

# Diseño de un Perfil de Movimiento Trapezoidal a la Implementación Física de un Prototipo de Robot Cartesiano 3P

Forero Salas, Andrés Felipe<sup>1</sup> (\*); Ramírez Montes, Juan Fernando<sup>1</sup>; Blanco Cañón, Robín Alfonso<sup>1</sup>

<sup>1</sup>Universidad Nacional de Colombia, Facultad de ingeniería, Bogotá, Colombia

**Resumen:** En este artículo se lleva a cabo el análisis detallado de un sistema robótico cartesiano 3P, que constituye un prototipo de parqueadero automatizado a escala, haciendo uso del microcontrolador de 8 bits PIC18F4550, el recorrido parte del desarrollo del perfil trapezoidal utilizado, su posterior discretización y parametrización con el fin de hacerlo implementable en el microcontrolador. Seguidamente, se muestra el algoritmo, así como las consideraciones a tener presentes dentro de la programación del dispositivo empleando el compilador de XC8®, junto con los resultados obtenidos. Finalmente se hace una comparativa entre las ventajas de utilizar el perfil trapezoidal y las posibles mejoras a implementar acorde con el hardware utilizado de tal forma que pueda ser aprovechado en entornos académicos donde se realizan los primeros acercamientos al diseño de perfiles de movimiento para trabajos con una capacidad de cómputo limitado.

**Palabras clave:** Modelado. Perfil de Movimiento Trapezoidal. Sistema Robótico Cartesiano. Microcontrolador PIC18F4550.

Recibido: 25 de noviembre de 2022. Aceptado: 20 de febrero de 2023

Received: November 25th, 2022. Accepted: February 20th, 2023

## Design of a Trapezoidal Motion Profile to the Physical Implementation of a 3P Cartesian Robot Prototype

**Abstract:** This paper presents a detailed overview of a 3P cartesian robotic system, represented in a scaled automated car park prototype, employing the 8 bits microcontroller PIC18F4550, the process started with the development of the used trapezoidal profile, continuing with its later discretization and its parameterization in order to implement it to the microcontroller. Secondly, it is shown the algorithm and the considerations considered during the programming of the dispositive with the XC8® compiler. Finally, it is made a comparative between the advantages of employing a trapezoidal profile and the possible enhancements to be implemented according to the available hardware in such a way that it can be used in academic environments especially during the designs of motion profiles for prototypes with limited computational capacity.

**Keywords:** Modelling. Trapezoidal Motion Profile. Cartesian Robotic System. Microcontroller PIC18F4550.

---

(\*) [anforero@unal.edu.co](mailto:anforero@unal.edu.co)

## 1. INTRODUCCIÓN

Los problemas afrontados en la actualidad con frecuencia requieren de un desarrollo transversal a diferentes ramas del conocimiento, un ejemplo de ello son los estacionamientos automatizados, una idea que se ha venido desarrollando durante los últimos años y que su propuesta se enfoca en solucionar el problema de espacio en las ciudades altamente pobladas permitiendo un correcto aprovechamiento de los lugares destinados para estacionar automóviles. De la misma forma reduciendo la exposición de los conductores a gases tóxicos como el monóxido de carbono que se concentran en los estacionamientos cerrados.

El presente informe, presenta el estudio del proceso matemático detrás de la definición de las curvas de movimiento seguidas por la grúa que ubica a los automóviles en un prototipo de un estacionamiento subterráneo automatizado, así como el proceso de interpolación que se realiza con el fin de mover de manera sincrónica los ejes horizontal y vertical del sistema controlado con un PIC18F4550.

## 2. MARCO TEÓRICO

Dentro del mundo de la automatización es frecuente hablar del control de movimiento, ya que buena parte de los procesos que se llevan a cabo en entornos industriales y productivos requieren del transporte y modificación mecánica de la materia a partir de unos procesos definidos. Dentro de las tareas a cumplir una de las más frecuentes es el transporte eficiente de productos por lo cual se hace indispensable el diseño de trayectorias y perfiles de movimiento (Yoon), los perfiles de movimiento han sido explorados a lo largo de la historia dado que a partir de la definición de una posición deseada en un tiempo dado es posible desarrollar un análisis completo haciendo uso del cálculo para obtener información valiosa como puede ser la velocidad, aceleración e incluso el Jerk, los cuales definen el comportamiento de los sistemas dinámicos, a la vez que nos dan una imagen del comportamiento posible de las fuerzas y esfuerzos implicados dentro del mecanismo, así como los efectos que el movimiento puede tener sobre la carga a ser transportada (Yu).

Teniendo en cuenta la importancia de los perfiles de movimiento, se hace necesaria la búsqueda de funciones que ofrezcan un impacto mínimo sobre el mecanismo y la carga en términos de fuerza a la vez que se cumple con un requerimiento de tiempo. Este problema ha sido ampliamente explorado, y si bien se tienen resultados que pueden minimizar el impacto sobre el mecanismo de manera satisfactoria aparece otro problema a resolver, y es el que se encuentra asociado a la capacidad de cómputo de los controladores más utilizados en los motores (Yoon). En el momento en que se busca una función más suavizada resulta inevitable el uso de funciones más complejas las cuales pueden llegar a tener un costo computacional tan alto que no justifique dicho desarrollo a causa de su poca practicidad en entornos de control de movimiento. Por esta razón se busca un equilibrio entre el coste computacional y el impacto que las funciones de

movimiento pueden tener sobre el mecanismo. La pregunta natural sería ¿Cómo se obtiene ese equilibrio? Pues bien, la realidad es que dependerá de cada aplicación, pero basados en la bibliografía es frecuente utilizar perfiles trapezoidales o trapezoidales suavizados (Heo), los cuales son relativamente fáciles de parametrizar, su costo computacional es bajo y a la vez cumple con los requisitos mínimos para mitigar el impacto que puede tener las fuerzas asociadas al movimiento sobre los mecanismos (Rodríguez).

Dentro de este trabajo se ha decidido utilizar un microcontrolador PIC18F4550 para implementar la trayectoria trapezoidal, ya que estos dispositivos se encuentran entre la gama de unidades de procesamiento más elementales que cumplen con múltiples tareas dentro de la industria. El desarrollo en esta plataforma busca mostrar la capacidad que pueden tener estos dispositivos dentro de las aplicaciones donde sea necesario definir perfiles de movimiento haciendo una implementación funcional a la vez que económica (Chekired), finalmente el aporte más importante que se espera lograr con este trabajo es acercar estos conceptos a espacios académicos como podría ser un curso de microcontroladores. Todo esto con el fin de que estudiantes de carreras relacionadas con la automatización puedan implementar este tipo de soluciones con un enfoque práctico, el cual será de utilidad en su vida profesional.

## 3. METODOLOGÍA

El proceso se realizó en tres etapas correspondientes al diseño cinemático del perfil, desarrollo del algoritmo de programación del microcontrolador y construcción física del prototipo.

### 3.1 Diseño cinemático del perfil de movimiento trapezoidal

Para establecer un modelo adecuado se parte del comportamiento de las curvas de movimiento, analizando principalmente la curva de velocidad. El objetivo es que el cabezal que lleva a los carros se mueva a lo largo del plano reduciendo los casos donde se presenten cambios muy bruscos en la velocidad, teniendo en cuenta que en un prototipo esto produce un mayor desgaste en los motores; y podría causar afectaciones mayores en caso de ser implementado a escala real (García).

Por lo anterior, se establece un perfil de movimiento trapezoidal en velocidad, según la Figura 1 para el movimiento en los ejes horizontal y vertical del cabezal. A la hora de implementar un perfil de movimiento de este tipo existe una infinidad de posibilidades, ya que al modificar cada uno de los tiempos, así como la velocidad cruceo ( $V_c$ ), se obtienen perfiles muy variados. Se decidió implementar un perfil en el que el proceso de aceleración y desaceleración tardaran el mismo tiempo. Adicionalmente, buscando un perfil de velocidad armónico, se decidió que el 50 % del desplazamiento se realizará a velocidad constante. De esta forma, se encuentra una relación para los tiempos que definen el perfil en las Ecuaciones (1) y (2).

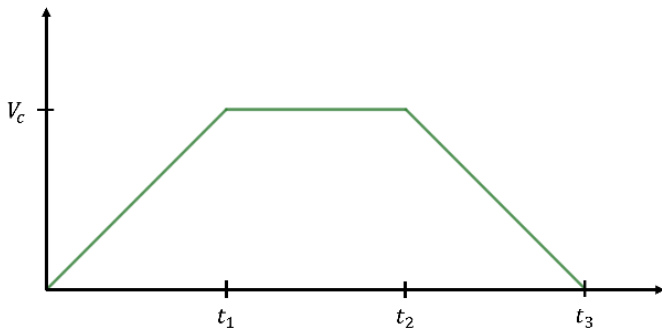


Figura 1. Perfil de velocidad trapezoidal. Fuente propia.

$$t_2 = 2 \cdot t_1 \quad (1)$$

$$t_3 = 3 \cdot t_1 \quad (2)$$

Para definir el parámetro de tiempo ( $t_1$ ), se deben tomar consideraciones respecto al desplazamiento más largo que realizará el mecanismo, la frecuencia máxima para la señal de paso permitida por el amplificador utilizado para los motores, en este caso el DRV8825, reportada en la hoja de datos como 250 kHz (Texas Instruments). Así como la velocidad máxima del motor Nema17, que dependiendo del modelo puede llegar hasta las 8 000 RPM (Cui Devices).

Teniendo en cuenta que se configuró el controlador para que el flanco ascendente de la señal cuadrada ordene al motor realizar una rotación de un octavo de paso, que para efectos de este escrito serán llamados indistintamente como pasos; se define la frecuencia máxima para la señal cuadrada según la Ecuación (3).

$$f_{\text{máx}} = 2\,400 \frac{\text{pasos}}{\text{s}} \quad (3)$$

Conocida esta frecuencia, se encuentra el valor límite para el tiempo  $t_1$ , al considerar que el desplazamiento máximo del cabezal se caracteriza en 20 000 pasos, obteniéndose la Expresión (4).

$$f_{\text{máx}} = \frac{\text{pasos}_{\text{máx}}}{2 \cdot t_1} \quad (4)$$

Al reemplazar los valores en la Ecuación 4 se encuentra el intervalo para el parámetro  $t_1$  según la Inecuación (5).

$$t_1 \geq 4,166 \text{ s} \quad (5)$$

Definiéndose así el tiempo  $t_1$  a partir de la Ecuación (6).

$$t_1 = 4,5 \text{ s} \quad (6)$$

Al establecer la duración de los diferentes intervalos del perfil con las Ecuaciones (1), (2) y (6); sólo queda parametrizar la velocidad cruce de este en base al número de pasos requeridos en cada una de las direcciones de movimiento, a través de la Expresión (7).

$$V_c = \frac{\text{pasos}}{2 \cdot t_1} \quad (7)$$

De esta forma, el parámetro de velocidad constante será independiente para los perfiles de movimiento horizontal y vertical, lo que permite la interpolación del movimiento al contar con los mismos intervalos temporales de aceleración, velocidad cruce y desaceleración.

### 3.2 Desarrollo e implementación del algoritmo de programación para el microcontrolador

La utilización de un microcontrolador supone la discretización del perfil de movimiento. De esta manera, los periodos de cambio de velocidad se dividen en 20 intervalos constantes, véase la Figura 2. Utilizando una aproximación rectangular de la curva de punto final para la aceleración y de punto inicial para la desaceleración, obteniéndose un perfil simétrico en aceleración y desaceleración.

Al discretizar el perfil, se omiten algunos pasos durante los procesos de cambio de velocidad, por lo que, para obtener el desplazamiento requerido, se debe aumentar la duración del intervalo de movimiento a velocidad constante. La duración de este periodo se calcula a través de la Ecuación (8).

$$t_{vc} = \frac{\text{pasos} - \text{pasos}_{\text{ace}} - \text{pasos}_{\text{des}}}{V_c} \quad (8)$$

Teniendo en cuenta, que durante la aceleración y desaceleración el desplazamiento es el mismo, la Ecuación (8) se reescribe bajo la Expresión (9) al tener en cuenta que los procesos de cambios de velocidad son idénticos.

$$t_{vc} = \frac{\text{pasos} - 2 \cdot \text{pasos}_{\Delta v}}{V_c} \quad (9)$$

En base a la discretización, el desplazamiento durante los cambios de velocidad puede ser encontrado fácilmente como una sumatoria del área de los rectángulos formados; y teniendo en cuenta la Ecuación (7), la Expresión (9) puede ser desarrollado a través de la Ecuación (10).

$$t_{vc} = \frac{2 \cdot t_1 \cdot V_c - 2 \cdot \sum_{i=1}^{19} A_i}{V_c} \quad (10)$$

El área de cada uno de los rectángulos se calcula a través de la Expresión matemática (11).

$$A_i = \frac{i \cdot t_1 \cdot V_c}{400} \quad (11)$$

Al reemplazar la Ecuación (11) en la Expresión (10), se encuentra el valor numérico para la duración del intervalo de velocidad constante, véase Expresión Matemática (12).

$$t_{vc} = \frac{21}{20} \cdot t_1 = 4,725 \text{ s} \quad (12)$$

El algoritmo de programación del microcontrolador inicia basado en el diagrama de flujo de la Figura 3, consultando los registros en la hoja de datos del microcontrolador (Microchip).

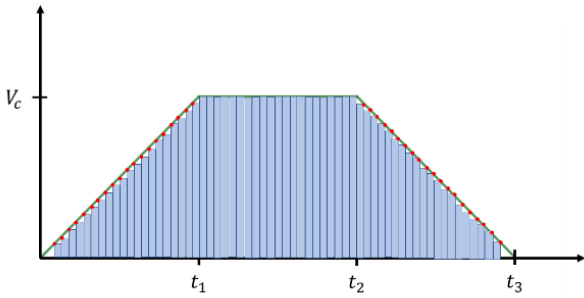


Figura 2. Perfil de velocidad trapezoidal discretizado. Fuente propia.

Aprovechando las bondades del microcontrolador PIC18F4550, la implementación del movimiento con el perfil desarrollado se realizará con la ayuda de 3 de los 4 temporizadores del dispositivo. El temporizador 0 (TMR0), implementa un periodo de interrupción de  $0,225\text{ s}$  ( $t_1/20$ ), para esto, dado que la frecuencia del oscilador del microcontrolador se encuentra definida en  $8\text{ MHz}$ , se emplea un preescalador de 8 y una resolución de  $16\text{ bits}$ , realizando precargas de 9 286. Por otro lado, los temporizadores 1 y 3 (TMR1 y TMR3), serán utilizados para el proceso de interpolación, para lo cual funcionarán como temporizadores variables. Esto se logra variando la precarga que se asigna a sus registros, teniendo en cuenta la frecuencia máxima de paso, véase Ecuación (4), se configuran estos temporizadores con una resolución de  $16\text{ bits}$  y un preescalador de 4. Adicionalmente, se deben configurar seis pines del microcontrolador como salidas digitales, que permitan enviar las señales de PASO y dirección de movimiento al amplificador DRV8825, para el caso de esta implementación, se utilizan los pines del puerto D. De la misma forma, se necesitan tres pines configurados como entradas digitales en los que se lee el estado de los fines de carrera dispuestos en el prototipo, para esto se emplean los pines del puerto B, liberándolos previamente de sus funciones dentro del módulo ADC.

Los valores de las precargas para cada eje, en los intervalos de aceleración-desaceleración, deben ser calculados previo al comienzo del movimiento. El diagrama de flujo de la Figura 4 muestra el proceso de cálculo para obtener los valores de precarga para cada uno de los ejes, esta función se encarga de modificar un arreglo, en el que se almacenan los 20 valores de precargas que se necesitan para la rutina. Esto se realiza a través de un ciclo, que comienza por revisar que la velocidad del intervalo sea diferente de 0, si esto se cumple, se calcula el periodo de la interrupción a través de la Ecuación (13).

$$T_{int} = \frac{V_{int}}{2} \tag{13}$$

Dado el periodo con el que se necesita hacer la interrupción, la precarga del registro puede ser calculada a través de la Expresión matemática (14).

$$precarga = 2^N - \frac{T_{int} \cdot F_{osc}}{4 \cdot PS} \tag{14}$$

Donde  $N$  y  $PS$  son la resolución y el preescalador del temporizador respectivamente, y  $F_{osc}$  la frecuencia del

oscilador. La dinámica del cambio de la precarga de los temporizadores 1 y 3, se explica a detalle en el diagrama de flujo de la Figura 5.

De la misma forma, el diagrama de la Figura 6 explica a detalle, el proceso de envío de pulsos para el *driver* de los motores que permiten llevar el cabezal del parqueadero a la posición especificada.

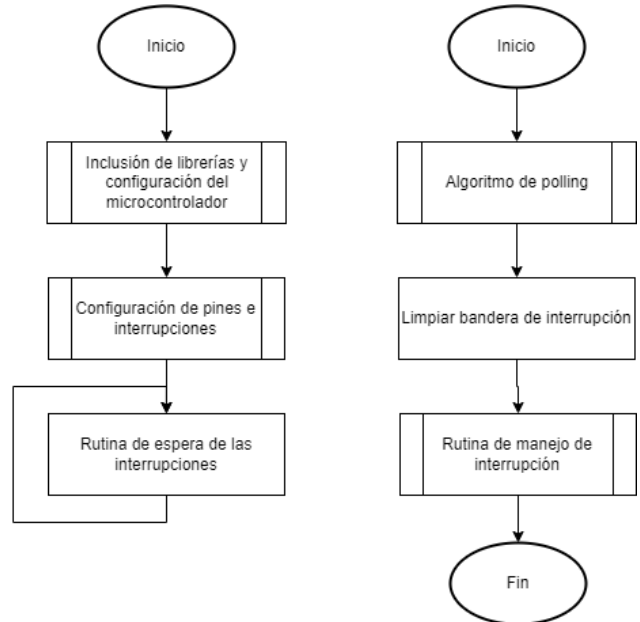


Figura 3. Diagrama de Flujo de ejecución del programa. Fuente propia.

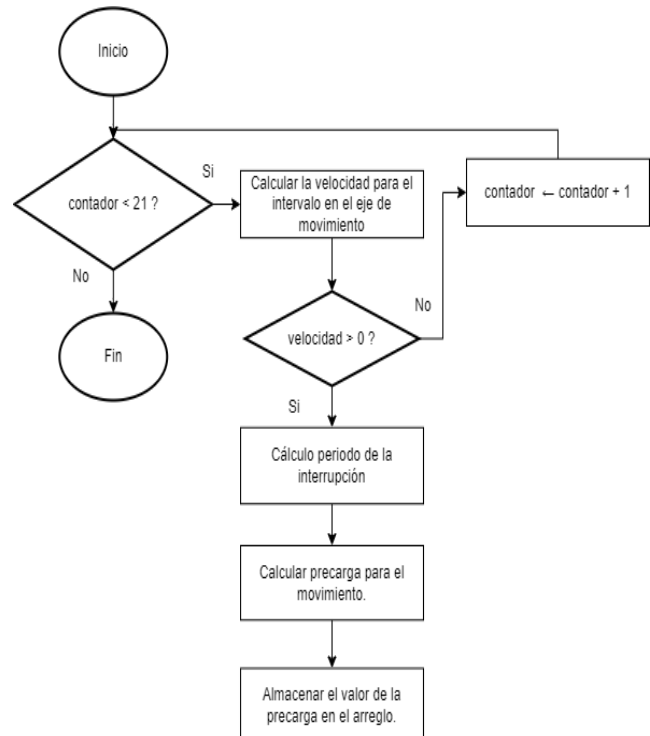
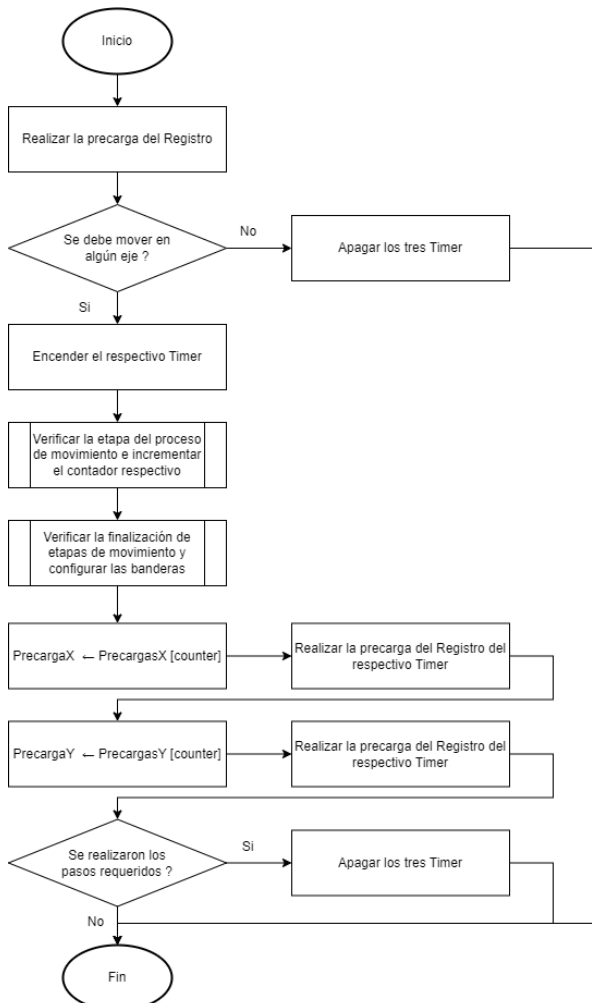
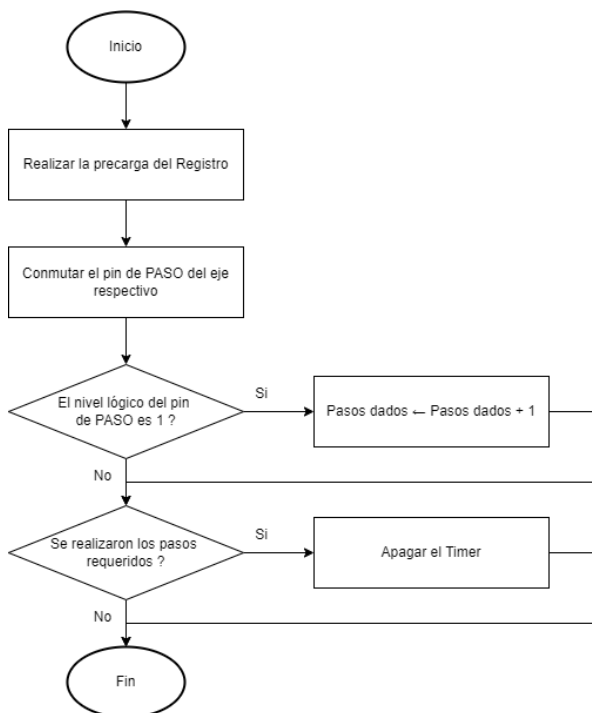


Figura 4. Diagrama de Flujo función de Precargas. Fuente propia.



**Figura 5.** Diagrama de Flujo manejo de la interrupción del temporizador TMR0. Fuente propia.



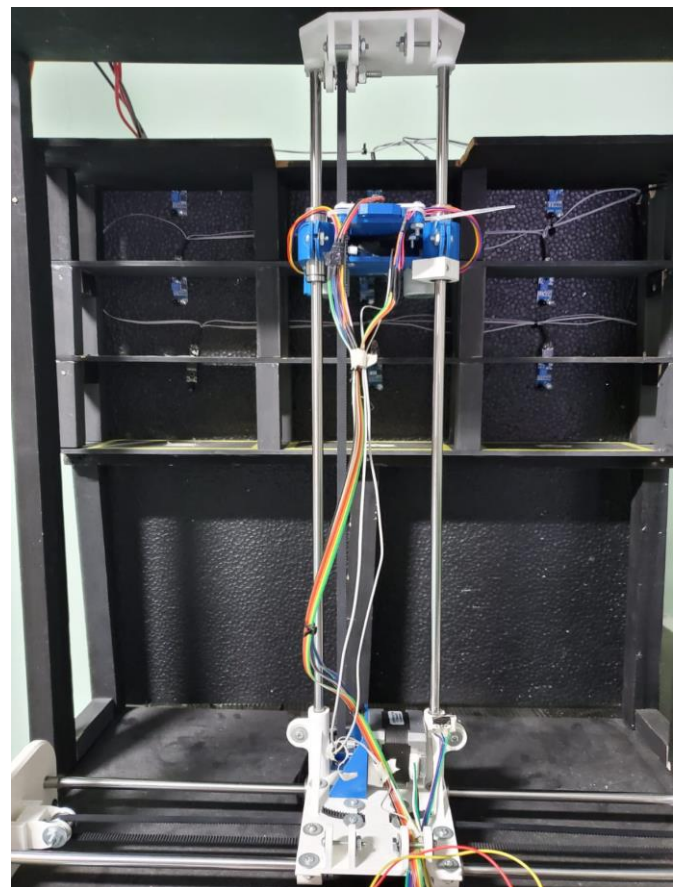
**Figura 6.** Diagrama de Flujo manejo de la interrupción de los temporizadores TMR1 y TMR3. Fuente propia.

### 3.3 Construcción Física del Prototipo

Para la construcción del prototipo se tuvieron en cuenta una serie de parámetros que garantizaran un desarrollo de prototipo rápido de implementar a la vez que robusto para poder hacer pruebas de manera consistente en el mismo, los materiales con los que se desarrolló el prototipo fueron en su mayoría impresión en 3D, madera MDF y palos de balsa obteniendo el resultado que se aprecia en la Figura 7.

Las dimensiones del prototipo global son de 54 cm de ancho y 60 cm de alto con 30 cm de profundidad. Así mismo se tiene un espacio de trabajo de 35 cm de ancho y 40 cm de alto con un desplazamiento de 10 cm de profundidad, permitiendo almacenar y sacar los automóviles. Cada compartimento para autos cuenta con unas dimensiones de 8 cm de alto 8 cm de profundidad y 15 cm de ancho. Las medidas se pueden apreciar en la Figura 8.

La primera etapa consistió en la disposición de los lugares de estacionamiento, se propone una matriz de estacionamiento con 3 niveles, para 3 automóviles, esto con el fin de explorar el desplazamiento en todas las direcciones posibles, contando también con distancias variables entre los lugares de estacionamiento. Adicionalmente se estableció un punto especial donde el automóvil sería recogido, cumpliendo la función de ser un punto de acceso por encima del suelo, donde empezaría el recorrido del cabezal.



**Figura 7.** Montaje físico del prototipo. Fuente propia.

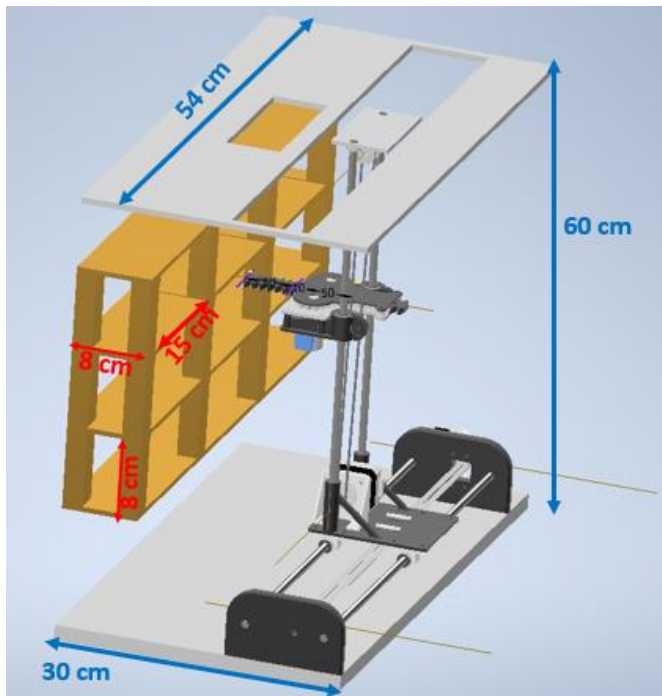


Figura 8. Diseño CAD del prototipo en Autodesk Inventor. Fuente propia.

Para simplificar el funcionamiento del sistema, se utilizaron motores paso a paso unidos por correas, los cuales se ubicaron en la base de la maqueta, punto donde se definen también las coordenadas de origen por medio de unos finales de carrera. Para el diseño del cabezal se tuvo en cuenta su peso, además que el diseño debía permitir la ubicación de los motores encargados de retirar y poner a los automóviles en un eje perpendicular al plano donde ocurre la mayor parte del movimiento, esto se hizo por medio de un par de motores paso a paso más pequeños que trabajan de forma coordinada para mover una pieza dentada.

Finalmente se tiene a disposición un espacio en la base donde toda la parte electrónica y de control se encuentra concentrada para facilitar el acceso en caso de que sea necesaria alguna modificación, la descripción más detallada se puede encontrar en el documento que explica el componente electrónico y de comunicación de este proyecto.

#### 4. RESULTADOS Y DISCUSIÓN

La implementación del algoritmo de interpolación de movimiento presentó diferentes retos para el equipo. En primer lugar, se decidió calcular la precarga de cada uno de los temporizadores, en tiempo real, dentro de la interrupción del TMR0. Esto supuso un inconveniente dado que el cálculo y la conversión de tipos de datos por parte del compilador XC8®, demandaban un periodo de tiempo extenso, lo que desencadenaba en un incorrecto funcionamiento del algoritmo.

Este primer problema fue solucionado realizando el cálculo de cada una de las precargas de manera anticipada como ya se comentó en la sección de metodología, dado que el factor tiempo no se presentaba tan crucial en dicho momento respecto a realizarlas dentro de los procesos de aceleración -

desaceleración. El resultado del proceso de interpolación se puede analizar en la Figura 9.



Figura 9. Proceso de interpolación de movimiento del cabezal. Fuente propia.

La Figura 9 muestra que la implementación del perfil es satisfactoria en términos de la posición del cabezal; sin embargo, sería deseable la utilización de un microcontrolador con un poder de cálculo más avanzado, lo que se lograría con un dispositivo de 16 o 32 bits, contrastando los 8 bits con los que cuenta el PIC18F4550. Esto permitiría obtener perfiles de movimiento de mayor complejidad que permita reducir la demanda sobre los motores en las derivadas superiores de las variables cinemáticas como el caso del *jerk*.

Lo anterior, debido a que a pesar de que el requerimiento en aceleración por parte del perfil se ve disminuido, al no presentarse aceleraciones y desaceleraciones “infinitas” por cambiar la velocidad del perfil de manera instantánea, el perfil cinemático del *jerk* si presentaría este comportamiento impulsivo. Perfiles apropiados para disminuir estos efectos nocivos durante la operación de los motores, son los perfiles en “S” o “S parcial”; o por otra parte, se podría pensar en la implementación de un perfil de movimiento polinómico de orden superior en velocidad. Para esto, se deben tener en cuenta algunas consideraciones.

En primer lugar, en la Figura 10, se observan los requerimientos de memoria impuestos por el proyecto

teniendo en cuenta la implementación del perfil de velocidad trapezoidal.

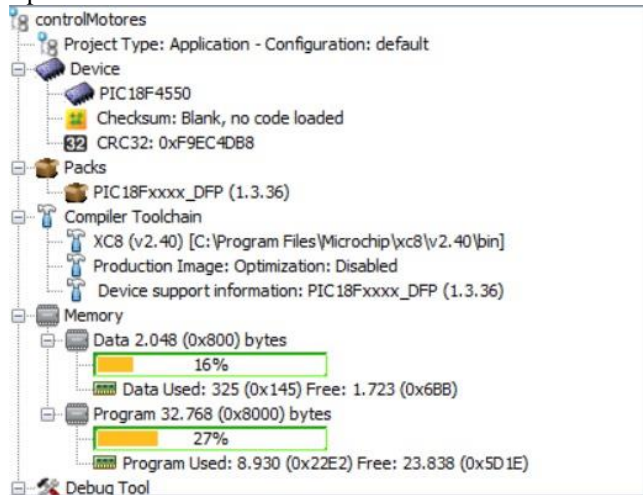


Figura 10. Requerimientos de memoria para el PIC18F4550..

A pesar de observarse una gran disponibilidad para el almacenamiento de nuevas variables, como se evidencia en la sección “Data”, la implementación de un perfil de mayor complejidad implicaría la creación de una gran cantidad de nuevas variables que permitan realizar apropiadamente los cálculos de las variables cinemáticas del perfil; no obstante, desde el punto de vista de la memoria empleada por el dispositivo, no se presentarían mayores inconvenientes.

Por otra parte, el verdadero problema a solucionar se presentaría con el desarrollo de los cálculos del perfil, generalmente, para perfiles en “S” se emplean curvas parametrizadas con 12 trozos, y en el caso de curvas polinómicas de mayor orden, la realización de potencias, implican la utilización de librerías, o funciones iterativas lo que no sólo incrementará la memoria “Program” utilizada, sino que su mayor reto se evidenciará a la hora de realizar los cálculos de manera ágil, requiriéndose de esta forma un microcontrolador con una mayor capacidad de cómputo, es decir, dispositivos de 16 o 32 bits.

Teniendo en cuenta lo anterior, y adicionalmente al considerar que el perfil de velocidad trapezoidal es altamente utilizado en el control de movimiento en la industria (Montalvo), se decide continuar con la implementación y el análisis de los resultados obtenidos con dicho perfil en el prototipo.

La utilización de temporizadores en este proceso presenta el segundo problema con el cual aún se debe lidiar, correspondiente al truncamiento de valores. Dada la imposibilidad de obtener precargas flotantes debido a la naturaleza entera de los registros de los temporizadores. Lo que hace que, dentro del proceso de interpolación, se pierden algunos pasos en cada intervalo.

Teniendo en cuenta lo anterior, se emplea el último intervalo, para corregir los posibles pasos que se hayan perdido. La Tabla 1. ilustra este hecho, al realizar la medición de los intervalos temporales para el perfil de movimiento de la Figura 9. Estas mediciones fueron realizadas con ayuda del simulador del IDE MPLAB® X y verificados con un cronómetro.

Tabla 1. Intervalos temporales del perfil de movimiento

Etapa de movimiento	Intervalo temporal [s]
Aceleración	4,5
Velocidad crucero	4,725
Desaceleración	6,8

De esta forma, los resultados de movimiento del cabezal dentro del parqueadero automatizado son sobresalientes. Para mejorar su desempeño temporal, la compensación de pasos podría realizarse adicionalmente en las otras dos etapas del perfil de movimiento

## 5. CONCLUSIONES

La utilización de curvas de velocidad trapezoidales representa una excelente implementación para el prototipo, ya que los cálculos que requiere hacer el PIC18F4550 son simples y el impacto que sufre el mecanismo debido a la aceleración es bajo. Disminuyendo de esta forma las vibraciones en el mecanismo, obteniéndose un movimiento armónico y fluido del cabezal en todo momento.

Adicionalmente, se encontró que el algoritmo de interpolación puede verse comprometido debido a los errores de truncamiento impuestos por los registros del microcontrolador en los que se incurren a lo largo de los cálculos. Por otra parte, la realización de operaciones aritméticas complejas puede demandar un lapso considerable para el microcontrolador, lo que compromete la correcta ejecución de otras tareas por parte de este.

Lo anterior evidencia la importancia de conocer las limitaciones de cada microcontrolador ya que en base a características como la memoria o el número de bits con los que trabaja el procesador, una referencia específica puede no ser adecuada para desempeñar la tarea que se espera; o en casos menos severos, implicar un cambio en el algoritmo de programación.

## REFERENCIAS

Chekired, F., Houtti, S., Bouroussis, C. A., Rahmani, A., Tilmatine, A., & Canale, L. (2020). Low Cost Automation System for Smart Houses based on PIC Microcontrollers. *2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)*, 1–5. <https://doi.org/10.1109/EEEIC/ICPSEurope49358.2020.9160808>

Cui Devices. (2022). *NEMA17-AMT112S Data Sheet*. <https://www.cuidevices.com/product/resource/nema17-amt112s.pdf>

García-Martínez, J. R., Rodríguez-Reséndiz, J., & Cruz-Miguel, E. E. (2019). A New Seven-Segment Profile Algorithm for an Open-Source Architecture in a Hybrid

Electronic Platform. *Electronics* 2019, Vol. 8, Page 652, 8(6), 652. <https://doi.org/10.3390/ELECTRONICS8060652>

Heo, H.-J., Son, Y., & Kim, J.-M. (2019). A Trapezoidal Velocity Profile Generator for Position Control Using a Feedback Strategy. *Energies*, 12(7), 1222. MDPI AG. <http://dx.doi.org/10.3390/en12071222>

Microchip. (2009). *PIC18F4550 Data Sheet*. <https://ww1.microchip.com/downloads/en/devicedoc/39632e.pdf>

Montalvo, V., Estévez-Bén, A. A., Rodríguez-Reséndiz, J., Macías-Bobadilla, G., Mendiola-Santibañez, J. D., & Camarillo-Gómez, K. A. (2020). FPGA-Based Architecture for Sensing Power Consumption on Parabolic and Trapezoidal Motion Profiles. *Electronics* 2020, Vol. 9, Page 1301, 9(8), 1301. <https://doi.org/10.3390/ELECTRONICS9081301>

Rodríguez-Núñez, E., & Martínez-Prado, M. Á. (2022). Evaluación del desempeño de perfiles de velocidad con base en la ejecución del controlador aplicado a un motor de corriente directa [Universidad Autónoma de Querétaro]. <http://ri-ng.uaq.mx/bitstream/123456789/3613/1/IGLIN-262790-0522-522-Emmanuel%20Rodr%20c%20adguez%20N%20c%20ba%20c%20bl%20e%20z%20-A.pdf>

Texas Instruments. (2014). *DRV8825 Stepper Motor Controller IC*. [https://www.ti.com/lit/ds/symlink/drv8825.pdf?ts=1662869166160&ref\\_url=%20https%253A%252F%252Fwww.google.com%252F](https://www.ti.com/lit/ds/symlink/drv8825.pdf?ts=1662869166160&ref_url=%20https%253A%252F%252Fwww.google.com%252F)

Yoon, H., Chung, S., Kang, Han., & Hwang, M. (2019). Trapezoidal Motion Profile to Suppress Residual Vibration of Flexible Object Moved by Robot.

Yu, Z., Han, C., & Haihua, M. (2015). A novel approach of tuning trapezoidal velocity profile for energy saving in servomotor systems. *Chinese Control Conference, CCC, 2015-September*, 4412–4417. <https://doi.org/10.1109/CHICC.2015.7260323>