electronics

MDPI

Article

Real-Time Embedded System-Based Approach for Sensing Power Consumption on Motion Profiles

Luis F. Olmedo-García, José R. García-Martínez *, Edson E. Cruz-Miguel, Omar A. Barra-Vázquez, Mario Gónzalez-Lee and Trinidad Martínez-Sánchez

Faculty of Electronics and Communications Engineering, Universidad Veracruzana, Poza Rica 93390, Mexico; lolmedo@uv.mx (L.F.O.-G.); edsoncruz@uv.mx (E.E.C.-M.); omabarra@uv.mx (O.A.B.-V.); mgonzalez01@uv.mx (M.G.-L.); trmartinez@uv.mx (T.M.-S.)

* Correspondence: romangarcia@uv.mx; Tel.: +52-782-121-0928

Abstract: This paper discusses the energy consumption of three parabolic, trapezoidal, and S-curve profiles when implemented using an embedded system. In addition, it presents an alternative methodology for implementing motion controllers using an Advanced RISC Machine (ARM) microcontroller, which computes the trajectory and performs the control action in hard real-time. We experimented using a linear plant composed of a direct current (DC) motor coupled to an endless screw where a carriage was mounted. It can move mechanically along a rail at a distance of 1.16 m. A 4096 pulses per revolution (PPR) encoder was connected to the motor to calculate position and angular velocity. A Hall-effect-based current sensor was used to assess energy consumption. We conducted 40 tests for each profile to compare the energy consumption for the three motion profiles, considering cases with and without load on the carriage. We determined that the parabolic profile provides 22.19% lower energy consumption than the other profiles considered in this study, whereas the S-curve profile exhibited the highest energy consumption.

Keywords: power consumption; trapezoidal profile; parabolic profile; S-curve; embedded systems; microcontroller

Citation: Olmedo-García, L.F.; García-Martínez, J.R.; Cruz-Miguel, E.E.; Barra-Vázquez, O.A.; Gónzalez-Lee, M.; Martínez-Sánchez, T. Real-Time Embedded System-Based Approach for Sensing Power Consumption on Motion Profiles. *Electronics* **2023**, *12*, 3853. https://doi.org/10.3390/electronics12183853

Academic Editors: Deok-Hwan Kim and Mehdi Pirahandeh

Received: 31 July 2023
Revised: 7 September 2023
Accepted: 8 September 2023
Published: 12 September 2023

1. Introduction

Scientists have studied motion profiles extensively for several decades since it has uncountable applications related to motion control, robotics, and commercial machinery, such as robot manipulators [1,2], conveyor belts, computer numerical control (CNC) machinery, 3D printers, and any system that needs to control movement based on electric motors [3–5]. The motion profiles define not only the position of the motor shaft but also the velocity, acceleration, and the third derivative of position (jerk) along the path or entire movement [6]. With constant movement phases, it is possible to generate smooth trajectories that reduce mechanical wear and aggressive current peaks to improve movements' precision [7,8]. So, motion profiles are focused on obtaining a function of time, which returns at each instant the position that the mechanical system must follow [9]. Constantly it is sought to have at least continuous acceleration [10], since discontinuities in acceleration, such as in a trapezoidal profile, can cause the jerk to be too large and residual vibrations are generated [11], which, in addition to affecting the precision of the movement, also degrade valuable life of the mechanical system [12]. On the other hand, in mechatronics systems, motion profiles are applied to execute point-to-point movements [13]. For instance, to move from an initial point to a final point, the trajectory must be smooth, according to the cinematic magnitude restrictions of the system, with which an exceeding of the speed, acceleration, or jerk limits supported by the motor is avoided [14].

Another aspect to consider besides the motion profiles is the control algorithm, which is designed to keep the tracking error as low as possible, guaranteeing efficiency in the execution of the movement [15]. The most commonly used control technique is the classical

Proportional-Integral-Derivative (PID) and PD controllers as observed in [16]. However, technological changes demand new control strategies that present a better response than traditional controllers. For example, in [17], the implementation of a controller based on fuzzy logic for a robot manipulator shows better performance in terms of robustness and trajectory following; in [18], an adaptive controller with neural networks is used, which improves the precision of the movement at the cost of greater computational demands.

In recent years, several works have been proposed that allow smooth movements to be executed using functions of the polynomial, exponential, trigonometric type, or a combination of these [12,19–21]. The study of the motion profiles has shown that, depending on the path implemented, it is possible to minimize execution times, reduce vibrations by minimizing the jerk (which translates into less wear on the mechanical structure), as well as reduce energy consumption [22]. Fang et al. use a 15-segment S-curve trajectory formed by pieces of sigmoid functions, which allows for obtaining successive continuous derivatives [23]. This results in a trajectory with vibration and reduced tracking error verified experimentally in a robotic arm. Similarly, Wu et al. propose a 15-segment trajectory with a shorter execution time. It is made up of pieces of functions of the trigonometric type, with constant motion phases [1]. Alpers employs a seven-segment motion profile based on chunks of polynomial functions [24]. However, discontinuities in successive derivatives can cause unwanted vibrations; Ref. [10] proposes an improvement with a 15-segment path, in which the snap is a continuous piece-wise function. Due to the complexity of the expressions, the time required to calculate the trajectory is longer compared to classical profiles. Lee and Han use nth-order polynomial motion profiles to achieve fast motion and minimal vibration in an industrial robot [12]; the motion controller is implemented on a 32-bit MCU. Jerk minimization has been achieved through the use of degree 7 polynomials, as shown by Wu and Zhang in [4], which is implemented in a robot manipulator. Similarly, Boryga et al. use polynomials of degrees 5 to 9 to obtain phases of continuous motion [3]. However, experimental results still need to be presented. For their part, Wang et al. propose a methodology for using degree 4 polynomials in an industrial robot [25]. On the other hand, the reduction of energy consumption has been analyzed in point-to-point movements using trapezoidal and cycloid profiles [2], as well as trajectories based on Chebyshev polynomials [15]. A review and classification of energy-saving optimization strategies for robotic and automatic systems are presented in [26], being a software approach based on motion planning, an effective and economical technique easy to implement on existing systems [27]. Montalvo et al. conducted a comparative analysis of the energy consumption of the trapezoidal and parabolic motion profiles on a Field-Programmable Gate Array (FPGA)-based motion control system, which allowed parallel processing of the algorithms [28]. Carabani and Vidoni conducted a similar study, adding higher-order polynomial and modified S-curve profiles [29]. However, Hosseini and Hahn obtained better energy saving with third-degree polynomials than with higher-degree trajectories [30]. Other authors have developed a hybrid approach. For instance, Nshama seeks, in addition to decreasing execution times, to reduce energy consumption in industrial feed drive systems [31], and Halinga with the same hybrid approach proposed a modified S-curve trajectory introducing harmonic functions into the constant jerk phases [32].

This work's motivation is to design a motion controller for precise tracking to reduce energy consumption through appropriate motion profiles. A trajectory generator and a motion controller were embedded in a 32-bit microcontroller with ARM architecture to calculate the trajectories in real time.

The main contributions of this paper are:

- An analysis of the energy consumption of three motion profiles on a linear plant when it has a load and when there is no load.
- A novel approach that allows designing and implementing open-source motion controllers in ARM microcontrollers.
- A methodology for selecting microcontrollers for use in motion control applications according to the project to be developed.

The rest of this paper is organized as follows. We present and explain the velocity profiles in Section 2; next, we describe the materials we used and the methodology we followed for conducting the experiments in Section 3. Also, we discuss relevant implementation guidelines in Section 4. We present the experimental results in Section 5. Finally, we present our conclusions in Section 6 and the relevant references.

2. Fundamental Analysis of Parabolic, Trapezoidal, and S-Curve Velocity Profiles

2.1. Motion Profiles

A movement is defined by deriving a model with parameters such as position, velocity, and acceleration for different time instants: initial time t_0, final time t_f, or specific times in the interval $[t_0, t_f]$. Based on the above, the resulting model is a function $x(t)$ that satisfies all the time constraints. This model consists of a polynomial function whose degree is related to the number of constraints for the system [33]. This polynomial is known as the motion profile.

As stated before, the main idea is to implement three motion profiles: the parabolic, trapezoidal, and S-curve profiles; we describe each in this section.

2.2. Parabolic Velocity Profile

This type of trajectory models a movement with continuous speed and constant acceleration [34], the velocity is defined using a quadratic polynomial (parabola) where the position is given by a third-degree polynomial of the form provided by Equation (1).

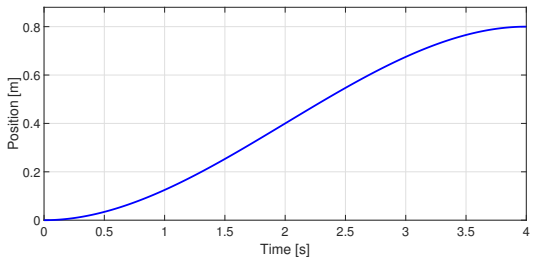
$$x(t) = k_0 + k_1(t - t_0) + k_2(t - t_0)^2 + k_3(t - t_0)^3 \tag{1}$$

$x(t)$ is the position of the profile. Successive derivatives of Equation (1) result in Equations (2) and (3) corresponding to the velocity and acceleration.

$$\dot{x}(t) = k_1 + 2k_2(t - t_0) + 3k_3(t - t_0)^2 \tag{2}$$

$$\ddot{x}(t) = 2k_2 + 6(t - t_0) \tag{3}$$

Here, $\dot{x}(t)$ and $\ddot{x}(t)$ are the velocity and acceleration of the profile, respectively. The constraints for the movement are $x(t_0) = x_0$, $x(t_f) = x_f$ and $\dot{x}(t_0) = \dot{x}(t_f) = 0$. On the other hand, $T = t_f - t_0$ is the total time of the movement, and $h = x_f - x_0$ is the total displacement. By evaluating Equations (2) and (3) at $t = t_0$, we determined $k_0 = x_0$ and $k_1 = 0$, whereas by evaluating at $t = t_f$, we determined that $k_2 = \frac{3h}{T^2}$ and $k_3 = -\frac{2h}{T^3}$. Figure 1 presents the parabolic motion profile.



(**a**)

Figure 1. *Cont.*

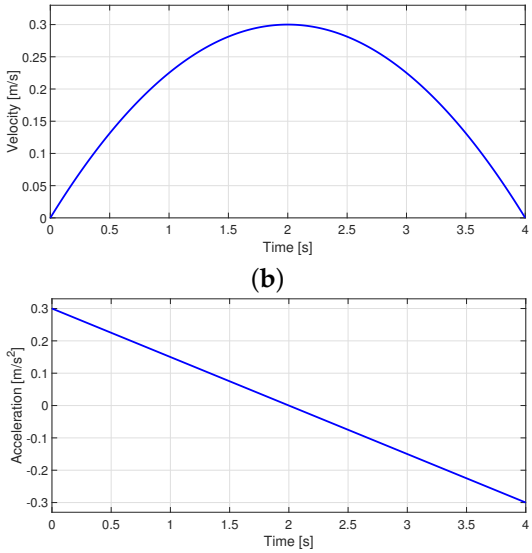


Figure 1. Parabolic velocity profile: (**a**) Position, (**b**) Velocity, and (**c**) Acceleration.

Figure 1b clarifies that the maximum velocity occurs at half the total time, that is, at $t = t_0 + \frac{T}{2} = \frac{t_0+t_f}{2}$. By evaluating Equation (3) at the said instant of time, we provide the maximum speed as $\dot{x}_{max} = \frac{3h}{2T}$. Computing Equation (1) in terms of the maximum speed allowed, we derived Equation (4). Equations (5) and (6) describe the velocity and acceleration of the movement at each instant of time.

- Position

$$x(t) = x_0 + \frac{2\dot{x}_{max}}{T}(t - t_0)^2 - \frac{4\dot{x}_{max}}{3T^2}(t - t_0)^3, \quad 0 < t \leq T \tag{4}$$

- Parabolic Velocity

$$\dot{x}(t) = \frac{4\dot{x}_{max}}{T}(t - t_0) - \frac{4\dot{x}_{max}}{T^2}(t - t_0)^2, \quad 0 < t \leq T \tag{5}$$

- Acceleration

$$\ddot{x}(t) = \frac{4\dot{x}_{max}}{T} - \frac{8\dot{x}_{max}}{T^2}(t - t_0), \quad 0 < t \leq T \tag{6}$$

Equations (7)–(9) relate the speed, acceleration, and execution time of the movement, respectively.

$$\dot{x}_{max} = \frac{3h}{2T} \tag{7}$$

$$\ddot{x}_{max} = \frac{6h}{T^2} = \frac{4\dot{x}_{max}}{T} \tag{8}$$

$$T = \frac{3h}{2\dot{x}_{max}} \tag{9}$$

2.3. Trapezoidal Motion Velocity

The trapezoidal velocity profile is one of the most popular in the industry due to its simplicity, which facilitates its implementation in motion controllers [35,36]. The trapezoidal motion profile combines a linear trajectory with curved sections, which allows obtaining movement at continuous speeds [7]. A target trajectory is divided into three stages with the same duration to implement it. The first stage is constrained to a constant

positive acceleration; therefore, the velocity increases linearly, and the position shape is parabolic. In the second stage, the movement is linear, with constant speed and zero acceleration. The movement in the last stage is continuous with negative acceleration, the velocity decreases linearly, and the position is modeled with a parabolic function [33]. Figure 2 shows the graphs for the trapezoidal profile's position, velocity, and acceleration.

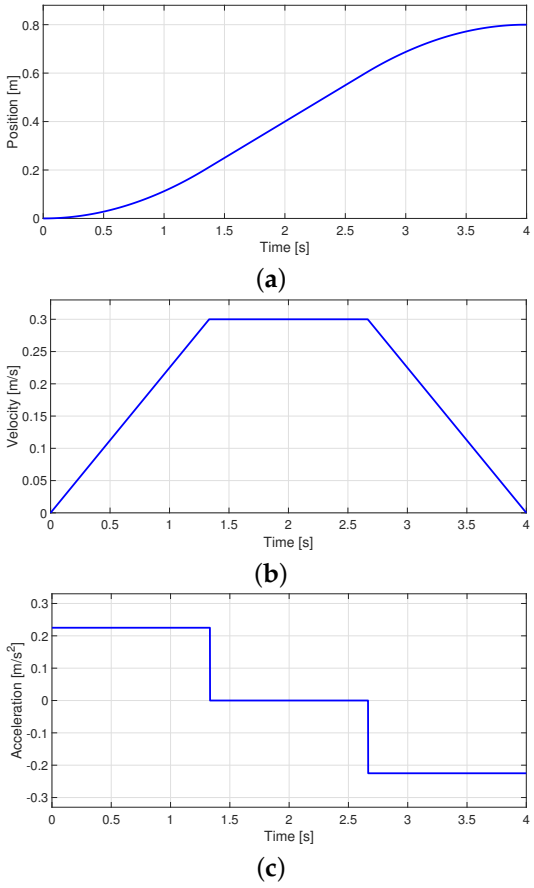


Figure 2. Trapezoidal velocity profile: (**a**) Position, (**b**) Velocity, and (**c**) Acceleration.

The mathematical model that defines the trajectory was obtained by analyzing the segments that make up the trajectory. The acceleration time T_a equals the deceleration time T_d; for this motion profile, Equations (10)–(12) determine the position, trapezoidal velocity, and acceleration of the trajectory, respectively.

- Position

$$x(t) = \begin{cases} x_0 + \frac{\dot{x}_{max}}{2T_a}t^2, & 0 \leq t < T_a \\ x_0 + \dot{x}_{max}(t - \frac{T_a}{2}), & T_a \leq t < 2T_a \\ x_f - \frac{\dot{x}_{max}}{2T_a}(T - t)^2, & 2T_a \leq t \leq T \end{cases} \tag{10}$$

- Trapezoidal Velocity

$$\dot{x}(t) = \begin{cases} \frac{\dot{x}_{max}}{T_a}t, & 0 \leq t < T_a \\ \dot{x}_{max}, & T_a \leq t < 2T_a \\ -\frac{\dot{x}_{max}}{T_a}t, & 2T_a \leq t \leq T \end{cases} \tag{11}$$

- Acceleration

$$\ddot{x}(t) = \begin{cases} \frac{\dot{x}_{max}}{T_a}, & 0 \leq t < T_a \\ 0, & T_a \leq t < 2T_a \\ -\frac{\dot{x}_{max}}{T_a}, & 2T_a \leq t \leq T \end{cases} \tag{12}$$

The maximum velocity $\dot{x}_{max}$, acceleration $\ddot{x}_{max}$, and displacement h are related by Equations (13) and (14).

$$\dot{x}_{max} = \frac{3h}{2T} \tag{13}$$

$$\ddot{x}_{max} = \frac{3\dot{x}_{max}}{T} = \frac{9h}{2T^2} \tag{14}$$

The total time T is determined using Equation (15). T_a is given by Equation (16).

$$T = \frac{3h}{2\dot{x}_{max}} = \sqrt{\frac{9h}{2\ddot{x}_{max}}} \tag{15}$$

$$T_a = \frac{T}{3} \tag{16}$$

2.4. S-Curve Velocity Profile

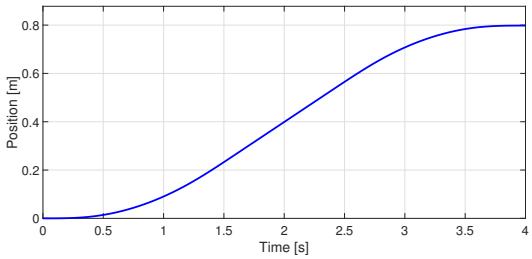
A disadvantage of previous motion profiles is that they exhibit discontinuities for acceleration, causing a considerable magnitude of jerk [11]. This paper defines the jerk as $\dddot{x}(t)$. To mitigate this drawback, we use a 7-segment S-curve profile with equal acceleration times T_a and deceleration times T_d to obtain a movement with continuous acceleration and limited jerk values. The parameters for $\dddot{x}_{max}$, $\ddot{x}_{max}$, $\dot{x}_{max}$ used are computed using Equations (17)–(19) proposed in [37].

$$\dot{x}_{max} = \frac{h}{(1-\alpha)T} \tag{17}$$

$$\ddot{x}_{max} = \frac{h}{\alpha(1-\alpha)(1-\beta)T^2} \tag{18}$$

$$\dddot{x}_{max} = \frac{h}{\alpha T_j(1-\alpha)(1-\beta)T^2} \tag{19}$$

where α is a constant factor in varying the acceleration time $T_a = \alpha T$, and β is a constant factor in altering the length of the jerk's pulse, $T_j = \beta T_a$. Figure 3 depicts the position, velocity, acceleration, and jerk of an S-curve profile with $\alpha = 0.4$ and $\beta = 0.25$.



(**a**)

Figure 3. *Cont.*

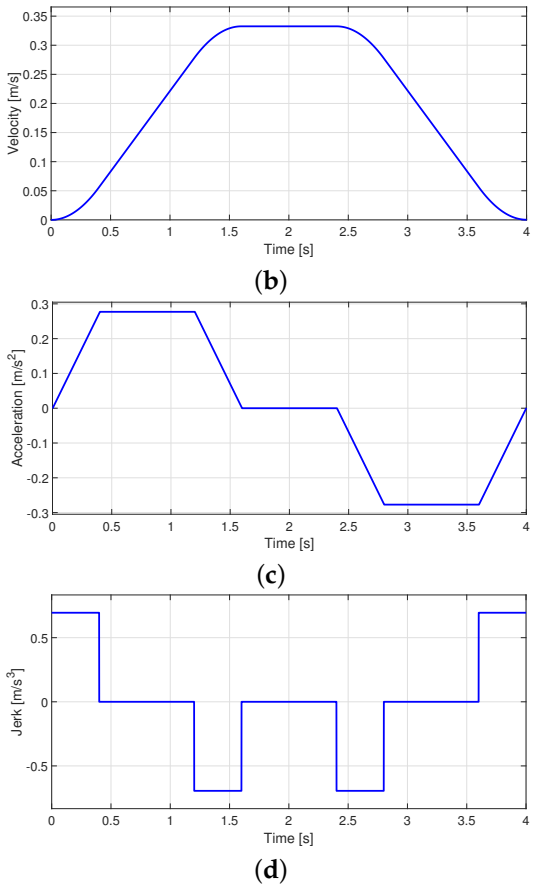


Figure 3. S-curve velocity profile: (**a**) Position, (**b**) Velocity, (**c**) Acceleration, and (**d**) Jerk.

Once the values for $\dot{x}_{max}$ and the time intervals of Equation (20) are calculated, the acceleration, velocity, and position for the trajectory are obtained by integration using Equations (21)–(23).

$$\dddot{x}(t) = \begin{cases} \dddot{x}_{max}, & t_0 \leq t < t_1 \\ 0, & t_1 \leq t < t_2 \\ -\dddot{x}_{max}, & t_2 \leq t < t_3 \\ 0, & t_3 \leq t < t_4 \\ -\dddot{x}_{max}, & t_4 \leq t < t_5 \\ 0, & t_5 \leq t < t_6 \\ \dddot{x}_{max}, & t_6 \leq t < t_7 \end{cases} \tag{20}$$

$$\ddot{x}(t) = \ddot{x}(t_0) + \int_{t_0}^{t} \dddot{x}(\tau) d\tau \tag{21}$$

$$\dot{x}(t) = \dot{x}(t_0) + \int_{t_0}^{t} \ddot{x}(\tau) d\tau \tag{22}$$

$$x(t) = x(t_0) + \int_{t_0}^{t} \dot{x}(\tau) d\tau \tag{23}$$

3. Materials and Methods

An approach to selecting the embedded system, sensor calibration, plant identification, and motion controller design is presented in Figure 4. The first step is selecting the most suitable embedded system according to the design constraints. The proper selection of embedded systems is of utmost importance since it is crucial to understand the capabilities of these devices to determine if they are suitable to solve a given problem. Thus, it is

essential to analyze cost-benefit trade-offs during the selection process of an embedded system to properly balance the costs associated with its implementation and the benefits achieved.

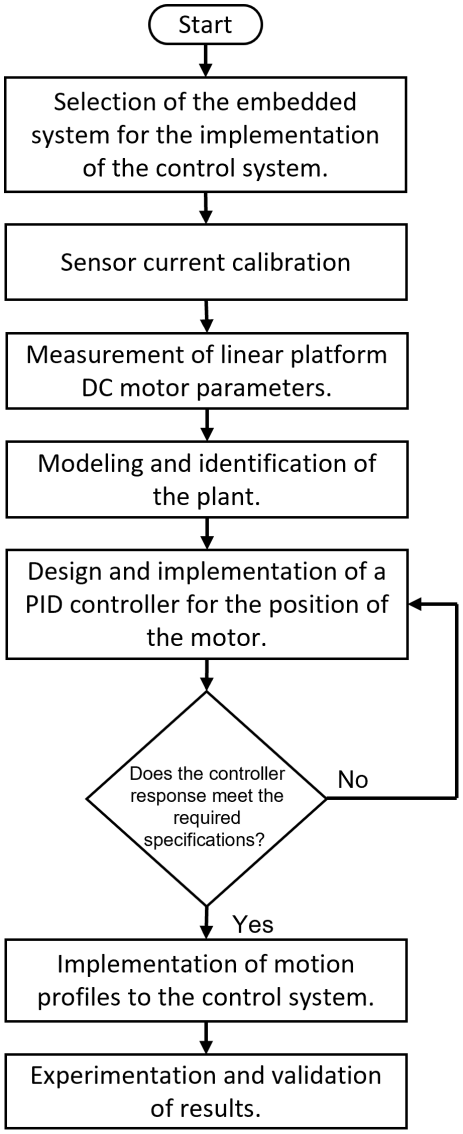


Figure 4. Design stages of a motion controller.

Once the designer selects an embedded system, the next step is obtaining the parameters for the motor voltage, current, and kinematic magnitudes. This is crucial to ensure that the platform operates within its limits. In the case of having information on these parameters in advance, they can be used directly. However, such values are missing in some cases; in those cases, designers require determining them experimentally.

The third step involves the designer deriving a mathematical model of the DC motor. This step, known as identification, involves obtaining a transfer function through experimentation to determine the system's dynamic response. To reach this goal, designers must follow a rigorous and methodological approach, comparing motor input and output data, such as applied voltage and resulting angular velocity converted in linear position over time, and then use these data to determine the motor's mathematical model. When designers assess the closed-loop model of the DC motor, they can design a controller to ensure the system's stability. This controller, which is usually proportional in action, plays a crucial role in rejecting disturbances that may affect the proper behavior of the motor. The proportional action of the controller also allows errors to be corrected and deviations to be

mitigated, which contributes to maintaining stability and improving the dynamic response of the plant. Figure 5 shows a block diagram of the system used for the identification process. The input is the target position the DC motor should reach, denoted as $X_{sp}(s)$. On the other hand, the system's output is the carriage's real position, denoted by $X_m(s)$. A control mechanism computes the control signal $U(s)$. In this case, it is the voltage applied to the motor to get the carriage as close to the target position as possible.

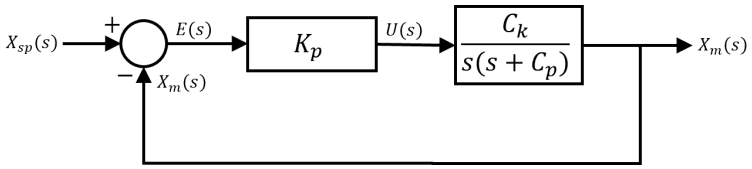


Figure 5. Block diagram of the system used for identification.

The closed-loop transfer function of the block diagram in Figure 5 is given by Equation (24).

$$X(s) = \frac{X_m(s)}{X_{sp}(s)} = \frac{K_p C_k}{s^2 + C_p s + K_p C_k} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2} \tag{24}$$

$X(s)$ is the transfer function of a second-order system, where C_k and C_p are parameters calculated with Equations (25) and (26).

$$C_p = 2\zeta\omega_n \tag{25}$$

$$C_k = \frac{\omega_n^2}{K_p} \tag{26}$$

According to modern control theory, when a step input is applied to a second-order system, the system response exhibits specific characteristics, such as rise time t_r and overshoot M_p, especially in the case of an overdamped system, where the damping coefficient is in the range $0 \leq \zeta < 1$. To achieve this overdamped behavior, it is necessary to adjust the value of the proportional coefficient K_p to be large enough. The parameters to be calculated based on the transient response are the damping constant ζ, the undamped natural frequency ω_n, and the damped natural frequency ω_d, expressed in Equations (27)–(29).

$$\zeta = \frac{|\ln M_p|}{\sqrt{\pi^2 + \ln^2 M_p}} \tag{27}$$

$$\omega_d = \frac{\pi - \arctan\left(\frac{\sqrt{1-\zeta^2}}{\zeta}\right)}{t_r} \tag{28}$$

$$\omega_n = \frac{\omega_d}{\sqrt{1-\zeta^2}} \tag{29}$$

The fourth step consists of the design of the controller, a classic PID-type controller shown in Figure 6 is proposed to eliminate errors in steady state.

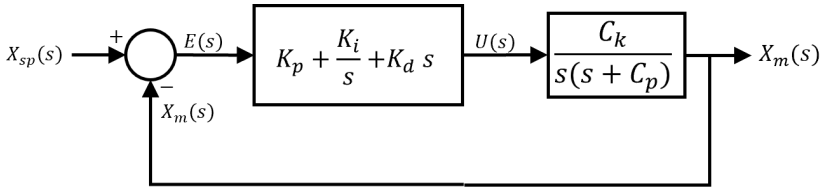


Figure 6. PID algorithm structure.

The controller parameters are obtained using the pole assignment method. This approach consists of selecting the desired poles of the closed-loop system and calculating the

controller parameters so that these poles are located at the desired positions. By assigning the poles appropriately, a desired behavior of the system in terms of stability, response time, and damping can be achieved. The closed-loop transfer function of the system is determined from the controller parameters. This transfer function, represented by Equation (30), describes the relationship between the system's input and output in the frequency domain.

$$\frac{X_m(s)}{X_{sp}(s)} = \frac{C_k(K_d s^2 + K_p s + K_i)}{s^3 + (C_p + C_k K_d)s^2 + K_p C_k s + K_i C_k} \tag{30}$$

The controller gains are selected to place specific poles in the denominator of Equation (30). The poles are calculated using the characteristic equation of a second-order system given in Equation (31), whose parameters are computed with Equations (27)–(29) according to the required design criteria.

$$s^2 + 2\zeta\omega_n s + \omega_n^2 \tag{31}$$

Once the controller meets the specifications, the previously described motion profiles are added to the control system. The function of the trajectory generator is to generate a series of values that specify the position that the motor must assume at each instant of time. Calculating the trajectory requires defining the magnitude of the displacement and the time to execute the movement. The last step consists of experimentation and validation of results, which is addressed in a later section.

4. Methodology of Implementation

4.1. Selecting an Embedded System

Following the process described in Figure 4, the TIVA C series TM4C123 evaluation platform has been selected. This low-cost platform includes two TM4C123GH6PM high-performance 32-bit microcontrollers based on the ARM Cortex-M4F architecture. The choice of this embedded system is based on its essential capabilities for implementing the project, such as motion controller, path generation, and accurate current measurement through the 12-bit analog-to-digital converter (ADC). In addition, it has a QEI (Quadrature Encoder Interface) module dedicated to reading the encoder, significantly simplifying the system configuration process and allowing it to implement the system without interruptions or parallel programming like in an FPGA-based system. Another essential advantage of this microcontroller is its ability to operate in hard real-time with a sampling rate of 0.001 s. This ensures that the entire system can respond quickly and accurately to the demands of the environment in which it is located, which is especially relevant for real-time embedded system applications. In addition, it allows connection to a computer via serial protocol, which facilitates efficient data collection from the plant. This communication capacity is invaluable for monitoring and analyzing the results obtained during the operation of the embedded system. The main characteristics of this microcontroller are presented in Appendix A.

As recommended in [38,39], an analysis of computational times is required to validate the embedded system selection for real-time applications. To prove the computational analysis, this section compares the backward Euler, trapezoidal rule, and Runge–Kutta numerical methods for giving a solution to the linear plant and motion controller system offline. Besides, an integration time of 0.001 s is considered the upper limit, so all the calculations under these data are adequate for real-time motion control systems. Figure 7 depicts the behavior of the microcontroller during 5000 iterations. The backward Euler and the trapezoidal rule obtained 0.00000155 se and 0.00000181 s per execution, respectively. The Runge–Kutta method shows variability in the execution time, where the average computed per execution was 0.00000362 s. Table 1 contains the execution time for the total number of iterations using the three integration methods and step sizes. These results demonstrate that the microcontroller implements hard real-time implementation requirements.

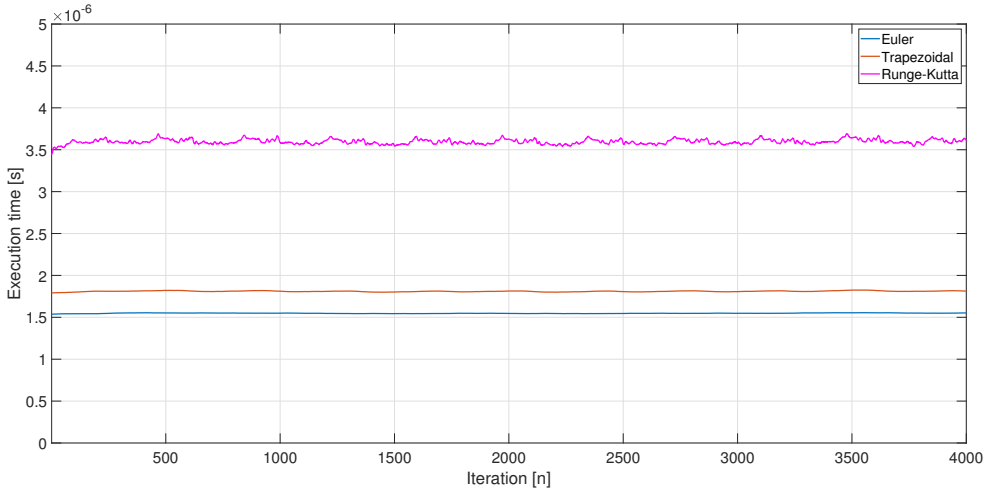


Figure 7. Execution time for each integration technique.

Table 1. Execution times using different step sizes.

Step Size (Δ)	Euler Method	Trapezoidal Method	Runge–Kutta Method
10^{-3} s	0.008502 s	0.010502 s	0.019379 s
10^{-4} s	0.085002 s	0.105002 s	0.193754 s
10^{-5} s	0.837502 s	1.037502 s	1.925004 s
10^{-6} s	8.375002 s	10.375002 s	19.250004 s

4.2. Embedded Motion Controller Implementation

This section describes the implementation of the motion control system that has been made. In this system, a classic PID controller and a motion profile generator have been integrated into a microcontroller to control the position of the carriage of a linear platform. To allow configuration and adjustment of the motion profile parameters, serial communication is established between the microcontroller and a computer. Figure 8 shows the general diagram of the blocks that comprise the proposed system.

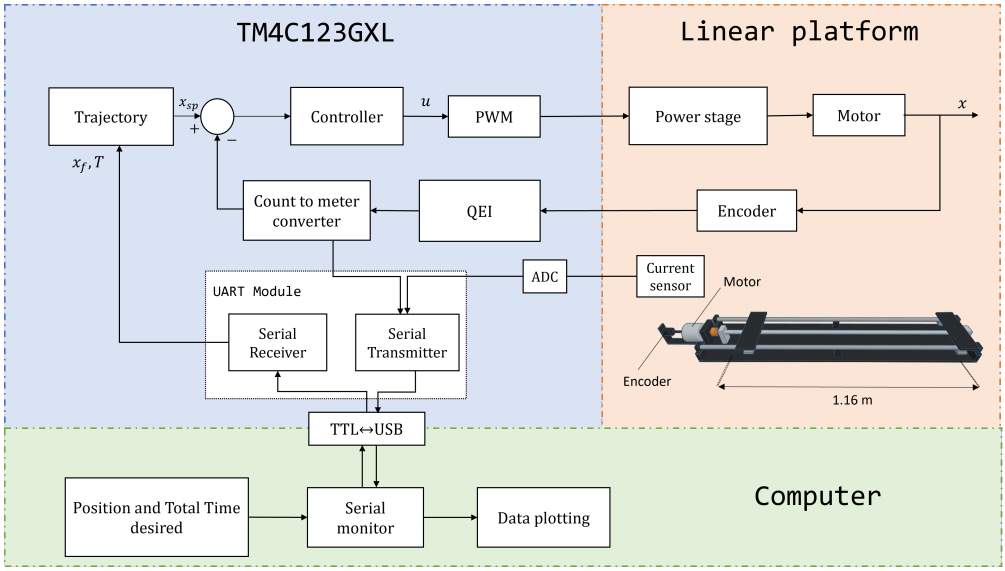


Figure 8. Approach of the entire system implementation.

The first block consists of the plant described in the last section. The second block is the embedded system. In this stage, the microcontroller schedules sensing tasks for measuring

both the current and the position of the encoder, which allows for obtaining information on the state of the motor. In addition, the microcontroller computes the motion profile using the received data to determine the carriage's speed, acceleration, and displacement. Once the motion profile has been calculated, the PID controller adjusts the trajectory using the position information. The control signal is applied to the motor driver, the HW-039 BTS7960, Hand on Tech manufacturer, via a Pulse-Width Modulation (PWM) signal.

The third component of the system is a computer, which is linked to the ARM microcontroller using the PL2303HX TTL-USB converter from Prolific Technology Inc. manufacturer in Taipei. During the tests, the computer sends data to the embedded system regarding the required displacement and the total time for the execution of the movement. These data define the parameters of the movement profile and the duration of the operation. On the other hand, the microcontroller sends data to the computer about the current position of the motor, the computed control signal, and the current consumed by the motor. These data allow real-time monitoring and visualization of the status and performance of the control system.

The direct manipulation of the microcontroller registers provides greater control and optimization of the system, adapting it to the specific needs of this project, one of which is the sampling time of the entire proposed system. The flowchart in Figure 9 shows the algorithm used to program the embedded system.

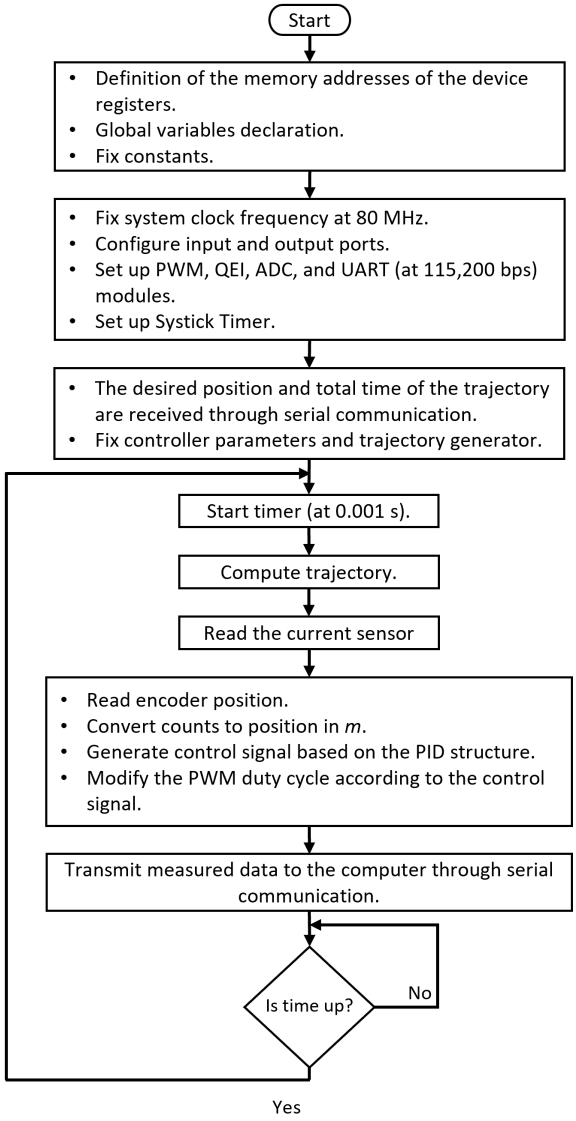


Figure 9. Flowchart of the algorithm used for the microcontroller programming.

4.3. Initial Definitions of the Microcontroller Memory Addresses

In the initial stage of the programming process, the memory addresses of the registers related to the device's configuration are defined. These registers are essential to establish and control the behavior of the different components of the microcontroller. In addition to memory addresses, global variables, and constant values are established. Global variables can be accessed and used anywhere in the program, making exchanging information between code sections easy. On the other hand, constant values are those that do not change throughout the execution of the program and are used to establish parameters or static configurations.

4.4. Initial Configuration

In this subsection, the configuration of the necessary registers for the correct operation of the microcontroller is carried out. By default, the microcontroller operates at a frequency of 16 MHz, but by appropriately modifying the corresponding registers, it is possible to reach a maximum operating frequency of 80 MHz. Once the operating frequency has been configured, proceed to the activation and configuration of input/output ports. These ports allow communication and connection with other external devices, facilitating the interaction of the microcontroller with its environment. In addition to the input/output ports, other necessary peripheral modules must be initialized. For example, the Quadrature Encoder Interface (QEI) module is configured to allow quadrature reading of the encoder, which provides greater accuracy in motor position measurement. In addition, the PWM module is configured to generate two PWM signals at a frequency of 5 kHz, allowing efficient control of the speed and power supplied to the motor. In the same way, the initialization of the ADC module is performed, which enables the conversion of analog signals to digital for further processing. On the other hand, the Universal Asynchronous Receiver/Transmitter (UART) module is configured to operate at a speed of 115,200 baud, which facilitates serial communication with other devices. The last module to configure is the Systick Timer. This 24-bit timer is loaded with RELOAD_VALUE, a value given by the Equation (32) to generate an interrupt every t seconds, which allows ensuring a time-fixed sampling. However, before starting the timer count, x_f and T are set according to the information received by serial communication.

$$\text{RELOAD\_VALUE} = t \times \text{Clock Frequency} \tag{32}$$

Within the program's main loop, a series of tasks are carried out periodically every $T_s = 0.001$ s, corresponding to the system sampling time. First, the trajectory calculation is performed, which involves determining the desired position or the path that the system should follow. Next, the current is measured, an important parameter to evaluate the performance and load of the motor. This measurement is carried out through an ACS712 model current sensor from Allegro Microsystems, Inc., Worcester Massachusetts. Subsequently, the control algorithm is executed, which uses the trajectory information to generate the appropriate control signal. The control algorithm is based on a discrete PID controller. Finally, the measured data is sent to the computer using the UART module.

4.5. Implementation of Motion Profiles

For the implementation of the velocity profiles, in each iteration, k, a point of the trajectory, is calculated for the current time instant $t = kT_s$. In the case of the parabolic profile, the trajectory is calculated in the microcontroller with Equation (33), which is a simplification of Equation (4) in which $x_0 = 0$ is considered and $t_0 = 0$.

$$x(t) = \frac{2\dot{x}_{max}}{T}t^2 - \frac{4\dot{x}_{max}}{3T^2}t^3, \quad 0 < t \leq T \tag{33}$$

Regarding the trapezoidal profile, Equation (10) is reduced to Equation (34) considering the initial position and time equal to 0.

$$x(t) = \begin{cases} \frac{\dot{x}_{max}}{2T_a}t^2, & 0 \leq t < T_a \\ \dot{x}_{max}(t - \frac{T_a}{2}), & T_a \leq t < 2T_a \\ x_f - \frac{\dot{x}_{max}}{2T_a}(T-t)^2, & 2T_a \leq t < T \end{cases} \tag{34}$$

The S-curve velocity profile is calculated with Equations (35)–(37), obtained after the discretization of Equations (21)–(23) by Euler's method backward with a step size of 0.001 s, corresponding to the sampling time of the system.

$$A(k) = A(k-1) + T_s J(k) \tag{35}$$

$$V(k) = V(k-1) + T_s A(k) \tag{36}$$

$$X(k) = X(k-1) + T_s V(k) \tag{37}$$

k is the sample number, whereas $J(k)$, $A(k)$, $V(k)$, and $X(k)$ are the current jerk, acceleration, velocity, and position, respectively.

4.6. PID Control Strategy Design

The Root Locus Method (RLM) was employed to tune the PID controller. The system's behavior was designed with a settling time of $t_s \leq 0.3$ s and overshoot $M_p < 20\%$. This corresponds to having dominant poles at $s = -13 \pm 25.5\,\mathrm{i}$. As shown in Figure 10, the Root Locus of the system's response with the PID controller and the location of the dominant poles are depicted according to the design.

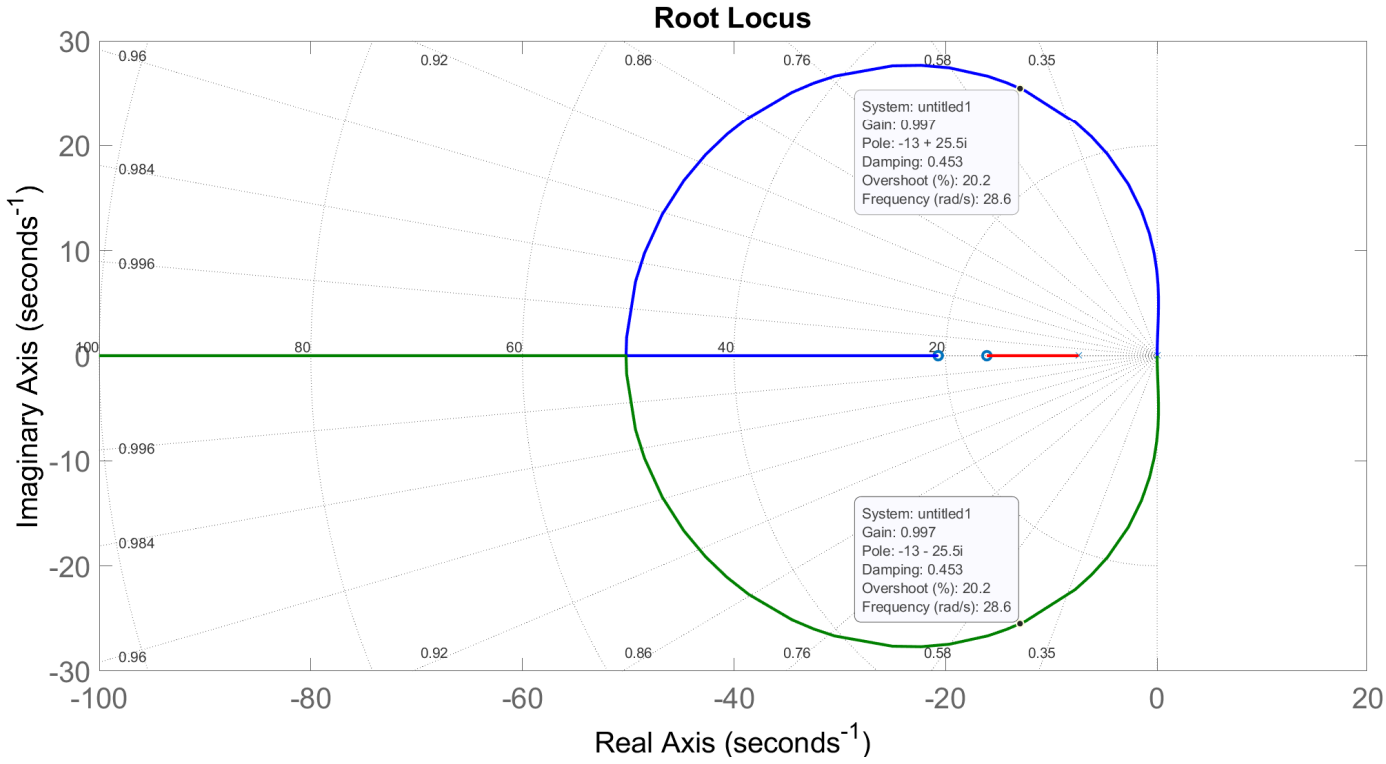


Figure 10. Root Locus of the control system.

Similarly, the stability analysis of the controller was analyzed through the Bode diagram according to Figure 11. Based on the phase margin, it can be concluded that the closed-loop system is stable.

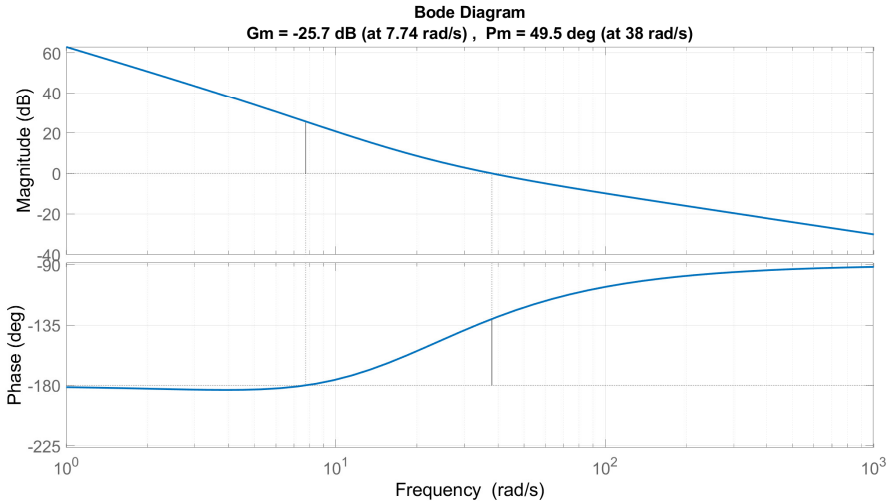


Figure 11. Bode diagram of the control system.

Figure 12 shows the system model's time response in simulation. It can be observed that the overshoot is 30% with a settling time of 0.3 s. It is evident that the closed-loop system is stable and meets the design requirements. However, as mentioned in the works [40,41], for the validation of the control system, it is necessary to verify its performance in the physical system and observe its behavior in the presence of disturbances and model uncertainties.

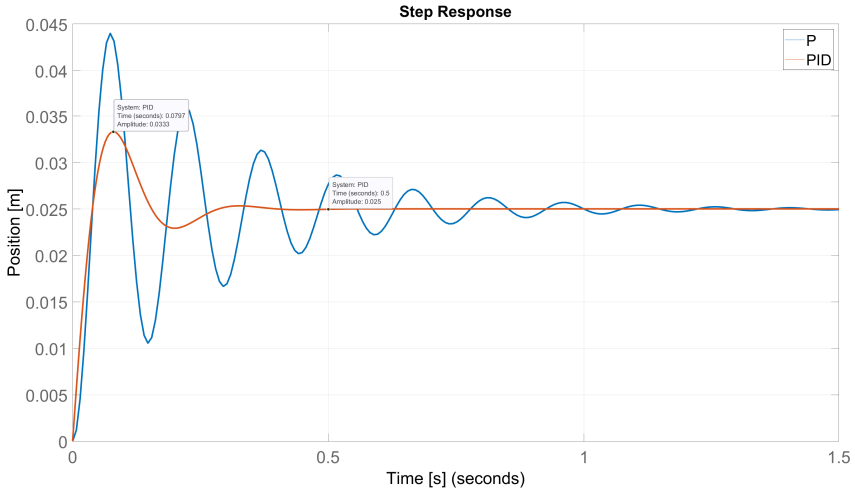


Figure 12. Time response of the system to a step input, applying a PID controller.

4.7. Implementation of the Control Algorithm

The control algorithm starts with the encoder reading. The value is converted to meters by multiplying the count by $\frac{1}{220500}$. The error is calculated as the deviation between the position in meters and the actual set point given by the trajectory generator. The control signal $U(k)$ is generated with Equations (38) and (39), obtained from the discretization of the PID controller by the backward Euler method.

$$U_i(k) = U_i(k-1) + K_i T_s E(k) \tag{38}$$

$$U(k) = K_p E(k) + U_i(k-1) + K_i T_s E(k) + K_d \frac{E(k) - E(k-1)}{T_s} \tag{39}$$

Controller gains were calculated as described in the materials and methods section. The controller gains tuned for this experiment are $K_p = 60.6464$, $K_i = 549.2290$, $K_d = 1.6480$.

The value obtained by the control signal represents a voltage in the $[-24, 24]$ V range, which is the DC motor's operating range. The control signal is scaled to the microcontroller's operating range $[-3.3, 3.3]$ V by modifying the duty cycle of the PWM signal, given by Equation (40).

$$Duty\ Cycle(\%) = 100\frac{\text{PWM\_CMP}}{\text{PWM\_LOAD}} = 100\frac{U(k)}{v_{Max}} \tag{40}$$

$$\text{PWM\_CMP} = \frac{\text{PWM\_LOAD}}{v_{Max}} * U(k) \tag{41}$$

where PWM_LOAD is the value to load to the configuration register to generate a 5 kHz PWM signal, v_{Max} is the maximum voltage (in this case 24 V), $U(k)$ is the control signal, and PWM_CMP is the value to load in the register that modifies the duty cycle.

4.8. Timer Ended

After sending the measurements to the computer, wait for the timer to expire. In this step, the account status indicator is checked. When it reaches zero, the counter resets and the loop starts again.

5. Experimental Results

This section presents the results obtained by applying the three velocity profiles to the position controller over a linear plant. First, the linear plant is a round rail linear guide system with end support that ensures correct carriage displacement on the rails. Table 2 describes the geometry of the plant used in the experiment.

Table 2. Geometrical parameters of the linear plant.

Geometrical Parameters	Dimensions
Total length of the plant	1.7 m
Plant width	0.30 m
Carriage shaft	0.020 m
Rail linear guides	1.16 m
Round shaft linear guides	0.020 m
Coupling shaft flexible	0.020 m
Bearing balls	0.020 m × 0.42 m × 0.012 m

5.1. Data Acquisition

The QEI module handles the encoder counts, which are held in a 32-bit integer register. The carriage position in meters is determined by multiplying the encoder counts with a fixed factor of 22.05, given that 220,500 counts correspond to 1 m. Monitoring the current drawn by the DC motor is achieved through the current sensor. The embedded system reads and stores the current sensor data in a 2-byte register, as the internal ADC has a resolution of 12 bits. The sample time of the experiment was 0.001 s, while the UART module was configured at a speed of 115,200 bps. Figure 13 shows the linear plant used for the experimentation. This figure details the main elements that make up the motion control system.

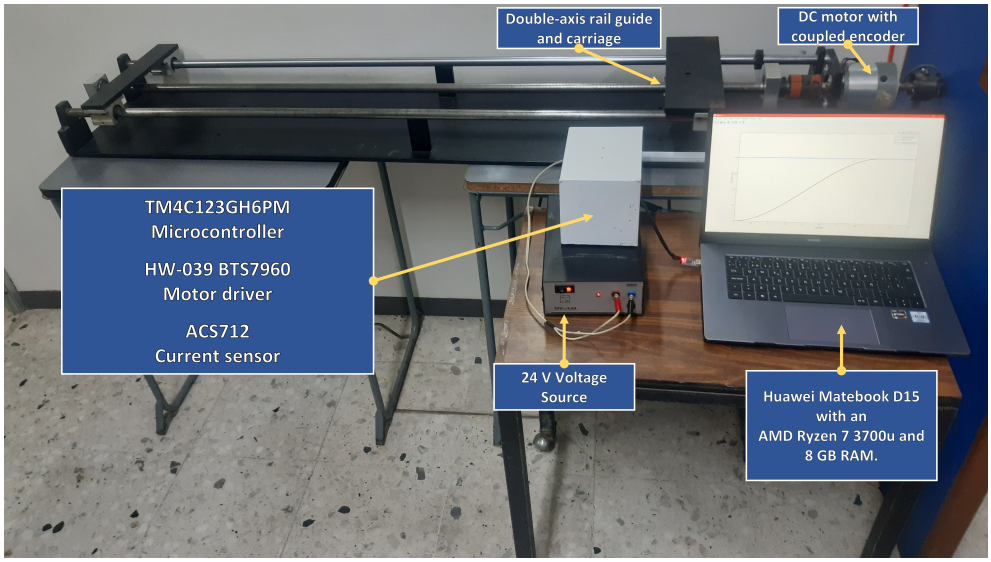


Figure 13. Linear plant used for the experimentation.

In Figure 14, the physical system's performance with a PID controller is depicted for different step inputs with amplitudes of 0.0025 m at time intervals of 5 s. Besides the robustness of the controller, it can be observed that the system is stable and meets the design specifications established, as mentioned in Section 4.6.

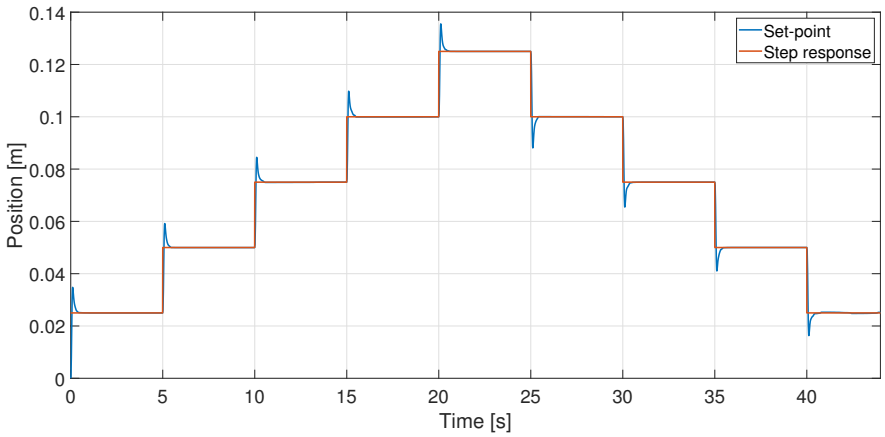


Figure 14. Test of steps response to prove the robustness of the controller.

5.2. Parabolic Profile without Load

The carriage of the plant is moved at a distance of 0.8 m in 4 s. Two cases are considered for each profile: the first without a load on the carriage and the second with a load of 9 kg. Table 3 summarizes the parameters for implementing the parabolic velocity profile.

Table 3. Parameters used to implement the parabolic motion profile.

Parameter	Value
Final position (x_f)	0.8 m
Total time for displacement (T)	4 s
Maximum velocity (x_{max})	0.3 $\frac{\text{m}}{\text{s}}$
Maximum acceleration ($\ddot{x}_{max}$)	0.3 $\frac{\text{m}}{\text{s}^2}$

Figure 15a compares the carriage position obtained experimentally versus the ideal trajectory of the velocity profile. In both graphs, the final position of the carriage corresponds

to the target position x_f in the required time T. In addition, the graph of the percentage error $\epsilon(\%)$ in Figure 15b shows a higher performance of the controller in terms of tracking trajectory as ϵ tends towards zero for most of the execution time of the movement. $\epsilon(\%)$ is obtained with Equation (42), where x_r is the actual position and x_p is the reference position calculated with the profile.

$$\epsilon(\%) = 100\left|\frac{x_r(k) - x_p(k)}{x_p(k)}\right| \tag{42}$$

The speed is calculated offline using central difference approximation given by Equation (43) for the first derivative of the experimental position.

$$V(k) = \frac{-X(k+2) + 8X(k+1) - 8X(k-1) + X(k-2)}{12T_s} \tag{43}$$

Figure 15c shows the estimated speed after applying a low-pass filter of the FIR type. The resulting graph's shape is close to that of the implemented velocity profile; the estimated maximum velocity of 0.3 $\frac{m}{s}$ agrees with the calculated maximum velocity. On the other hand, the acceleration is obtained with the same numerical method of Equation (43). The acceleration computed and that of the profile are compared in Figure 15d, which is notorious for the abrupt change in acceleration that the velocity profile presents at the beginning and end of the movement.

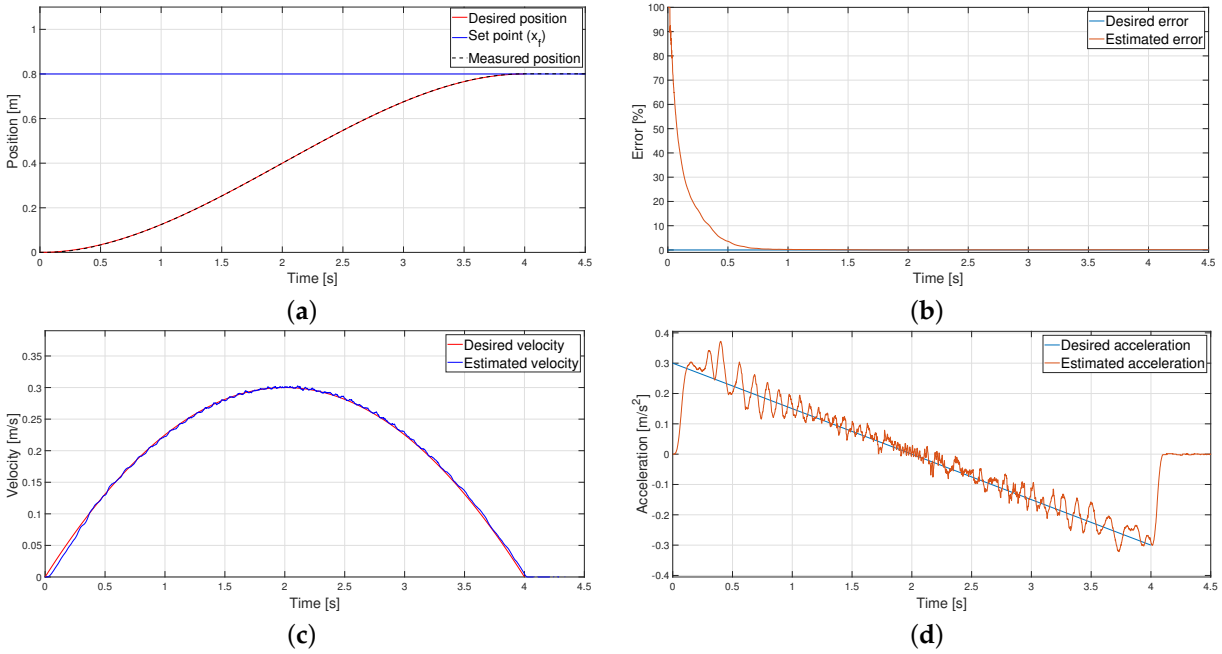


Figure 15. Parabolic velocity profile: (**a**) Position, (**b**) Error (**c**) Velocity, and (**d**) Acceleration.

The control signal applied to the system is depicted in Figure 16a. A maximum voltage of 9.8 V is observed for the no-load case. It is important to note that the shape of the control signal closely follows the speed modeled by the profile. Figure 16b shows the current consumed during the execution of the movement. The maximum current value obtained was 1.12 A. Figure 16c shows the instantaneous power. The energy, calculated from integrating the instantaneous power, is shown in Figure 16d. The total energy consumed was 23.7044 J.

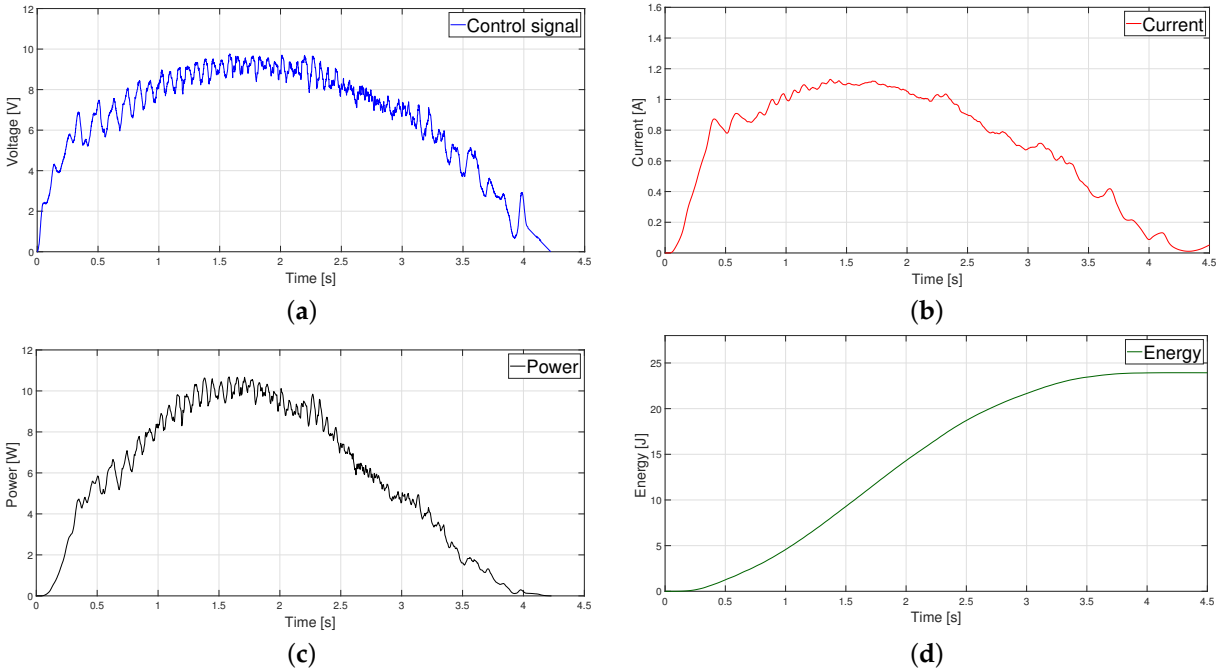


Figure 16. Parabolic velocity profile: (**a**) Control signal, (**b**) Current (**c**) Power, and (**d**) Energy.

5.3. Trapezoidal Profile without Load

Regarding the trapezoidal velocity profile, Table 4 shows the parameters used to match the same displacement and execution time as the parabolic profile. The calculated maximum velocity of 0.3 $\frac{\text{m}}{\text{s}}$ is the same as in the trapezoidal velocity profile. However, the maximum acceleration is reduced from 0.3 $\frac{\text{m}}{\text{s}^2}$ to 0.225 $\frac{\text{m}}{\text{s}^2}$.

Table 4. Parameters used to implement the trapezoidal motion profile.

Parameter	Value
Final position (x_f)	0.8 m
Total time for displacement (T)	4 s
Acceleration Time (T_a)	1.333 s
Maximum velocity (x_{max})	0.3 $\frac{\text{m}}{\text{s}}$
Maximum acceleration ($\ddot{x}_{max}$)	0.225 $\frac{\text{m}}{\text{s}^2}$

Figure 17a compares the experimental and ideal position of the profile. It can be seen, in Figure 17b, that the error drops to less than 1% in 0.7 s. On the other hand, Figure 17c shows that the estimated velocity is numerically close to that determined by the profile. Figure 17d offers the estimated acceleration; four abrupt changes are observed. The first occurs at the beginning of the acceleration phase at $t = 0$, where the acceleration rises from 0 to $\ddot{x}_{max}$. The second one happens at the beginning of the constant velocity phase at $t = T_a$, where the acceleration drops to 0. The third abrupt change in acceleration rises from 0 to $-\ddot{x}_{max}$ at $t = 2T_a$, where the deceleration phase starts. The last one is presented at the end of the movement at $t = T$. The acceleration suddenly changes from $-\ddot{x}_{max}$ to 0.

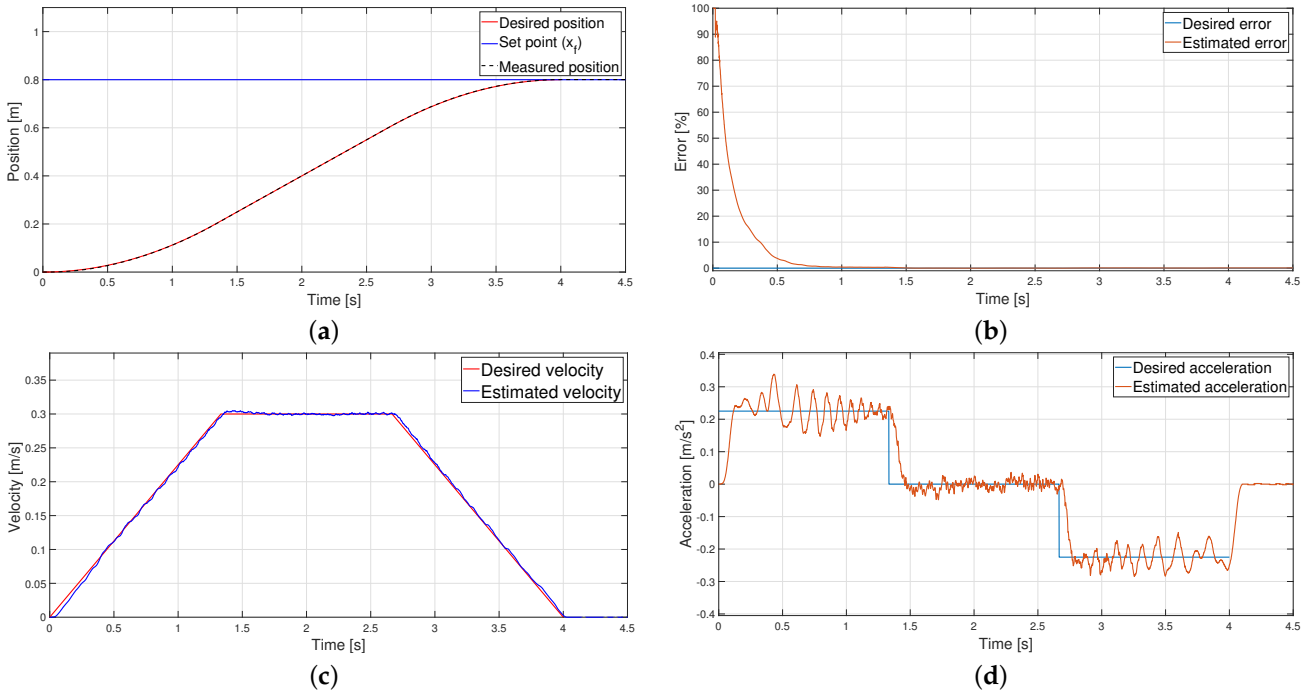


Figure 17. Trapezoidal velocity profile: (**a**) Position, (**b**) Error (**c**) Velocity, and (**d**) Acceleration.

From Figure 18a, it can be seen that at the beginning of the constant speed phase in $t = T_a$, the control signal presents a maximum peak voltage of 10.45 V, caused by the abrupt change in the acceleration is subsequently maintained at 9.6 V until the beginning of the deceleration phase at $t = 2Ta$, where it begins to drop to zero.

The current presents a similar behavior. Figure 18b shows that in $t = T_a$, a maximum current of 1.43 A is measured. During the constant speed phase, the current consumed is 1.04 A. Since instantaneous power is calculated as the product of voltage times current, the power graph in Figure 18c shows its maximum value at $t = T_a$. According to Figure 18d, the total energy consumed is 25.964 J.

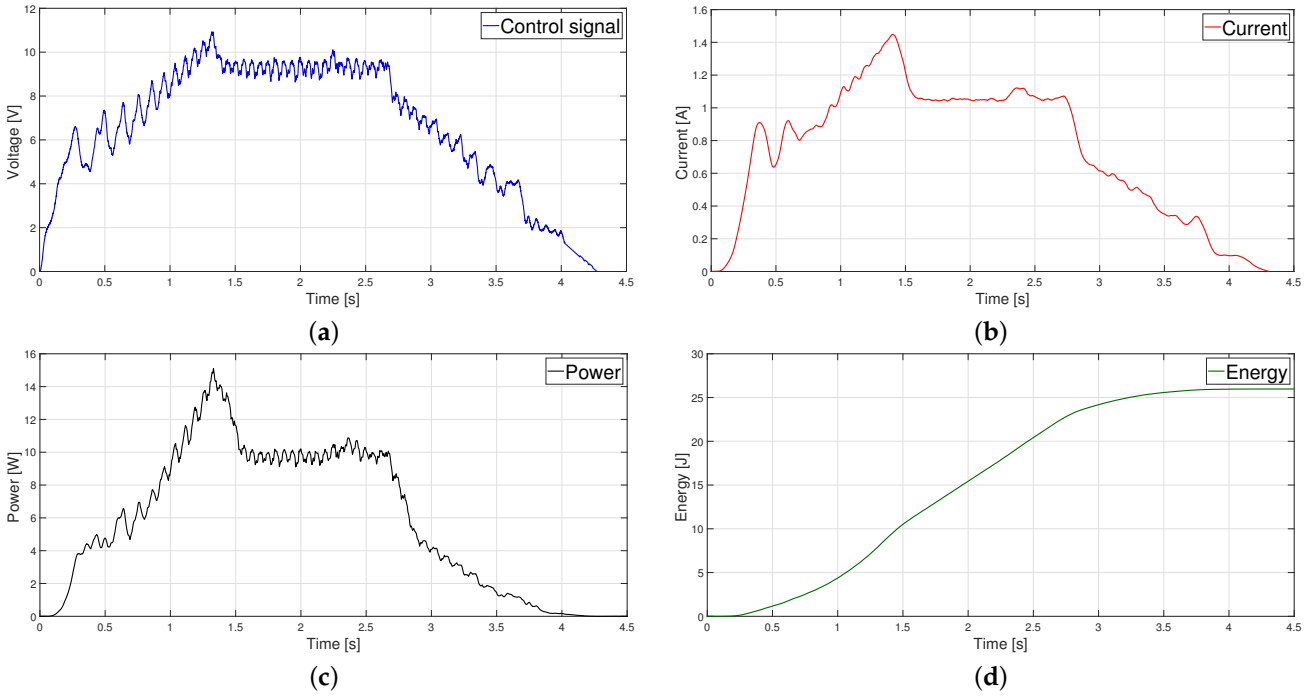


Figure 18. Trapezoidal velocity profile: (**a**) Control signal, (**b**) Current (**c**) Power, and (**d**) Energy.

5.4. S-Curve Profile without Load

Table 5 presents the selected and calculated parameters for generating the S-curve velocity profile. The maximum velocity obtained from 0.333 $\frac{m}{s}$ is the largest of the three profiles. In contrast, the maximum acceleration of 0.277 $\frac{m}{s^2}$ is more significant than in the trapezoidal but less than in the parabolic profile.

Table 5. Parameters used to implement the S-curve velocity profile.

Parameter	Value
Final position (x_f)	0.8 m
Total time for displacement (T)	4 s
Time factor for acceleration (α)	0.4
Time factor for jerk phase (β)	0.25
Acceleration Time (T_a)	1.6 s
Jerk pulse width (T_j)	0.4 s
Maximum velocity (x_{max})	0.333 $\frac{m}{s}$
Maximum acceleration ($\ddot{x}_{max}$)	0.277 $\frac{m}{s^2}$
Maximum jerk ($\dddot{x}_{max}$)	0.694 $\frac{m}{s^3}$

Figure 19a shows an approximation of the position of the carriage to that of the calculated profile, which due to numerical integration, has a 0.4% concerning the final position. However, a reduction of the following error to less than 1% in 0.8 s is presented in Figure 19b. Similarly, in Figure 19c, the maximum velocity of 0.35 $\frac{m}{s}$, is close to 0.333 $\frac{m}{s}$ of the profile of speed. Regarding the acceleration graphs of Figure 19d, the acceleration measured does not exhibit the abrupt changes observed in the two previous profiles.

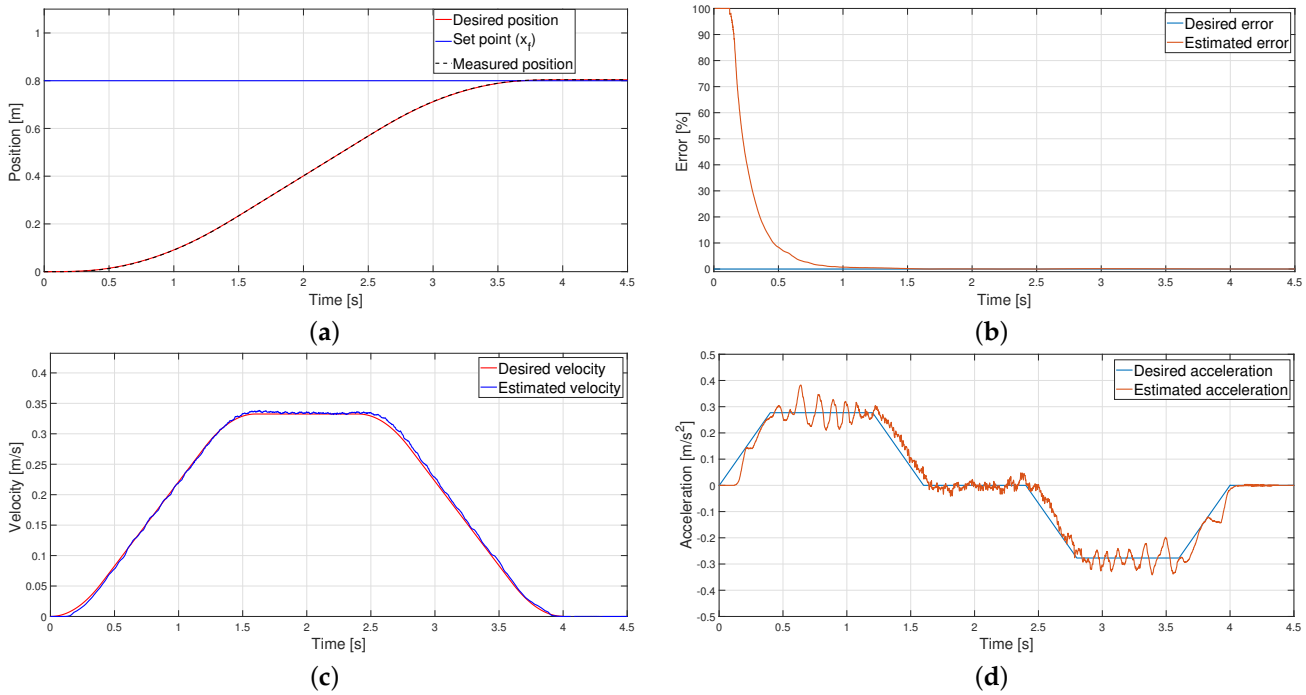


Figure 19. S-curve velocity profile: (**a**) Position, (**b**) Error (**c**) Velocity, and (**d**) Acceleration.

The control signal generated for the S-curve velocity profile in Figure 20a has a maximum voltage of 11 V. This is the profile with the highest maximum voltage in the control signal and the one that reaches the highest speed during the execution of the movement.

As for the current shown in Figure 20b, the maximum peak presented was 1.54 A, which gradually reduced to 1.1 A during the constant velocity phase. The power and energy graphs are shown in Figure 20c and Figure 20d, respectively. The total energy consumed is 26.952 J.

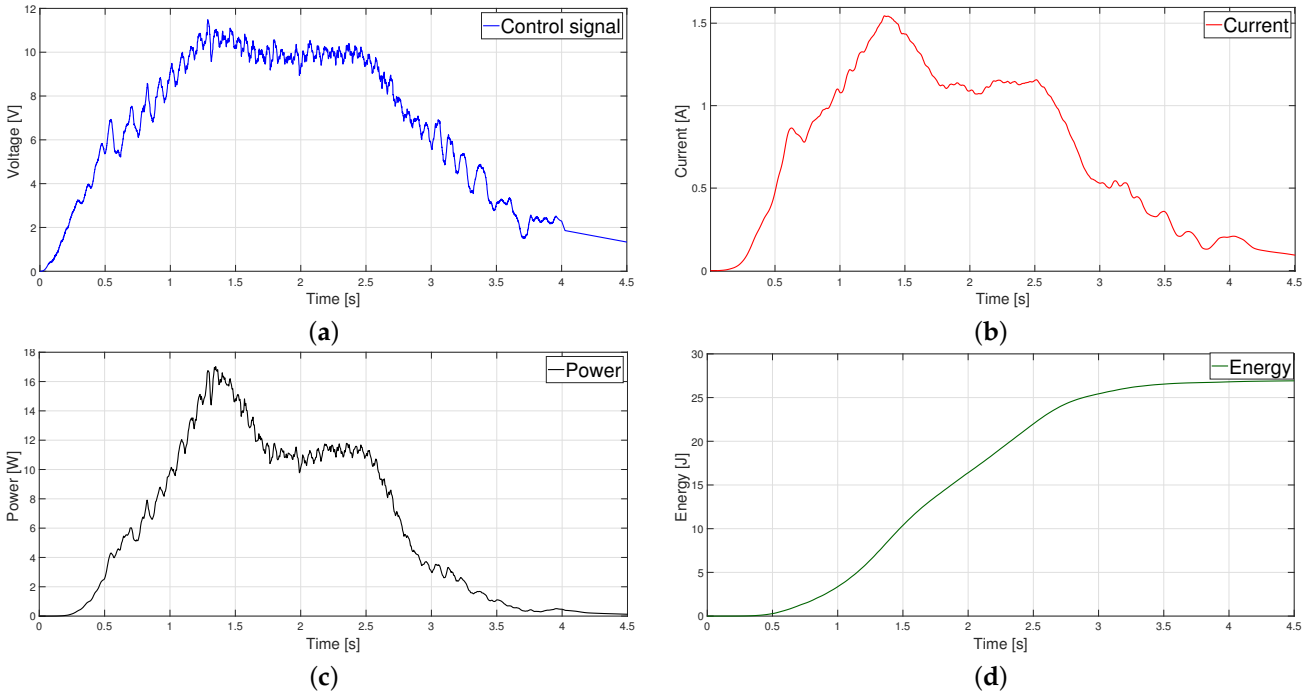


Figure 20. S-curve velocity profile: (**a**) Control signal, (**b**) Current (**c**) Power, and (**d**) Energy.

Table 6 displays the average execution time of the entire system compounded by the motion profiles computed in real time. The S-curve is the profile that requires the most computational time, with an average of 0.000007207 s, while the profile that requires the least processing time is the trapezoidal, with an average time of 0.000004179 s.

Table 6. Execution times for each motion profile.

Distance (Δx)	Parabolic Profile	Trapezoidal Profile	S-Curve Profile
80 m	0.00000425 s	0.000004179 s	0.000007207 s

5.5. Experimentation with Load

This section reports the experimental results of adding a load of 9 kg to the carriage. First, the tracking error for each profile is compared in Figure 21. The error in the S-curve profile shows that an error of 0.003 m at the final position is obtained, which is less than 0.5% of the total displacement. As for the parabolic and trapezoidal profile, the error is less than 1%. In addition, Figure 22 shows the controller's robustness since, despite adding a load to the system, the carriage maintains the position for each profile under test.

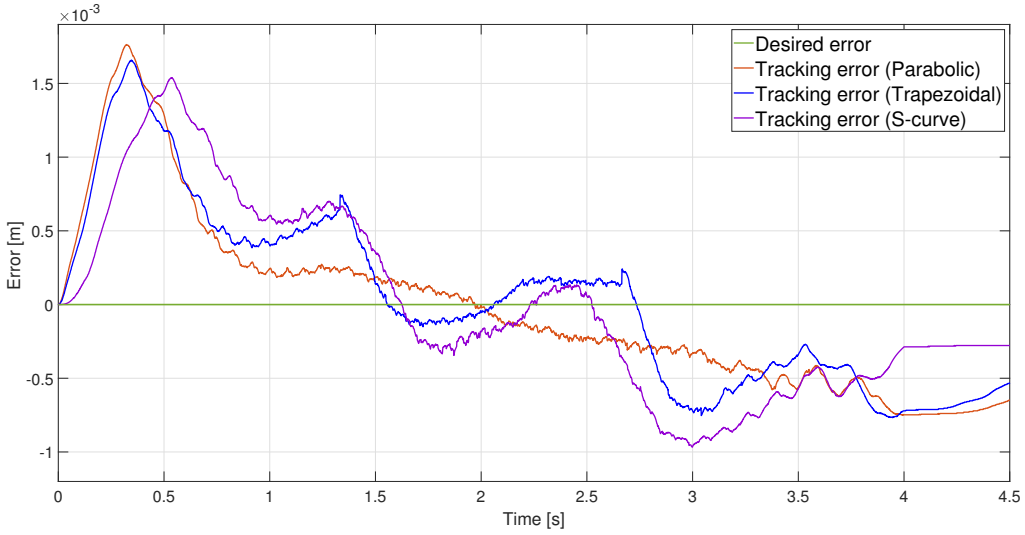


Figure 21. Real error position with a load of the three position profiles.

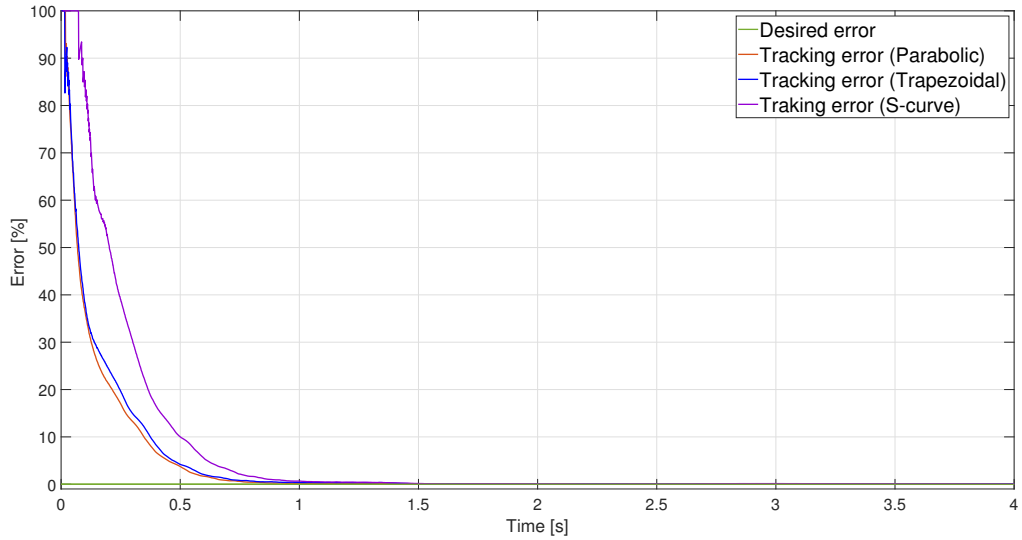


Figure 22. Error with a load of the three position profiles.

The control signal computed for each of the three profiles when a load is placed on the carriage is presented in Figure 23. In the case of the parabolic profile, the control signal reaches a maximum value of 9.8 V. For the trapezoidal profile, a maximum voltage of 11 V is presented at $t = T_a$ due to the discontinuity in acceleration. The voltage delivered is 9.6 V in the constant speed phase. The S-curve profile shows the values of voltage maximums: 11.2 V in the acceleration phase and 10.2 V in the constant speed phase. By comparing these values with those of the no-load case, we observed that the control signal of the loaded case presented higher voltage values than the no-load case.

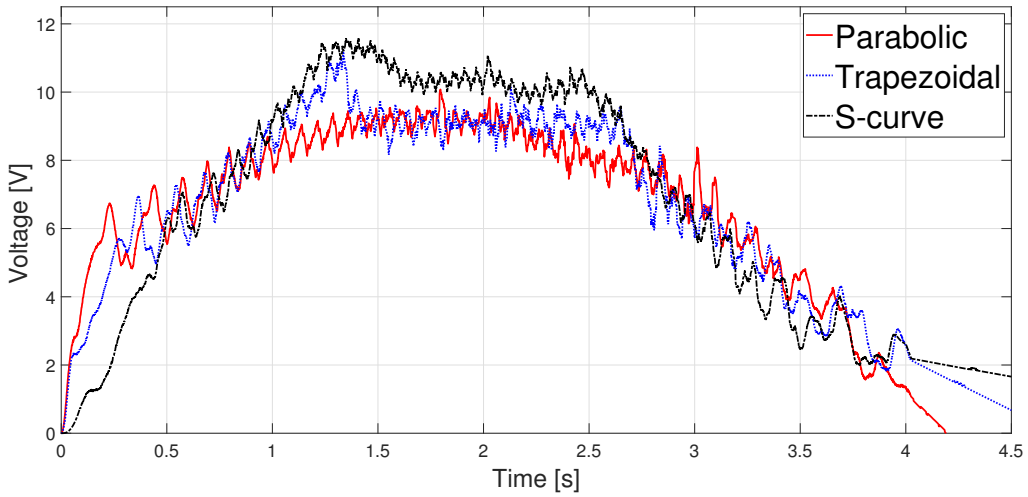


Figure 23. Control signal with a load of 9 kg.

Figure 24 shows the current consumption of the three implemented velocity profiles. The graph of the current corresponding to the parabolic profile shows that the maximum current is 1.11 A. While for the trapezoidal velocity profile, the maximum current is 1.58 A. Finally, using the S-curve type trajectory, a maximum current of 1.77 A has the highest current amplitude of the three profiles.

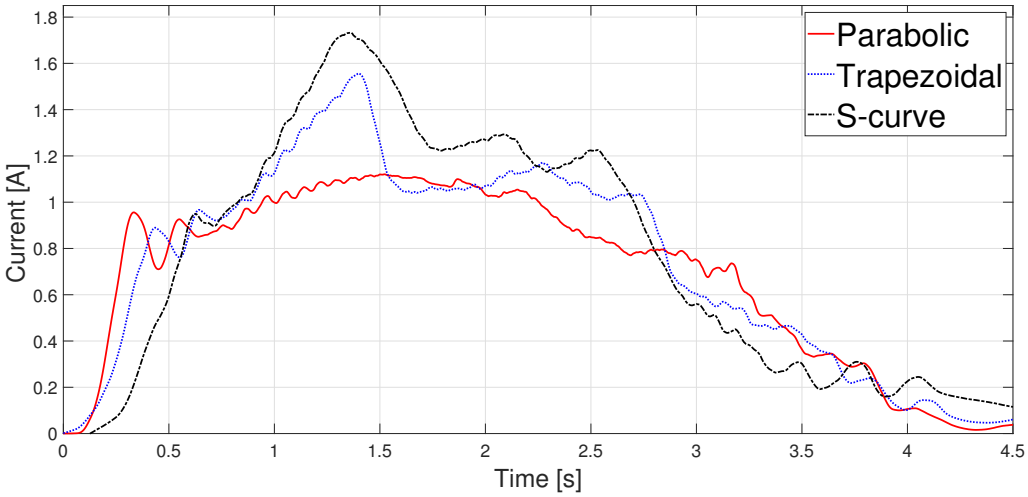


Figure 24. Current measured with a load of 9 kg.

Figure 25 depicts the instantaneous power of the three profiles. It is observed again that the S-curve velocity profile gives the highest dissipated power, while the parabolic profile is the lowest since the latter presented lower voltage and current consumption than the other two profiles.

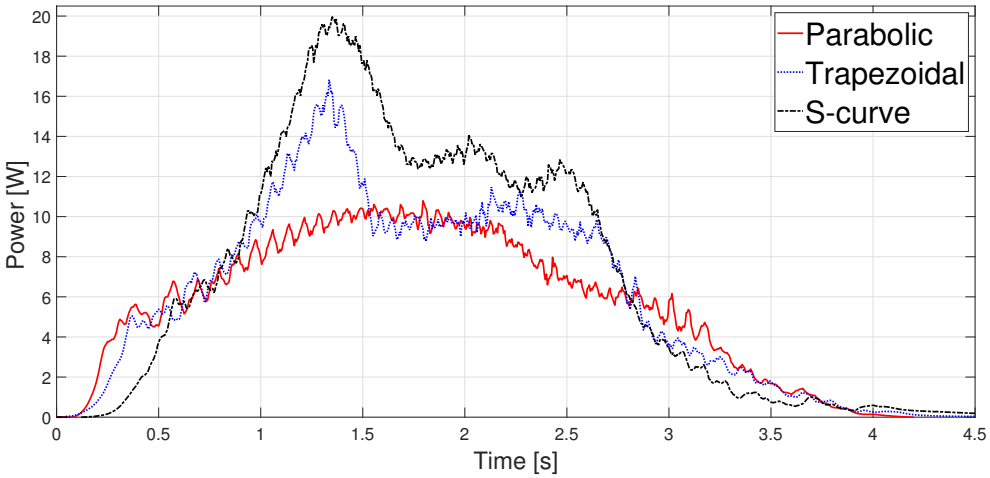


Figure 25. Power estimated with a load of 9 kg.

Finally, Figure 26 displays the energy consumption of the three motion profiles. From the graph, it can be seen that the profile with the highest energy consumption is the S-curve with 30.645 J, followed by the trapezoidal speed profile with 26.761 J and the parabolic with 23.845 J, the latter being the one with the lowest consumption of the three profiles implemented.

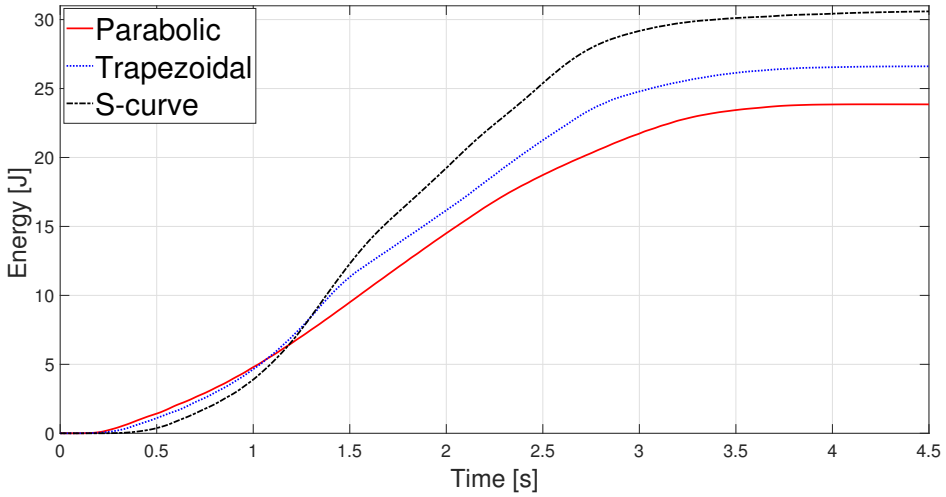


Figure 26. Energy estimated with a load of 9 kg.

Table 7 presents a comparative chart of the total energy consumed in each profile when no load was placed on the carriage and when it was used. In both circumstances, 40 tests for each profile were carried out. As can be seen, the energy consumption is higher in the S-curve profile and lower in the parabolic velocity profile. Comparing the results presented in [7,28], the system's robustness is demonstrated due to the load tests carried out in this work.

Table 7. Energy comparison of the three motion profiles.

Motion Profile	Energy (without Load)	Energy (with Load)
Parabolic	23.7044 J	23.845 J
Trapezoidal	25.9644 J	26.7616 J
S-curve	26.9525 J	30.6456 J

6. Conclusions

This paper focused on a linear platform that underwent an open structure design for a motion controller and a profile generator. These implementations were carried out on an embedded system, using C/C++ programming languages, to analyze energy consumption concerning three specific motion profiles. The paper also includes tests to evaluate the strength and robustness of the control system using a series of step responses, motion profiles with and without load, and when a specific weight is applied. Furthermore, a methodology is proposed that allows the designing and implementation of an open-source motion control algorithm for a linear plant composed of a direct current motor coupled to an endless screw to measure the energy consumption of trajectories. The control algorithm is based on a classic PID controller whose reference signal is given by a motion profile generator. The selected embedded system allowed a sampling time of 0.001 s, with which an adequate system response for controlling direct current motors is obtained. The motion profiles used in this work were of the polynomial type: trapezoidal, parabolic, and S-curve. The same displacement length h was considered for calculating each of the trajectories and the exact total movement time T. The kinematic magnitudes of the three profiles, such as position, velocity, and acceleration, were compared during the experimentation. The S-curve profile presented the most significant magnitude in speed, and the parabolic profile had the highest acceleration. However, the main characteristic analyzed was the energy consumption, which required analyzing the control signal, the measured current, and the instantaneous power to be estimated using numerical integration techniques. The results showed that the parabolic speed profile presented the lowest amplitude in the control signal, current, and power. In contrast, the S-curve speed profile was the one that presented the highest amplitude in the same magnitudes mentioned. Table 7 shows that the highest energy consumption was dissipated using the S-curve profile, while the parabolic trajectory consumed less than the other two. In fact, after carrying out 40 tests for each profile, the energy consumption in the parabolic profile was 10.89% less than the trapezoidal profile and 22.19% less than the S-curve profile. In a second case, the same analysis was carried out, placing a load of 9 kg on the carriage; the robustness of the controller allowed for obtaining similar results in position, velocity, and acceleration for each profile compared to the case with no load. The difference identified was in the control signal and current consumed, translating into higher energy consumption. Nevertheless, from Table 7, it is clear that the parabolic velocity profile also had the lowest energy consumption despite adding the load. Finally, the iteration time for the parabolic, trapezoidal, and S-curve motion profile was 0.00000425 s, 0.000004179 s, and 0.000007202 s, respectively, making the embedded system suitable for hard real-time motion control applications. Based on the results, trajectory planning is an effective technique easy to implement in low-cost microcontrollers to reduce energy consumption in motion control systems. In future studies, the S-curve parameters will be modified to reduce the maximum velocity and analyze the energy consumption behavior. Finally, a fuzzy controller will be used so as not to require the mathematical model of the plant.

Author Contributions: Conceptualization, L.F.O.-G. and J.R.G.-M.; methodology, L.F.O.-G.; software, L.F.O.-G. and E.E.C.-M.; validation, L.F.O.-G., J.R.G.-M. and E.E.C.-M.; formal analysis, J.R.G.-M.; investigation, L.F.O.-G. and O.A.B.-V.; resources, J.R.G.-M., E.E.C.-M., O.A.B.-V., M.G.-L. and T.M.-S.; data curation, M.G.-L. and T.M.-S.; writing—original draft preparation, J.R.G.-M., M.G.-L. and T.M.-S.; writing—review and editing, J.R.G.-M.; visualization, M.G.-L. and T.M.-S.; supervision, J.R.G.-M.; project administration, J.R.G.-M. and L.F.O.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by CONAHCYT grant number 1271936.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The first author thanks the Faculty of Mechanics and Electrical of the Universidad Veracruzana, Poza Rica–Tuxpan Region, for the support provided by the Master of Science in Engineering, of which he is a student during the period February 2023–December 2024. In addition, all the authors thank the Faculty of Electronics and Communications Engineering, Universidad Veracruzana, for using the Control and Robotics Laboratory to conduct the experimentation.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PID	Proportional-Integral-Derivative
DC	Direct Current
ARM	Advanced RISC Machine
ADC	Analog-to-Digital converter
PWM	Pulse Width Modulation
GPIO	General-Purpose Input/Output
GPTM	General-Purpose Timer
UART	Universal Asynchronous Receivers/Transmitter
QEI	Quadrature Encoder Interface
PPR	Pulses per revolution

Appendix A

Table A1. Main features of the TM4C123GH6PM microcontroller.

Feature	Description
Core	ARM Cortex-M4F
Performance	80-MHz operation
Universal Asynchronous Receivers/Transmitter (UART)	8 Modules
General-Purpose Timer (GPTM)	6–16/32-bit GPTM blocks and six 32/64-bit Wide GPTM blocks
General-Purpose Input/Output (GPIO)	6 physical GPIO blocks
Pulse Width Modulator	2 PWM modules, each with four PWM generator blocks and a control block, for 16 PWM outputs.
Quadrature Encoder Interface (QEI)	2 QEI modules
Analog-to-Digital Converter (ADC)	2–12-bit ADC modules, with a maximum sample rate of one million samples/second

References

1. Wu, Z.; Chen, J.; Bao, T.; Wang, J.; Zhang, L.; Xu, F. A Novel Point-to-Point Trajectory Planning Algorithm for Industrial Robots Based on a Locally Asymmetrical Jerk Motion Profile. *Processes* **2022**, *10*, 728. [CrossRef]
2. Carabin, G.; Scalera, L. On the trajectory planning for energy efficiency in industrial robotic systems. *Robotics* **2020**, *9*, 89. [CrossRef]
3. Boryga, M.; Kołodziej, P.; Gołacki, K. The Use of Asymmetric Polynomial Profiles for Planning a Smooth Trajectory. *Appl. Sci.* **2022**, *12*, 12284. [CrossRef]
4. Wu, G.; Zhang, S. Real-time jerk-minimization trajectory planning of robotic arm based on polynomial curve optimization. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2022**, *236*, 10852–10864. [CrossRef]
5. Sang, W.; Sun, N.; Zhang, C.; Qiu, Z.; Fang, Y. Hybrid Time-Energy Optimal Trajectory Planning for Robot Manipulators With Path and Uniform Velocity Constraints. In Proceedings of the 2022 13th Asian Control Conference (ASCC), Jeju, Republic of Korea, 4–7 May 2022; pp. 334–340.
6. Kröger, T. *On-Line Trajectory Generation in Robotic Systems: Basic Concepts for Instantaneous Reactions to Unforeseen (Sensor) Events*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 11–29.

7. Martínez, J.R.G.; Reséndiz, J.R.; Prado, M.Á.M.; Miguel, E.E.C. Assessment of jerk performance s-curve and trapezoidal velocity profiles. In Proceedings of the 2017 XIII International Engineering Congress (CONIIN), Santiago de Queretaro, Mexico, 15–19 May 2017; pp. 1–7.
8. Ha, C.W.; Lee, D. Analysis of Embedded Pre-filters in Motion Profiles. *IEEE Trans. Ind. Electron.* **2017**, *65*, 1481–1489. [CrossRef]
9. Mejerbi, M.; Knani, J. Influence of the motion profile on the performance of a flexible arm. In Proceedings of the 2017 International Conference on Control, Automation, and Diagnosis (ICCAD), Hammamet, Tunisia, 19–21 January 2017; pp. 76–81.
10. Alpers, B. On Fast Jerk-Continuous Motion Functions with Higher-Order Kinematic Restrictions for Online Trajectory Generation. *Robotics* **2022**, *11*, 73. [CrossRef]
11. Nguyen, K.D.; Chen, I.M.; Ng, T.C. Planning algorithms for s-curve trajectories. In Proceedings of the 2007 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Zurich, Switzerland, 4–7 September 2007; pp. 1–6.
12. Lee, D.; Ha, C.W. Optimization process for polynomial motion profiles to achieve fast movement with low vibration. *IEEE Trans. Control Syst. Technol.* **2020**, *28*, 1892–1901. [CrossRef]
13. Nam, N.Q. Characteristics of S-curve motion profile for all ranges of motion length and limits. In Proceedings of the 2020 International Conference on Advanced Mechatronic Systems (ICAMechS), Hanoi, Vietnam, 10–13 December 2020; pp. 214–220.
14. Rubio, F.; Lopis-Albert, C.; Valero, F.; Suñer, J.L. Industrial robot efficient trajectory generation without collision through the evolution of the optimal trajectory. *Robot. Auton. Syst.* **2016**, *86*, 106–112. [CrossRef]
15. Van Oosterwyck, N.; Vanbecelaere, F.; Knaepkens, F.; Monte, M.; Stockman, K.; Cuyt, A.; Derammelaere, S. Energy Optimal Point-to-Point Motion Profile Optimization. *Mech. Based Des. Struct. Mach.* **2022**, 1–18. [CrossRef]
16. Chamraz, Š.; Balogh, R. Optimal structure for the trajectory controller. In Proceedings of the 2016 Cybernetics & Informatics (K&I), Levoca, Slovakia, 2–5 February 2016; pp. 1–5.
17. Al-khayyt, S.Z. Comparison between fuzzy logic-based controllers for robot manipulator trajectory tracking. In Proceedings of the 2012 First National Conference for Engineering Sciences (FNCES), Baghdad, Iraq, 7–8 November 2012; pp. 1–6.
18. Lin, C.Y. Neural network adaptive control and repetitive control for high-performance precision motion control. In Proceedings of the SICE Annual Conference 2010, Taipei, Taiwan, 18–21 August 2010; pp. 2843–2844.
19. Biagiotti, L.; Melchiorri, C. Optimization of generalized S-curve trajectories for residual vibration suppression and compliance with kinematic bounds. *IEEE/ASME Trans. Mechatronics* **2020**, *26*, 2724–2734. [CrossRef]
20. Rymansaib, Z.; Iravani, P.; Sahinkaya, M.N. Exponential trajectory generation for point to point motions. In Proceedings of the 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Wollongong, NSW, Australia, 9–12 July 2013; pp. 906–911.
21. Scalera, L.; Carabin, G.; Vidoni, R.; Gasparetto, A. Minimum-energy trajectory planning for industrial robotic applications: Analytical model and experimental results. In *Advances in Service and Industrial Robotics: Results of RAAD*; Zeghloul, S., Laribi, M.A., Arevalo, J.S.S., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 334–342.
22. Gasparetto, A.; Boscariol, P.; Lanzutti, A.; Vidoni, R. Path planning and trajectory planning algorithms: A general overview. In *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*; Carbone, G., Gomez-Bravo, F., Eds.; Springer International Publishing: Switzerland, 2015; pp. 3–27.
23. Fang, Y.; Hu, J.; Liu, W.; Shao, Q.; Qi, J.; Peng, Y. Smooth and time-optimal S-curve trajectory planning for automated robots and machines. *Mech. Mach. Theory* **2019**, *137*, 127–153. [CrossRef]
24. Alpers, B. On Fast Jerk–, Acceleration– and Velocity–Restricted Motion Functions for Online Trajectory Generation. *Robotics* **2021**, *10*, 25. [CrossRef]
25. Wang, H.; Wang, H.; Huang, J.; Zhao, B.; Quan, L. Smooth point-to-point trajectory planning for industrial robots with kinematical constraints based on a high-order polynomial curve. *Mech. Mach. Theory* **2019**, *139*, 284–293. [CrossRef]
26. Carabin, G.; Wehrle, E.; Vidoni, R. A Review on Energy-Saving Optimization Methods for Robotic and Automatic Systems. *Robotics* **2017**, *6*, 39. [CrossRef]
27. Assad, F.; Rushforth, E.; Ahmad, B.; Harrison, R. An Approach of Optimising S-curve Trajectory for a Better Energy Consumption. In Proceedings of the 2018 IEEE 14th International Conference on Automation Science and Engineering (CASE), Munich, Germany, 20–24 August 2018; pp. 98–103.
28. Montalvo, V.; Estévez-Bén, A.A.; Rodríguez-Reséndiz, J.; Macias-Bobadilla, G.; Mendiola-Santíbañez, J.D.; Camarillo-Gómez, K.A. FPGA-Based Architecture for Sensing Power Consumption on Parabolic and Trapezoidal Motion Profiles. *Electronics* **2020**, *9*, 1301. [CrossRef]
29. Carabin, G.; Vidoni, R. Energy-saving optimization method for point-to-point trajectories planned via standard primitives in 1-DoF mechatronic systems. *Int. J. Adv. Manuf. Technol.* **2021**, *116*, 331–344. [CrossRef]
30. Hosseini, S.; Hahn, I. Energy-efficient motion planning for electrical drives. In Proceedings of the 2018 IEEE Electrical Power and Energy Conference (EPEC), Toronto, ON, Canada, 10–11 October 2018; pp. 1–7.
31. Nshama, E.W.; Msukwa, M.R.; Uchiyama, N. A trade-off between energy saving and cycle time reduction by Pareto optimal corner smoothing in industrial feed drive systems. *IEEE Access* **2021**, *9*, 23579–23594. [CrossRef]
32. Halinga, M.S.; Nyobuya, H.J.; Uchiyama, N. Generation of Time and Energy Optimal Coverage Motion for Industrial Machines Using a Modified S-Curve Trajectory. In Proceedings of the 2023 IEEE/SICE International Symposium on System Integration (SII), Atlanta, GA, USA, 17–20 January 2023; pp. 1–6.

33. Biagiotti, L.; Melchiorri, C. *Trajectory Planning for Automatic Machines and Robots*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 1–13.
34. Ali, S.A.; Annuar, K.A.M.; Miskon, M.F. Trajectory planning for exoskeleton robot by using cubic and quintic polynomial equation. *Int. J. Appl. Eng. Res.* **2016**, *11*, 7943–7946.
35. Moritz, F.G. *Electromechanical Motion Systems: Design and Simulation*, 1st ed.; John Wiley & Sons: West Sussex, UK, 2014; pp. 167–194.
36. Heo, H.-J.; Son, Y.; Kim, J.-M. A Trapezoidal Velocity Profile Generator for Position Control Using a Feedback Strategy. *Energies* **2019**, *12*, 1222. [CrossRef]
37. García-Martínez, J.R.; Cruz-Miguel, E.E.; Carrillo-Serrano, R.V.; Mendoza-Mondragón, F.; Toledano-Ayala, M.; Rodríguez-Reséndiz, J. A PID-type fuzzy logic controller-based approach for motion control applications. *Sensors* **2020**, *20*, 5323. [CrossRef]
38. Dini, P.; Saponara, S. Processor-in-the-loop validation of a gradient descent-based model predictive control for assisted driving and obstacles avoidance applications. *IEEE Access* **2022**, *10*, 67958–67975. [CrossRef]
39. Dini, P.; Ariaudo, G.; Botto, G.; La Greca, F.; Saponara, S. Real-time electro-thermal modelling & predictive control design of resonant power converter in full electric vehicle applications. *IET Power Electron.* **2023**, *16*, 2045–2064.
40. Bernardeschi, C.; Dini, P.; Domenici, A.; Palmieri, M.; Saponara, S. Formal verification and co-simulation in the design of a synchronous motor control algorithm. *Energies* **2020**, *13*, 4057. [CrossRef]
41. Dini, P.; Saponara, S. Design of adaptive controller exploiting learning concepts applied to a BLDC-based drive system. *Energies* **2020**, *13*, 2512. [CrossRef]