Control Predictivo Basado en Modelo (MPC) simplificado en una ESP32 para el seguimiento de una trayectoria simulada

Alexis Fernando Cruz Baños, Omar Amaro Hernandez, Luis Alberto Tovar Ortiz
Fernando Ivan Antonio Hernandez Diaz, Juan Carlos Herrera Lozada, Dr.

Instituto Politécnico Nacional

CIDETEC

acruzba2100@alumno.ipn.mx, oamaroh1400@alumno.ipn.mx
ltovaro2100@alumno.ipn.mx fhernandezd0200@alumno.ipn.mx
jlozada@ipn.mx

Referencia de este artículo [1].

ABSTRACT

This work addresses the implementation of a simplified Model Predictive Controller (MPC) on a low-cost embedded system based on the ESP32-S3. A first-order linear system and an MPC with a short prediction horizon are proposed in order to significantly reduce the computational load without compromising the controller's performance in tracking a simulated reference trajectory. The system's functionality was validated using a simulated reference generated by a potentiometer connected to an ADC port on the ESP32, and by introducing step-type disturbances at the system output. The results demonstrate that the proposed controller is a feasible solution for embedded applications with limited resources in first-order systems.

I.- Introducción

Hoy en día el uso de estrategias de control son una tendencia para lograr un desempeño óptimo de sistemas en general, sin embargo, así como las estrategias se desarrollan para ser cada vez más robustas, también hay un notable incremento de capacidad computacional requerida para ejecutarlas. Es por esto que surge la necesidad de comprobar la viabilidad de utilizar sistemas embebidos económicos como el ESP32, para la ejecución de estos. El control predictivo basado en modelos (MPC) ha ganado relevancia en diversos campos de desarrollo tecnológico ya que tiene la capacidad de anticipar comportamientos futuros para responder de manera óptima, a diferencia de un control tradicional como el control proporcional integral derivativo (PID), el MPC ofrece una ejecución robusta y flexible para resolver restricciones o posibles perturbaciones que un control tradicional difícilmente logra. En el presente trabajo

se aborda una implementación simplificada de un MPC sobre una ESP32, para determinar qué tan factible es utilizar este sistema embebido de bajo costo para la ejecución de estrategias complejas de control.

II.- Control Predictivo Basado en Modelos

El control predictivo basado en modelos o MPC (Model Predictive Control), es un sistema avanzado de control que parte de un modelado del sistema a controlar para predecir su comportamiento futuro [1]. El controlador parte de estas predicciones para calcular una secuencia de acciones de control resolviendo un problema de optimización durante un horizonte de predicción. A partir de esto, se aplica la primera acción de control para repetir el proceso para el siguiente instante de muestreo.

Para utilizar el MPC es necesario contar con diversos elementos comenzando con el Modelo del sistema que debe representar el comportamiento dinámico del sistema, puede ser un sistema lineal o no lineal, el horizonte de predicción (N) que es el número de pasos a futuro para los que se calcula la evolución del sistema, y la función de costo (J) que se encarga de penalizar la diferencia entre la salida y la referencia a seguir, así como la acción de control. La función de costo generalmente se define como:

$$\boldsymbol{J = \sum_{I=1}^{N} (r[k+i] - y[k+i])^2 + \lambda(\Delta u)^2}$$

Donde:

$r[k+i]$: Referencia futura.

$y[k+i]$: Salida predicha.

$\Delta u = u[k] - u[k-1]$: Acción de control.

λ: Penalización de cambio de control.

Control predictivo basado en modelos simplificado

El MPC simplificado es una alternativa del MPC tradicional diseñada como alternativa a sistemas con un procesamiento computacional reducido o limitado [2]. En esta alternativa se busca reducir la complejidad computacional utilizando simplificación del modelo del sistema y obteniendo una ley de control explicita, evitando así la resolución de problema de optimización en tiempo real. La principal diferencia en comparación con un MPC normal, es que este suele utilizarse para sistemas de baja complejidad como sistemas de primer orden o de segundo orden sin restricciones complejas. Además, el horizonte de predicción (N) generalmente es corto, es decir la predicción esta basada con 2 o 3 pasos adelantes, simplificando la parte analítica.

Función de costo. La función de costo en un MPC simplificado, busca penalizar la desviación entre la salida de predicción y la referencia, así como los cambios bruscos en la acción de control [3]. La función de costo está definida por:

$$\boldsymbol{J(u) = \sum_{I=1}^{N} (y[k+i] - r)^2 + \lambda(\mathbf{u[k]} - \mathbf{u[k-1]})^2} \qquad \textbf{(2.1)}$$

Donde:

N: Horizonte de predicción.

r: Entrada de referencia.

λ: Penalización de entrada de control.

u: Variable de decisión.

y: predicción de la salida en el paso i.

Para obtener las próximas salidas (predicciones futuras), se supone que la señal de control se mantiene constante durante el horizonte de predicción con el objetivo de reducir la complejidad de la función de costo. Para el sistema de primer orden propuesto anteriormente las posibles salidas.

Suponiendo un modelo discreto de primer orden sin retardo:

$$\boldsymbol{y[k+1] = Ay[k] + Bu[k]} \qquad \textbf{(2.2)}$$

Donde:

$y[k]$: Salida del sistema en el k-ésimo instante.

$u[k]$: Es la señal de control aplicada en el k-ésimo instante.

A, B: Coeficientes reales del modelado del sistema.

Se expresa también como:

$$\boldsymbol{y_i = \psi_i + \phi_i u} \qquad \textbf{(2.3)}$$

Donde:

ψ_i: Valores del modelo inicial.

ϕ_i: Ganancia de entrada sobre cada predicción.

Las salidas considerando $N = 3$ son:

$$\boldsymbol{y_1 = Ay[k] + Bu = \psi_1 + \phi_1 u} \qquad \textbf{(2.4)}$$

$$\boldsymbol{y_2 = Ay_1 + Bu = A^2 y[k] + ABu + Bu = \psi_2 + \phi_2 u} \qquad \textbf{(2.5)}$$

$$\boldsymbol{y_3 = Ay_2 + Bu = A^3 y[k] + A^2 Bu + ABu + Bu = \psi_3 + \phi_3 u} \qquad \textbf{(2.6)}$$

Entonces:

$$J(u) = \sum_{i=1}^{3} (\psi_i + \phi_i u[k] - r)^2 + \lambda(u[k] - u[k-1])^2 \qquad (2.7)$$

Si se expande la función de costo (2.7), y se expresa en una función cuadrática simple, se tiene:

$$J(u) = Au^2 + Bu + C \qquad (2.8)$$

Con:

$A = A_1 + \lambda$

$B = B_1 - 2\lambda u[k-1]$

$C = C_1 + \lambda(u[k-1])^2$

Finalmente, para obtener la ley de control, se minimiza la función de costo derivando respecto a u e igualando a 0. Entonces la ecuación que muestra la ley de control (u^*) es:

$$\frac{dJ(u)}{du} = 2Au + B = 0 \Rightarrow u^* = -\frac{B}{2A} \qquad (2.9)$$

III.- Implementación del controlador en ESP32-S3

Para comprobar el funcionamiento de un MPC simplificado en una ESP32-S3 se plantea un sistema lineal de primer orden como se presentó en la ecuación (2.2).

$$y[k+1] = Ay[k] + Bu[k] \qquad (2.2)$$

Se asignan valores arbitrarios para A y B siendo 0.1 y 0.9 respectivamente. Para observar el comportamiento del controlador MPC simplificado, se realiza un algoritmo que genere una trayectoria simulada a partir de la entrada analógica de la ESP32 utilizando un potenciómetro, de esta forma se tiene una señal de referencia a la cual el controlador debe de llegar [4][5]. Además, dentro del algoritmo se implementó una función que permite simular perturbaciones del tipo escalón con el objetivo de observar el comportamiento y respuesta del controlador frente a perturbaciones que modifican la dinámica del sistema. El algoritmo utilizado en la ESP32 se muestra en la figura 1.

El valor de penalización λ puede variar dependiendo de la respuesta que se requiera del controlador, en valores pequeños de λ el control tiene a ser más rápido ofreciendo una respuesta más agresiva, pero puede provocar oscilaciones o señales de control abruptas, en el caso contrario, el control tiende a una respuesta mas lenta y con menor sobreimpulso, pero se puede tener un error mayor en la respuesta del sistema. En el caso particular de este problema el valor de penalización fue bajo, pero ofreció una respuesta adecuada.

```
const float a = 0.9; //Valores arbitrarios del sistema A
const float b = 0.1; //Valores arbitrarios del sistema B
const float lambda = 0.0061; //Variable de penalización de entrada de control.

float y = 0.0;      // Salida del sistema
float u = 0.0;      // Señal de control u[k]
float u_prev = 0.0;  // Señal de control anterior u[k-1]
float r = 0.0;      // Variable para referencia

const int potPin = 5; // Pin para el potenciómetro
const int boton = 27; // Pin para conexión del botón para perturbación.
bool per_act = false; //Variable de apoyo para perturbación.

void setup () {
  pinMode(boton, INPUT_PULLUP);
  Serial.begin(115200);
  delay(1000);
  Serial.println("r,y,u");
}

void loop() {
  // Escalamiento de la señal de entrada del potenciómetro.
  int potValue = analogRead(potPin);
  r = map(potValue, 0, 4095, 0, 100);

  //Lectura de botón para perturbación.
  if (digitalRead(boton) == LOW){
    per_act = true;
  }

// Predicción de salidas futuras
  float y1 = a * y + b * u;  // y[k+1]
  float y2 = a * y1 + b * u; // y[k+2] (Se supone u[k+1] ≈ u[k])
  float y3 = a * y2 + b * u; // y[k+3]

  // Simplificación de la derivada de J.
  float phi1 = b;
  float phi2 = a * b;
  float phi3 = a * a * b;

  float psi1 = a * y;
  float psi2 = a * psi1;
  float psi3 = a * psi2;

  // Construcción de la función cuadrática J(u) = A*u^2 + B*u + C
  float A = phi1*phi1 + phi2*phi2 + phi3*phi3 + 3*lambda;
  float B = 2 * (phi1 * (psi1 - r) + phi2 * (psi2 - r) + phi3 * (psi3 - r))
+ 2*lambda*(u_prev);

  u = -B / (2 * A); //Función de control u*

  // Salida del sistema.
  y = a * y + b * u;
  if (pert_act == true){
    y += 10; //Perturbación tipo escalón.
  }

  // Mostrar valores
  Serial.print(r); Serial.print(",");
  Serial.println(y); Serial.print(",");
  //Serial.println(u);

  // Actualización de valor anterior.
  u_prev = u;

  delay(50); // Frecuencia de muestreo.
}
// Fin
```

Figura 1. Algoritmo ejecutado en la ESP32.

IV.- Pruebas y resultados

Para demostrar el funcionamiento del controlador, se realizó una prueba para observar el seguimiento de trayectoria y el comportamiento de la salida con perturbaciones simuladas. En la figura 2, se observa una gráfica mostrando el comportamiento de la salida con respecto a la referencia (generada por un potenciómetro) y la relación con la acción de control.

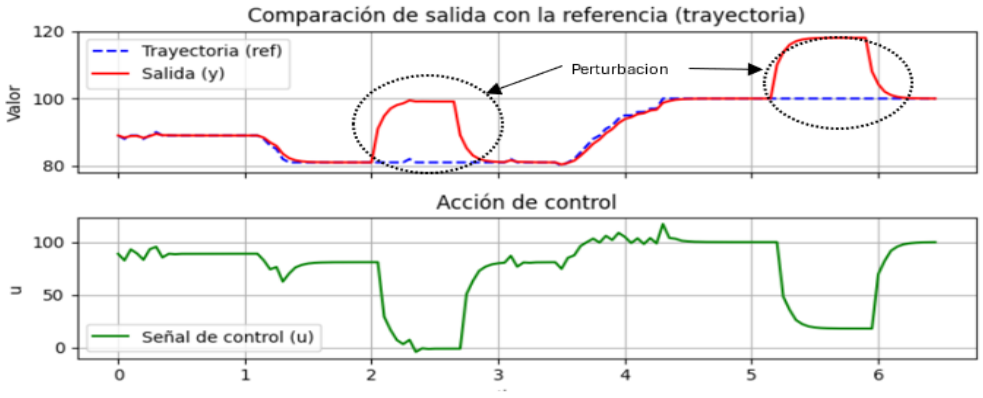

Figura 2. Respuesta del controlador para el seguimiento de una trayectoria simulada.

Además, se observa que el controlador permite seguir la referencia de manera óptima incluso en las variaciones o "ruido" que el potenciómetro pudiese generar, la salida del controlador compensa esas ligeras variaciones. Se observa que en el momento que se tienen perturbaciones, la acción de control es grande para compensar ese desfase con respecto a la trayectoria deseada, esto indica que el controlador se está comportando de manera óptima permitiendo que la salida regrese al valor de la referencia demostrando el rechazo de la perturbación.

V.- Conclusiones

El controlador MPC simplificado es una herramienta útil y computacionalmente viable para ser utilizado en sistemas embebidos básicos o con poca capacidad de procesamiento como lo es la ESP32-S3, utilizando horizontes de predicción pequeños, o reduciendo las operaciones analíticas a una forma más simple.

A pesar de su simplicidad, este tipo de controlador permite un seguimiento de trayectoria básico gracias a su capacidad de predicción, además, se muestra que el controlador tiene un comportamiento robusto frente a posibles perturbaciones indicando que es posible utilizar este modelo de control para sistemas lineales de primer orden como control de temperatura, control de flujo o control para velocidad de motores DC.

Como trabajo futuro, se propone analizar el desempeño del mismo controlador en sistemas de mayor orden o evaluar el impacto de utilizar horizontes de predicción mayores lo cual podría mejorar el comportamiento del controlador.

VI.- Referencias bibliográficas

J. B. Rawlings and D. Q. Mayne, *Model Predictive Control: Theory and Design*. Madison, WI: Nob Hill Publishing, 2009.

Zhuang, Y. Chen, and X. Chen, "A new simplified modeling method for model predictive control in a medium-sized commercial building: A case study," *Building and Environment*, vol. 127, pp. 1–12, 2018.

A. Bemporad, M. Morari, and N. L. Ricker, *Model Predictive Control*. Trento, Italy: University of Trento, 2010.

Espressif Systems, *ESP-IDF Programming Guide*. [En linea]. Disponible: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/

Arduino, *Arduino Core for the ESP32*. [En linea]. Disponible: https://docs.arduino.cc/cores/esp32/

Referencia del artículo

Cruz, A., Amaro, O., Hernandez, F. & Herrera, J. *(septiembre - octubre, 2025).* Control Predictivo Basado en Modelo (MPC) simplificado en una ESP32 para el seguimiento de una trayectoria. *Boletín UPIITA. año 20, (110) 2025*
https://www.boletin.upiita.ipn.mx/index.php/ciencia/1098-cyt-numero-110/2442-control-predictivo-basado-en-modelo-mpc-simplificado-en-una-esp32-para-el-seguimiento-de-una-trayectoria-simulada