VISIÓN ELECTRÓNICA

Algo más que un estado sólido

https://doi.org/10.14483/issn.2248-4728

Quadruped Robot Prototype for Agricultural Mobile Robotics

Prototipo de robot cuadrúpedo para robótica móvil agrícola

Gissel Daniela Cristiano Ayala[1], Samuel Alejandro Sabogal Mora[2], David Esteban Fuquen Garces[3], Emily Angelica Villanueva Serna[4], Henry Borrero Guerrero[5]

Abstract

Agricultural robots with wheels have contributed significantly to agriculture, generating interest in developing robots with improved maneuverability in crop fields where wheels do not work well [1]. This has led to the development of legged mobile robots as a viable alternative to wheeled robots [2,3,4], and for this reason, a study group at the Universidad Francisco José de Caldas Distrital in Colombia has been developing a small-scale quadruped robot in order to examine its maneuverability capabilities in difficult terrain. In this paper, the corresponding robot and its results are described. Using 3D printing, we developed a robot structure that could accommodate eight servomotors. Using the CircuitPython programming language, we programmed a Raspberry Pico-W microcontroller as the control unit [5]. Thanks to a graphical user interface developed in HTML, a quadruped robot could be controlled remotely from a computer or even a mobile phone. As reported in the results, our quadruped robot approach

[1] Electronic Engineering student, Universidad Distrital Francisco José de Caldas. E-mail: gdcristianoa@udistrital.edu.co ORCID: https://orcid.org/0009-0000-5043-2897

[2] Electronic Engineering student, Universidad Distrital Francisco José de Caldas. E-mail: sasabogalm@udistrital.edu.co. ORCID: https://orcid.org/0009-0009-7706-5025

[3] Electronic Engineering student, Universidad Distrital Francisco José de Caldas. E-mail: defuqueng@udistrital.edu.co. ORCID: https://orcid.org/0009-0008-3184-9126

[4] Electronic Engineer, Universidad Distrital Francisco José de Caldas. E-mail: eavillanuevas@udistrital.edu.co

[5] Electronic Engineer, Universidad de los Llanos, Colombia. Doctoral degree on Mechanical Engineering, University of Sao Paulo, Brazil. Professor: Universidad Distrital Francisco José de Caldas, Colombia. E-mail: hbguerrero@udistrital.edu.co

has demonstrated promising results in its maneuverability on off-road soils, opening the possibility of developing autonomous quadruped robots for agricultural use.

Keywords: Agricultural Robot, Legged Locomotion, CircuitPython, Mobile Robotics, Quadruped Robot, Raspberry Pi Pico.

Resumen

Los robots agrícolas con ruedas han contribuido significativamente a la agricultura, lo que ha generado interés en desarrollar robots con mejor maniobrabilidad en campos de cultivo donde las ruedas no funcionan adecuadamente. Esto ha dado lugar al desarrollo de robots móviles con patas como una alternativa viable a los robots con ruedas. Por esta razón, un grupo de estudio en la Universidad Distrital Francisco José de Caldas, en Colombia, ha estado desarrollando un robot cuadrúpedo a pequeña escala con el fin de examinar sus capacidades de maniobra en terrenos difíciles. En este artículo se describe el robot correspondiente y sus resultados. Utilizando impresión 3D, se diseñó una estructura capaz de alojar ocho servomotores. Mediante el lenguaje de programación CircuitPython, se programó un microcontrolador Raspberry Pi Pico-W como unidad de control. Gracias a una interfaz gráfica desarrollada en HTML, el robot cuadrúpedo puede ser controlado de forma remota desde un computador o incluso desde un teléfono móvil. Según lo reportado en los resultados, nuestro enfoque con el robot cuadrúpedo ha demostrado resultados prometedores en su maniobrabilidad sobre suelos irregulares abriendo la posibilidad de desarrollar robots cuadrúpedos autónomos para uso agrícola.

Palabras clave: Locomoción con patas, CircuitPython, Robótica agrícola, Robótica móvil, Robot cuadrúpedo, Raspberry Pi Pico.

1. Introduction

During the execution of a research project aimed at developing autonomous navigation for agricultural mobile robotics [6], we noticed that wheeled robots are not suitable for certain environments such as the Colombian Andes, where agriculture plays a key role in economic development [7,8].

According to the information provided in [9,10], deploying and operating a mobile robot in outdoor environments is often challenging due to their unstructured nature, which demands robust locomotion and significant computational capabilities. References [10] state that quadruped robots are generally preferred due to their mobility, ease of control, and stability [11].

Given that wheeled agricultural robots have sparked interest in the development of robots with improved maneuverability in crop fields where wheels do not perform well, a review of the literature on wheeled robots reveals that this type of robot is not widely used in Colombia. This validates the interest in developing this field of knowledge in the region. For this reason, a group of students from the Universidad Distrital Francisco José de Caldas in Colombia has been developing a small-scale quadruped robot to examine its maneuverability capabilities in simulated terrains.

The following are the stated objectives:

General Objective

- To develop a quadruped robot prototype to evaluate its functionality on different terrains and its integration into agricultural assistance.

Specific Objectives:

- To implement motion control using the Raspberry Pi Pico microcontroller.
- To evaluate the prototype's performance on simulated terrains.

- To implement remote control of the robot through connection with a computer or mobile device.

2. Theoretical Framework

2.1. Over time, the design of quadruped robots has evolved to become more efficient, compact, and capable. Early designs were often large and relied on hydraulic systems, which limited their practical applications due to size, weight, and power constraints [12, 13,14]. Advances in electric actuators and battery technology enabled the development of lighter and more agile robots with extended operating times.

2.2. When the task assigned to the robot requires mobility, Quadruped locomotion is commonly used. [15].

When implementing legged locomotion in a robot, it is essential to consider its position and speed, but also to ensure that the robot remains balanced and does not fall [16,17], relying solely on joint movement through motors [18,19].

2.3. CircuitPython is a programming language designed to simplify experimentation and learning to code on a low-cost microcontroller. It includes an organized set of libraries developed by Adafruit that complement CircuitPython. These libraries enable control of sensors, displays, motors, communications, and more [20].

2.4. The Raspberry Pi Pico W is a development board based on the RP2040 microcontroller, designed and built by the engineers at Raspberry Pi. It has been designed as a flexible, low-cost development platform with a 2.4 GHz wireless interface [21].

2.5. The SG90 servo is one of the most versatile and widely used servomotors in all types of robotics projects. It is small, yet it provides considerable torque of 1.8 kg/cm, making it suitable for a wide range of robots [22].

2.6. 3D printing is a process of creating objects by depositing layers of material on top of each other. It is referred to as additive manufacturing (AM), as opposed to traditional subtractive methods such as CNC milling, particularly when used for industrial production.

This technology has existed for about four decades, having been invented in the early 1980s. Although 3D printing initially began as a slow and expensive technique, significant technological advances have made modern AM technologies more affordable and faster than ever before [23].

2.7. Quadruped locomotion is the technique by which a four-legged robot is given movement, aiming to mimic the various forms of locomotion found in animals through nature [24]

2.8. The kinematic diagram of the leg makes it possible to identify the equations of the kinematic chain in both Cartesian and joint coordinates, considering only geometric aspects. Dynamic analysis, on the other hand, allows for the determination of forces, as well as the velocities and accelerations generated in the various joints due to these forces, whether caused by moments of inertia, centers of gravity, external forces, among others [25, 26, 27].

3. Results

3.1. Energy

Table 1: Energy consumption by type of movement.

Movement	Total Time (s)	Average current (mA)	Energy consumed (mAh)	Observations
Forward	1.33	520	0.19	Stable running, average consumption
Backward	1.34	580	0.22	Retraction with moderate load
Right	1.1	460	0.14	Moderate expenditure, 3 active legs
Left	1.06	470	0.14	Similar to right, slight variation
Dance	0.69	800	0.15	Fast movement, high consumption
Greet	16.5	650	2.92	Prolonged movement with an active but stable neck

Source: own.

3.2. Speed

Table 2: Locomotion speed by movement.

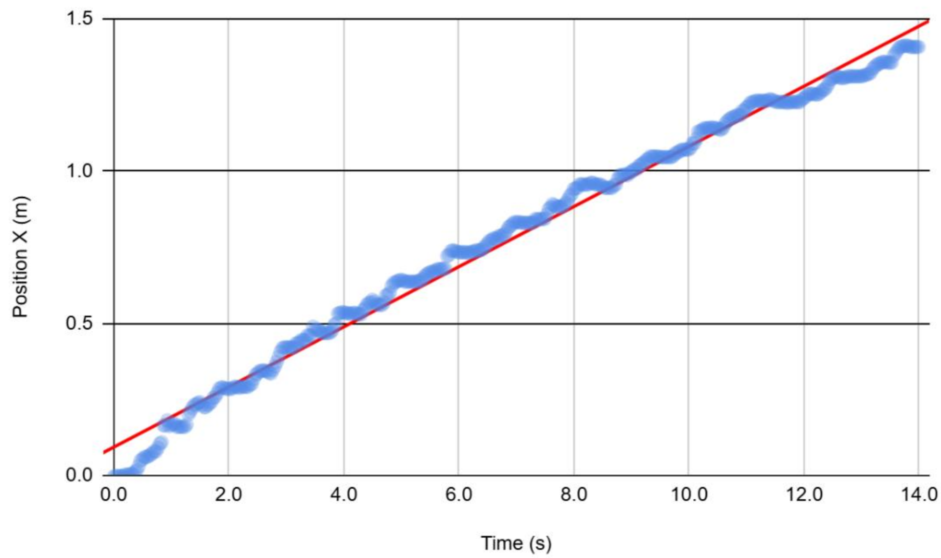
Movement	Time between steps (s)	Full cycle (s)	Frequency (Hz)	Distance (cm)	Total time (s)	Speed (cm/s)
Forward	0.3	1.33	0.75	100	14.12	7.082
Backward	0.3	1.34	0.75	100	12.21	8.19

Source: own.

3.3. Coordinates

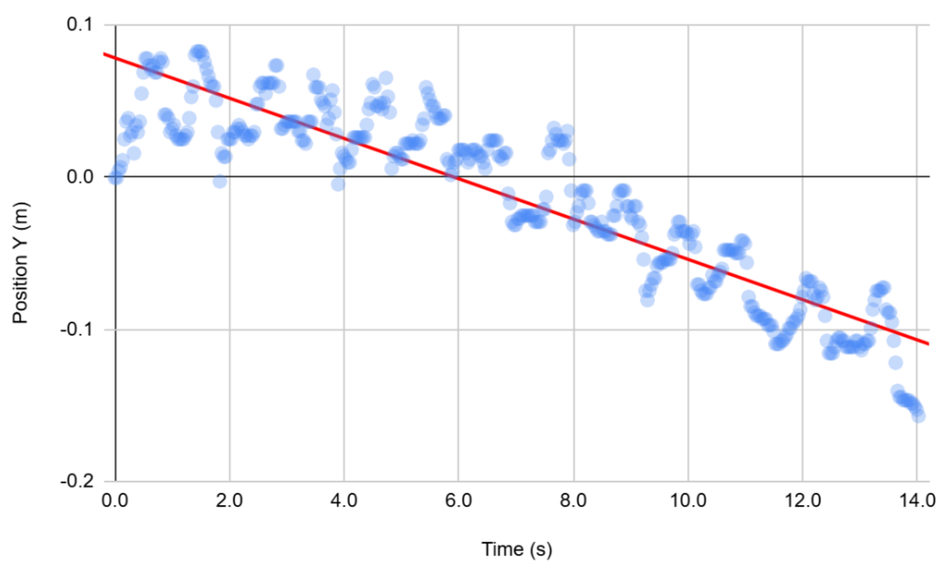
3.3.1. Forward

Figure 1: Time VS Position X "Forward".



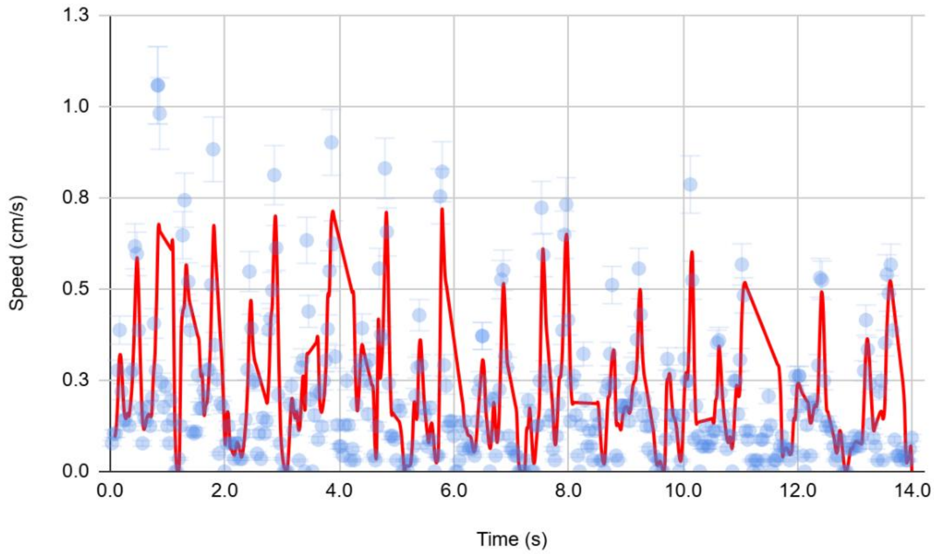
Source: own.

Figure 2: Time VS Position Y "Forward".



Source: own.

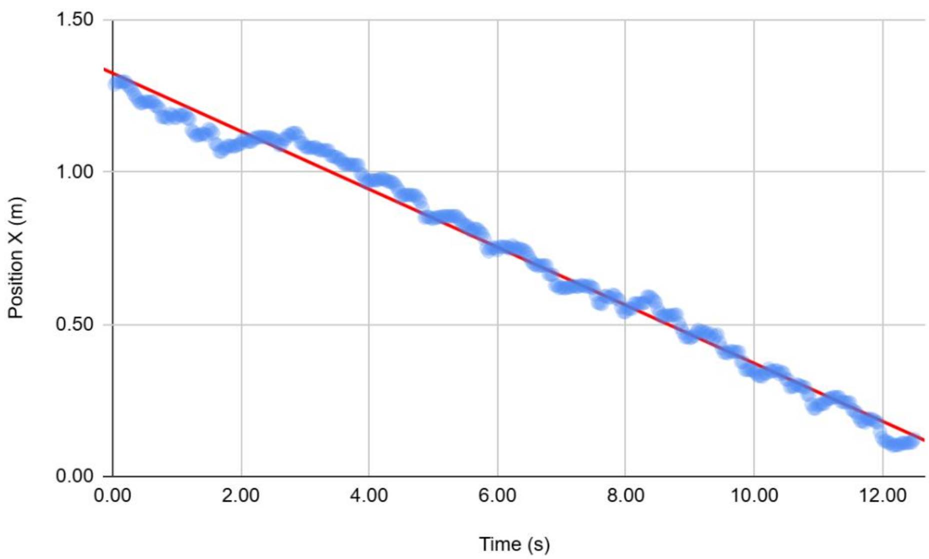
Figure 3: Time VS Speed "Forward".



Source: own.

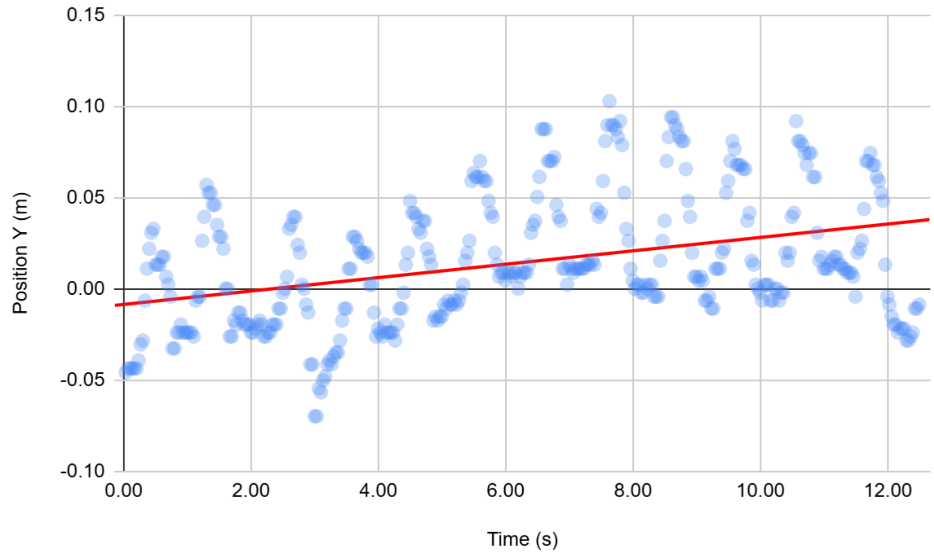
3.3.2. Backward

Figure 4: Time VS Position X "Backward".



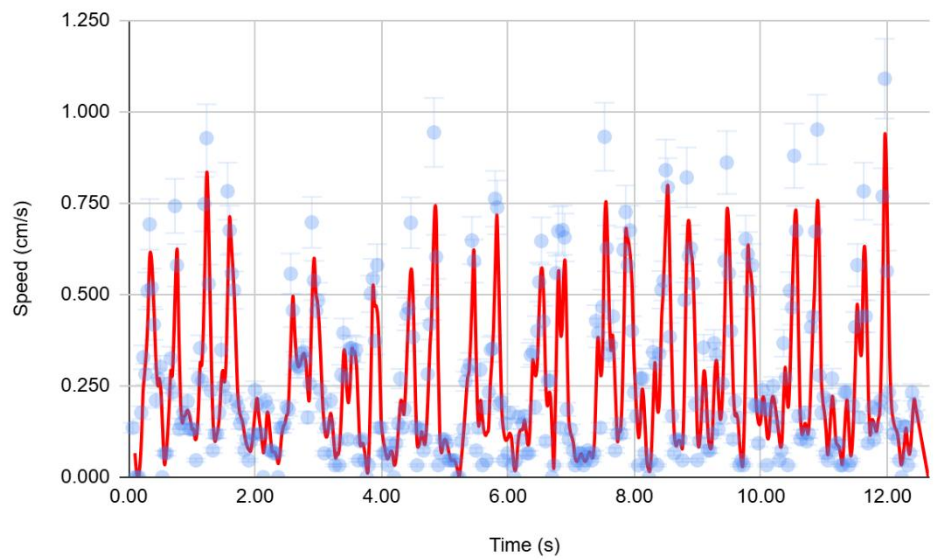
Source: own.

Figure 5: Time VS Position Y "Backward".



Source: own.

Figure 6: Time VS Speed "Backward".



Source: own.

3.4. **Gait Efficiency**

Table 3: Gait Efficiency (walking pattern)

Movement	Frequency (Hz)	Speed (cm/s)	Average current (mA)	Stability (Visual)
Forward	0.75	75.2	520	Tilt to one side
Backward	0.75	74.6	580	Tilt to one side
Right	0.9	20(rotation)	460	Stable
Left	0.94	20.3(rotation)	470	Stable
Dance	1.5	0.0(fixed)	800	Stable
Greet	0.12	0.0(fixed)	650	Stable

Source: own.

4. Methodology

The development of our quadruped robot began with a prototyping process inspired by the design and fabrication notes found on the blog *Burari Web* [28], which documents the construction of a 4-legged walking robot using low-cost components and 3D printing techniques. This material provided us with a useful visual reference to define both the mechanical design and the functional structure of the model. Based on this initial idea, we proposed manufacturing robots using 3D printing, a process we carried out in our university laboratories, taking advantage of available resources and adapting the design to our technical needs.

To control the system, we selected a Raspberry Pi Pico W, initially programmed in MicroPython, due to its low power consumption, wireless connectivity, and ease of programming. Regarding the locomotion system, we used SG90 micro servomotors in each joint of the robot, given their availability, affordable price, and suitable dimensions for small-scale projects.

During the initial testing phases, we faced several technical challenges. One of the main ones was the frequent disconnect of the servo cables, caused by sudden movements and the mechanical stress generated by the robot's movement. Additionally, we detected voltage fluctuations and power supply inadequacies, which negatively affected the system's performance. To address these issues, we replaced the cables with heavier gauges, which allowed for more stable current conduction and improved the overall system response.

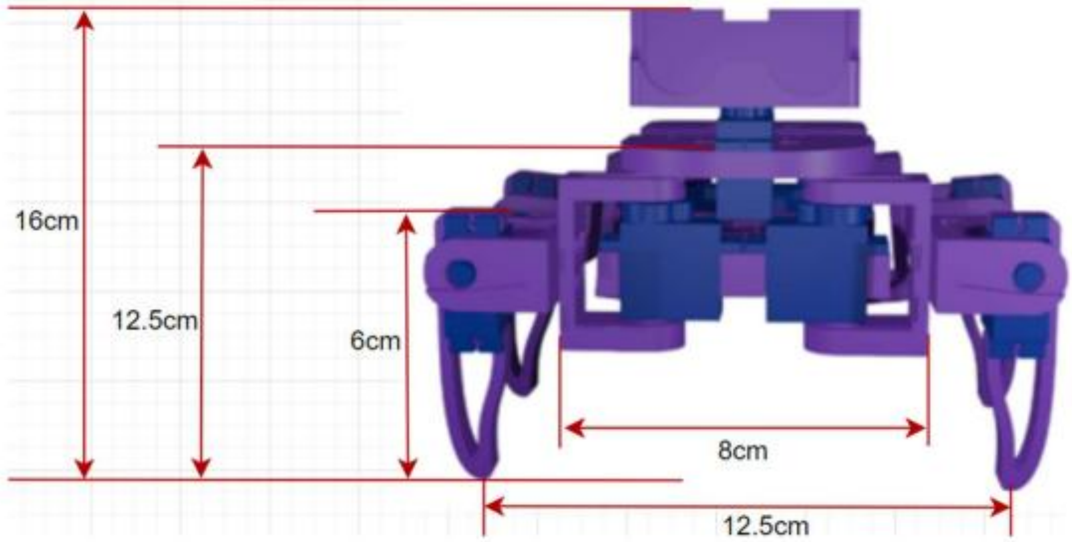
Based on these observations, we recommend incorporating a PCA9685 servo controller, which facilitated the simultaneous management of multiple motors and significantly improved control accuracy. This change also led us to miGreet the programming environment from MicroPython to CircuitPython, allowing us to write more modular, readable, and efficient code, thereby optimizing the robot's behavior.

This project is still in development and has become a comprehensive educational experience, spanning from the design and manufacturing phase to system programming and optimization. In addition to its educational value, the prototype offers a versatile platform for future testing on rough terrain, with potential applications in academic, research, or even low-cost robotic exploration scenarios.

5. Discussion

In terms of design, the implemented prototype, called Moradito, features a compact structure approximately 16 cm tall and 14 cm wide as seen in the Figure 7, with eight SG90 servomotors distributed across its four limbs, providing two degrees of freedom per leg.

Figure 7: Robot measurements.

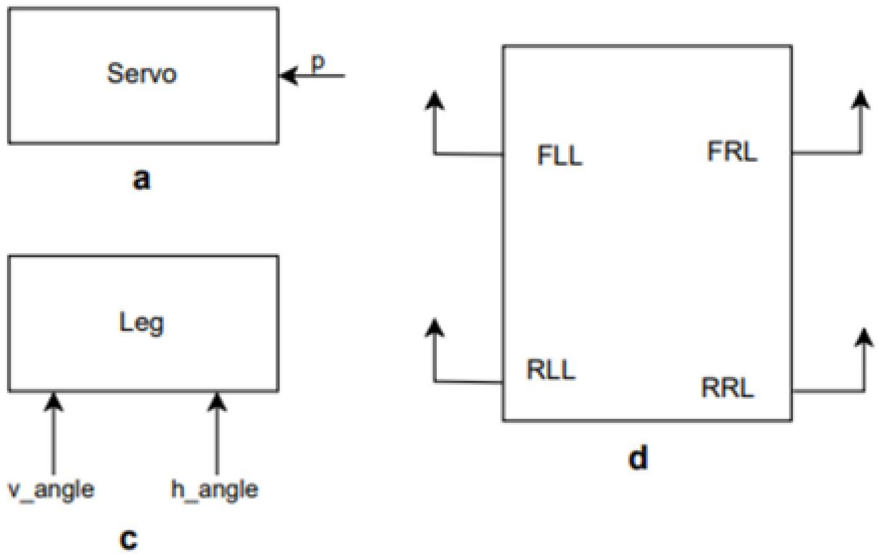


Source: own.

This configuration allows it to perform basic movements such as moving forward, backward, turning sideways, and performing preprogrammed sequences such as "Dance" or "Greet" During experimental testing, several technical successes were identified: the robot proved functional under remote control via a web interface developed in HTML, exhibited stable gait patterns on flat surfaces, and achieved effective displacements with speeds of up to 8.19 cm/s. Additionally, the object-oriented programming-based [29] control system enabled a modular and scalable architecture.

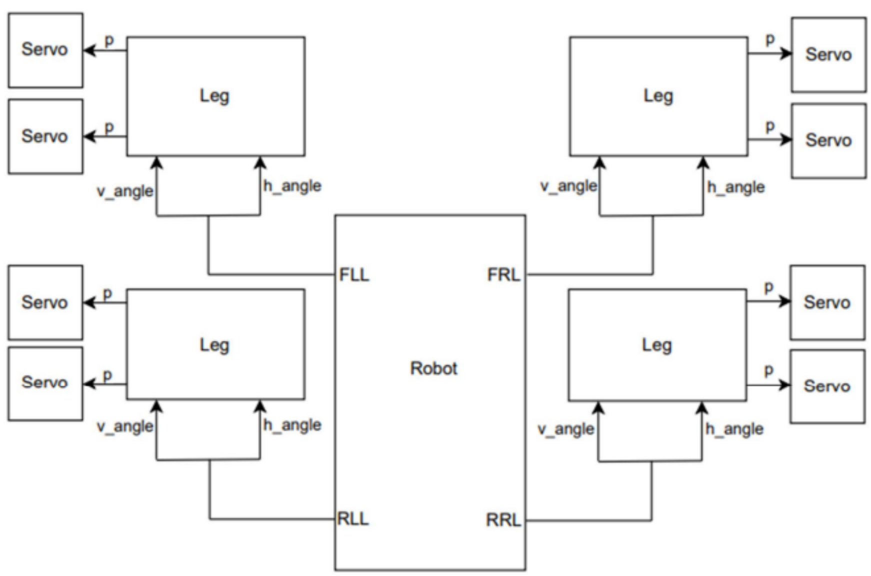
However, significant limitations were also present. A noticeable tilt was detected during forward and backward movements, which compromises stability on more demanding terrain. Furthermore, the SG90 motors exhibited torque limitations, particularly during prolonged movements or under additional load, as in the case of the “greet”. The absence of proximity or navigation sensors also restricts the potential for autonomy.

Figure 8: Blocks representing the CircuitPython modules.



Source: own.

Figure 9: Instantiated classes and modules.



Source: own.

The results obtained suggest that this type of platform may be viable for real-world agricultural environments, provided that aspects such as actuator torque, sensor integration (LIDAR), and an improvement in the gait algorithm are optimized.

Figure 8 illustrates a set of blocks. Each block represents a Circuitpython Module, in accordance the block denominated as servo (Figure 8a) represents a fundamental module that was conceived to control the position for just one servo, this module contains a class which we denominated as Servo, Servo corresponds to a fundamental class which is used just to position a servo according to the angle in which it is required, consequently, this class has a method for that purpose which in our code we denominated as setServo(p) in which the parameter p corresponds to the angle in which a servo has to be positioned. According to the block diagram then the angle parameter is informed externally to its corresponding block in this case from another Circuitpython module. Referring to the blocks in Figure 8, Figure 7b shows a block known as Leg, which has a class named Leg, which executes two methods we referred to as h_servo(h_angle) and v_servo(v_angle) respectively. The mnemonics h_servo and v_servo indicate the vertical and horizontal servos at each robot leg, whereas the v_angle and h_angle indicate the desired angles at which the servos should be positioned.

Figure 10: Instance class Robot.

```python
class Robot:
    def __init__(self):
        self.patas = {
            'ad_d': Pata(servo0, servo1),  # Delantera derecha
            'ad_i': Pata(servo2, servo3),  # Delantera izquierda
            'at_d': Pata(servo4, servo5),  # Trasera derecha
            'at_i': Pata(servo6, servo7),  # Trasera izquierda
        }

    def mover_pata(self, pata, h_angle, v_angle):
        if pata in self.patas:
            self.patas[pata].mover(h_angle, v_angle)
```

Source: own.

Figure 11: Instance class Pata.

```python
class Pata:
    def __init__(self, servo_horizontal, servo_vertical):
        self.servo_horizontal = servo_horizontal
        self.servo_vertical = servo_vertical

    def mover(self, h_angle, v_angle):
        self.servo_horizontal.angle = h_angle
        self.servo_vertical.angle = v_angle
        sleep(0.01)
```

Source: own.

Figure 10 and Figure 11 illustrate the internal software structure implemented in CircuitPython to control the quadruped robot's movements.

Figure 10 presents the implementation of the Robot class. This class serves as a controller that manages the movement of each of the robot's four legs. Each leg is identified by a unique key: 'ad_d' (Front Right Leg), 'ad_i' (Front Left Leg), 'at_d' (Rear Right Leg), and 'at_i' (Rear Left Leg). These keys are associated with instances of the Pata class (Spanish for “leg”), each receiving two servos (horizontal and vertical) as parameters. The method mover_pata() receives a leg identifier and target angles for both axes and delegates the command to the appropriate leg instance.

Figure 11 shows the implementation of the Pata class. This class encapsulates control of the two servos associated with one leg. The method mover() accepts horizontal (h_angle) and vertical (v_angle) angles, directly updating each servo’s position. A small delay (sleep(0.01)) ensures smooth and stable movement between sequential commands.

These two classes represent the core logic behind coordinated leg control in the robot and abstract away the lower-level servo control. By modularizing each leg as an independent instance, the Robot class can manage complex walking gaits and motion patterns in a scalable and structured manner.

Figure 8 depicts vertical and horizontal servos. In Figure 8c, you can see an illustration of the robot block. A class named Robot is included in this module. This class executes methods that reuse the methods from the class Leg in order to manage the movements of the four prototype legs. Two methods are implemented in this class in order to manage each leg of the robot. All legs in the class Robot are designated by the following mnemonics: FLL (Frontal Left Leg), RLL (Rear Left Leg), FRL (Frontal Right Leg), RRL (Rear Right Leg), correspondingly from the Robot class, the Leg class is instantiated four times as shown in Figure 9, as well as the parameters h_angle and v_angle. According to Figure 9, the class Servo is instantiated eight times.

5.1. **Quantitative analysis**

From the collected data, it is observed that the "Greet" movement presents the highest energy consumption (2.92 mAh), which is consistent with its long duration (16.5 s) and the active use of the neck. In contrast, the "Dance" movement presents a high instantly consumption (800 mA) in a short interval (0.69 s), indicating a high instantaneous current requirement.

In terms of speed, the backward movement ("Backward") reaches 8.19 cm/s, exceeding the forward movement, which is 7.08 cm/s. This difference may be related to the support pattern and retraction force, as observed in the diagrams (Figure 1 to Figure 6).

Finally, it is notable that the lateral movements (Right and Left) show low speed but high visual stability, which could be beneficial in precision maneuvers. This is reinforced by the gait efficiency data (Table 3), where these movements require less current (460–470 mA) and present stable patterns.

6. Conclusions

The development of a quadruped mobile robot prototype, as presented in this work, yielded positive results regarding its ability to move across the required type of simulated terrains. Through the implementation of a simple locomotion system based on servomotors programmed in CircuitPython, it was possible to perform coordinated “trotting” movements in each of its legs, ensuring both stability and continuous motion on uneven surfaces. This validates the use of such a system in environments where wheeled and differential drive systems face insurmountable limitations.

One of the key contributions of this project is its low cost, as it inteGreets inexpensive electronic components, 3D printing, and a modular, replicable control architecture. This makes it an attractive solution for the agricultural sector, particularly in rural environments with limited resources for implementing advanced automation.

Unlike high-budget solutions intended for use in factories, this prototype aims to address practical needs such as inspection, monitoring, and support for light logistics, without requiring complex or costly infrastructure.

To achieve simple, functional, and time-based servo activation sequences, the robot's design included both mechanical modeling and the implementation of the locomotion system from scratch. Although features such as autonomous navigation and energy autonomy have not yet been addressed, the system provides a solid foundation for future developments.

In this context, several opportunities for expanding the project have been identified, including the integration of environmental perception sensors and the optimization of the power system through solar panels or higher-capacity batteries. A suggested next step would be to validate the robot’s behavior in real field scenarios—such as mud, slopes, vegetation, or unpredictable obstacles—as its performance under actual agricultural conditions has not yet been investigated.

7. References

[1] A. O. Hourani y M. Z. Iskandarani, “Design, Modelling, and Analysis of Legged Robot for Terrains Exploration”, Int. J. Adv. Sci., Eng. Inf. Technol., vol. 13, n. º 3, pp. 1127–1128, junio de 2023. https://doi.org/10.18517/ijaseit.13.3.19000

[2] S. Bazeille et al., “Quadruped robot trotting over irregular terrain assisted by stereo-vision”, Intell. Service Robot., vol. 7, n. º 2, pp. 67–69, marzo de 2014. https://doi.org/10.1007/s11370-014-0147-9

[3] A. O. Baturone, Robótica: Manipuladores y Robots Móviles. Marcombo, 2005.

[4] X. Gao et al., "Review of Wheeled Mobile Robots’ Navigation Problems and Application Prospects in Agriculture," in IEEE Access, vol. 6, pp. 49248-49268, 2018, doi: 10.1109/ACCESS.2018.2868848.

[5] Robotics at Home with Raspberry Pi Pico: Build autonomous robots with the versatile low-cost Raspberry Pi Pico controller and Python. Packt Publ., 2023.

[6] N. Pudchuen, C. Deelertpaiboon, W. Jitviriya and A. Phunopas, "VENRiR: Vision Enhance for Navigating 4-legged Robot in Rough Terrain," 2020 59th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), Chiang Mai, Thailand, 2020, pp. 1410-1415, doi: 10.23919/SICE48898.2020.9240424.

[7] C. E. López Rodríguez and Y. L. Vargas Castiblanco, “Importancia de los procesos de automatización en el sector agrícola colombiano,” ID EST – Revista Investigación, Desarrollo, Educación, Servicio y Trabajo, vol. 4, no. 2, p. 4, 2024. https://revista.fundes.edu.co/index.php/revista/article/view/264.

[8] Desarrollo de la agricultura colombiana. Fedesarrollo, 2014.

[9] Bellicoso, C. D., Bjelonic, M., Wellhausen, L., Holtmann, K., Günther, F., Tranzatto, M., ... & Hutter, M. (2018). Advances in real-world applications for legged robots. Journal of Field Robotics, 35(8), 1311-1326.

[10] C. Quail, E. Emonot–de Carolis and F. Auat Cheein, "Legged Robots in the Agricultural Context: Analysing Their Traverse Capabilities and Performance," IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society, Singapore, Singapore, 2023, pp. 01-07, doi: 10.1109/IECON51785.2023.10312233.

[11] “Development of a quadruped robot platform for optimizing wheat and corn field for phenotyping”. University of Minnesota driven to discover. https://conservancy.umn.edu/items/69f6a205-2789-4095-8c6d-df2e227bf462.com

[12] M. F. Silva y J. A. Tenreiro Machado, “A Historical Perspective of Legged Robots”, J. Vib. Control, vol. 13, n. º 9-10, pp. 1447–1455, septiembre de 2007. https://doi.org/10.1177/1077546307078276

[13] Q. Li, F. Cicirelli, A. Vinci, A. Guerrieri, W. Qi y G. Fortino, “Quadruped Robots: Bridging Mechanical Design, Control, and Applications”, Robotics, vol. 14, n. º 5, pp. 5–8, abril de 2025. https://doi.org/10.3390/robotics14050057

[14] M. Aguilera Hernández, M. Bautista y J. Iruegas, “Diseño y Control de Robots Móviles”, Asoc. Mex. Mecatronica AC., pp. 1–6, 2003.

[15] J. M. Robles Atuesta. “Diseño y prototipado del mecanismo de locomoción para un robot cuadrúpedo”. Universidad de los Andes Colombia. https://hdl.handle.net/1992/45035

[16] J. Fu y F. Gao, “Dynamic stability analyzes for a parallel–serial legged quadruped robot”, Int. J. Adv. Robotic Syst., vol. 19, n. º 5, pp. 3–5, septiembre de 2022. https://doi.org/10.1177/17298806221132081

[17] J. Li, J. Wang, S. X. Yang, K. Zhou y H. Tang, “Gait Planning and Stability Control of a Quadruped Robot”, Comput. Intell. Neurosci., vol. 2016, pp. 1–13, 2016. https://doi.org/10.1155/2016/9853070

[18] Q. Cong et al., “Stability Study and Simulation of Quadruped Robots with Variable Parameters”, Appl. Bionics Biomechanics, vol. 2022, pp. 1–9, enero de 2022. https://doi.org/10.1155/2022/9968042

[19] P.-B. Wieber, R. Tedrake y S. Kuindersma, “Modeling and Control of Legged Robots,” en Springer Handbook of Robotics, 2ª ed., B. Siciliano y O. Khatib, eds., Springer, 2016, pp. 1213–1223. https://doi.org/10.1007/978-3-319-32552-1_48.

[20] R. Zwetsloot. “Control servos with CircuitPython and Raspberry Pi”. Raspberry Pi Official Magazine. https://magazine.raspberrypi.com/articles/control-servos-circuitpython-raspberry-pi?.com

[21] C. Bell, Beginning MicroPython with the Raspberry Pi Pico. Berkeley, CA: Apress, 2022. https://doi.org/10.1007/978-1-4842-8135-2

[22] Automacion, “Servomotores: control, precisión y velocidad”, AADECA, vol. 4, p. 2, 2017.

[23] J. Kim, T. Kang, D. Song y S.-J. Yi, “Design and Control of a Open-Source, Low Cost, 3D Printed Dynamic Quadruped Robot”, Appl. Sci., vol. 11, n. º 9, p. 3762, abril de 2021. Accedido el 15 de julio de 2025. https://doi.org/10.3390/app11093762

[24] E. Garcia, P. González-de-Santos y J. Estremera, Quadrupedal Locomotion: An Introduction to the Control of Four-legged Robots. Springer, 2006.

[25] Z. Gacovski, Ed., Mobile Robots - Current Trends. InTech, 2011. https://doi.org/10.5772/2305

[26] G. Raghavendra, B. B. V. L. Deepak y M. Gupta, Eds., Recent Advances in Mechanical Engineering, Volume 1. Singapore: Springer Nature Singap., 2024. https://doi.org/10.1007/978-981-97-0918-2

[27] Z. Y. Du y B. Liu, Advanced Mechanical Engineering. Trans Tech Publ. LTD, 2010. https://doi.org/10.4028/b-v26glh

[28] 東京バード. (2021, 12 de julio). *Arduinoで動かす4足歩行ロボット製作ノート！Arduino学習に便利なロボくんなので使って下さい！【STLデータ公開】*. ぶらり＠web走り書き. https://burariweb.info/electronic-work/4legged-waking-robot-production-notes.html

[29] M. Stefik and D. G. Bobrow, “Object-Oriented Programming: Themes and Variations”, *AIMag*, vol. 6, no. 4, p. 40, Dec. 1985. https://doi.org/10.1609/aimag.v6i4.508

8. Annexes

You can find the annexes in the attached document.