. 1149-1153, 2015, doi: 10.1109/ICCAS.2015.7364800.

[8] 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,* no. 1, 2015, doi: 10.1155/2015/963496.

[9] E. Hashemi and A. Khajepour, "Kinematic and three-dimensional dynamic modeling of a biped robot," *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics,* vol. 231, no. 1, pp. 57-73, 2017, doi: 10.1177/1464419316645243.

[10] A. M. Thakkar and V. J. Patel, "Dynamic Simulation of a 12 DoF Biped Robot with Newton-Euler Method using Unit Vector Approach," *Procedia Computer Science,* vol. 230, pp. 935-945, 2023, doi: 10.1016/j.procs.2023.12.133.

[11] F. Gonçalves, T. Ribeiro, A. F. Ribeiro, G. Lopes and P.

Flores, "Dynamic modeling of a human-inspired robot based on a Newton-Euler approach," in *ROMANSY 24 - Robot Design, Dynamics and Control*, CISM International Centre for Mechanical Sciences, vol 606. Springer, Cham., 2022, pp. 79-90, doi: 10.1007/978-3-031-06409-8_8.

[12] H. F. Al-Shuka, B. Corves and W. H. Zhu, "Dynamic modeling of biped robot using Lagrangian and recursive Newton-Euler formulations," *International Journal of Computer Applications,* vol. 101, no. 3, pp. 1-8, 2014, doi: 10.5120/17664-8485.

[13] X. Bajrami, A. Dermaku, A. Shala and R. Likaj, "Kinematics and dynamics modelling of the biped robot," *IFAC Proceedings,* vol. 46, no. 8, pp. 69-73, 2013, doi: 10.3182/20130606-3-XK-4037.00032.

[14] D. Kim, S. J. Jorgensen, J. Lee, J. Ahn, J. Luo and L. Sentis, "Dynamic locomotion for passive-ankle biped robots and humanoids using whole-body locomotion control," *The International Journal of Robotics Research,* vol. 39, no. 8, pp. 936-956, 2020, doi: 10.1177/02783649209180.

[15] S. Kolathaya, "Local stability of PD controlled bipedal walking robots," *Automatica,* vol. 114, 2020, doi: 10.1016/j.automatica.2020.108841.

[16] D. I. H. Putri and C. Machbub, "Gait controllers on humanoid robot using kalman filter and PD controller," *In 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV),* pp. 36-41, 2018, doi: 10.1109/ICARCV.2018.8581061.

[17] J. R. Guadarrama-Olvera, S. Kajita, F. Kanehiro and G. Cheng, "Contact Stability Control of Stepping Over Partial Footholds Using Plantar Tactile Feedback," *International Conference on Intelligent Robots and Systems,* 2024, doi: 10.1109/IROS58592.2024.10802686.

[18] K. Kaneko, F. Kanehiro, M. Morisawa, K. Miura, S. I. Nakaoka and S. Kajita, "Cybernetic human HRP-4C," *International Conference on Humanoid Robots,* pp. 7-14, 2009, doi: 10.1109/ICHR.2009.5379537.

[19] G. Gupta and A. Dutta, "Trajectory generation and step planning of a 12 DoF biped robot on uneven surface," *Robotica,* vol. 36, no. 7, pp. 945-970, 2018, doi: 10.1017/S0263574718000188.

[20] A. Sarkar and A. Dutta, "Optimal trajectory generation and design of an 8-dof compliant biped robot for walk on inclined ground," *Journal of Intelligent & Robotic Systems,* vol. 94, no. 3, pp. 583-602, 2019, doi: 10.1007/s10846-018-0882-9.

[21] D. Hein, M. Hild and R. Berger, "Evolution of biped walking using neural oscillators and physical simulation," *In Robot Soccer World Cup,* pp. 433-440, 2007, doi: 10.1007/978-3-540-68847-1_45.

[22] A. A. Saputra, J. Botzheim, I. A. Sulistijono and N. Kubota, "Biologically inspired control system for 3-D locomotion of a humanoid biped robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 46, no. 7, pp. 898-911, 2015, doi: 10.1109/TSMC.2015.2497250.

[23] Mandava, R. K., & Vundavilli, P. R. "Whole body motion generation of 18-DOF biped robot on flat surface during SSP & DSP". International Journal of Modelling, Identification and Control, 29(3), 266-277, 2018, doi: 10.1504/IJMIC.2018.10012300.

[24] D. I. H. Putri and C. Machbub, "Maintaining Trajectory of CoM for Stable Locomotion of Humanoid Robot Using Kalman Filter and Fuzzy Logic Controller," *In 2019 IEEE International Conference on Signals and Systems (ICSigSys),* pp. 97-102, 2019, doi: 10.1109/ICSIGSYS.2019.8811090.

[25] J. R. Cerritos-Jasso, K. A. Camarillo-Gómez, J. A. Monsiváis-Medina, G. Castillo-Alfaro, G. I. Pérez-Soto and J. A. Pámanes-García, "Kinematic Modeling of a Humanoid Soccer–Player: Applied to BIOLOID Premium Type A Robot.," in *In FIRA RoboWorld Congress*, Berlin, Heidelberg, Springer Berlin Heidelberg, 2013, pp. 49-63, doi: 10.1007/978-3-642-40409-2_5.

[26] J. Tacué, C. Rengifo and D. Bravo, "An experimental energy consumption comparison between trajectories generated by using the cart-table model and an optimization approach for the Bioloid robot," *International Journal of Advanced Robotic Systems,* vol. 17, no. 2, 2020, doi: 10.1177/1729881420917808.

[27] A. B. Krishnan, S. Aswath and G. Udupa, "Real Time Vision Based Soccer Playing Humanoid Robotic Platform," *In Proceedings of the 2014 International Conference on Interdisciplinary Advances in Applied Computing,* pp. 1-8, 2014, doi: 10.1145/2660859.2660966.

[28] Y. L. Hwang, C. H. Chen, S. J. Hwang and N. Xuan Mai, "The dynamic analysis of humanoid robot system," *Applied Mechanics and Materials,* vol. 373, pp. 242-245, 2013, doi: 10.1007/978-3-030-01054-6_60.

[29] J. V. Nunez, A. Briseno, D. A. Rodriguez, J. M. Ibarra and V. M. Rodriguez, "Explicit analytic solution for inverse kinematics of bioloid humanoid robot," *In 2012 Brazilian Robotics Symposium and Latin American Robotics Symposium,* pp. 33-38, 2012, doi: 10.1109/SBR-LARS.2012.62.

[30] F. Gonçalves, T. Ribeiro, A. F. Ribeiro, G. Lopes and P. Flores, "A Recursive Algorithm for the Forward Kinematic Analysis of Robotic Systems Using Euler Angles," *Robotics,* p. 20, 2022, doi: 10.3390/robotics11010015.

[31] B. Siciliano, L. Sciavicco, L. Villani and G. Oriolo, "Robotics: Modelling, Planning and Control", Springer, 2009, pp. 56-109, doi: 10.1007/978-1-84628-642-1.

[32] R. N. Jazar, "Theory of applied robotics", Springer, 2010, pp. 71-149, doi: 10.1007/978-1-4419-1750-8.

Miguel Angel Ortega Palacios is a student of PhD Program of Language & Knowledge Engineering at the Faculty of Computer Science of the Benemérita Universidad Autónoma de Puebla (BUAP). He received his M. Sc degree automation in 2016 from Benmérita Universidad Autónoma de Puebla (BUAP). His areas of interest include automation, robotics, and mechatronics systems.

Amparo Palomino Merino received her Ph.D. (2005) in Automatic Control from the HEUDIASYC laboratory of the University of Technology of Compiegne (UTC), France; a MsC. In Automatic Control from CINVESTAV – IPN, Mexico and an electronic engineer degree obtained at Benemérita Universidad Autónoma de Puebla, Mexico. Dr. Palomino skills

include research, modeling and control of navigation systems and inertial systems, autonomous robotics systems, Nonlinear Control and Experimental Platforms.

Fernando Reyes Cortés received his B. S. and M. Sc degrees in electronics engineering in 1984 and 1989 from Benemérita Universidad Autónoma de Puebla (BUAP) and INAOE, respectively. He received the Ph. D. in electronics at CICESE Center Research, Mexico in 1997. Since 1980, Dr. Reyes has been working at BUAP. Research interests: control of robot manipulators, parameter identification, design and built of robots' prototypes and mechatronics systems.