COBEM 2021 Florianópolis - Brasil
26th ABCM International Congress of Mechanical Engineering
November 22-26, 2021. Florianópolis, SC, Brazil

COB-2021-1375
A FOUR LEG ROBOT STABILITY ANALYSIS THROUGH A VIRTUAL ENVIRONMENT SIMULATION BASED IN DIRECT AND INVERSE KINEMATICS

Frederico Gomes Pires Azzolini
Lucas da Silva Rigobello
Thiago Boaventura
Glauco Augusto de Paula Caurin
Marcelo Becker
University of São Paulo, São Carlos School of Engineering
Av. Trab. São Carlense, 400 - Parque Arnold Schimidt, São Carlos - SP, 13566-590
frederico.azzolini@usp.br
lucasrigobello@usp.br
tboaventura@usp.br
gcaurin@sc.usp.br
becker@sc.usp.br

Abstract. *The replication of nature's characteristics through human technology and systems is denominated as Biomimetics and, when it comes to robotics, is one of the most complex challenges. This is easily illustrated when we look at quadrupedal movement, inspired by four-legged animals. Despite the high complexity involved in developing a system that functions on a four-legged posture, this kind of robot presents some outstanding mobility characteristics, like changing directions by shifting foot support, a higher degree of freedom for the robot's body, an advantage in moving forward through uneven terrain and overcoming obstacles. Based on this idea also aiming to provide an initial contribution for future studies, this work presents a stability analysis of the robot platform displacement, through the development and the simulation of a quadruped robot in a virtual environment. For this purpose, some simplified techniques of locomotion and control were applied. Initially, for environment development, a four-legged model was created in the software Gazebo, along with a communication structure based on ROS. Additionally, the detailed calculations of the forward and inverse kinematics were presented and implemented in C ++. To achieve the robot's movement, a simplified movement pattern, also known as gait, was defined and implemented in the code utilizing an oscillatory function, e.g. sine function, which calculates the height and length of step for each paw. Finally, considering this previous development, the stability analysis for the four-legged robot locomotion is performed by varying the following parameters: height and step length, step frequency, and also the control PD gains. Next, the discussion of the results is based on the displacement, path deviation, height alternation, orientations, actual and set joint positions. Based on that, it was enlightened about the importance of the controller and body dynamics for successful robot walking. Furthermore, under the assumption of the defined robot model and analyzed parameter range, it was possible to observe that the height and length of the step are limited by the body and legs dimension and that they influence the body dynamics as well. In addition, the results also showed that the frequency range between 1 Hz and 2 Hz resulted in the most stable walking movements.*
Keywords: *quadruped robot, simulation, Gazebo, ROS*

1. INTRODUCTION

Science and technological development always observe nature for inspiration, looking for new ways to improve. In this regard, on robot technological development, the legged robot locomotion is inspired mainly by mammal animals, and humans. This robot group is usually classified according to its number of legs: biped, quadruped, hexapod, or octopuses. Initially, there were statements about the high complexity of legged robots due to their mechanism necessities, electrical systems, detection algorithms, and control, that were overcome with new advanced technologies. The locomotion and direction change are standout capabilities for legged robots, performed by altering foot support, independently of body orientation, gives it a higher degree of liberty. Although, the leg movement is substantially more complex than wheels or tracks. This problem is solved inspired, again, in nature. The quadruped animals have many walking gaits, acquired over biological and behavioral evolution, perfectly adapted and optimized (De Santos *et al.*, 2007).

The successful improvements achieved in walking abilities for quadruped robots have benefited from simulations

in their development. Plenty of work has been published on simulators and simulation tools, ranging from comparisons between different systems to detailed testing of specific tools (Saglam and Papelis, 2020; Harris and Conrad, 2011; Torres-Torriti *et al.*, 2014; Noori *et al.*, 2017; Pitonakova *et al.*, 2018; Echeverria *et al.*, 2011; Rohmer *et al.*, 2013; Pinciroli *et al.*, 2012; Todorov *et al.*, 2012; Gerkey *et al.*, 2003; Erez *et al.*, 2015).

Moreover, different approaches to modeling and control methods are constantly being explored throughout academia. Renawi *et al.* (2017) presents the kinematic and dynamic models of the Kobuki robot while validating a trajectory tracking controller. Takaya *et al.* (2016) briefly explains how to set up a simulation environment on Gazebo. Sadeghian *et al.* (2017) presents the implementation of a vision controller for a spherical robot, along with mathematical modeling. Vyavahare *et al.* (2019) thoroughly explains how to model a complex four-legged robot in order to enable it for Gazebo-ROS simulation and control. Yeoh *et al.* (2020) presented a detailed description to construct a new legged robot package for simulation in Gazebo, using ROS.

Having in mind all challenges and complexities involved in developing a quadruped robot and its simulations, the main goal of this paper is to contribute in an initial stage, bringing a description of the mathematical model and simulation in a virtual environment for a four-legged robot, through basic gait defined movements. Therefore, to accomplish the desired movements, it is developed from scratch the direct and inverse kinematic modeling, obtained through the Denavit-Hartenberg principles, rather than to make usage of some already developed open-source packages such as the *RobCoGen* (Frigerio *et al.*, 2016) and *Pinocchio* (Carpentier *et al.*, 2019). This formulation is used to calculate each incremental step based on a foot gait pattern. Based on that, the simulation is constructed to use the software Gazebo combined with the ROS library and packages. Finally, a stability analysis is presented and discussed, exploring some different initial dispositions of the robot legs and walking pace conditions, such as pace frequency, pace length, and height.

2. DEVELOPMENT METHODOLOGY

Initially, this paper was inspired by Yeoh *et al.* (2020), which described the development of a ROS package from scratch. The robot model was adopted from Sen *et al.* (2017), which also presented inverse kinematics to provide the orientation and positions of the body and angular joints. Therefore, a step forward is given by the virtual simulation in this paper. For this purpose, two development paths are defined: kinetic modeling and the structuring of the virtual model.

The first path deals with the development of the proposed robot kinematic modeling, through transformation matrices, Denavit-Hartenberg method, and direct kinematics, to determine the positions of each of the paws, once it is given each joint angle and the central position of the robot's body. Then, to perform the inverse kinematics, the Jacobian matrix is calculated, in such a way that it becomes possible to perform the incremental angular calculation, required for each joint rotation, providing a direction of displacement of the legs. Thus, by performing incremental calculation steps for the joint displacement, it is possible to move the paw from a current point to a desired new one. In parallel, the second path develops the virtual model and packages with the mathematical calculation, employing the Gazebo Software and ROS. Therefore, the walking movements are executed based on the planned feet movement that through the defined kinetic model informs the joints positions to the dynamic model represented by the virtual environment simulation. The code developed in C++ is available on GitHub (Azzolini and Rigobello, 2021), for a more detailed description and testing.

2.1 Virtual Environment Setup

To build up the virtual environment using ROS and Gazebo, three complementary steps are necessary to successfully set up the simulation: correctly model its sensors and actuators; insert the model into a physically accurate environment; access and control those sensors and actuators. Figure 1 depicts an overview of features integration between the robot model (Robot), control system (Node), and the access to the sensors and actuators (Configuration).

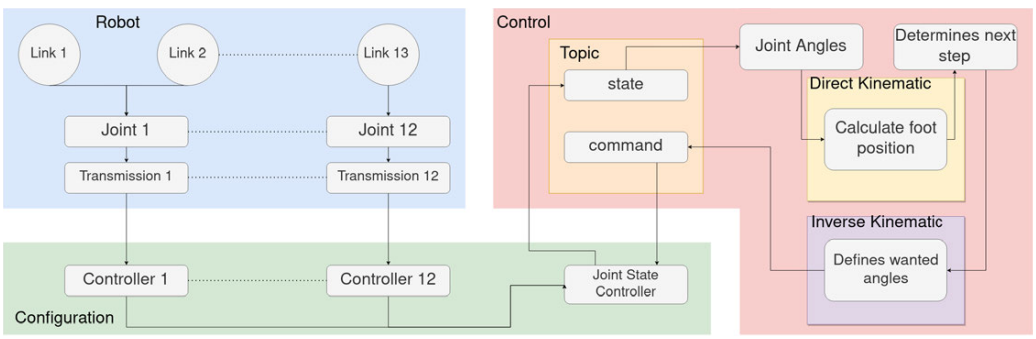


Figure 1: Simulation environment overall view

2.2 Robot Modeling

Kinematic modeling turns out to be essential to enable stability analysis and trajectory planning. Through the direct kinematics, the robot body position is obtained according to the provided values for the joints variables. Already, the inverse kinematics allows the determination of the necessary parameters for each joint upon setting the desired robot position. Although, previously, it must be defined the robot model to be referenced. Thereby, the four-legged robotic model applied in this work is shown in Figure 2, and its physical dimensions are defined in Table 1 (Sen *et al.*, 2017), and the inertia calculation is computed based on the primitive's geometry (cylinder, box, sphere).

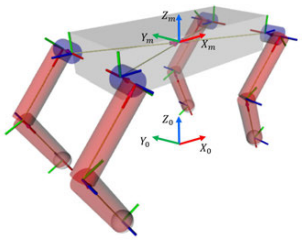


Figure 2: Robot model deployed

Table 1: Robot's physical dimensions and properties

	Length	Width	Radius	Mass
Body	$L = 1\ m$	$W = 0.4\ m$	-	15 kg
Shoulder link	$L_1 = 0.1\ m$	-	-	0.5 kg
Upper leg	$L_1 = 0.4\ m$	-	-	3 kg
Lower leg	$L_1 = 0.4\ m$	-	-	3 kg
Feet	-	-	$0.05m$	0.75 kg

For the presented model, the body coordinate system is given through the reference inertial coordinate system X_0, Y_0, Z_0 (representing the origin [0,0,0] position in the simulation), and the center body system X_m, Y_m, Z_m. Those coordinate systems can be correlated to the determination of the rotation body parameters: steering angle (Φ), inclination angle (Ψ), and rotation angle (ω). The leg's coordinate system is given in Table 2, followed by the parameters defined in Table 3.

Table 2: Coordinate systems for each model

(X_1, Y_1, Z_1)	Shoulder rotation joint
(X_2, Y_2, Z_2)	Shoulder hinge joint
(X_3, Y_3, Z_3)	Elbow joint
(X_4, Y_4, Z_4)	Paw system

Table 3: Parameters

θ_1	Shoulder hinge angle
θ_2	Shoulder rotation angle
θ_3	Elbow hinge angle

2.3 Direct Kinematics

Initially, through the homogeneous transformations matrices, the position and orientation of the center of the robot body must be determined according to the position and rotation angles (Φ, Ψ and ω) obtained by the rotation matrices R_x, R_y and R_z (Eq. 1). Combining these rotation matrices with the center of mass, results in the body's transformation matrix (T_m), as represented by Eq. 2. Next, the transformation matrices given in Table 4 represent the coordinate system transfer to the shoulder of each single leg, specific to each lateral swing joint system, subscribed by: rb (right and back, rf (right and front), lb (left and back) and lf (left and front).

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & cos(\omega) & -sen(\omega) & 0 \\ 0 & sen(\omega) & cos(\omega) & 0 \\ 0 & 0 & 0 & 1] \end{bmatrix} R_y = \begin{bmatrix} cos(\Phi) & 0 & sen(\Phi) & 0 \\ 0 & 1 & 0 & 0 \\ -sen(\Phi) & 0 & cos(\Phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} R_z = \begin{bmatrix} cos(\Psi) & -sen(\Psi) & 0 & 0 \\ sen(\Psi) & cos(\Psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$T_m = R_x R_y R_z \times \begin{bmatrix} 1 & 0 & 0 & X_M \\ 0 & 1 & 0 & Y_M \\ 0 & 0 & 1 & Z_M \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Having defined the Denavit-Hartenberg parameters in Table 5, they are applied to obtain the homogeneous transformation matrices (${}^{j-1}A_j$), given by Eq. 3 following the Denavit-Hartenberg definition between each coordinate system. In other words, this transformation matrix is calculated from a link to the next (Corke, 2011). The sequential multiplication of homogeneous transformation matrices results in matrix T ($T = {}^0A_1\ {}^1A_2\ {}^2A_3$) which represents the change between coordinate systems. In such a way that, when the coordinate system matrix of each shoulder, as presented in Table 4, is multiplied by each respected leg T transformation matrix, the first 3 lines of the 4^{th} column provide the point of interest vector, called "p". Also the orientation of the respective z-rotation axis, in terms of the reference inertial coordinate system, is given by the vector "q", composed by the first 3 lines of the 3^{rd} column.

Table 4: Transformation matrices to lateral swing joint system

$T_{rb} = T_m \times \begin{bmatrix} 0 & 0 & 1 & -L/2 \\ 1 & 0 & 0 & -W/2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$T_{rf} = T_m \times \begin{bmatrix} 0 & 0 & 1 & L/2 \\ 1 & 0 & 0 & -W/2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$T_{lf} = T_m \times \begin{bmatrix} 0 & 0 & -1 & L/2 \\ -1 & 0 & 0 & W/2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$T_{lb} = T_m \times \begin{bmatrix} 0 & 0 & -1 & -L/2 \\ -1 & 0 & 0 & W/2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Table 5: Denavit-Hartenberg parameters

Link	a_{i-1}	α_{i-1}	d_{i-1}	β_{i-1}
0-1'	$-L_1$	0	0	θ_1
1'-1	0	$\pi/2$	0	$\pi/2$
1-2	L_2	0	0	θ_2
2-3	L_3	0	0	θ_3

$$ {}^{i-1}A_i = \begin{bmatrix} cos(\theta_i) & -sen(\theta_i)cos(\alpha_{i-1}) & sen(\theta_i)sen(\alpha_{i-1}) & a_{i-1}cos(\theta_i) \\ sen(\theta_i) & cos(\theta_i)cos(\alpha_{i-1}) & -cos(\theta_i)sen(\alpha_{i-1}) & a_{i-1}sen(\theta_i) \\ 0 & sen(\alpha_{i-1}) & cos(\alpha_{i-1}) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3) $$

2.4 Inverse Kinematics

The inverse kinematics calculation is a differential representation relating the joints with the desired movement to the point of interest. This relationship is governed by the Jacobian matrix (J), its calculation depending on the configuration preferred by the handler (Sciavicco and Siciliano, 2012). The linear and angular speeds given by $\dot{p}$ and ω, respectively, compose the joints velocity $\dot{q}$, represented by $\dot{q} = [\dot{p}\, \omega]$ $(Eq.\ 4)$, defining a direct relation with the end-effector velocity v, given by $v = J(q)\dot{q}$ $(Eq.\ 5)$. Therefore, for this formulation, the Jacobian matrix can be interpreted in two parts: the first responsible for the linear movements $J_{P(3\times n)}$ and the second accountable for the rotations $J_{O(3\times n)}$.

The strategy applied is to find the geometric Jacobian matrix, obtained from the transformation matrices (T), previous defined in the direct kinematics. For each robot joint "i", it must have a corresponding Jacobian matrix row, which can be obtained by the following operations: $J_P = z_{i-1} \times (P - p_{i-1})$ $(Eq.\ 6)$; and $J_O = z_{i-1}$ $(Eq.\ 7)$. Where P is the point of interest position in the referential coordinate system; and p_{i-1} is the rotation position of each joint "i", obtained from the vector p of the matrix T to the respective joint. This position representation is given by $p_{i-1} = A_1^0(q_1) \cdots A_{i-1}^{i-2}(q_{i-1})\, p_0$ $(Eq.\ 8)$, defining $p_0 = [0\ 0\ 0\ 1]^T$.

The rotation axis of the joint (z_{i-1}), corresponding to the Denavit-Hartenberg representation, is obtained from the vector q of the matrix T to the respective joint. This axis representation is given by $z_{i-1} = R_1^0(q_1) \cdots R_{i-1}^{i-2}(q_{i-1})\, z_0$ $(Eq.\ 9)$. When having the origin as the starting point, by standard, it is defined $z_0 = [0\ 0\ 1]^T$.

Thus, based on the previous equations relating the movement of the manipulator's endpoint to the movements of the joints, the inverse kinematics can be interpreted as an inverse problem, where the common solution path is getting the inverted Jacobian. Turns out, it can be more difficult than it seems. Therefore, usually, the use of the Jacobian pseudo-inverse ($J^+ = J^T(JJ^T)^{-1}$) is required, then the problem is stated by $\dot{q} = J^+ v$ $(Eq.\ 10)$.

2.5 Description of Simulation Proposal

The simulation objective is to execute a walking movement with the virtual model, to discuss the sensitivity of some defined parameters. In beforehand, the considered parameters are the gait frequency (f_{gait}); the step length and height (L_{step}, H_{step}); and proportional and derivative control parameters. Based on that, 8 cases are defined and will be detailed in the first section of the discussion chapter.

First, the walking movement is produced by moving the robot's leg joint. Thereby, to accomplish and provide the appropriate joint actuation, a PD control is used. As for the walking movement, a standard movement, similar to those identified in small quadrupeds, is defined. According to literature, the movement pattern is chosen based on the desired speed and biological characteristics (Hildebrand, 1965; Abourachid, 2003; DATT and FLETCHER, 2012). Figure 3 depicts the proposed simulated gait pattern.

(a) (b)

Figure 3: Applied gait (a), ground contact representation in black; (b) gait sequence

In Fig. 3, marks in blank represent the elevation, and the forward movement of feet and the marks in black are the push back foot movements. To promote displacement of the robot, the movement of each foot is defined by combining positions on "x" and "z" directions. As illustrated in Fig. 4 the feet position target is governed by a harmonic function in the x-direction, defined through the step length (L_{step}), interchanging with holding positions; also, the z-direction target position is governed by harmonic function, defined through the step length (H_{step}), combined with instantaneous shifting to holding position in coordinate zero.

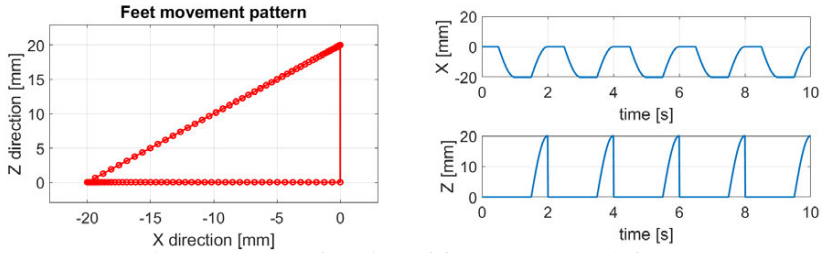


Figure 4: Graph function of feet movement defined

The defined gait and feet movement are related in such a way that the step frequency (f_{step}), for each foot, is dependent on gait frequency (f_{gait}). Additionally, there is a correlation between the step parameters and the step frequency, that defines an estimation for the robot's speed, given by $V_c = L_{step} \times f_{step}$ $(Eq.\ 11)$, and consequently, travel distance is also defined by $\Delta S_c = V_c \times time$ $(Eq.\ 12)$.

3. EXPERIMENTS, RESULTS AND DISCUSSIONS

The quadrupedal robot walking simulation results and discussions are based on the displacement, path deviation, height alternation, orientations, actual and set joint positions. Although, there are some setup features to be mentioned and explained. The parameter range chosen was based on the common walking frequency and step height and length proportional to the body and leg dimensions. In addition, the control gains were tested beforehand, by trial and error, and through observation, the dynamic parameters were adjusted to provide stability for the robot and an adequate walking capability, in which the PD gains were higher on the back leg joints than the front ones, with more detailed information given in this section. For the simulation, the feet trajectories were updated based on a frequency of 1000 Hz, as well as for the joints PD controllers. The actuator saturation (joint torque and velocity) is also essential in legged locomotion, in such a way that it was defined a saturation of 1000 Nm and 10 m/s for all simulations.

3.1 Case studies: general characteristics and results

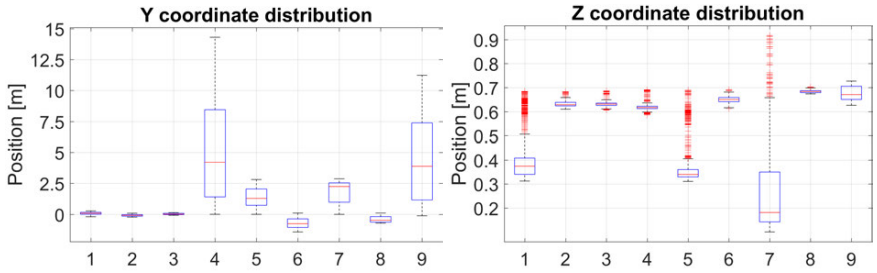
Nine case studies were proposed, with the parameters varying accordingly to Table. 6. In the same table, it is also presented some results: the simulation time, the simulated distance traveled, the expected calculated distance, similarly, the speeds, and the related variation in percent. Farther, Fig. 5 brings, for each case, the distribution of the path deviation represented by the robot position on "y" coordinate; the height alternation, represented by the z coordinate position of the robot's center of mass; and the body's orientation, informing the body rotation in all three axes.

3.2 Pace frequency analysis

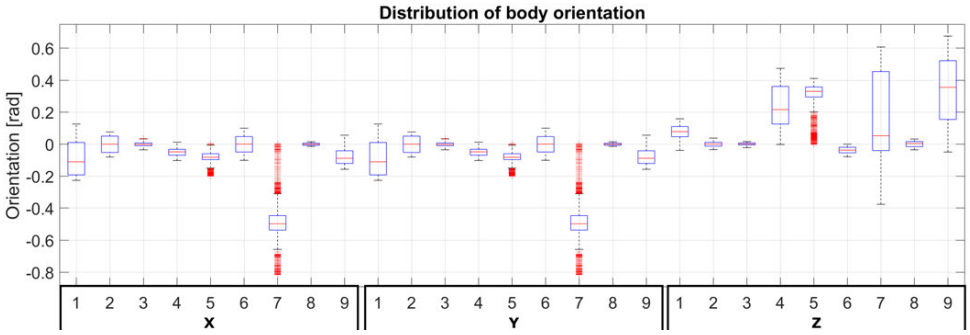
The pace frequency analysis is based on the five first cases. The increasing of this parameter (f_{step}), going from 0.5 Hz up to 3.5 Hz, while the length and height of the step are fixed, allows to define its best practical value. Case 2 and 3, respectively, using f_{step} of 1 Hz and 2 Hz, are highlighted as the most successful cases, since they presented low variation ($5.52\ \%$) on traveled distance and speed, compared with what was expected, although in this respect, Case 2 presented the lower variation. Figure 6(b) illustrates the robot executing a constant and expected motion of Case 2. This analysis is also asserted by other results in Fig. 5, which shows a path deviation close to zero, and a stable height position, and

Table 6: General cases characteristics and results

	Parameters				Simulated results					
	f_{gait}	f_{step}	L_{step}	H_{step}	$time$	ΔS_s	ΔS_c	V_s	V_c	Δ
1	0.25 Hz	0.5 Hz	0.2 m	0.2 m	58.57	2.93 m	5.86 m	0.05 m/s	0.1 m/s	-50.04 %
2	0.5 Hz	1 Hz	0.2 m	0.2 m	57.48	12.13 m	11.50 m	0.21 m/s	0.2 m/s	5.52 %
3	1 Hz	2 Hz	0.2 m	0.2 m	58.69	25.10 m	23.48 m	0.43 m/s	0.4 m/s	6.895 %
4	1.25 Hz	2.5 Hz	0.2 m	0.2 m	59.11	24.69 m	29.55 m	0.42 m/s	0.5 m/s	-16.70 %
5	1.75 Hz	3.5 Hz	0.2 m	0.2 m	57.07	7.83 m	39.95 m	0.14 m/s	0.7 m/s	-80.40 %
6	0.5 Hz	1 Hz	0.4 m	0.4 m	57.52	19.55 m	23.01 m	0.34 m/s	0.4 m/s	-15.05 %
7	0.5 Hz	1 Hz	0.6 m	0.6 m	58.23	-1.06 m	34.94 m	-0.02 m/s	0.6 m/s	-103.05 %
8	1.75 Hz	3.5 Hz	0.2 m	0.2 m	58.41	45.95 m	40.89 m	0.79 m/s	0.7 m/s	12.3743 %
9	0.5 Hz	1 Hz	0.2 m	0.2 m	58.25	9.42 m	11.65 m	0.16 m/s	0.2 m/s	-19.11 %



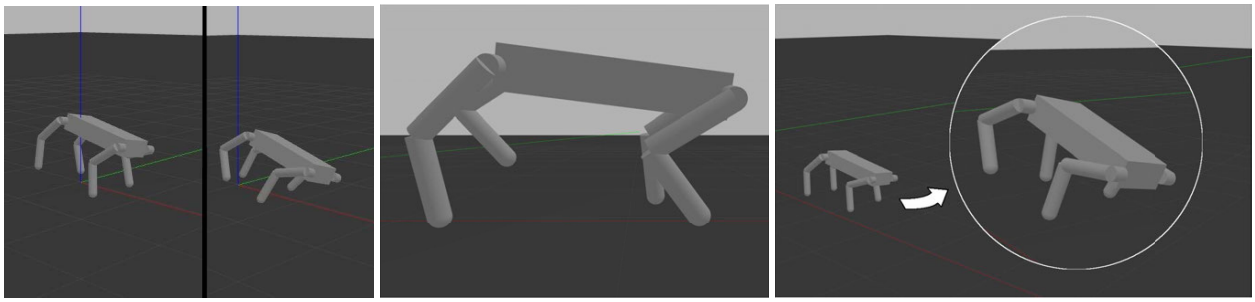
(a) (b)



(c)

Figure 5: (a) Path deviation distribution; (b) height alternation; and (c) body orientation

orientation in all directions; although in this matter, the best results were from Case 3.



(a) (b) (c)

Figure 6: (a) Case 1; (b) Case 2; and (c) Case 5

Regarding Case 1, the high variation (-50.04 %) from the expected values resulted in the robot's failure to walk, as depicted in Fig. 6(a), falling at the first steps. Also perceived through high variation and lower height position, given by information on Fig. 5. The lower stepping frequency (0.5 Hz) utilized is reasonably deemed as a possible reason for failure, since by slowing the movement of the feet it could not properly support the robot in the swinging motion imposed by the push of the rear legs. The robot also fell in Case 5, $f_{step} = 3.5$ Hz, failing to walk as illustrated in Fig. 6(c). In this

case, however, it can be affirmed that the control was not capable to actuate on the joints as desired, to promote adequate feet movement. This aspect is discussed in the following subsection, regarding control influence.

Next, a variation of -16.70 % is presented by Case 4, showing successful walking performance, although quite unstable. The information in Fig. 5 demonstrates a considerable path deviation while walking. In other words: the robot did not walk in a straight line. All other factors, however, can be considered satisfactorily stable.

Therefore, the step frequency of 1 Hz was selected as the best to compose the following cases of simulation, since it showed the most successful walking behavior, providing adequate stability and almost no path deviation.

3.3 Pace length and height analysis

The analysis of step length and height is based on Cases 6 and 7, assuming as reference the success of Case 2. In this regard, both parameters are increased simultaneously, going from 0.2 m up to 0.6 m, resulting in a poorly walking performance, keeping fixed the gait frequency at 1 Hz, as previously mentioned.

Similar to Case 4, the variation of -15.05 % presented by Case 6 shows successful, but unstable, walking performance. Based on the information presented in Fig. 5, this case also presented considerable path deviation while walking.

In Case 7 there was a high variation of -103.05 % between the simulated and expected displacement, causing the robot to fail to walk, falling in its first steps, which is also confirmed noticing the high variation and lower height position, as shown in Fig. 5. The reason for this failure is revealed through an inspection on the set joint positions, given in Fig. 7, in which the informed chaotic pattern is resulted from the inverse kinematic, indicating that it could not find a real solution to the desired foot position, once this point would be out of the workspace.

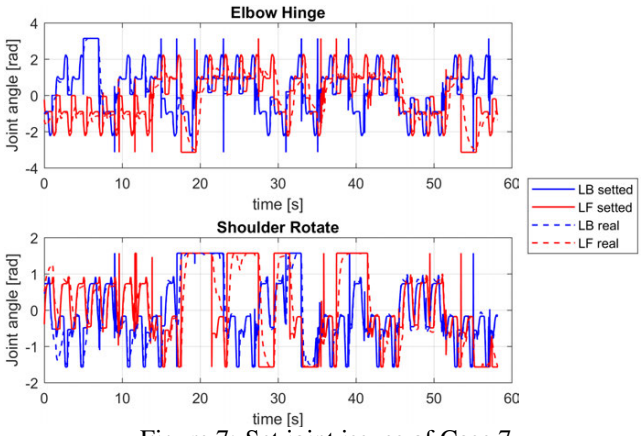


Figure 7: Set joint issues of Case 7

3.4 Control influence

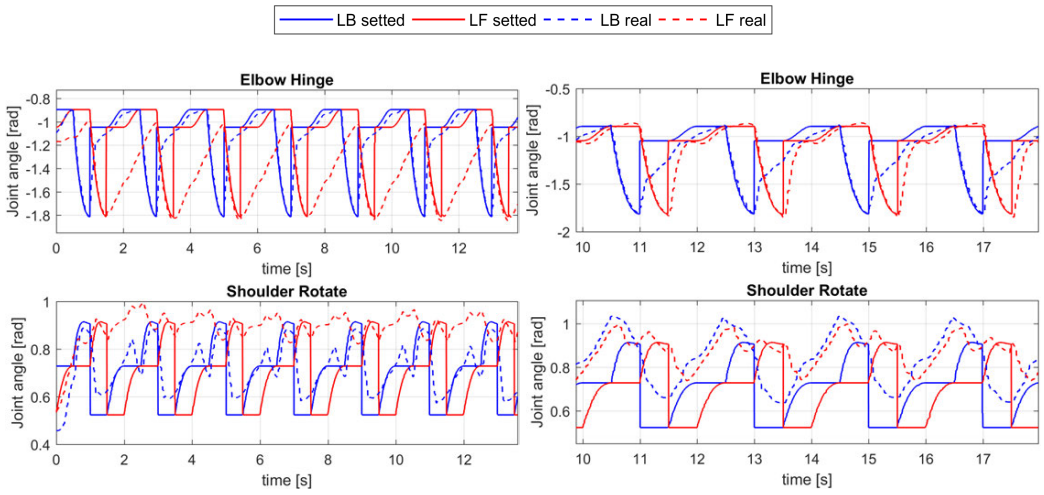
The control is a potential cause of some of the failures presented in the previous analyses, in such a way that increasing the frequency and/or the parameters of the steps makes it difficult to actuate the joints in the desired way due to requiring greater effort, and consequently, greater controller action. Both Cases 8 and 9 compose this analysis, representing the controller's importance on movement behavior.

Previously, for all other cases here addressed were applied a proportional (P) and derivative (D) controller, with the following configuration: frontal elbows rotation are $P = 100$; $D = 50$; and on back with $P = 500$ and $D = 50$; the frontal shoulder rotation are $P = 100$; $D = 32$; and on back with $P = 200$ and $D = 30$; and finally, the shoulder hinge $P = 500$ and $D = 100$, with no distinction.

Case 8 starts from the same configuration utilized in Case 5 (critical failure), but the intention was to raise the proportional gain of the front actuators until they were higher than the values of the back actuators. Therefore, it was adapted to $P = 300$ on the shoulders rotation and $P = 700$ on the elbows rotation. This adaptation made it possible for the legs to withstand and keep up with the faster movement, which was impossible before transforming a negative 80.40 % variation to a positive 12.3743 % (Table 6). The improved walking skills are presented in Fig. 10(a). The back legs were maintained at $P = 200$ (shoulders) and $P = 500$ (elbows);

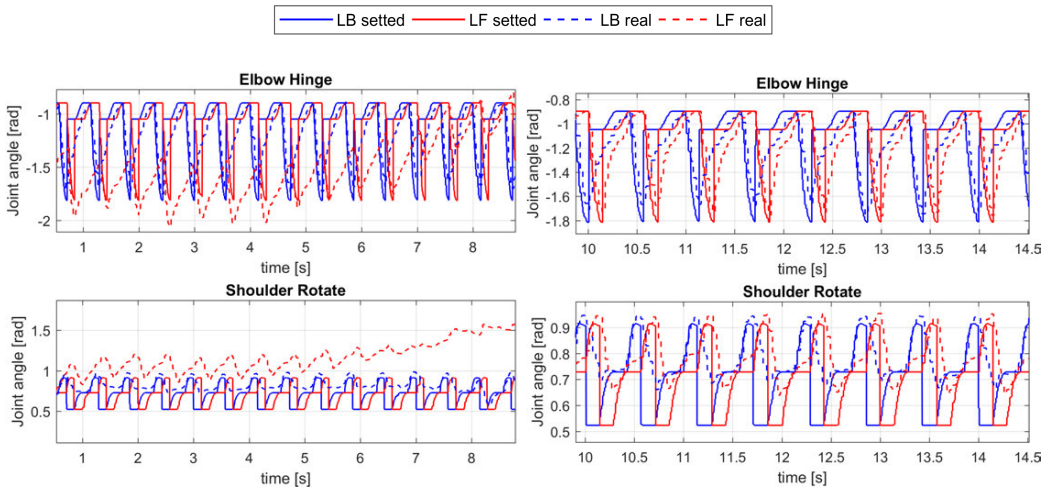
Case 9, on the other hand, is based on the considerably stable Case 2. Now, even though the objective remained to set the front legs with the highest proportional gains, the strategy was to also lower the back leg values. Therefore, the new configuration became $P = 200$ (shoulders) and $P = 500$ (elbows) for the front legs and $P = 100$ (shoulders) and $P = 250$ (elbows) for the back legs. With these gains, the robot was still able to walk, however, the back legs were not able to fully bear the weight of the body while walking, illustrated in Fig. 10(b), elevating the absolute variation, in compression to Case 2 (Table 6).

The effects of control values manipulation can be more clearly seen on Figures 8 and 9, when analyzing how successfully each joint achieved its wanted position. Observing the control influences under the stability and ability to walk, an interesting aspect of body dynamics is revealed, playing an important role in conducting an appropriate forward movement. This aspect will be further discussed in the next sub-section of this chapter.



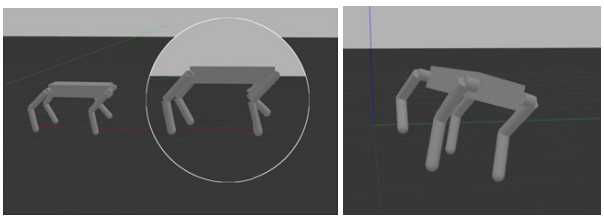
(b) Case 2 (c) Case 9

Figure 8: Results of control gains changes (a) Case 2 and (b) Case 9



(b) Case 5 (c) Case 8

Figure 9: Improvements on walking and stability increasing the control: (a) Case 5 and (b) Case 8



(a) (b)

Figure 10: (a) Case 8; and (b) Case 9

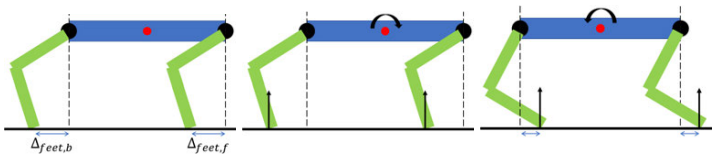
3.5 Body dynamics

According to what has been observed in the previous discussion, body dynamics plays an important role in the walking skill of the robot. When analyzing the body dynamics through the walking movement, some key aspects can be high-

lighted. Regarding feet placement, there is a clear relation between robot stability and its distance from the shoulder line, as illustrated in Figure 11(a). In cases where the foot crosses that line and is fixed below the robot's body, it is generated a momentum as depicted in Figure 11(c).

This momentum, however, is not always bad for the movement in such cases as presented by Figure 11(b). In the gait chosen for these tests, for example, the front feet are expected to be constantly put behind that line. This is utilized due to generating a sort of pendulum effect that propels the robot forward, facilitating continuous movement. This is very easily seen when observing Case 2, as presented by Figure 6(b), especially if taking into consideration the robot's body forward inclination. Although, it is important to certify that the front legs have enough speed and strength in its actuators to withstand the forward push and maintain pace, if not, the robot might end up falling or presented instability, similar to what happened with Case 5, depicted in Figure 6(c).

On the other hand, if the back feet are positioned across that line and under the body as schematized by 11(c), the momentum is formed in the opposite direction of the movement, generating an elevated amount of force applied on the back legs. As a result, in the best cases the robot will present a higher difficulty walking, and in the worst cases, fall upon its back legs. This dynamic is equivalent to what happened in Cases 6 and 7.



(a) (b) (c)

Figure 11: Investigation on body dynamics (a) feet position; (b) resultant force and momentum; and (c) dynamics of foot position inversion

4. CONCLUSION

This work could satisfactorily demonstrate the initial steps for developing a robot model from scratch. Even though a highly limited and simplified simulation, it was possible for the model to move without falling on the ground. In such a way that the simulation results showed that Cases 2 and 3 provided the most stable movements, indicating a better range for the step frequency between 1 Hz and 2 Hz.

The observation regarding the controller's influence clarified its importance for successful walking, demonstrated by the improvements achieved on a critical case when increasing the proportional control gains. In contrast, other demonstrations were given through the reduction of controller gains on a successful case, degrading its performance. Furthermore, it was possible to observe that the height and length of the step are limited mainly by the dimensions of the body and legs, as such as the desired balance dynamics of the body.

Based on those essential clarifications of the most important features of a quadrupedal robot walking mainly on the controller and body dynamics, it is suggested to get a step further on future works, considering dynamic body control, employing forward/inverse dynamics and centroidal dynamics, and providing some trajectory optimization. In that case, some open-source packages may help, such as the *Towr* and *Pinnochio* (Winkler *et al.*, 2018; Carpentier *et al.*, 2019).

Additionally, an interesting suggestion considered for future work is to investigate the performance that could be obtained by using other simulation software. For contact simulation, as it occurs during contact of the feet with the ground in a quadrupedal robot, Gazebo is not the most advanced. Even considering that no simulation will be as accurate as in the real world, the suggestion, in this respect, would be to use MuJoCo (Todorov *et al.*, 2012), that according to Erez *et al.* (2015), stood out as the most accurate and fastest in robotics constrained systems, performing stable grasping at a larger time step.

5. REFERENCES

Abourachid, A., 2003. "A new way of analysing symmetrical and asymmetrical gaits in quadrupeds". *Comptes Rendus Biologies*, Vol. 326, No. 7, pp. 625 – 630. ISSN 1631-0691. doi:https://doi.org/10.1016/S1631-0691(03)00170-7. URL `http://www.sciencedirect.com/science/article/pii/S1631069103001707`.

Azzolini, F.G.P. and Rigobello, L.S., 2021. "Github - controller". URL `https://github.com/fredgpa/quadrupedal_tests/tree/artigo-congresso`.

Carpentier, J., Saurel, G., Buondonno, G., Mirabel, J., Lamiraux, F., Stasse, O. and Mansard, N., 2019. "The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives". In *IEEE International Symposium on System Integrations (SII)*.

Corke, P., 2011. *Robotics, Vision and Control - Fundamental Algorithms in MATLAB®*, Vol. 73 of *Springer Tracts in Advanced Robotics*. Springer. ISBN 978-3-642-20143-1.

DATT, V.L. and FLETCHER, T.F., 2012. *Gaits Web Site*. Supported by University of Minnesota College of Veterinary Medicine. URL `http://vanat.cvm.umn.edu/gaits/index.html`.

De Santos, P.G., Garcia, E. and Estremera, J., 2007. *Quadrupedal locomotion: an introduction to the control of four-legged robots*. Springer Science & Business Media.

Echeverria, G., Lassabe, N., Degroote, A. and Lemaignan, S., 2011. "Modular open robots simulation engine: Morse". In *2011 IEEE International Conference on Robotics and Automation*. pp. 46–51. doi:10.1109/ICRA.2011.5980252.

Erez, T., Tassa, Y. and Todorov, E., 2015. "Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx". In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. pp. 4397–4404. doi:10.1109/ICRA.2015.7139807.

Frigerio, M., Buchli, J., Caldwell, D.G. and Semini, C., 2016. "RobCoGen: a code generator for efficient kinematics and dynamics of articulated robots, based on Domain Specific Languages". Vol. 7, No. 1, pp. 36–54.

Gerkey, B., Vaughan, R. and Howard, A., 2003. "The player/stage project: Tools for multi-robot and distributed sensor systems". *Proceedings of the International Conference on Advanced Robotics*.

Harris, A. and Conrad, J.M., 2011. "Survey of popular robotics simulators, frameworks, and toolkits". In *2011 Proceedings of IEEE Southeastcon*. pp. 243–249. doi:10.1109/SECON.2011.5752942.

Hildebrand, M., 1965. "Symmetrical gaits of horses". *Science*, Vol. 150, No. 3697, pp. 701–708. ISSN 00368075, 10959203. URL `http://www.jstor.org/stable/1717082`.

Noori, F.M., Portugal, D., Rocha, R.P. and Couceiro, M.S., 2017. "On 3d simulators for multi-robot systems in ros: Morse or gazebo?" In *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*. pp. 19–24. doi:10.1109/SSRR.2017.8088134.

Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Caro, G.D., Ducatelle, F., Birattari, M., Gambardella, L. and Dorigo, M., 2012. "Argos: a modular, parallel, multi-engine simulator for multi-robot systems". *Swarm Intelligence*, Vol. 6, pp. 271–295.

Pitonakova, L., Giuliani, M., Pipe, A. and Winfield, A., 2018. "Feature and performance comparison of the v-rep, gazebo and argos robot simulators". ISBN 978-3-319-96727-1. doi:10.1007/978-3-319-96728-8_30.

Renawi, A., Jaradat, M.A. and Abdel-Hafez, M., 2017. "Ros validation for non-holonomic differential robot modeling and control: Case study: Kobuki robot trajectory tracking controller". In *2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*. pp. 1–5. doi:10.1109/ICMSAO.2017.7934880.

Rohmer, E., Singh, S.P.N. and Freese, M., 2013. "V-rep: A versatile and scalable robot simulation framework". In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 1321–1326. doi:10.1109/IROS.2013.6696520.

Sadeghian, R., Zarei, M., Shahin, S. and Masouleh, M.T., 2017. "Vision based control and simulation of a spherical rolling robot based on ros and gazebo". In *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*. pp. 0304–0309. doi:10.1109/KBEI.2017.8324991.

Saglam, A. and Papelis, Y., 2020. "Scalability of sensor simulation in ros-gazebo platform with and without using gpu". In *2020 Spring Simulation Conference (SpringSim)*. pp. 1–11. doi:10.22360/SpringSim.2020.CIAAS.001.

Sciavicco, L. and Siciliano, B., 2012. *Modelling and control of robot manipulators*. Springer Science & Business Media.

Sen, M.A., Bakircioglu, V. and Kalyoncu, M., 2017. "Inverse kinematic analysis of a quadruped robot". *International journal of scientific & technology research*, Vol. 6, No. 9, pp. 285–289.

Takaya, K., Asai, T., Kroumov, V. and Smarandache, F., 2016. "Simulation environment for mobile robots testing using ros and gazebo". In *2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*. pp. 96–101. doi:10.1109/ICSTCC.2016.7790647.

Todorov, E., Erez, T. and Tassa, Y., 2012. "Mujoco: A physics engine for model-based control". In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 5026–5033. doi:10.1109/IROS.2012.6386109.

Torres-Torriti, M., Arredondo, T. and Castillo-Pizarro, P., 2014. "Survey and comparative study of free simulation software for mobile robots". *Robotica*, Vol. -1, pp. 1–32. doi:10.1017/S0263574714001866.

Vyavahare, P., Jayaprakash, S. and Bharatia, K., 2019. "Construction of urdf model based on open source robot dog using gazebo and ros". In *2019 Advances in Science and Engineering Technology International Conferences (ASET)*. pp. 1–5. doi:10.1109/ICASET.2019.8714265.

Winkler, A.W., Bellicoso, D.C., Hutter, M. and Buchli, J., 2018. "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization". *IEEE Robotics and Automation Letters (RA-L)*, Vol. 3, pp. 1560–1567. doi:10.1109/LRA.2018.2798285.

Yeoh, C.E., Kim, D.B., Won, Y.B., Lee, S.R. and Yi, H., 2020. "Constructing ros package for legged robot in gazebo simulation from scratch". In *2020 20th International Conference on Control, Automation and Systems (ICCAS)*. pp. 94–99. doi:10.23919/ICCAS50221.2020.9268358.

6. RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.