

Conference Paper Title

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

Abstract—With the growth of robotics, mobile robots are increasingly becoming a common aspect in key sectors of the economy. When compared to wheeled and tracked robots, legged robots present greater mobility and maneuverability, but less locomotion stability, which in turn demands more complex control systems. This paper aims to develop a quadruped robot capable of walking when teleoperated and assess its locomotion performance based on experiments. The first part of the methodology was dedicated to research and the definition of the robot's concept. The following stage involved the development of the robot's software, simulation, and, in parallel, the design, fabrication and assembly of the prototype. After both stages were done, the software was integrated into the prototype. After the integration stage, the locomotion experiments were conducted. Based on these experiments, it was possible to conclude that the robot was able to perform gaits of different sizes and walk in both plane and uneven terrains. Furthermore, it was observed that the body's rotation PID controller contributed to reducing the oscillation of the robot's body in both roll and pitch during walking in both terrains.

Index Terms—Quadruped robot; Locomotion; Control.

I. INTRODUCTION

As robotics advances, mobile robots are gaining ground in key economic sectors such as commercial, industrial, and military. Bio-inspired robots that move with legs have proven to be more efficient at moving over uneven, sloping, and slippery terrains, as well as overcoming obstacles [1]. Compared to wheeled robots, robots with legs still have better mobility and maneuverability in complex environments, which makes it possible to move along paths that are not necessarily continuous. Among legged robots, quadrupeds are gaining prominence due to their greater stability and simple structure when compared to bipeds and hexapods.

This work aims to prototype a quadruped robot platform for research: develop its kinematic, control, and locomotion algorithms, manufactory and test. Through experiments and statistical analysis, the prototype locomotion performance and stability will be evaluated, both on flat and uneven terrains, to check the efficacy of a PID-based stabilization controller.

This paper is divided as follows: section X presents the development of the robot, focusing on the gait planner, the kinematic model, the stabilization control and the trajectory planner implementations. In section X, the experiments conducted to evaluate and validate the effectiveness of the designed controller through statistical analysis are explained.

Finally, section X presents the conclusions and suggestions for future work.

II. ROBOT DEVELOPMENT

INTRO

A. Prototype

A small quadruped was designed and developed for research (Fig. X). During the prototyping phase, the hardware architecture and physics structure was designed. Following that, the structure of the robot was printed and integrated with the hardware components. The robot's actuators are servo motors of the model dynamixel MX-28, and a Raspberry Pi 4 was chosen as the processing unit. Furthermore, it contains an inertial sensor, model MPU6050. All the software was developed with the Robot Operating System 2 Humble (ROS 2) [2].

IMAGEM DO ROBO

The robot's structure is mammal-type, which is characterized by the presence of two joints in the sagittal plane, similar to animals such as dogs and horses. The robot's legs follow a configuration called full-elbow, in which the four legs are oriented backward, as shown in Fig. X. Furthermore, it features 3 degrees of freedom (DoF) per leg, providing the legs with substantial freedom of movement. The robot was designed to favor the mass balance between the body and the robot's legs, that is, the majority of its mass is located in the body or as close to it as possible. Lighter legs can move quickly without significantly changing the center of gravity of the robot, which increases stability and requires less control complexity. On the other hand, the legs should be strong enough to support the robot's weight and the impact on the ground.

The internal components, which include sensors, the processing unit, and the communication interface with the actuators were arranged symmetrically in the robot's body. The actuators (components that contribute to most of the system's mass) were positioned as close as possible to the body. The motor that acts on the tibia joint was placed in the upper part of the femur, to reduce the moment of inertia of the leg. This choice required the addition of a transmission system between the engine and the tibia link, consisting of a rigid metal rod with a ball joint at each edge.

B. Gait Planner

Quadruped robots move according to a gait pattern. The gait implemented for the robot follows the sequence of phases demonstrated in Fig. 1, in which the white rectangles represent the swing phase and the gray rectangles represent the stance phase. During the stance phase, the legs are on the ground and push the robot in the desired direction. In the swing phase, the legs are raised and moved to the next support point. This gait pattern is called trot and is based on the concept of periodic and symmetrical gaits. The periodic gaits are characterized by the continuous repetition of the movements in the same instants. Symmetry is a characteristic of gaits that move a pair of legs together, leaving and returning to the ground in a synchronized way. Although the structure of the robot makes it possible to perform other types of gaits, in this work, only the trot was considered. To reduce the control complexity, a discontinuous gait was adopted, that is, the robot's body moves only when all the legs are on the ground, resulting in intermittent movement. In Fig. 1, it is noticeable that the same pair of diagonal legs always move at the same instant. Between two consecutive swing phases, there is a moment when all legs are in the stance phase, which is when the robot body is displaced in the desired direction of locomotion.

Fig. 1: Gait pattern diagram

Frontal Direita				
Frontal Esquerda				
Traseira Esquerda				
Traseira Direita				

REESTRUTURAR ESSE PARAGRAFO;

The gait planner is responsible for controlling each leg and the robot's body. It defines where each foot will step and where the body should move based on the desired velocity command. Through a trajectory planner, it calculates the trajectory each foot and body should follow based on the velocity command. The gait planner then reads the pitch and roll angles of the body from the stabilization controller

The gait planner then sends commands to each joint based on the planned trajectories and the control efforts from the stabilization controller of the body at a frequency of 50 Hz. To convert both trajectories' points and the control efforts into the angles of the joints, a kinematic model is used, which is responsible for mapping the three-dimensional position of one foot to the joints' angles of its respective leg.

C. Trajectory Planner

The trajectory planner is responsible for calculating the trajectory that each foot must perform. For both the stance and swing phases, a cycloidal curve is adopted. As shown in [3], a cycloidal curve can be defined in the three-dimensional space by (1) to (4) between the points (x_o, y_o, z_o) and (x_f, y_f, z_f) as a function of time t , step height H , and period T .

$$x = (x_f - x_o) \frac{K - \sin(K)}{2\pi} + x_o \quad (1)$$

$$y = (y_f - y_o) \frac{K - \sin(K)}{2\pi} + y_o \quad (2)$$

$$z = H \frac{1 - \cos(K)}{2} + z_o \quad (3)$$

$$K = \frac{2\pi t}{T} \quad (4)$$

The graph of a cycloidal curve in 3D space is shown in Fig. 2a. The same trajectory is illustrated in Fig. 2b as a function of time. It is noticeable that the curve has the first null derivative when the foot touches the ground, which is favorable to open loop control, since the smoother the landing, the fewer disturbances are caused in the system.

In addition to the step height, the x and y translations, and the period, the robot's trajectory planner also considers two parameters to make the landing of the foot smoother. Since the robot's actuators are servo motors controlled by position, the torque is proportional to the displacement it must perform between the points of the trajectory. Therefore, the higher the resolution of the trajectory, the smoother the movement. However, the resolution of the trajectory N is fixed, given as a function of the step period and the frequency of the gait planner (50 Hz) $N = 50T$. Thus, the strategy adopted is to distribute the N points unevenly throughout the step period, so that there are more points near the moment the foot lands on the ground and fewer points near the moment the foot takes off. The P_T parameter is a fraction of the total step period, and the P_N is the fraction of the total number of points that should be between 0 and $P_T \cdot T$. In other words, if $P_T = 0.66$ and $P_N = 0.33$, 33% of N will be in the first two-thirds of the period, while the remaining 67% will be in the final one-third. The trajectory, considering these parameters, is illustrated in Fig. 2c.

D. Kinematic Model

The kinematic model of a quadruped robot describes the relationship between the position of the foot in the three-dimensional space and the rotation of each joint of its corresponding leg. This model is used to solve the forward kinematics, which provides the position (x, y, z) of the foot as a function of the leg joint's angles, and the inverse kinematics, which provides the rotation of each joint to achieve the desired (x, y, z) position of the foot. Moreover, this model can be extended to encompass the kinematics of the quadruped's body.

The *tf2* package, an available *ROS2* feature that can manage transformations between coordinate axes, was used to solve the robot's forward kinematics, and the inverse kinematics was solved based on geometric analysis. The variables θ_1 , θ_2 and θ_3 express the angular position of each robot leg joint (angles of hip, femur and tibia, respectively), and they can be calculated with the help of (5) to (10), as a function of the desired position (x_{IK}, y_{IK}, z_{IK}) of the foot and the lengths L_1 , L_2 and L_3 (Fig. 3).

$$\theta_1 = \arctan\left(\frac{x_{IK}}{y_{IK}}\right) - \arctan\left(\frac{L_1}{a}\right) \quad (5)$$

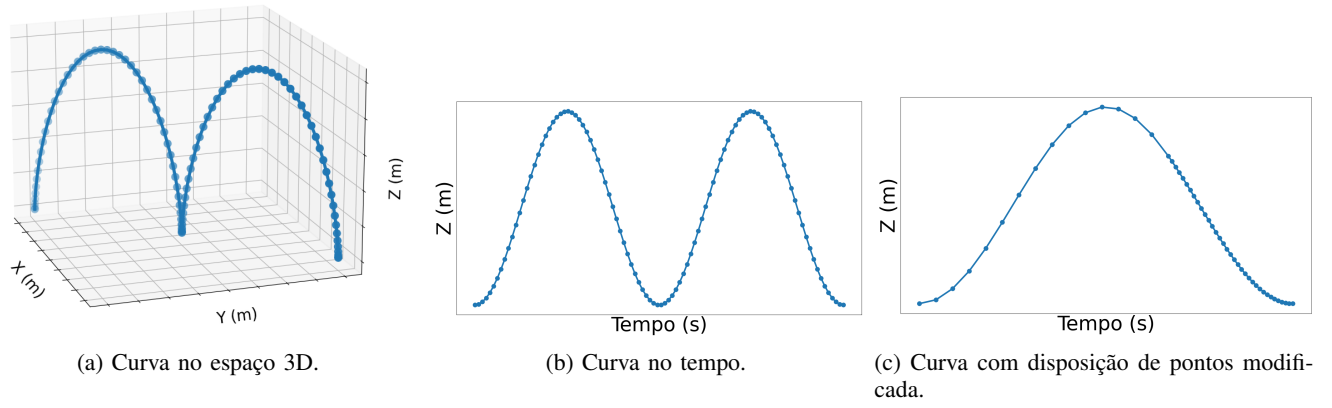


Fig. 2: Trajetórias cicloidais para o passo de robô.
Fonte: autores.

$$\theta_2 = \frac{\pi}{2} - \arctan\left(\frac{a}{z_{IK}}\right) - \arctan\left(\frac{\sqrt{1-A^2}}{A}\right) \quad (6)$$

$$\theta_3 = \arctan\left(\frac{\sqrt{1-B^2}}{B}\right) \quad (7)$$

$$a = \sqrt{x_{IK}^2 + y_{IK}^2 - L_1^2} \quad (8)$$

$$A = \frac{a^2 + z^2 + L_2^2 - L_3^2}{2L_2\sqrt{a^2 + z_{IK}^2}} \quad (9)$$

$$B = \frac{a^2 + z_{IK}^2 - L_2^2 - L_3^2}{2L_2L_3} \quad (10)$$

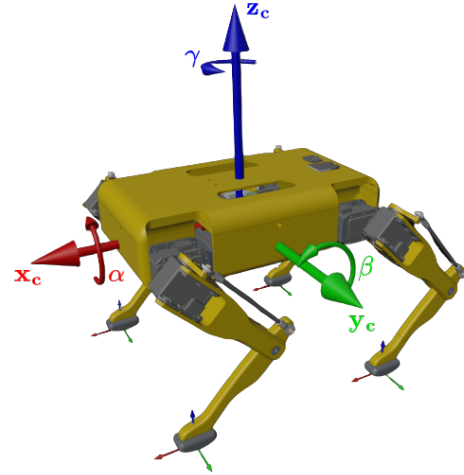


Fig. 4: Robot axes in resting position.

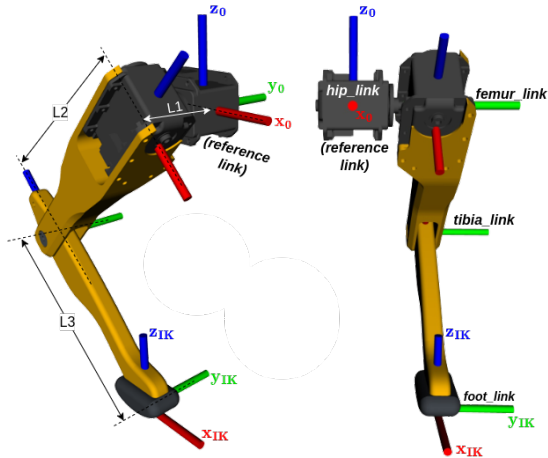


Fig. 3: Links from the robot leg.

These equations are useful for calculating only one foot's position but are insufficient for the robot's body kinematics. For that, a center frame called *base_link* (Fig. 4) is taken as a reference. The transformation matrix T_M is created with the desired body translations (x_c, y_c, z_c) and rotations (α, β, γ). The transformations T_{FR} , T_{FL} , T_{BL} and T_{BR} of each of the shoulders (*hip_links*) with respect to the *base_link* are also required.

The calculation of the joints' angles of each leg is then done using as inputs the values resulting from each of the transformations, according to (11). While $[x, y, z, 1]^T$ in the right-hand side of (11) represents the foot position relative to the robot's *base_link*, $[x_{IK}, y_{IK}, z_{IK}, 1]^T$ on the left-hand side represents the same position relative to the leg's *hip_link*, which can be used as input in equations (5) to (10).

$$\begin{bmatrix} x_{IK} \\ y_{IK} \\ z_{IK} \\ 1 \end{bmatrix} = (T_M \cdot T_{FR})^{-1} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (11)$$

The same calculation can be done for the other legs, using T_{FL} , T_{BL} , and T_{BR} . In this way, the equation (11) allows computing the angles θ_1 , θ_2 and θ_3 of each leg from the desired position of each foot and the desired translations and rotations of the body, using the same frame as reference (*base_link*). It means one can control all the six DoFs of the body together with the positions of the feet at the same time.

E. Control Subsystem

The control subsystem of the robot is composed of two main components: the joints' controllers, and the body's stabilization controller. The joints' controllers, each one a PID, are responsible for controlling each joint of the robot independently. Meanwhile, the body's stabilization controller is responsible for controlling the pitch and roll angles of the robot, maintaining it at 0 degrees.

¡APROFUNDAR JOINT CONTROLLER¡

The stabilization controller is formed by two PID controllers for the roll and pitch rotation of the robot's body. They act independently, stabilizing the rotation in each axis simultaneously. Both controllers are identical and were implemented following the block diagram in Fig. 5.

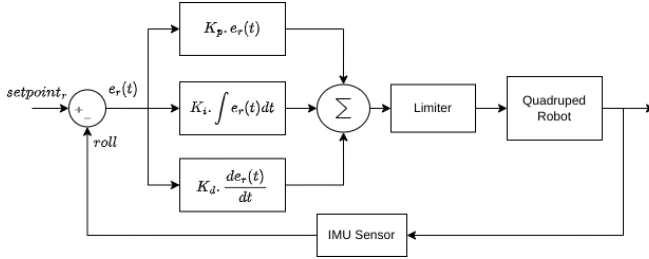


Fig. 5: Designed PID controller.

The IMU sensor is responsible for measuring the rotation of the robot's body, providing feedback on the system's outputs. The limiter was added to avoid body rotations that exceed the limits of the joints.

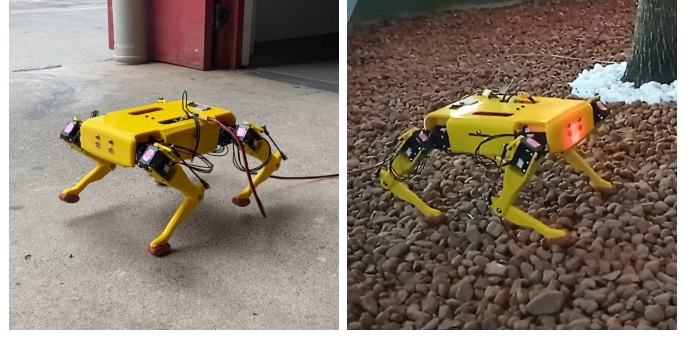
III. RESULTS

¡COLOCAR OUTRO EXPERIMENTO¡

To assess the robot's locomotion performance with the implemented stabilization controller, the robot's capability of following a forward velocity setpoint and the oscillation of its body while walking were evaluated. An experiment was conducted, in which the robot should walk in a straight line of 1.5 meters in both flat and uneven terrain, with and without the stabilization controller, totalizing four tests. The flat terrain consisted of a cement floor and the uneven terrain, a ground formed by small loose stones. Both are illustrated in Fig. 6a and 6b, respectively.

For each test, the robot received a forward velocity command of 0.05 m/s, and its mean velocity, as well as its pitch and roll oscillation, were recorded. The mean velocity was calculated based on the time it took to travel 1.5 meters and the oscillation is the difference between the maximum and minimum readings of the IMU. The calculated velocity was compared to the desired velocity sent to the robot and the stability data was analyzed statistically: first, the Shapiro-Wilk test [4] was applied to the distribution to check whether the distribution was normal; lastly, the results were compared by performing an Analysis of Variance (ANOVA) [5].

For all the tests, the robot's gait settings consisted of a height of 5 cm, a period of 0.5 seconds, a resolution of 25



(a) Flat.

(b) Uneven.

Fig. 6: Terrains used for the experiment

points, $P_T = 0.66$, and $P_N = 0.33$. For each test type, 10 samples were collected.

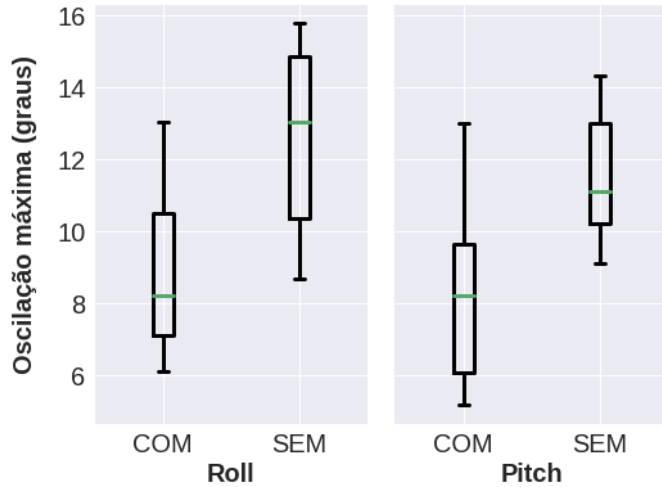
Table I summarizes the collected results. The robot did not reach a velocity of 0.05 m/s in any test. This result is due to errors in the final position of each leg during the swing phase: the legs could not reach the desired (x, y) position required to maintain the desired velocity. The fact that the final position of one foot is a function of its motors' final orientation indicates that the motors did not reach their desired orientation. Before the experiment, the gait movement was tested with the legs off the ground (almost no load), and no significant errors were noted. Since the errors only appeared while walking, it is perhaps caused by the fact that the motors were not powerful enough to support the robot's weight appropriately while walking.

TABLE I: Velocity and stability results

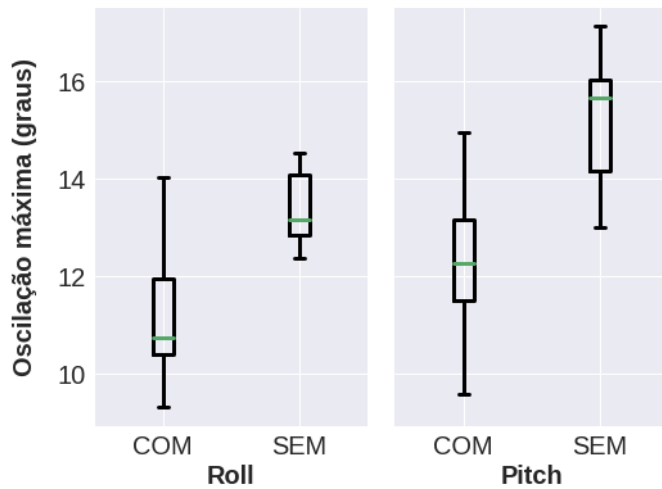
Test	1	2	3	4
Ter.	Flat	Flat	Uneven	Uneven
S. C.	No	Yes	No	Yes
Vel. (cm/s)	2.15	3.68	2.03	2.39
σ_{Vel} (cm/s)	0.040	0.075	0.016	0.047
Δ_{Roll}	12.59°	8.81°	13.37°	11.13°
σ_{Roll}	2.54°	2.36°	0.78°	1.47°
Δ_{Pitch}	11.55°	8.31°	15.19°	12.09°
σ_{Pitch}	1.89°	2.63°	1.44°	1.66°

In addition to the velocity results, the pitch and roll oscillations were compared to determine whether or not the stabilization controller lowered the robot's body oscillation. The Shapiro-Wilk test was applied to the data from all the tests, after removing outliers, and every test returned a p-value > 0.05 , which indicates that all the distributions followed a normal curve. The ANOVA was, then, applied to compare the roll and pitch oscillations from tests 1 and 2. The same process was applied to the tests 3 and 4 as well. The ANOVA revealed that the roll and pitch distributions of the tests with and without stabilization control were significantly different. Table I shows that the mean value of the tests with stabilization control was less than the mean value of the test without it, which proves that it indeed lowered the robot's body

oscillations in both terrain types. Fig. 7 better pictures the data distribution. Despite the more scattered distribution presented, 3 out of 4 of the tests with stabilization control had its third quartile less than the first quartile of the distribution without the controller for the same terrain type, demonstrating that the controller lowered the oscillations in 75% of the collected test samples.



(a) Terreno plano.



(b) Terreno irregular.

Fig. 7: Oscilação do corpo em ambos os tipos de terreno.
Fonte: autores.

IV. CONCLUSION

This paper addressed the design, prototyping and testing of a quadruped robot. An experiment was held to assess the robot's walking performance. The results showed that the robot did not reach the desired velocity in any of the tests. This was due to errors during the gait's swing phase, which indicates that the motors were unable to support the robot's weight while walking. These errors were not compensated, because the gait's trajectory tracking is open-loop (only the individual joint's controllers are closed-loop). In addition to that, it

was possible to conclude that the implemented stabilization controller was able to lower the robot's body oscillations in pitch and roll while walking in both flat and uneven terrains.

Reviewing the robot's actuators is recommended for future work. Perhaps a model with more torque is required for more precise control performance.

The robot's project is open source and is available on GitHub [6].

REFERENCES

- [1] X. Zhang, L. Lang, J. Wang, and H. Ma, 'The quadruped robot locomotion based on force control', Proceedings of the 2015 27th Chinese Control and Decision Conference, CCDC 2015, pp. 5440–5445, 2015.
- [2] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, 'Robot Operating System 2: Design, architecture, and uses in the wild', Science Robotics, vol. 7, no. 66, p. eabm6074, 2022.
- [3] Y. Shi, S. Li, M. Guo, Y. Yang, D. Xia, and X. Luo, 'Structural design, simulation and experiment of quadruped robot', Applied Sciences (Switzerland), vol. 11, 2021.
- [4] V. B. Leotti, A. R. Birck, and J. Riboldi, 'Comparação dos Testes de Aderência à Normalidade Kolmogorov-smirnov, Anderson-Darling, Cramer-Von Mises e Shapiro-Wilk por Simulação', Anais do 11º Simpósio de Estatística Aplicada à Experimentação Agronômica, 2005.
- [5] E. L. Cano, J. M. Moguerza, and A. Redchuk, Six sigma with R: statistical engineering for process improvement, vol. 36. Springer Science & Business Media, 2012.
- [6] B. S. de Alencar, F. M. S. M. Barreto, and L. L. Souza, 'Caramel', GitHub repository, 2022.