



UNIVERSIDAD DE BUENOS AIRES  
FACULTAD DE INGENIERIA  
2018 - 2<sup>do</sup> Cuatrimestre

## 86.22 - LABORATORIO DE CONTROL AUTOMÁTICO

### TP2

Integrantes-

Funes, Pablo Nicolás - 94894

Eichenbaum, Daniel - 95233  
18 de noviembre de 2018

## Índice

<b>1. Objetivo</b>	<b>2</b>
<b>2. Introducción</b>	<b>2</b>
<b>3. Acondicionamiento de las señal del encoder</b>	<b>3</b>
<b>4. Algoritmo del PID</b>	<b>4</b>
<b>5. Comunicación Serie</b>	<b>5</b>
5.1. Recepción . . . . .	5
5.2. Emisión . . . . .	5
<b>6. Integración de la planta y Ensayos</b>	<b>7</b>
6.1. Sintonización del PID . . . . .	8
6.2. Ensayo a una perturbación externa . . . . .	9
6.3. Identificación de la planta . . . . .	10
<b>7. Conclusión</b>	<b>11</b>

## 1. Objetivo

En el siguiente documento, se controlará la velocidad de un motor de corriente continua cuya velocidad es sensada con un encoder. Finalmente se desea estimar un modelo para el mismo. Todo el proyecto debe estar ejecutándose en tiempo real comunicándose contra una computadora quien representará los datos leídos.

## 2. Introducción

El driver utilizado responde al siguiente circuito Fig. 1

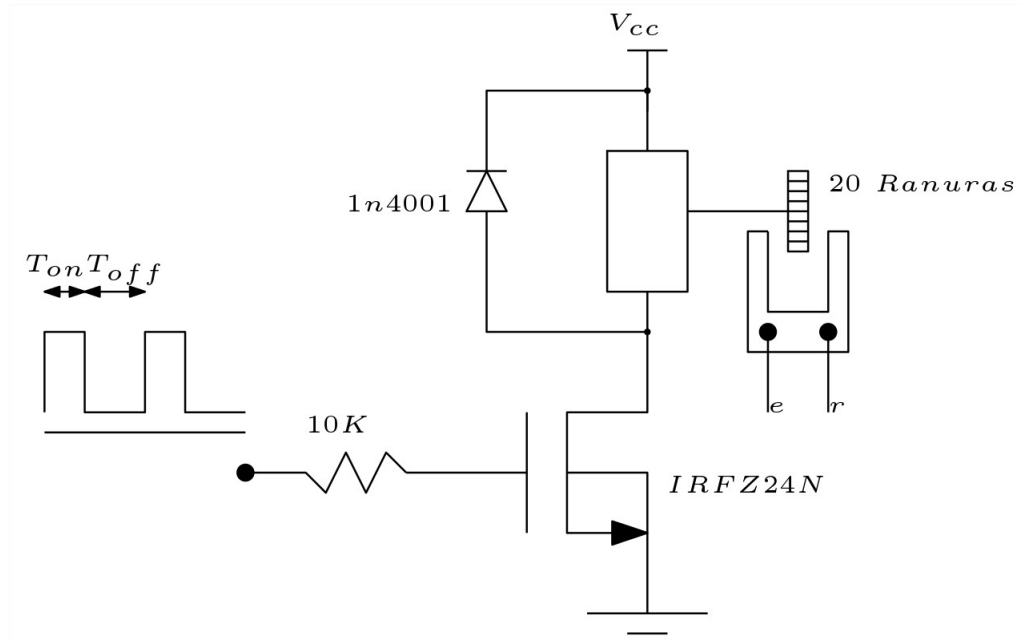


Figura 1: Circuito implementado para controlar a la planta

El circuito está penado para soportar una carga de 2A a temperatura ambiente sin disipador, la desventaja es que no puede invertir el sentido de giro del motor.

Como entrada, recibe una señal cuadrada modulada (Actuación PWM) y como señal de realimentación está el encoder que genera un pulso cuando el haz fue interrumpido.

### 3. Acondicionamiento de las señal del encoder

El primer obstáculo en la realización del proyecto, consiste en transformar el tren de pulsos producido por la serie de interrupciones del haz en el encoder, debido al giro del motor.

En la Fig. 2 se observa el algoritmo de lectura de velocidad.

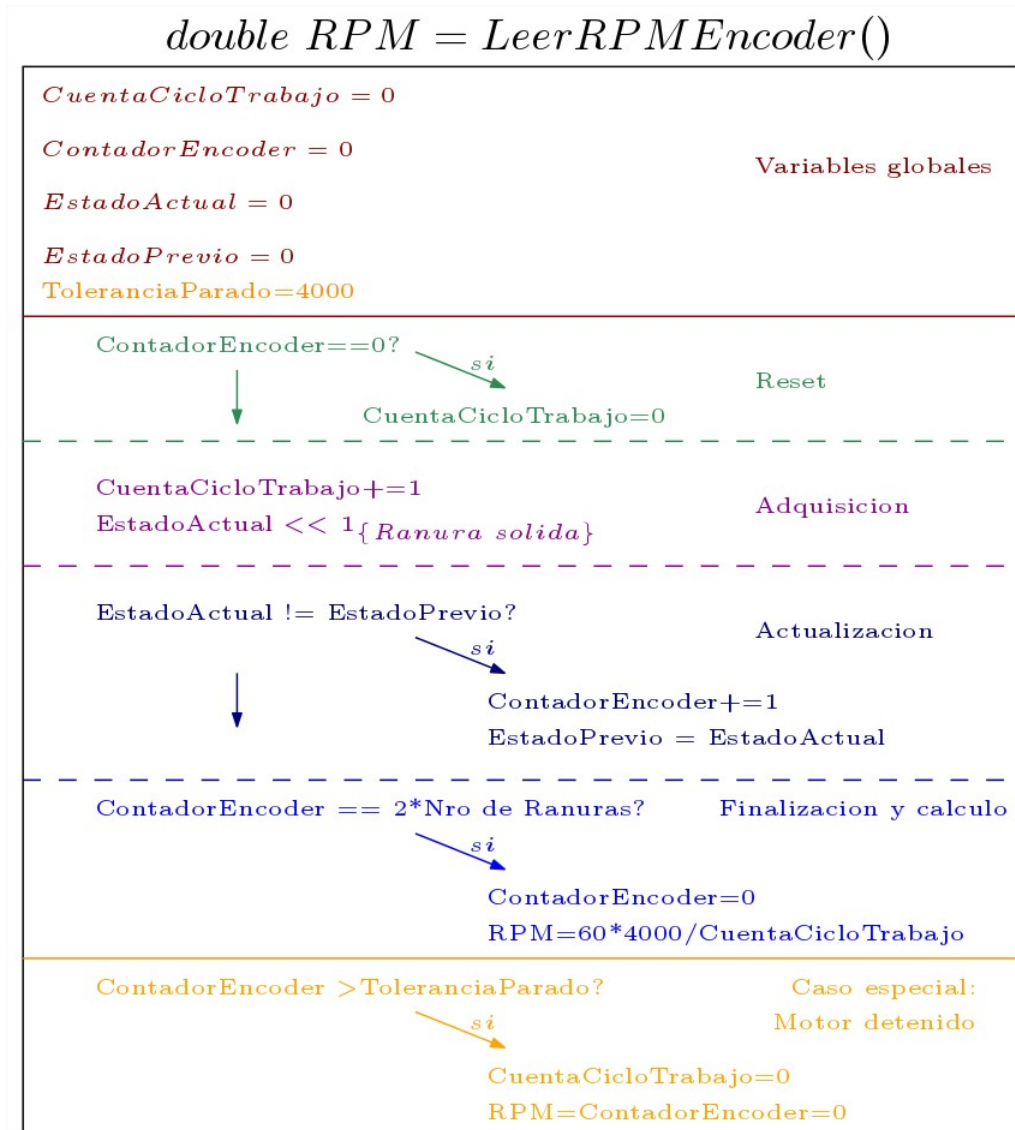


Figura 2: Diagrama del algoritmo de adquisición de la velocidad del motor

Las variables globales, consisten en *CuentaCicloTrabajo* que cuenta cuanto tiempo ocurre entre revolución y revolución del motor. *ContadorEncoder* que cuenta cuantos flancos positivos y negativos van ocurriendo hasta que se resetea. *EstadoActual* que indica si actualmente el sensor detecta "Hueco." o "Sólido". *EstadoPrevio* que recuerda si el estado anterior era hueco o no.

En actualización, el algoritmo va contando flancos positivos y negativos; Cuando la cantidad de flancos contados (sean positivos o negativos) alcanza a dos veces la cantidad de ranuras menos una, se calcula las revoluciones y resetean los contadores.

Finalmente, como la tasa de actualización del encoder es  $T_s = 1/4000$  ( $f_{enc} = 4kHz$ ). Ahora las revoluciones se calculan como  $RPM = \frac{60}{N \cdot T_s}$ . Donde N es la cantidad de veces que se llamo a dicha función al dar un giro el motor.

En el caso especial en que el motor se quedó bloqueado/no arrancó, se indicará que la velocidad es 0.

## 4. Algoritmo del PID

El algoritmo del PID discreto, coincide con el del trabajo práctico 1. El mismo se recuerda en la Fig. 3. Cabe aclarar que el mismo será llamado a una frecuencia  $f_{PID} = 4kHz$

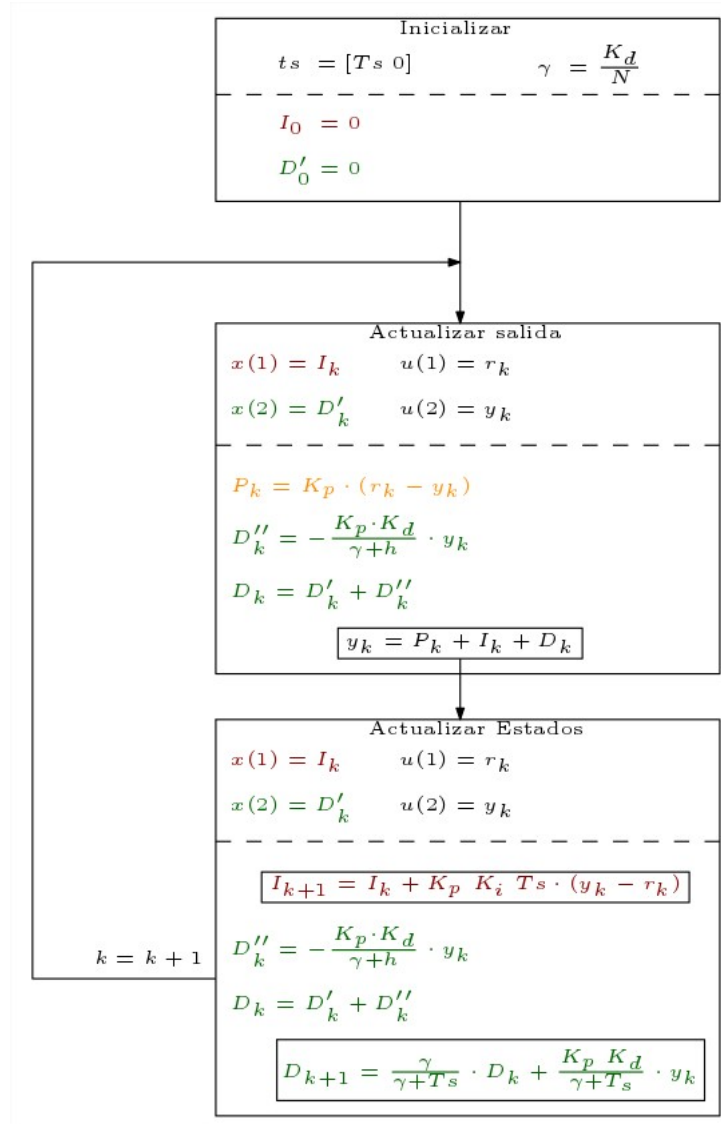


Figura 3: Algoritmo de cálculo del PID.

El PID incluye un saturador a 255 que cuando esto ocurre se apaga el efecto Integrador (efecto anti Wind Up).

## 5. Comunicación Serie

La comunicación serie consta de dos situaciones de funcionamiento, por un lado los parámetros del dispositivo deben poder ser recalibrados a distancia y por otro se desearía visualizar su estado de funcionamiento en tiempo real.

Se implementa el sistema Punto Fijo (8,5) Fig. 4 para la transmisión de aquellos símbolos que pesan 1 byte; Con ello se podrá representar números enteros desde 0 hasta 7 con parte fraccionaria múltiplos de 0.03125.

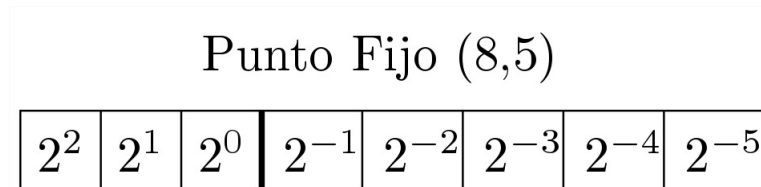


Figura 4: Punto Fijo

En cambio, para los símbolos que pesan 2 bytes, se implementa Punto Fijo (16,6) que representan enteros desde 0 hasta 1023 con parte fraccionaria múltiplos de 0.015625.

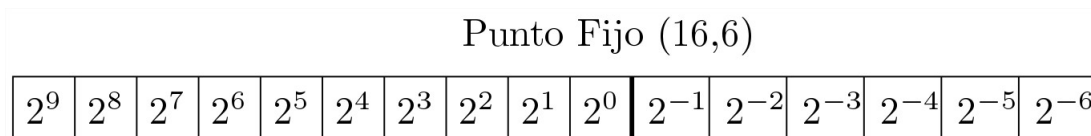


Figura 5: Punto Fijo para simbolos de 16 bits

### 5.1. Recepción

En la Fig. 6 se observa la trama de recepción de datos del dispositivo. En el se encuentra el encabezado, cuya función es la de verificar que la transmisión está funcionando correctamente. Los nuevos parámetros del PID que deberán ser actualizados. La velocidad de referencia. El parámetro del filtro para la parte derivativa del PID N y se incluye un parámetro opcional "Bias" que sirve para luego realizar la identificación.

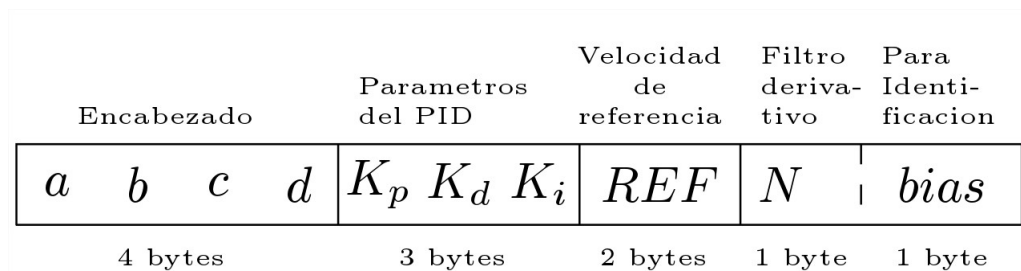


Figura 6: Trama de recepción del dispositivo

### 5.2. Emisión

La transmisión se realiza a una frecuencia de 100Hz para visualizar de forma continua el estado del controlador.

En la Fig. 7 se observa la trama que responde a la emisión de datos del dispositivo hacia la interfaz HMI.

Encabezado				Parametros del PID	Velocidad de referencia	Senal de actua- cion	Lectura de velocidad
<i>e</i>	<i>f</i>	<i>g</i>	<i>h</i>	$K_p$ $K_d$ $K_i$	<i>REF</i>	<i>PWM</i>	<i>vel</i>
4 bytes				3 bytes	2 bytes	1 byte	2 bytes

Figura 7: Trama de emision del dispositivo hacia la computadora

En el se encuentra un encabezado diferente, los parámetros del PID actuales; La velocidad de referencia actual; La señal de actuación *PWM* y la velocidad leída desde el encoder.

## 6. Integración de la planta y Ensayos

La interfaz HMI propuesta responde a la Fig. 8; A la izquierda se introducirán los diferentes parámetros a transmitir (1 Seg. luego de introducidos) y a la izquierda el microcontrolador responde con la trama que indica el estado del motor en tiempo real (Delay de 0.1 Seg).

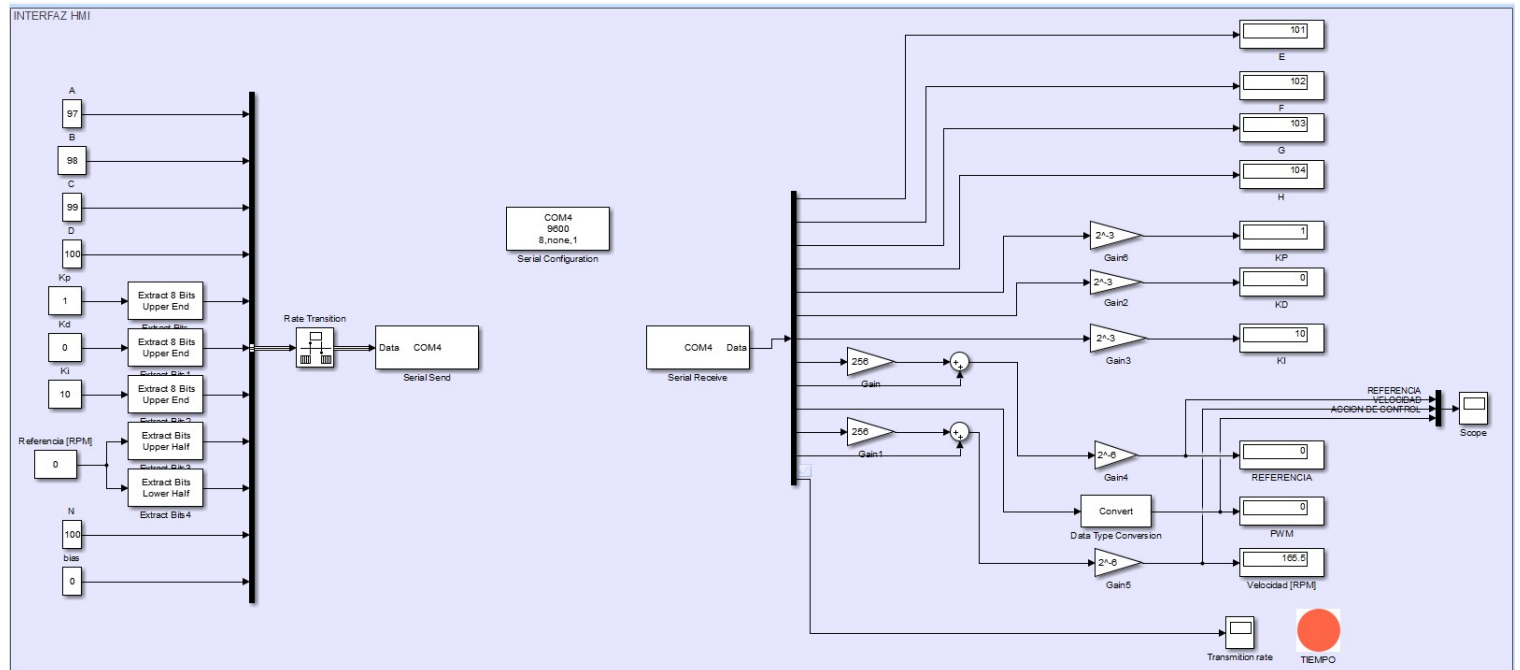


Figura 8: Interfaz HMI



## 6.1. Sintonización del PID

En este ensayo se sintonizará el controlador de modo que siga la referencia impuesta por software. El método empleado es la sintonización de Ziegler-Nichols que consiste en la siguiente regla:

$$\begin{array}{ccc} K_P & K_I & K_D \\ K_c \cdot 0,56 & 1,18 \cdot \frac{K_c}{T_c} & 0,074 \cdot K_c \cdot T_c \end{array}$$

Donde se define  $K_c$  como la ganancia  $K_P$  crítica que obtendrá una respuesta oscilatoria y  $T_c$  es el período de estas oscilaciones.

Sin embargo, debido a la no linealidad del arranque del motor debido en parte al rozamiento estático este modelo produjo malos resultados y se prefirió hacer un ajuste a mano. Los parámetros obtenidos fueron:

$$\begin{array}{ccc} K_P & K_I & K_D \\ 0,2 & 5 & 0,01 \end{array}$$

Originando la Fig. 9 la cual muestra que el sistema se comporta en el límite del régimen crítico.

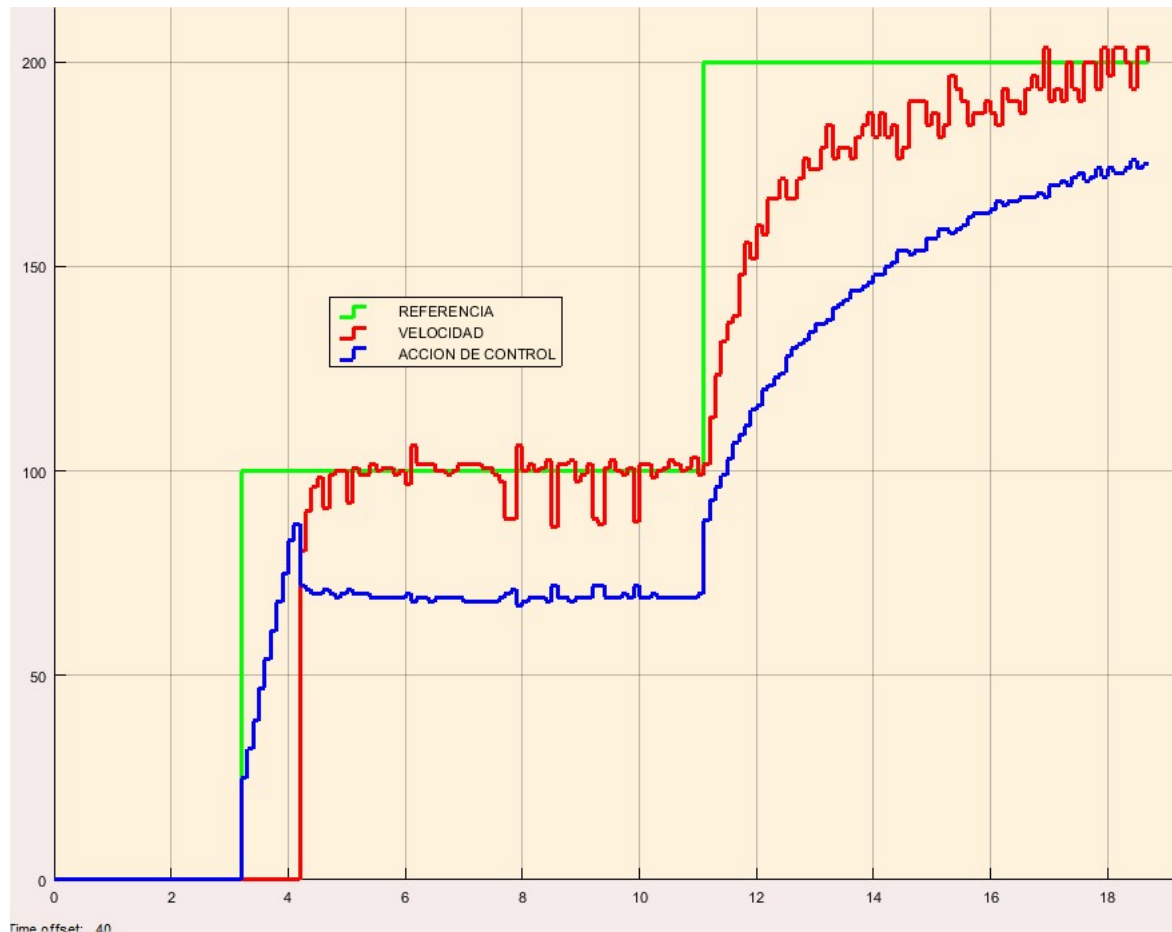


Figura 9: Sintonización del PID

## 6.2. Ensayo a una perturbación externa

En este ensayo, se desea ver como responde el controlador ante un aumento de carga.

En la Fig. 10, se ha establecido una referencia en 150 RPM, y a partir del segundo 10 se ha intentado frenar bruscamente al motor; Al segundo 14 se liberó la carga.



Figura 10: Ensayo de carga variable al motor

Se observa como a partir de que aumentó la carga, la acción de control se incrementó de forma lineal hasta casi equilibrarse al segundo 14 y lograr la velocidad de referencia; Momento en que se libera la carga y la compensación ahora resulta ser negativa.

### 6.3. Identificación de la planta

Para la identificación de la planta, se utiliza el parámetro *Bias* con todos los parámetros del PID en cero, de modo que la señal de actuación proviene 100 % del kit de identificación.

El modelo identificado fué

$$H_{cc}(S) = \frac{-0,3662 \cdot s + 637}{s^2 + 51,89 s + 734,5}$$

Que discretizando se obtiene:

$$H_{cc}(Z) = \frac{0,7476 \cdot z^{-1}}{1 - 0,161 \cdot z^{-1} + 0,02637 \cdot z^{-2}}$$

Se desea comparar el modelos en tiempo real con el motor verdadero, la señal de referencia y la acción de control en la Fig. 11.



Figura 11: Identificación del modelo del motor

Se observa que el modelo identificado se aproxima bastante bien al motor real, puesto que en esta simulación los datos fueron generados en tiempo real

Obs. Se observa una oscilación del modelo del motor al comienzo, esto es debido a que el filtro que modelaba el motor, no tenía ninguna entrada y su comportamiento resultó errático.

## 7. Conclusión

Se ha implementado un controlador simple para un motor de corriente continua, que ha traído diversas complejidades entre ellas, la dificultad de sensar rápidamente la velocidad mediante un encoder o la no linealidad que presenta el motor al arranque haciéndolo difícil de controlar. Por otro lado se tuvo un primer acercamiento a interfaces HMI, a modelar y controlar dispositivos en tiempo real.