www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 17(1), Jan-April 2026, 400-410. ISSN: 2007-1558.
https://doi.org/10.61467/2007.1558.2026.v17i1.1159

Frugal mechatronic design of a mobile self-balancing robot

Erick Axel Padilla-García[1], *Raúl Dalí Cruz-Morales*[2], *Jaime González-Sierra*[3], *David Tinoco-Varela*[2]
[1] Tecnológico Nacional de México / Tecnológico de Estudios Superiores de Ecatepec (TESE). Av. Tecnológico, Valle de Anáhuac, Ecatepec 55210, Estado de México, México
[2] Universidad Nacional Autónoma de México (UNAM), Facultad de Estudios Superiores Cuautitlán (FESC), Departamento de Ingeniería. Cuautitlán Izcalli 54714, México.
[3] UPIIH, Instituto Politécnico Nacional, Carretera Pachuca—Actopan Kilómetro 1+500, Distrito de Educación, Salud, Ciencia, Tecnología e Innovación, San Agustín Tlaxiaca 42162, México.
E-mails erickaxelpadillagarcia@gmail.com , rdcruz@comunidad.unam.mx , jagonzalezsi@ipn.mx y dativa19@hotmail.com

Abstract. Mobile robots are commonly used for tasks such as monitoring, mapping, and transportation. Among them, the self-balancing robot stands out, as it is capable of maintaining balance and adapting to different slopes, which makes it particularly suitable for material distribution. However, commercial robots are often costly, which limits their practical implementation. Rapid Manufacturing (RM), through additive manufacturing (AM), offers a potential approach by enabling low-cost and reproducible designs. This approach is characteristic of frugal innovation, allowing the robot's structure and component selection to be scalable and low-cost, thereby supporting more efficient use of available resources. This manuscript describes a frugal mechatronic design for a self-balancing mobile robot with a 5 kg payload, based on a dynamic mechatronic model. A feasible solution was obtained through CAD development, appropriate component selection, and instrumentation for the synthesis of the physical configuration, while dynamic model simulations were used to parameterise inertial parameters. The resulting design is presented as a reproducible and affordable proposal.
Keywords: Mechatronic design, frugal innovation, mobile robots, self-balancing robots, rapid manufacturing.

Article Info
Received June 26, 2025
Accepted Aug 26, 2025

1 Introduction

The development of mechatronic systems requires a strategy that enables the combined integration of various engineering disciplines, including mechanical engineering, electronics, computing, and control engineering, and that defines a specific artifact (Hernández, 2018). These strategies have served as a methodological reference for design, enabling the development of feasible prototyping solutions for complex, innovative projects (Bartalucci, 2023) and the design of robotic systems (Morales, 2021).

These mechatronic design strategies are commonly three: sequential, iterative, and concurrent strategies, which have been widely used in the scientific community for the design of new systems (Zheng, 2023), and by using them, it is possible to consider all stages of the configuration process as an interdisciplinary design to achieve a specific system operation (Herrera, 2024). A design approach that facilitates the creation of economical, accessible, scalable, and flexible products and systems using limited resources is known as frugal design (Trejo, 2024). This approach has been referenced as a new paradigm in product creation and manufacturing, enabling the development of innovative solutions that are simultaneously sustainable, cost-effective, and environmentally friendly (Santiago, 2024).

Taking advantage of new tools such as artificial intelligence, open-source software, Computer Aided-Design (CAD), Computer Aided-Analysis (CAE), and Computer Aided-Manufacturing (CAM), it is possible to satisfy the needed balance of a specific design and operation with complex functioning systems, such as mobile robots, while selecting and proposing available materials (Lira Hernández, 2022).

The self-balancing robot is one of the most versatile mobile robots due to its numerous applications, particularly in logistics and warehousing, as it is capable of autonomously and quickly transporting products and materials within factories and distribution centers. (Velandia, 2023). This mobile robot is an interesting system, particularly for the scientific community, which looks for solutions for stabilization and control strategies to achieve a more balanced operation of the robot. Self-balancing can maintain its equilibrium on the support point and adapt to various road and slope conditions, whether transporting loads or distributing materials on a production line. However, commercial models are usually expensive, which limits their availability for academic projects or initiatives with limited budgets. In such cases, the need arises for new robot designs of this type, focusing on frugal innovation — scalable, low-cost, and accessible prototypes that utilize materials available in the environment. (Trujillo, 2025).

This work presents a methodological proposal for designing mechatronic systems using a multidisciplinary approach, incorporating CAD, CAM, RM, AM, and artificial intelligence tools to address one or more design stages. An application case study is then presented in the combined design process to achieve the frugal innovation proposal of a mobile Self-balancing robot.

The structure of this article is organized as follows: Section 2 presents the methodological strategy for designing mechatronic systems based on the mathematical model that allows for defining a specific configuration. Section 3 presents a design case study, showing the CAD design obtained by solving the dynamic model equations. Section 4 presents reflections and discussions on the advantages, disadvantages, and costs of the developed mechatronic system, and Section 5 concludes with the key findings of the presented proposal.

2 Frugal Mechatronic Design Methodology

The concept of "design" can be defined as the process of configuring a system within a field of study (Díaz, 2023). In this configuration process, decision variables are identified and established, enabling a specific solution to be identified within a model. Design parameters, on the other hand, are those values that cannot be changed but are necessary in the configuration process, such as the value of gravity, inertial parameters of a material, among others. Thus, from a more formal perspective, we can identify, through mathematical models (Morales, 2021), the variables in combination that enable us to determine a specific design within a configuration process, which is expressed as:

$$y_D = f(\,P_D\,;\,V_D) \tag{1}$$

Here y_D is the mathematical expression that gives synthesis to a specific physical configuration. For this, the requires the variables V_D and parameters P_D to achieve a feasible solution, that is, set parameters to our different parts design parts (mechanical, electrical, software, among others), which sizes the kinematics of the mechanism, subject to dimensional constraints of feasible ranges, which results in the structural mechanical design of the Self-balancing robot. Additionally, during the instrumentation process, it is necessary to select affordable sensors, components, and actuators based on factors such as weight, size, price, and electrical operating ranges, among others. One way to obtain the dimensional design is through CAD, as it enables us to define dimensions. Simulation of the dynamics mathematical model can get the inertial parameters of the system. Later, the final proposed design can be analyzed via CAE and manufactured via CAM. In this proposal, SolidWorks was used for structural analysis and the instrumentation process. The inertial parameters, such as centers of mass, weights, and other relevant parameters, were obtained using the dynamic model simulation of the Self-balancing robot. This dynamic model can be expressed by using differential equations of the type:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(\boldsymbol{t}))\boldsymbol{x}(\boldsymbol{t}) + \boldsymbol{g}(\boldsymbol{x}(\boldsymbol{t}))\boldsymbol{u}(\boldsymbol{t}) \tag{2}$$

Where $\boldsymbol{x}$ represents those state variables of the system that determine its behavior, $\boldsymbol{f}(\boldsymbol{x})$ is the process function of the differential equation model, and $\boldsymbol{g}(\boldsymbol{x})$ is the function that depends on the inputs $\boldsymbol{u}(\boldsymbol{t})$. Given a system input, a behavior is obtained using the expression $h(x(t))$ that defines the response and the system under analysis. The reader can find more details about the dynamical systems model in Zill (2022). This proposal utilized the dynamic model of a balancing robot, which, through the assignment of control and design variables, was able to achieve the desired balancing behaviors and subsequently provide a specific solution for them.

Both synthesis and analysis models (see Fig. 1) enable the creation of a feasible design that can be built as a physical configuration, as well as to analyze the system's performance (Durango, 2018; Pedraza, 2017).

A methodology widely used in the scientific community for the design of mechatronic systems is through the use of a mechatronic design strategy, which can be sequential or iterative, with which we can ensure an integrated design of the mechanical, electrical, and control proposals in an integrated manner (Cervantes, 2021), as can be seen in Fig. 2.

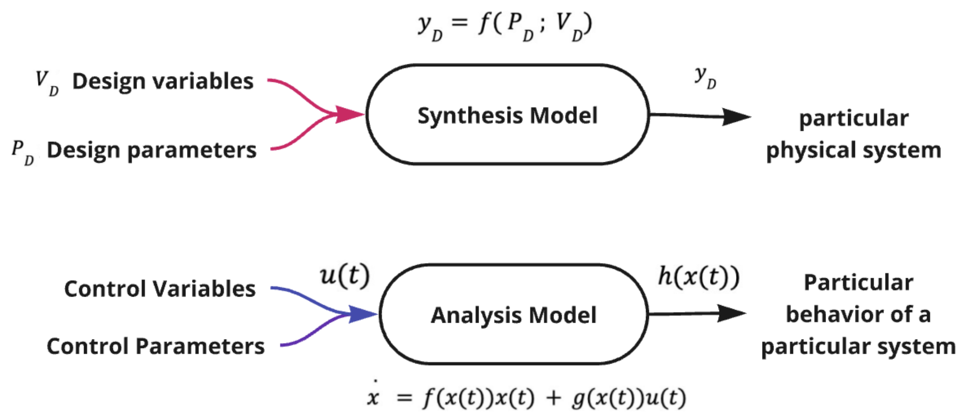

Fig. 1. Model-based systems design.

In a sequential mechatronic model, both synthesis and analysis models are combined. One specifies the structural design or physical configuration of the system, while the other allows the evaluation of operational performance. Being a multidisciplinary design problem, traditionally, the structure or synthesis of the prototype to be developed is defined first (considering elements with a frugal approach). The behavior of the combined system is then evaluated (Abrego Preza, 2021).

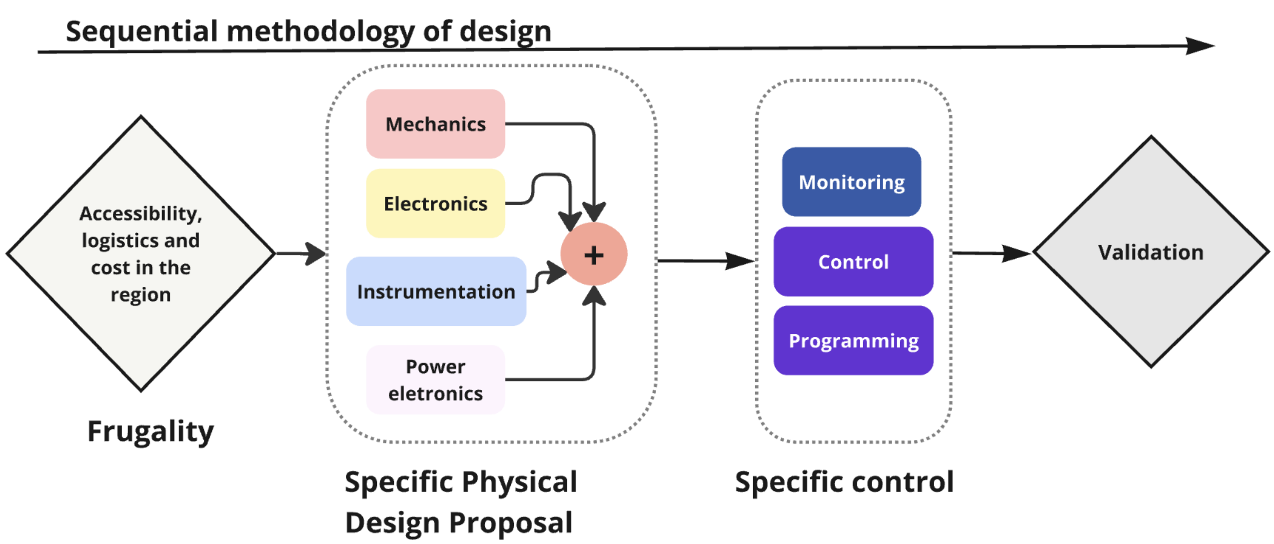

Fig. 2. Diagram of design methodology.

2.1 Case Study: Self-balancing Robot Design

This proposal employs traditional mechatronic design methodology to achieve the robot design; however, to validate the design's suitability, the system is modeled in a closed-loop configuration. Thus, it is possible to determine the appropriate parameters through model reconfiguration, as shown in Fig. 3.

This section presents the mathematical modeling of the system. The Euler-Lagrange formulation is used to derive the dynamic equations of the robotic system. For the generalized coordinates selected as $\boldsymbol{q} \in \{\boldsymbol{x}, \boldsymbol{\theta}\}$, where the chosen generalized coordinates x regard the wheels in motion, and θ consider the angular motion of platforms as an inverted pendulum of the system. Here,

$\boldsymbol{Q_q} \in \{\boldsymbol{Q_x = u, Q_\theta = 0}\}$ where $\boldsymbol{Q_x}$ represents the external force from the motor to move the robot, while $\boldsymbol{Q_\theta}$ is the passive joint of the inverted pendulum. Let us define the kinetic energy of the system as:

$$T = T_{car} + T_{pend.} = \frac{1}{2}M\dot{\boldsymbol{x}}^{\boldsymbol{2}} + \frac{1}{2}m(\dot{\boldsymbol{x}}_{\boldsymbol{p}}{}^{\boldsymbol{2}} + \dot{\boldsymbol{y}}_{\boldsymbol{p}}{}^{\boldsymbol{2}}) = \frac{1}{2}(M+m)\dot{\boldsymbol{x}}^{\boldsymbol{2}} + \boldsymbol{mL\dot{x}\dot{\theta}\cos\theta} + \frac{1}{2}mL^2\dot{\theta}^{\boldsymbol{2}} + \frac{1}{2}mI\dot{\theta}^{\boldsymbol{2}} \quad (3)$$

Where M is the wheel's mass, $\boldsymbol{m}$ is the mass of the pendulum, $\boldsymbol{L}$ is the pendulum's length, and I is the inertia given by the center of mass, and it is computed as $\boldsymbol{I} = \frac{1}{2}mL^2$ **.** The potential energy is given as:

$$V = V_{pend.} = mgL\cos\theta \quad (4)$$

With g the acceleration of gravity. Then, the Lagrangian is obtained by subtracting the kinetic energy (3) from the potential energy (4), as follows:

$$L_a(\boldsymbol{x}, \boldsymbol{\theta}, \dot{\boldsymbol{x}}, \dot{\boldsymbol{\theta}}) = \frac{1}{2}(M+m)\dot{\boldsymbol{x}}^{\boldsymbol{2}} + \boldsymbol{mL\dot{x}\dot{\theta}\cos\theta} + \frac{1}{2}(m\boldsymbol{L^2} + \boldsymbol{I})\dot{\boldsymbol{\theta}}^{\boldsymbol{2}} - \boldsymbol{mgL\cos\theta} \quad (5)$$

Thus, the Euler-Lagrange formulation $\frac{d}{dt}\frac{\partial L_a}{\partial \dot{q}} - \frac{\partial L_a}{\partial q} = Q_q$ is used to obtain the equations of a self-balancing robot (which behaves like an inverted pendulum on a wheel or cart), obtaining the following equations:

1. First equation (car motion):

$$(\boldsymbol{M} + \boldsymbol{m})\ddot{\boldsymbol{x}} + \boldsymbol{mL\ddot{\theta}\cos\theta} - \boldsymbol{mL\,\dot{\theta}^2\sin\theta} = \boldsymbol{u} \quad (6)$$

2. Second equation (rotation of inverted pendulum):

$$\boldsymbol{mL\ddot{\theta}\cos\theta} + (\boldsymbol{mL^2} + \boldsymbol{I})\ddot{\boldsymbol{\theta}} - \boldsymbol{m\,gL\,\sin\theta} = \boldsymbol{0} \quad (7)$$

Fig. 3. shows the side view and the coordinate system on which the mathematical model for the two-wheeled inverted pendulum is constructed. Furthermore, the system parameters are shown in Table 1.

Table 1. Self-balancing robot parameters

Parameter	Description	Unit
$\boldsymbol{g}$	Acceleration of gravity	m$/s^2$
$\boldsymbol{M}$	Wheel mass	kg
$\boldsymbol{R}$	Wheel radius	m
J	Moment of inertia of the wheels	kgm$/s^2$
$\boldsymbol{L}$	Length of pendulum	m
$\boldsymbol{m}$	Mass of the pendulum	kg

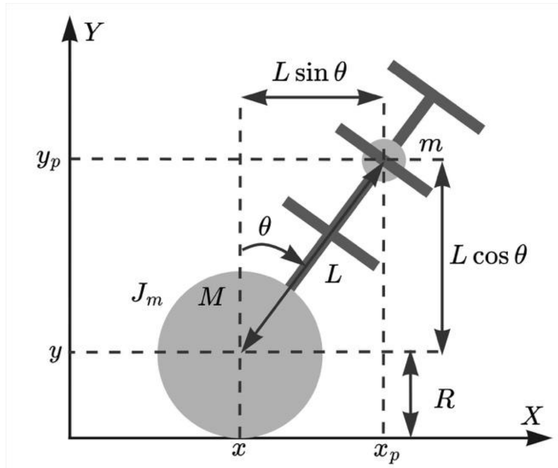

Fig. 3. Reference diagram of the self-balancing robot

Derived from the equations and parameters; to model the system and simulate its behavior, the following considerations are taken into account:

- $\boldsymbol{m}$=5 kg, being the desired mass of the load.
- $\boldsymbol{g}$=9.81 m/s^2, being the acceleration due to gravity.
- θ ≈ 0 corresponds to vertical equilibrium.
- For small oscillations around equilibrium (θ ≈ 0), the following approximation can be used: sin(θ) ≈θ, cos(θ) ≈1.
- The linearized model around the equilibrium point shows the behavior of the system.
- It is also considered to add viscous friction $b_x\dot{x}$ for the car and $b_\theta\dot{\theta}$ the pendulum.

Now, if we wish to describe the system in terms of state space, let $X = (x, \dot{x}, \theta, \dot{\theta})$, then the model becomes $\dot{X} = AX + Bu$, as

$$\dot{X} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{(mL^2+I)b_x}{D} & \frac{m^2gL^2}{D} & -\frac{mLb_\theta}{D} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{mLb_x}{D} & \frac{(M+m)mgL}{D} & -\frac{(M+m)b_\theta}{D} \end{bmatrix} X + \begin{bmatrix} 0 \\ \frac{mL^2+I}{D} \\ 0 \\ \frac{mL}{D} \end{bmatrix} u \tag{8}$$

Where $D = (M + m)(mL^2 + I) - (mL)^2$.

The robot's balance is achieved through a Proportional-Integral-Derivative (PID) controller control algorithm, which is the most widely used controller in the industry, specifically due to its real-time response capabilities (Borase, 2021). The PID controller is a feedback control system whose objective is to eliminate the error between the reference signal and the output signal as time approaches infinity. It also anticipates future behavior through a derivative action, which predicts the process output (Joseph, 2022).

Considering the frugal design approach based on the prototyping environment

● **Actuator:** Select an inexpensive DC motor or servomotor capable of generating the necessary torque.
● **Sensor:** Use an affordable encoder to measure θ near the equilibrium point.
● **Controller**: A PID may be sufficient to stabilize the system.

2.2 Programming

A pseudocode is presented in Fig. 4, which can be used for programming in any code to simulate the dynamic equations of the self-balanced robot. By using MATLAB, the code simulates the behavior of a self-balancing robot with a 5 kg load using a linearized state-space model. By using this code and assigning the appropriate dimensions in the CAD design of the proposed self-balancing robot, it is possible to define the physical parameters of the system, such as mass, gravity, friction, and inertia, along with the corresponding mathematical model. However, to run the simulation, it is necessary to establish the control inputs.

Algorithm 1 PID Control for Self-Balancing Robot with Friction

Initialize physical parameters: M, m, L, I, g
Initialize friction coefficients: b_x, b_θ
Compute common denominator D for dynamic equations
Define positive PID gains: K_p, K_i, K_d
Initialize initial conditions: x, $\dot{x}$, θ, $\dot{\theta}$
Initialize PID variables: $integral = 0$, $prev\_error = 0$
Define simulation time t_{end} and time step dt
Create vectors for storing results
for $t = 0$ **to** t_{end} **step** dt **do**
    Compute error: $error = \theta$
    Update integral term: $integral = integral + error \times dt$
    Compute derivative term: $derivative = \frac{error - prev\_error}{dt}$
    Update previous error: $prev\_error = error$
    Compute PID output without sign:

$$u_{pid} = K_p \times error + K_i \times integral + K_d \times derivative$$

    Apply negative sign for control input: $u = -u_{pid}$
    Compute accelerations (linearized model with friction):

$$\begin{cases} \ddot{\theta} = \dfrac{(M+m)(mgL\theta - b_\theta\dot{\theta}) - mL(u - b_x\dot{x})}{D} \\ \ddot{x} = \dfrac{(mL^2+I)(u - b_x\dot{x}) - mL(mgL\theta - b_\theta\dot{\theta})}{D} \end{cases}$$

    Update velocities and positions using Euler integration:

$$\begin{cases} \dot{x} = \dot{x} + \ddot{x} \times dt \\ x = x + \dot{x} \times dt \\ \dot{\theta} = \dot{\theta} + \ddot{\theta} \times dt \\ \theta = \theta + \dot{\theta} \times dt \end{cases}$$

    Store states and control input for analysis
end for
Plot results and optionally animate the system

Fig. 4. Pseudocode for programming the dynamics equation of the self-balancing robot.

3. Results

3.1 Synthesis and instrumentation results

After the heuristic process of component and material selection, and simulation, the CAD result shown in Figure 8 is obtained, which proposes the following:

- **Structure and support:** The base and chassis are made of plywood, 3D printed PLA, acrylic, or lightweight metal, allowing for good robot balance. The dimensions are 200x100 mm. M3/M4 screws are proposed for assembling 3D-printed joints and spacers between the robot bases.

- **Mechanical components:** Rubber wheels with a diameter of 60-80 mm. Castor-type wheels for balancing the robot, and general cables and switches.

- **Actuators:** DC motors with plastic gearboxes, with an operating voltage of 6 to 12 V. An L298N driver and an Arduino UNO or NANO microcontroller are proposed. - Sensors: An MPU6050 accelerometer/gyroscope would be used to measure the actual reference.

- Power: 18650 rechargeable batteries with a DC-DC converter for a 5V bank.

To achieve this, a PID controller is employed using the pseudocode proposed in MATLAB, which yields the obtained results for testing, modifying the CAD dimensions, and validating the robot's dynamic response. Using CAD and analyzing the dynamic model, the system state is iteratively updated, recording the following values for the system parameters: $\boldsymbol{m = 5\ kg}$, $\boldsymbol{M = 1\ kg}$, $\boldsymbol{L = 0.5\ m}$, $\boldsymbol{I = 0.006 \frac{kgm}{s^2}}$, $\boldsymbol{b_x = 0.1 \frac{kg\ m^2}{s}}$ y $\boldsymbol{b_\theta = 0.1 \frac{kg\ m^2}{s}}$. Furthermore, the gains for the PID controller were adjusted as $K_p = 100$ $K_d = 1$ $K_i = 20$.

3.1 Test 1: Vertical equilibrium point

Since the system starts at the desired equilibrium point, and there is no disturbance or initial error, the error is zero, so the control input no force is applied to the wheels, then, no torque and no acceleration is produced, which means that the simulation in the equilibrium point is stable and can maintain a payload of **5** kg in the body of the robot. The dynamic responses of the wheels, pendulum, and control input are shown in Fig. 5.

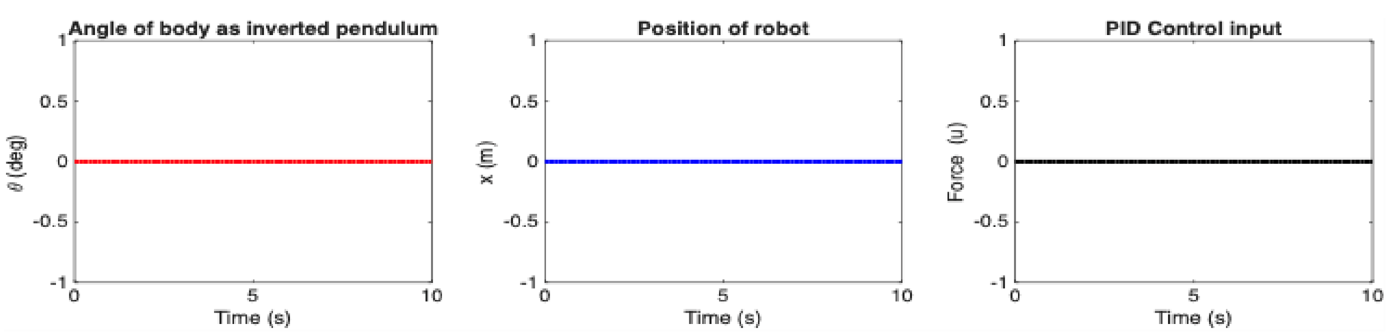

Fig. 5. Balance of the self-balancing robot in the vertical equilibrium point.

3.2 Test 2: Balance with non-zero initial angle of the pendulum

In this test, the pendulum's angle is initialized at 10 degrees with respect to the robot's vertical position. It can be seen from Fig. 6 that the system is balanced, but any minor disturbance would cause it to start falling. The PID controller adjusts the input signal; if a slight deviation occurs, the controller responds and attempts to correct it. However, the simulation achieves perfect precision and eliminates noisy motion from the system.

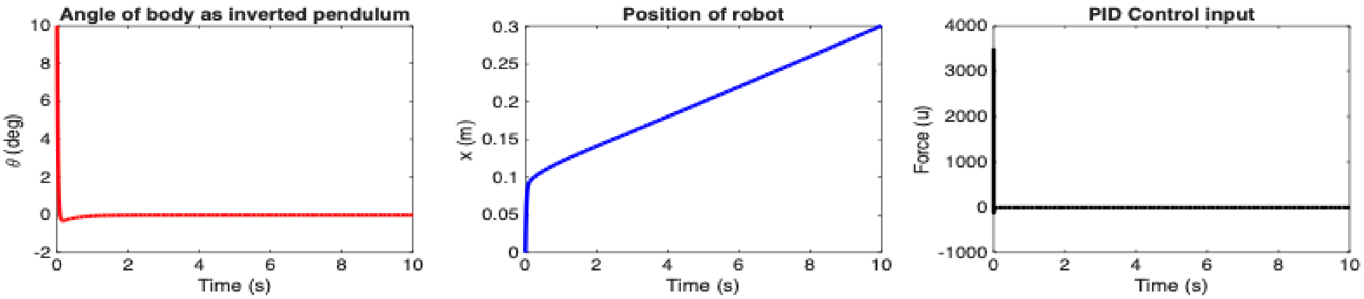

Fig. 6. Balance of the self-balancing robot in the vertical equilibrium point.

3.3 Test 3: Balance with a non-zero initial angle of the pendulum and follow a sinusoidal trajectory

In this test, the complete robot should follow a desired sinusoidal path given by $x_d(t) = A \ sin \ (2\pi f t)$ with A=0.3 m and f=0.2 Hz. Then, the pendulum controller's PID should be modified and tuned to $K_p = 500$ $K_d = 10$, and $K_i = 200$ the pendulum's angle should be balanced, disregarding the robot's motion on the wheels. Notice that it is about ten times more energy. Moreover, to move the robot to a desired path, another controller should be added for the wheels, with gains $K_{px} = 20$ $K_{dx} = 100$and $K_{ix} = 10$. Thus, the control input can be stated as $u = u_p + u_b$ being composed of two controllers, one to balance the pendulum u_p and the other to take the robot to a desired point or path u_b. Therefore, the robot not only has to balance its body, but it also follows a desired path. The dynamic response can be seen in Fig. 7.

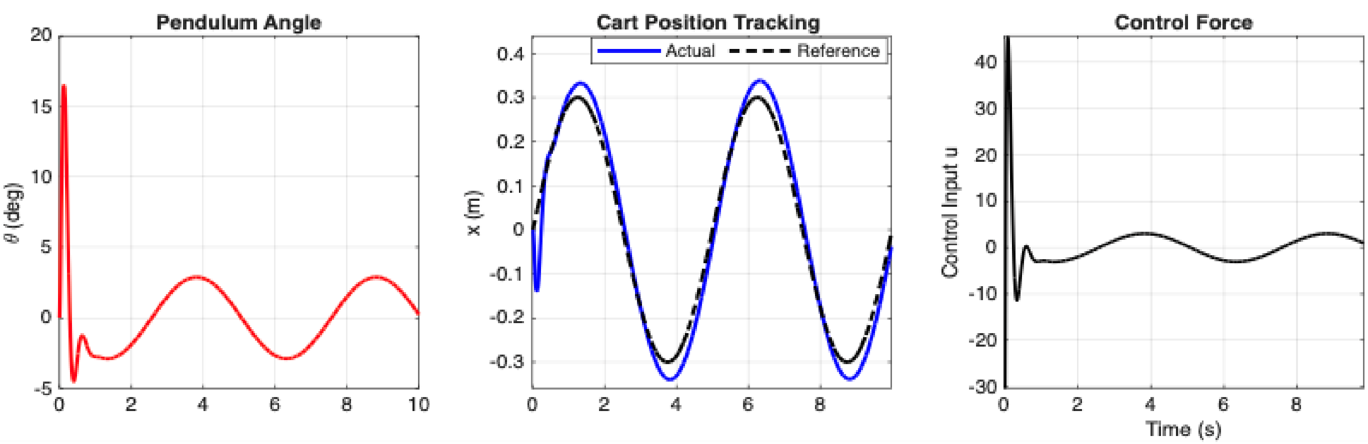

Fig. 7. Balance of pendulum and tracking to a desired path of the self-balancing robot

Now, it is clear that we have found a set of inertial parameters for the robot that validates the payload can be moved along a desired path, with the designed values for mass, length, and other parameters.

3.4 CAD results

First, the main base was created, where the motors would be fitted, as well as the components that included the battery and drivers.

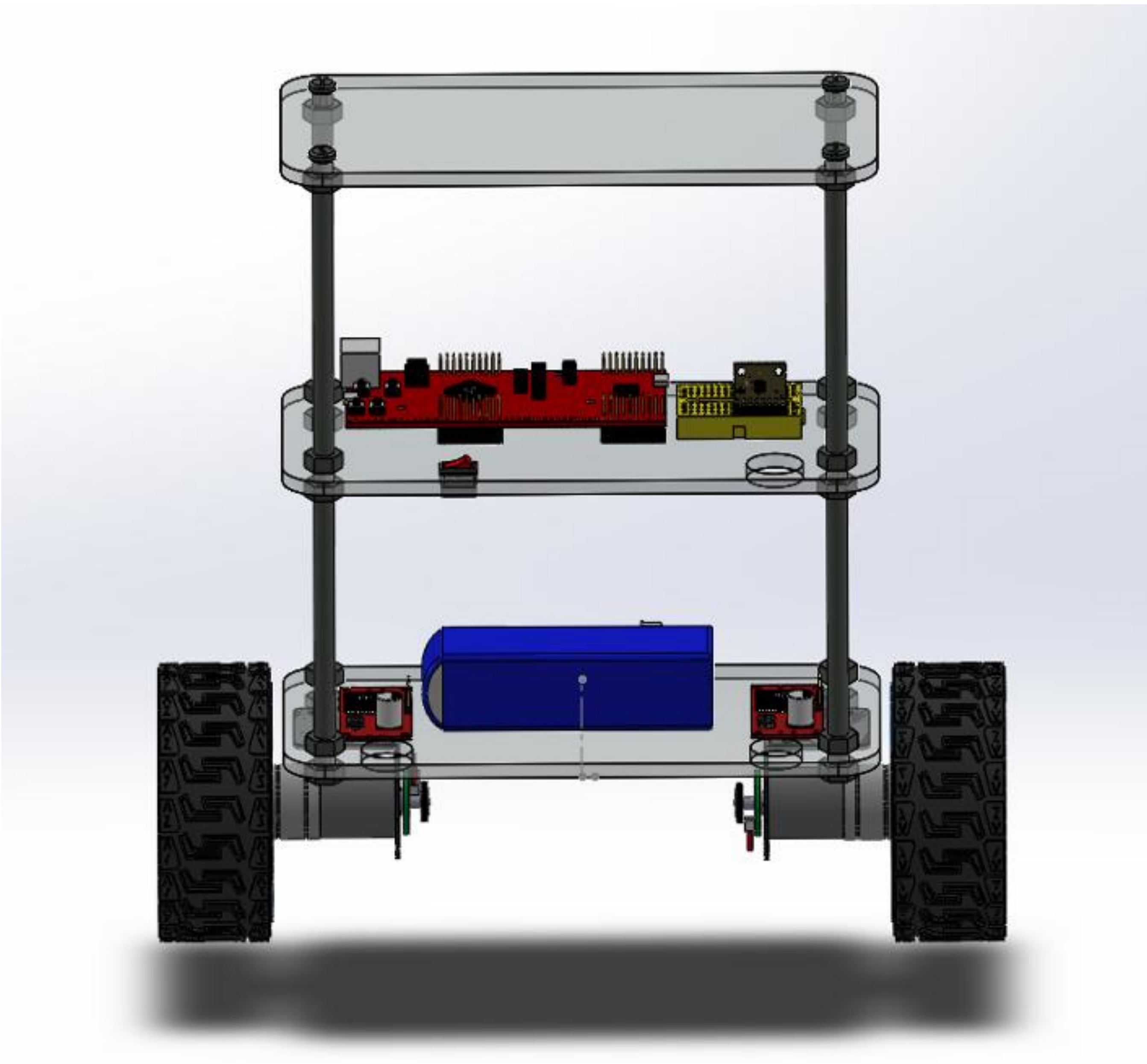

Fig. 8. Proposed design of the self-balancing robot.

The second base was created to accommodate components such as the microcontroller, the robot's activation button, and the MPU6050 accelerometer, as shown in Figure 6. Finally, the base was created to support the robot's load, with a capacity of 5 kg, without compromising the robot's automatic balance in a vertical position.

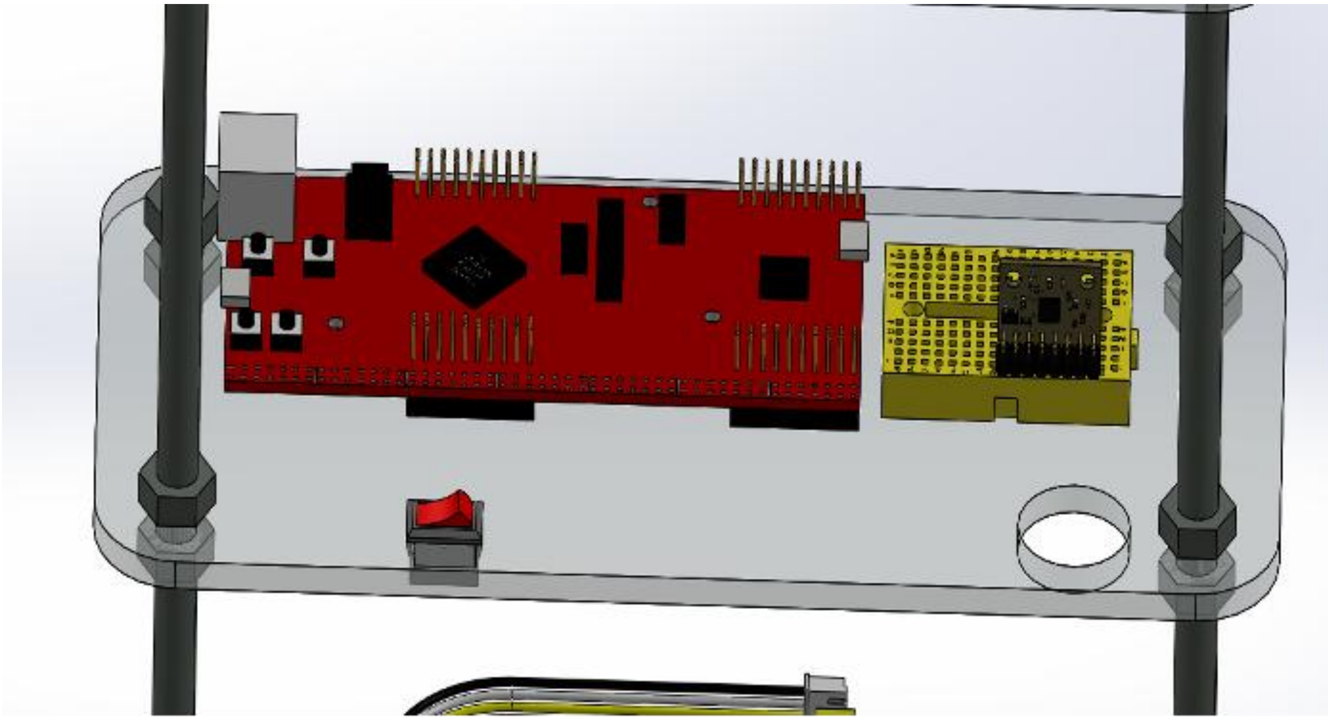

Fig. 9. Second base of the self-balancing robot

4 Discussions

The design of the self-balancing robot with a 5 kg payload partially meets the established objectives. Regarding its load capacity, tests show that it can support up to 5 kg without compromising stability, maintaining balance even in the face of moderate disturbances. This is due to an adequate distribution of the center of mass and an efficient structural design, utilizing lightweight yet resilient materials, which also contributes to maintaining a low cost of approximately 2,500 Mexican pesos. This price may vary depending on location, local prices, delivery availability, and other logistical factors. In the specific case of the northern part of the State of Mexico, this cost is quite affordable compared to other robot kits available in local and online stores. Nevertheless, it may depend on the delivery time and location.

The control system, based on a PID algorithm, performs satisfactorily in controlled environments but exhibits limitations in dynamic situations, such as terrain inclinations or strong disturbances. The sensors used, primarily low-cost gyroscopes and accelerometers, offer adequate performance; however, their accuracy could be improved to enhance reliability during the balancing process. The motors and actuators, on the other hand, meet the power requirements, but their high energy consumption affects the robot's autonomy. From an economic perspective, the design achieves a significantly lower cost than commercial solutions, thanks to the use of accessible components and a simplified manufacturing process. However, this entails some trade-offs in terms of the durability of some components, which could limit the robot's lifespan.

5 Conclusions

A cost-effective solution was proposed for designing a self-balancing robot, utilizing a mechatronic design methodology through CAD and parameterizing the robot using a mathematical model. This approach allowed for the development of a viable, economical, and functional solution. The self-balancing robot, designed to support a 5 kg load, demonstrates the feasibility of creating accessible and efficient robotic systems for specific applications.
By optimizing resources and using standard components, an appropriate balance was achieved between cost, functionality, and performance. To achieve the appropriate parameters in the CAD design, specific configurations were employed in the robot modeling, which significantly reduced costs without compromising objectives.

This project opens the door to future improvements, such as the integration of artificial intelligence systems for autonomous decision-making and the development of additional capabilities, such as object manipulation and adaptation to various surfaces. Furthermore, the cost-effective approach makes this robot a viable solution for small and medium-sized businesses, educational institutions, or research laboratories.

Acknowledgements

The authors are grateful for the support of the Instituto Politécnico Nacional Project SIP: 20253806, UNAM through PAPIME PE104125, PAPIME PE107225, and CI 2446 projects. Also, the authors thank the support of Universidad Politécnica de Atlacomulco for the CAD design proposed by the Estadia projects of the University.

References

Bartalucci, L. S. (2023). An original mechatronic design of a kinaesthetic hand exoskeleton for virtual reality-based applications. *Mechatronics, 90*, 102947.

Borase, R. P. (2021). A review of PID control, tuning methods and applications. *International Journal of Dynamics and Control, 9*, 818–827.

Cervantes, C. H. (2021). Concurrent design of a 2 DoF five-bar parallel robot: A hybrid design of rigid and flexible links. *IEEE Access, 9*, 17450–17462.

Díaz, D. J. (2023). Diseño, implementación, simulación y control de un brazo robótico para el Rover UVG [unpublished].

Durango, I. S. (2018). Tecnologías avanzadas de mecatrónica, diseño y manufactura. En *Memorias del IV Congreso AMDM 2018*.

Hernández, G. C. (2018). *Enfoques, teorías y perspectivas de la ingeniería de sistema y sus programas académicos*. Corporación Universitaria del Caribe.

Herrera, C. C. (2024). Desarrollo de competencias a través de prototipos y simuladores en un entorno interdisciplinario de física-matemática. *Revista Oratores, 20*, 78–102.

Joseph, S. B. (2022). Metaheuristic algorithms for PID controller parameters tuning: Review, approaches and open problems. *Heliyon, 8*(5), e09526.

Lira Hernández, I. A. (2022). *Sistema híbrido para asistir los procesos de diseño de ingeniería hacia la industria 4.0* [Tesis de licenciatura/posgrado, Universidad Autónoma Metropolitana]. https://doi.org/10.24275/uama.5810.8747

Morales, C. C. (2021). An innovative optimization design procedure for mechatronic systems with a multi-criteria formulation. *Applied Sciences, 11*(19), 8900.

Pedraza, Á. J. (2017). *Implementación de una metodología de diseño mecatrónico con herramientas modernas de simulación[unpublished].*

Santiago, L. M. (2024). Creando más con menos: La innovación frugal como nuevo paradigma de diseño para la sostenibilidad. *Revista de Estudios Interdisciplinarios del Arte, Diseño y la Cultura* (Especial: Intersecciones entre Sustentabilidad, Artes y Diseño), 38–55.

Trejo, L. O. (2024). Control industrial de un motor a pasos usando tecnología frugal. *AvaCient, 4*(2), 11–21.

Trujillo, M. S. (2025). El fenómeno de la innovación frugal: Productos asequibles, funcionales y minimalistas para atender el segmento bajo del mercado. *Innovar, 35*(95), e101621.

Velandia, L. J. (2023). *Sistema de control self-balancing robot (SBR) [unpublished].*

Zheng, C. A. (2023). Knowledge-based engineering approach for defining robotic manufacturing system architectures. *International Journal of Production Research, 61*(5), 1436–1454.