

Analysis, Prototyping and Locomotion Control of a Quadruped Robot

1st Lucas Souza

*Autonomous Robotics Research Center
Technology Innovation Institute
Abu Dhabi, UAE
lucas.souza@tii.ae*

2nd Felipe Mohr

*Robotics Departament
SENAI CIMATEC
Salvador, Brazil
felipe.barreto@fieb.org.br*

3rd Brenda Alencar

*Robotics Departament
SENAI CIMATEC
Salvador, Brazil
brenda.alencar@fieb.org.br*

Abstract—With the growth of robotics, mobile robots are increasingly becoming common 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 with a PID stabilization controller to compensate for the body oscillation during the walk. For that, the gait planner and the kinematic model algorithm were implemented. After designing and manufacturing the robot, experiments were conducted to assess the robot's locomotion performance when using the stabilization controller in flat and uneven terrains. It was observed that the stabilization controller contributed to reducing the oscillation of the robot's body in both roll and pitch angles in both terrains.

Index Terms—Quadruped robot, locomotion control, kinematic model, gait planner.

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 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, manufacture it and test it. Through experiments and statistical analysis, the prototype locomotion performance and stability were evaluated, both on flat and uneven terrains, to check the efficacy of a PID-based stabilization controller.

This paper is divided as follows: section II presents the development of the robot. Section III shows the results of the experiments conducted to evaluate and validate the effectiveness of the designed controller. Finally, section IV presents the conclusions and suggestions for future work.

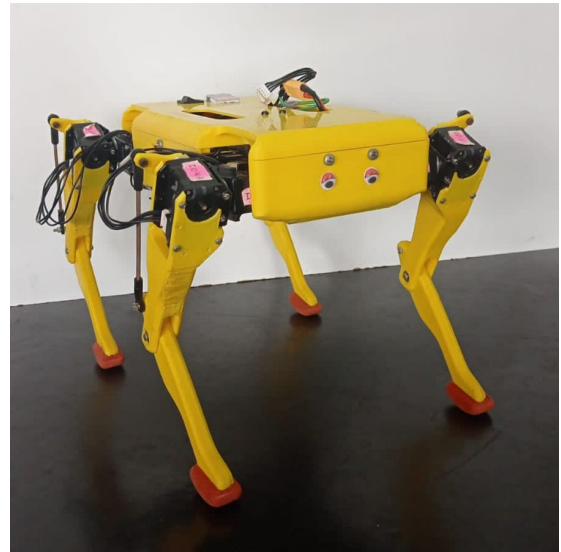


Fig. 1: The quadruped robot.

II. ROBOT DEVELOPMENT

The robot platform designed for this research is presented in Fig. 1. The development encompassed the prototyping and the implementation of the gait planner, the trajectory planner, the kinematic model, and the control subsystem.

A. Prototype

During the prototyping phase, the hardware architecture and physics structure was designed. Following that, the structure of the robot was printed in ABS (Acrylonitrile Butadiene Styrene) 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 of the model MPU6050. All the software was developed with the Robot Operating System 2 Humble (ROS 2) [2].

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 [3]. The robot's legs follow a configuration called full-elbow, in which the four legs have their lower joints (the tibia joint) oriented backward, as shown in Fig. 1. 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 [4].

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, which then 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. 2, 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's body in the desired direction. In the swing phase, the legs are raised and moved to the next support point. The gait pattern illustrated in Fig. 2 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 [5]. 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. 2, 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 motion.

For the robot to move according to the adopted gait pattern, there is a gait planner responsible for commanding the robot's legs. The gait planner defines where each foot will step and where the body should move based on the desired velocity of

the robot. A trajectory planner calculates the trajectory each foot and body should follow. Then, the gait planner reads, from the stabilization controller, the corrections in pitch and roll angles and applies them before sending the commands to each joint, 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 was adopted. As shown in [6], 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 plot of a cycloidal curve in 3D space is shown in Fig. 3a. The same trajectory is illustrated in Fig. 3b as a function of time. It is noticeable that the curve has a null first 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, y) translations, and the period, the robot's trajectory planner also considers two parameters to make the landing of the foot even 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 N of the trajectory 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 $P_N = 0.33$, then 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. 3c.

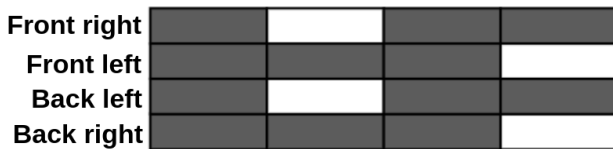


Fig. 2: Gait pattern diagram.

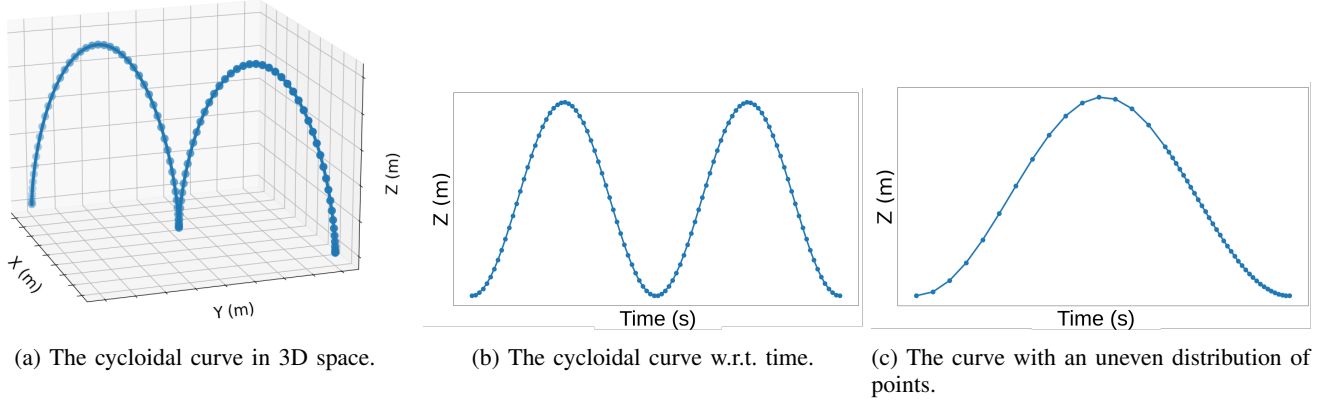


Fig. 3: Cycloidal trajectories by the trajectory planner.

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 package tf2 [7], an available ROS 2 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 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 of the foot w.r.t. to the hip link (x_{IK}, y_{IK}, z_{IK}) and the lengths L_1 , L_2 and L_3 (Fig. 4).

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

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

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

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

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

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

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. 5) 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) .

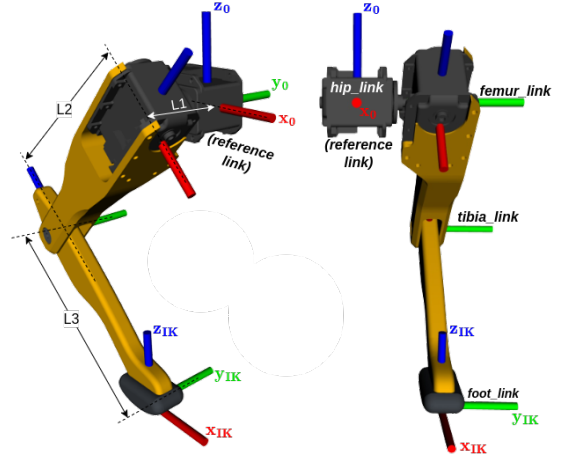


Fig. 4: Links from the robot leg.

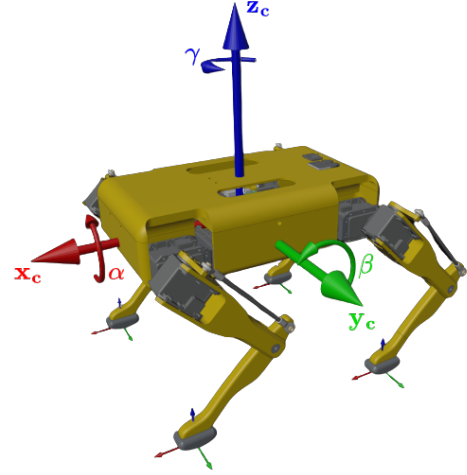


Fig. 5: Robot axes in resting position.

The transformations T_{FR} , T_{FL} , T_{BL} and T_{BR} of each of the four hip_links with respect to the base_link are also required.

The calculation of the joints' angles of each leg is then performed 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 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, by substituting T_{FR} by T_{FL} , T_{BL} , or T_{BR} in (11). In this way, (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 are the built-in Dynamixel position PID controllers, and they control each joint of the robot independently. Meanwhile, the body's stabilization controller is responsible for controlling the pitch and roll angles of the robot [6], [8], maintaining it at 0 degrees.

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. 6 (the output of the "IMU Sensor" block is either roll or pitch).

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

To assess the robot's locomotion performance, three experiments were conducted. The first experiment aimed to test the trajectory tracking of one single leg in a no-load scenario, a more ideal case. The second experiment evaluated the robot's capability of following a forward velocity setpoint and the amplitude of its body oscillations' while walking in a straight line, i.e. stability, in flat and uneven terrain. Therefore, in the first experiment, the fundamental functionality of performing

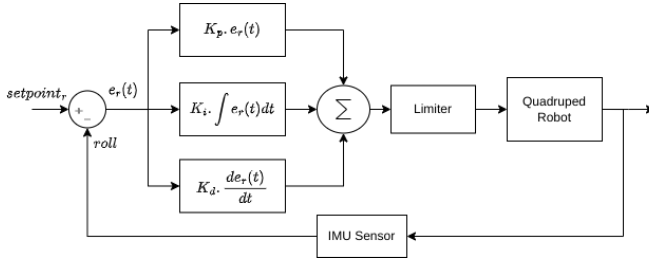


Fig. 6: Designed PID controller.

one gait step with one leg was tested, while in the second the system was submitted to a realistic operation scenario where the gait planner must manage the movement of all legs and the body. The third experiment was similar to the second but focused on the yaw motion of the robot and was performed only on uneven terrain.

For the first experiment, the robot's body was supported on a platform so that its leg would not touch the floor, lowering the load applied to the motors. The experiment consisted of sending gait trajectories to a single leg and evaluating how precise the leg could follow it. The trajectories were cycloidal curves with a height of 0.05 cm, a period of 0.5 seconds and a resolution of 25 equally distributed points ($P_T = P_N = 1.0$). The distances in the (x, y) directions were (5.0, 3.0) and (3.0, 5.0) cm, and, for each distance pair, 30 samples were collected. Table I shows the average and standard deviation of the time taken to complete the trajectory, the height, and the distance reached by the leg in the (x, y) direction. Test 1 refers to the trajectories of the first distance pair and Test 2 to the second. Fig. 7 shows the z-axis (height) tracking of one sample, where the blue curve is the trajectory commands (in meters) over time (in seconds) and the red curve is the trajectory performed by the foot.

It is noticeable that there was a delay of approximately 57 milliseconds in the execution time and that the final height was off by approximately 0.25 centimeters in both tests. To confirm that the system's performance during Test 1 and 2 was similar, i.e. the difference in (x, y) distance did not affect the overall performance, two ANOVA [9] tests were conducted comparing the execution time and gait height distributions of both tests. Before the ANOVA, the outliers of these distributions were removed and the Shapiro-Wilk [10] test was used to check the data normality. The result of the latter was a p-value greater than 0.05 for all distributions, which indicates that they followed a normal curve. The ANOVA tests were, then, conducted and showed a p-value of 0.7212 for the execution

TABLE I: Results of the gait trajectory tracking experiment.

Test	1	2
$time(s)$	0.557	0.557
$\sigma_{time}(s)$	0.008	0.008
$\bar{z}_{max}(cm)$	4.760	4.755
$\sigma_{z_{max}}(cm)$	0.023	0.028
$\bar{x}_{final}(cm)$	5.034	3.024
$\sigma_{x_{final}}(cm)$	0.009	0.013
$\bar{y}_{final}(cm)$	2.924	4.915
$\sigma_{y_{final}}(cm)$	0.013	0.004

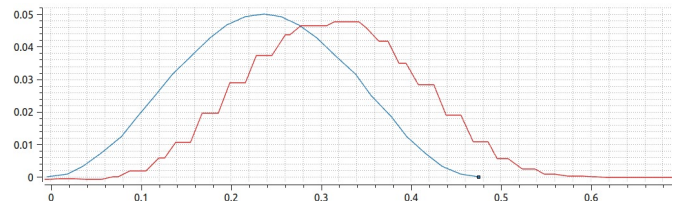


Fig. 7: Trajectory executed by the foot.

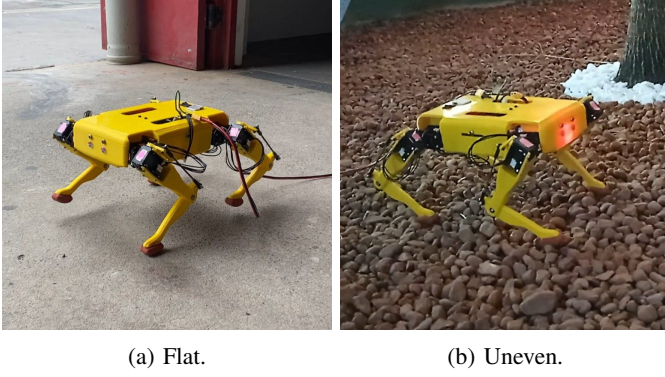


Fig. 8: Terrains used for the experiment.

time comparison and 0.5070 for the gait height comparison, which means that the difference in the (x, y) distance did not affect the trajectory tracking performance. In addition to that, the final average error for x and y was less than 0.1 centimeters and the standard deviation was small, meaning the gait distance was followed correctly and that this result was consistent throughout all samples.

For the second experiment, 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. 8a and 8b, respectively.

For each test, the robot received a forward velocity command of 0.05 m/s. 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 oscillation data was analyzed statistically following the same procedure of the trajectory tracking experiment.

For all the tests, the robot's gait settings consisted of a height of 5 centimeters, a period of 0.5 seconds, a resolution of 25 points, $P_T = 0.66$, and $P_N = 0.33$. For each test type, 10 samples were collected.

Table II summarizes the collected results. The robot did not reach a velocity of 0.05 m/s in any test. This result is due to steady state errors in the final position of each leg during the swing phase: the legs could not reach the desired (x, y) position required during the gait period, 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. And since no significant errors were noted during the previous experiment (a more ideal situation), they were perhaps caused by a lack of power from the motors, which were not able to support the robot's weight appropriately while walking.

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

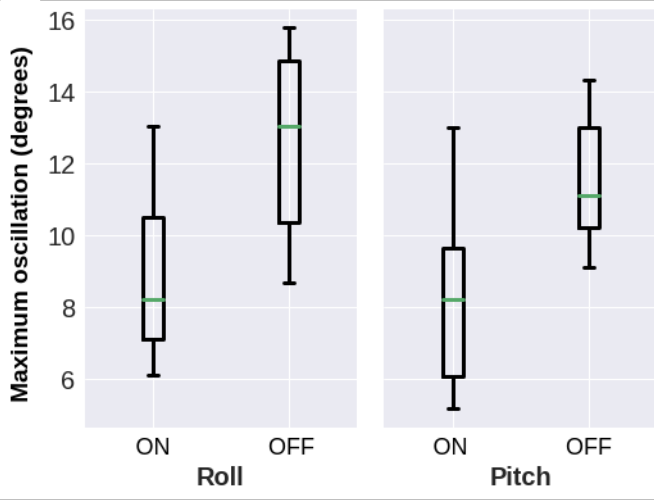
TABLE II: Velocity and stability results.

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°

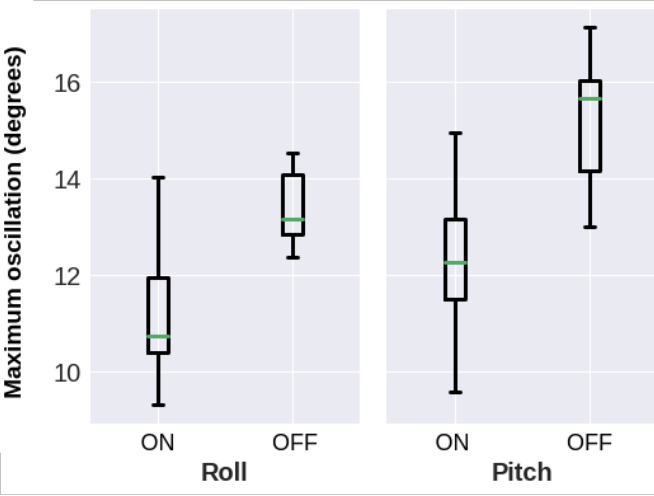
tests, after removing outliers, and every test returned a p-value greater than 0.05, which indicates that the distributions followed a normal curve. The ANOVA was, then, applied to compare the roll and pitch oscillations from tests 1 and 2 (flat terrain). The same process was applied to the tests 3 and 4 as well. The ANOVA revealed that the roll and pitch distributions of the tests with and without stabilization control were significantly different. Table II shows that the mean value of the oscillations in both axes of the tests with stabilization control was less than the mean value of the tests without it, indicating that the stabilization controller lowered the robot's body oscillations in both terrain types. Fig. 9 better pictures the data distribution. Despite the more scattered distribution, 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 at least 75% of the collected test samples.

Finally, in the third experiment, the robot was placed in the same uneven terrain and was commanded to follow a yaw motion of 0.25 rad/s. The experiment was conducted until the robot had completed a full rotation around its z-axis. Again, the time, roll oscillation and pitch oscillation were recorded along 10 samples with and without the stabilization controller.

Table III summarizes the collected results. The robot did not reach the angular velocity setpoint of 0.25 rad/s but maintained a steady 0.19 rad/s in both tests with a very low standard deviation, suggesting no significant differences. The body rotation analysis was also performed as before. The Shapiro-Wilk test presented a p-value greater than 0.05, which led to the application of the ANOVA test. The ANOVA test between the roll oscillation distributions with and without the stabilization controller resulted in a p-value of approximately 0.5, supporting the thesis that the controller did not affect the yaw motion significantly. The ANOVA test for the pitch oscillation, on the other hand, shows that the pitch oscillation was slightly bigger with the stabilization controller, which explains a much lower p-value, around 0.07. This result, though different from the previous two, still indicates no significant changes in the stabilization controller while the robot performs a yaw movement.



(a) Flat terrain.



(b) Uneven terrain.

Fig. 9: Robot body's oscillation in both terrains.

TABLE III: Angular velocity and stability results.

Test	1	2
Ter.	Uneven	Uneven
S. C.	No	Yes
Angular vel. (rad/s)	0.19	0.19
$\sigma_{Ang. Vel.}$ (rad/s)	0.006	0.007
Δ_{Roll}	16.16°	16.84°
σ_{Roll}	1.88°	2.6°
Δ_{Pitch}	11.79°	12.89°
σ_{Pitch}	1.05°	1.41°

IV. CONCLUSION

This paper addressed the design, prototyping and testing of a quadruped robot. Two experiments were 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 a straight line in both flat and uneven terrains, but did not affect the yaw motion of the quadruped in uneven terrain.

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 [11].

ACKNOWLEDGMENT

The authors' acknowledgments.

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. [Online]. Available: <https://www.science.org/doi/abs/10.1126/scirobotics.abm6074>
- [3] S. Kitano, S. Hirose, A. Horigome, and G. Endo, "Titan-xiii: sprawling-type quadruped robot with ability of fast and energy-efficient walking," *ROBOMECH Journal*, vol. 3, pp. 1–16, 2016.
- [4] Y. Zhong, R. Wang, H. Feng, and Y. Chen, "Analysis and research of quadruped robot's legs: A comprehensive review," 3 2019.
- [5] P. G. De Santos, E. Garcia, and J. Estremera, *Quadrupedal locomotion: an introduction to the control of four-legged robots*. Springer, 2006, vol. 1.
- [6] 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.
- [7] T. Foote, E. Marder-Eppstein, and W. Meeussen, "Tf2 package," [Online]. Available: <https://index.ros.org/p/tf2/github-ros2-geometry2/>, (Accessed in Jun. 17, 2023).
- [8] N. Kau, A. Z. Schultz, and M. Bowers, "Stanford quadruped," [Online]. Available: <https://github.com/stanfordroboticsclub/StanfordQuadruped>, (Accessed in Nov. 24, 2022).
- [9] E. L. Cano, J. M. Moguerza, and A. Redchuk, *Six sigma with R: statistical engineering for process improvement*. Springer Science & Business Media, 2012, vol. 36.
- [10] 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.
- [11] B. S. de Alencar, F. M. S. M. Barreto, and L. L. Souza, "Caramel," [Online]. Available: <https://github.com/Brazilian-Institute-of-Robotics/black-mouth>, 2022, (Accessed in Nov. 28, 2022).