Esta práctica consiste en controlar un circuito RC real, mediante la utilización de un microcontrolador y un sistema operativo de tiempo real. El circuito a controlar es el siguiente:

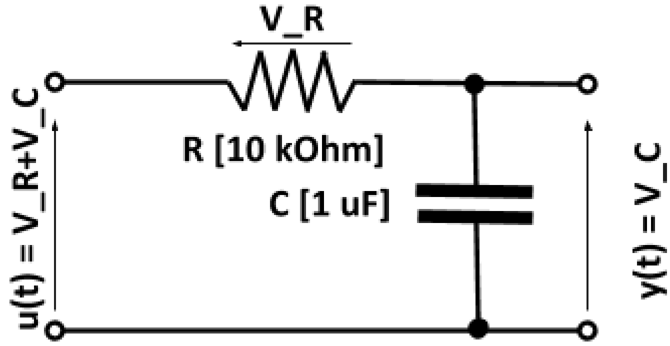


Figura 1. Circuito RC

Ejercicio 1

Obtener matemáticamente la transferencia de tiempo continuo del circuito (en función de R y C).

Las ecuaciones eléctricas del circuito son:

- $v_R(t) = i(t) * R$
- $i(t) = C * \frac{d}{dt} v_C(t)$

Que en base a la Figura 1 puede reescribirse como:

$$u(t) = i(t) * R + y(t) \Rightarrow u(t) = RC * \frac{d}{dt} y(t) + y(t)$$

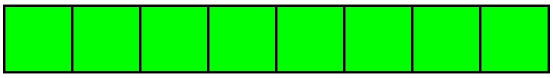
Y aplicando la transformada de Laplace:

$$U(s) = RC * sY(s) + Y(s) \Rightarrow U(S) = (RCs + 1)Y(S)$$

Operando para obtener el cociente:

$$H(s) = \frac{Y(S)}{U(S)} = \frac{1}{RCs+1} = \frac{\frac{1}{RC}}{s+\frac{1}{RC}}$$

Que corresponde a un sistema de orden 1 con polo en $\frac{-1}{RC}$



A partir de la transferencia en tiempo continuo obtener una representación en variables de estado en tiempo continuo (en función de R y C)

En base a la ecuación anterior:

- $u(t) = i(t) * R + y(t) \Rightarrow u(t) = RC* \frac{d}{dt}y(t) + y(t)$

Ya que $y(t) = V_C(t)$ $\Rightarrow$ ya que $V_C(t) = \frac{1}{C}\int_0^t i(t)dt \Rightarrow y(t) = \frac{1}{C}\int_0^t i(t)dt = \frac{1}{C}x(t)$ si tomamos como estado a $x(t) = \int_0^t i(t)dt$ y sabiendo que $\frac{d}{dt}x(t) = i(t)$ se puede reemplazar en la ecuación (3) para obtener:

- $u(t) = i(t) * R + y(t) \Rightarrow u(t)\frac{1}{R} - y(t)\frac{1}{R} = i(t)$
- Usando $i(t) = \frac{d}{dt}x(t)$ y $y(t) = \frac{1}{C}x(t)$: $\frac{d}{dt}x(t) = -\frac{1}{RC}x(t) + \frac{1}{R}u(t)$

$$\frac{d}{dt}x(t) = -\frac{1}{RC}x(t) + \frac{1}{R}u(t)$$

Por lo tanto las ecuaciones de estado son:

- $\frac{d}{dt}x(t) = -\frac{1}{RC}x(t) + \frac{1}{R}u(t)$
- $y(t) = \frac{1}{C}x(t)$

Y las matrices del sistema:

- $[A] = [-\frac{1}{RC}]$
- $[B] = [\frac{1}{R}]$
- $[C] = [\frac{1}{C}]$
- $[D] = [0]$
- $\frac{d}{dt}x(t) = [A]\,x(t) + [B]\,u(t)$
- $y(t) = [C]\,x(t) + [D]\,u(t)$

Ejercicio 2

A partir de la representación en variables de estado en tiempo continuo, obtener matemáticamente la representación en variables de estado en tiempo discreto (en función de R, C y H). Utilizar la aproximación ZOH.

Las matrices que representan el sistema continuo se representaron en el Ejercicio 1 entre corchetes y mayúsculas. Para las que corresponden al sistema discreto se harán igualmente entre corchetes, pero en minúscula.

Utilizando la aproximación ZOH:

- $[a] = e^{-\frac{h}{RC}}$
- $[b] = C(1 - e^{-\frac{h}{RC}})$
- $[c] = [C]$
- $[d] = [D]$

En el modelo discreto tengo que realizar los siguientes cambios además:

- $\frac{d}{dt}x(t) \Rightarrow x[k+1]$
- $x(t) \Rightarrow x[k]$

Entonces las ecuaciones de estado son:

- $x[k+1] = e^{-\frac{h}{RC}} x[k] + C(1 - e^{-\frac{h}{RC}})u[k]$
- $y[k] = \frac{1}{C} x[k]$

Obtener la ecuación de recurrencias del circuito RC.

Partiendo de la ecuaciones de estado y aplicando la transformada Z en ambas se obtiene:

- $zX[z] = e^{-\frac{h}{RC}} X[z] + C(1 - e^{-\frac{h}{RC}})U[z]$
- $Y[z] = \frac{1}{C} X[z] \Rightarrow X[z] = C * Y[z]$

Reemplazando todas las $X[z]$ para dejar todo en función de $Y[z]$ y $U[z]$:

$$zC * Y[z] = Ce^{-\frac{h}{RC}} Y[z] + C(1 - e^{-\frac{h}{RC}})U[z]$$

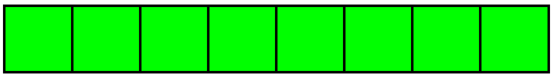
$$Y[z](z - e^{-\frac{h}{RC}}) = (1 - e^{-\frac{h}{RC}})U[z]$$

$$H[z] = \frac{Y[z]}{U[z]} = \frac{1-e^{-\frac{h}{RC}}}{z-e^{-\frac{h}{RC}}}$$

Si lo escribimos nuevamente como $Y[z](z - e^{-\frac{h}{RC}}) = (1 - e^{-\frac{h}{RC}})U[z]$ y operamos:

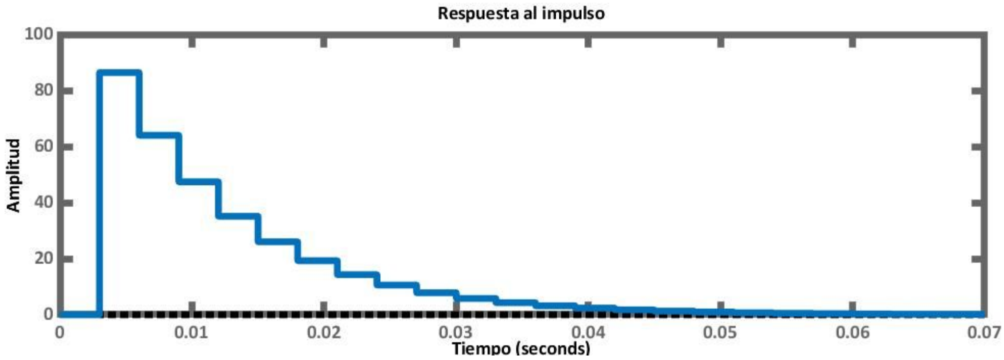
$$Y[z]z - Y[z]e^{-\frac{h}{RC}} = (1 - e^{-\frac{h}{RC}})U[z]$$

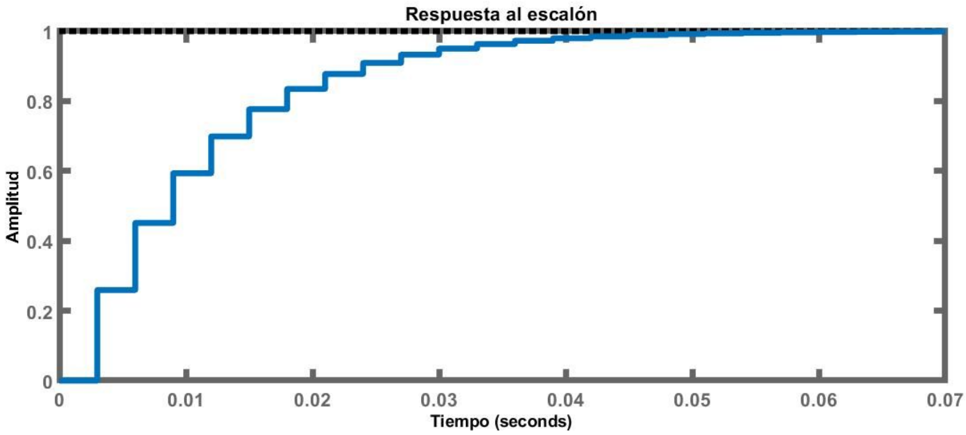
Al antitransformar:

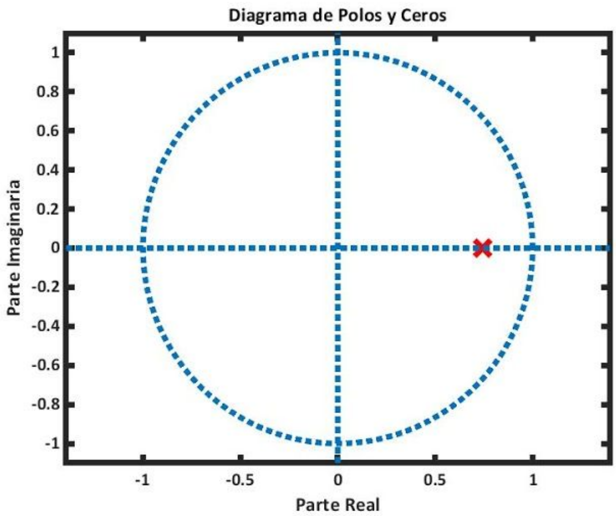


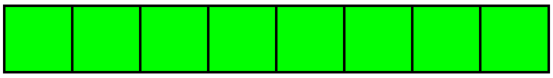
$$y[k+1] = e^{-\frac{h}{RC}} y[k] + (1 - e^{-\frac{h}{RC}})u[k]$$

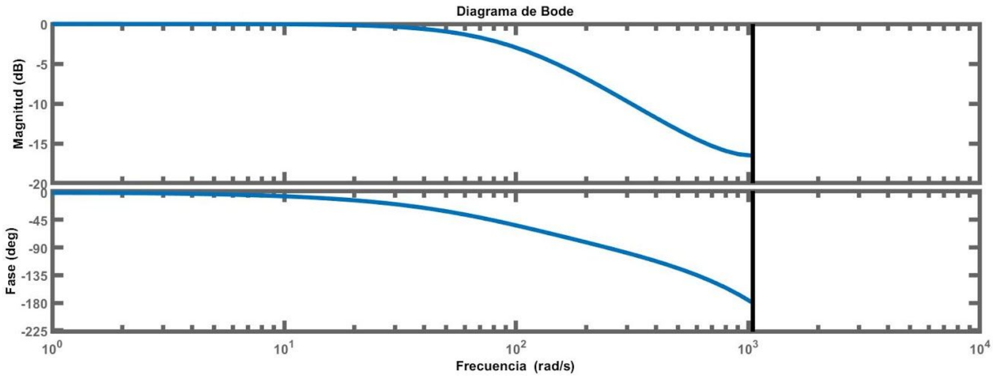
El modelo discretizado de la planta es simulado en MatLab y se obtuvieron los siguientes gráficos:











Ejercicio 3

Seleccionar un tiempo de muestreo que cumpla con todas las restricciones prácticas vistas en clase. Para esto, medir el tiempo de cómputo del controlador PID en la práctica.

Podemos aplicar dos de los criterios vistos en la práctica:

- $20\,\tau_c \le h \le \frac{1}{60 f_{BW}}$
- $\frac{\tau_r}{10} \le h \le \frac{\tau_r}{4}$

En la primera ecuación, τ_c es el tiempo de cómputo que al modelar la planta en una EDU-CIAA con procesador Cortex-M4 de 200 MHz es equivalente a unos 25 μseg.

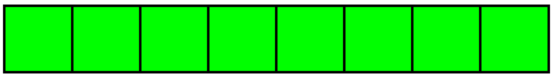
En la segunda ecuación se tiene $\tau_c = 2.2 * R * C = 2.2 * (10K\Omega) * (1\ \mu F) = 22\ mseg$

Reemplazando todo: $2.2\ mseg \le h \le 5.5\ mseg \Rightarrow h$ ~ 3 mseg

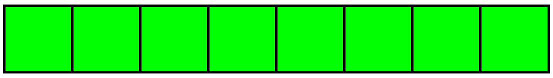
Ejercicio 4

Inyectar en el microcontrolador una señal cuadrada de 10 Hz como señal de referencia.

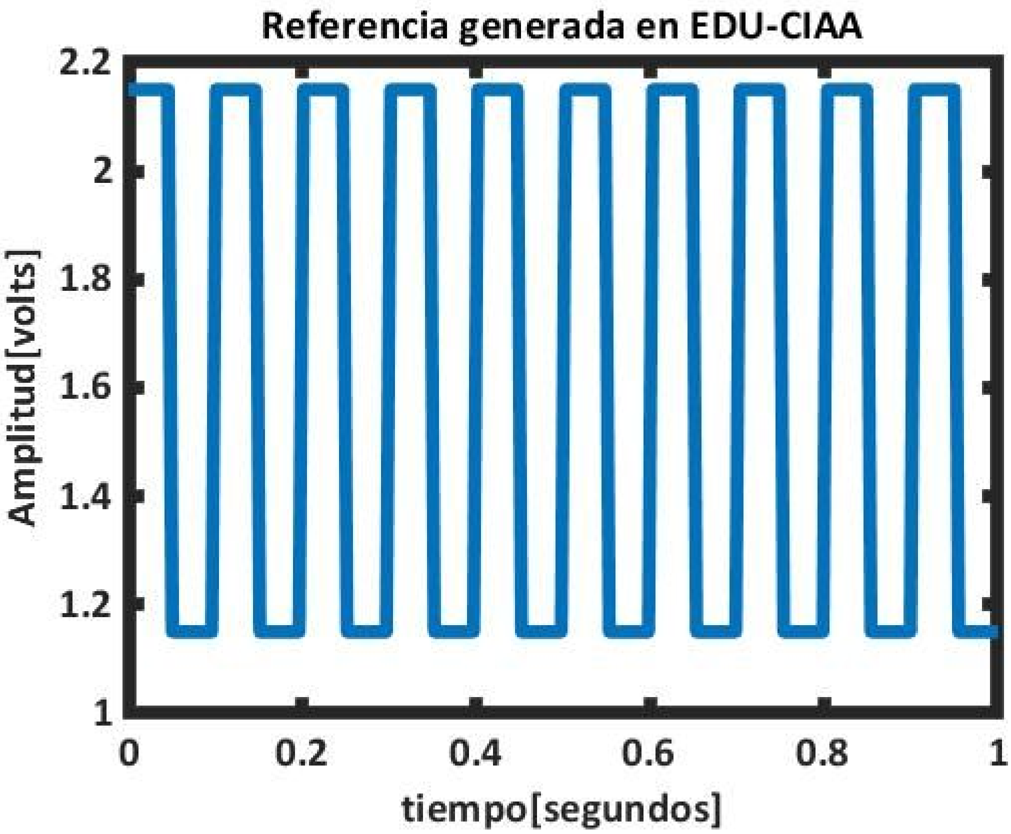
Se generó la señal de referencia (r[k]) internamente en la EDU-CIAA con el siguiente código:



```
void generar_referencia( void* taskParmPtr )
{
  // ---------- CONFIGURACIONES ------------------------------
  printf( "Tarea generadora de señal de referencia.\r\n" );

  // Tarea periodica cada 50 ms
  portTickType xPeriodicity =  50 / portTICK_RATE_MS;
  portTickType xLastWakeTime = xTaskGetTickCount();


  uartWriteString( UART_PC, "\n\rr[k] =        [" );

  // ---------- REPETIR POR SIEMPRE --------------------------
  while(TRUE) {

            if ( i_R < N_SMUESTRAS)
            {
                        dacWrite( DAC, lastDacValue );

                        if( lastDacValue == DAC_MAX){
                         if  (i_r == n_MUESTRAS)
                         {
                                    lastDacValue = DAC_MIN;
                                    i_r = 0;
                         }
                         R[i_R] = (float)DAC_MIN * 3.3/1023.0;
                         gpioWrite( LED1, OFF );
                         uartWriteString( UART_PC, float_to_string( R[i_R] ) );
                         i_R++;
                         i_r++;
                        } else{
                                   if  (i_r == n_MUESTRAS )
                                   {
                                    lastDacValue = DAC_MAX;
                                    i_r = 0;
                                   }
                         R[i_R] = (float)DAC_MAX * 3.3/1023.0;
                         gpioWrite( LED1, ON );
                         uartWriteString( UART_PC, float_to_string( R[i_R] ) );
                         i_R++;
                         i_r++;
                        }

                        if (i_R < N_MUESTRAS )
                         uartWriteString( UART_PC, ";");

                        if( i_R == N_MUESTRAS )
                        {
                          uartWriteString( UART_PC, "];\r\n");
                          i_R++;
                        }

            }
            // Envia la tarea al estado bloqueado durante xPeriodicity (delay periodico)
            vTaskDelayUntil( &xLastWakeTime, xPeriodicity );
  }
}
```



La planta se modeló en un circuito RC externo en un protoboard a la cual se le inyecta en la entrada la señal r[k] y se lee la tensión de su capacitor (y[k]), ambos por la EDU-CIAA por medio de los puertos de expansión para el manejo de señales analógicas.



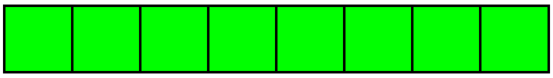
Se generaron 200 muestras, con 10 muestras por estado, cada 5ms. Además cada señal fue enviada al DAC adjudicando 1.15 V al estado bajo y 2.15 V al estado alto (el DAC va entre 0 y 3.3 V)

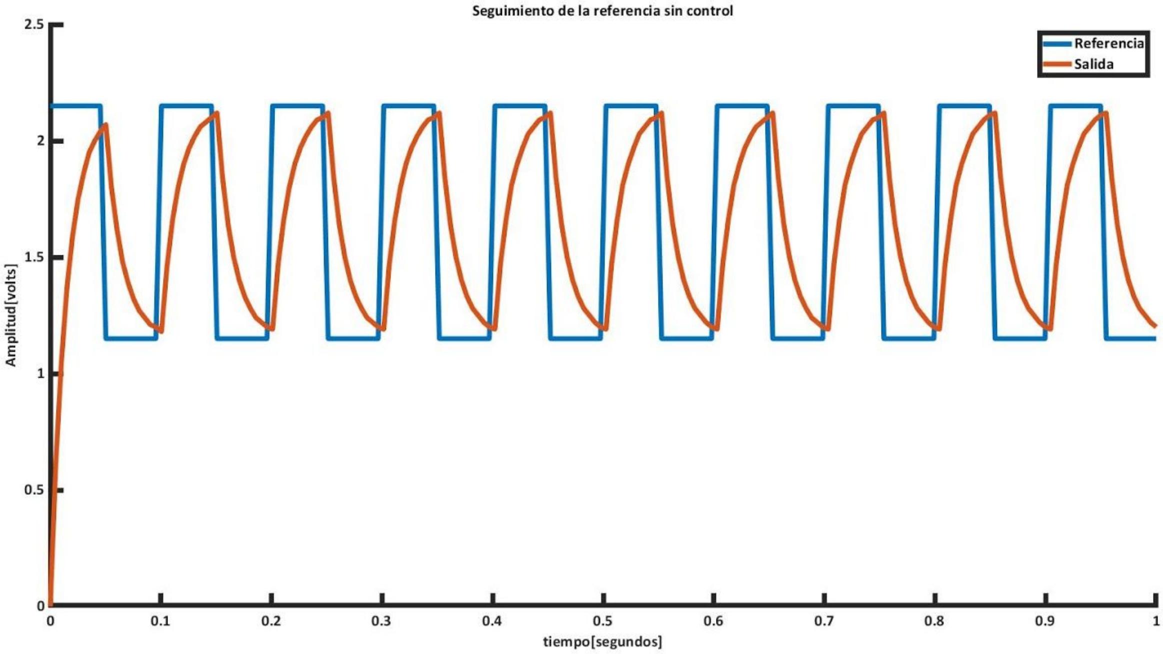
Sin aplicar ninguna técnica de control, aplicar la misma señal como acción de control para obtener la respuesta al escalón del circuito RC.

Se aplicó la señal r[k]=u[k] generada internamente a uno de los puertos analógicos para inyectar en la planta. Luego la señal de salida de la planta es leída en otro puerto analógico y ambos resultados (u[k] , y[k]) son impresos en pantalla para ser exportados a matlab.

Obtener el gráfico de la respuesta al escalón.

Los valores obtenidos fueron graficados en matlab para obtener la siguiente figura.





Como se puede apreciar, la salida termina dentro del rango de la referencia pero la morfología de la señal dista mucho de ser similar.

Ejercicio 5

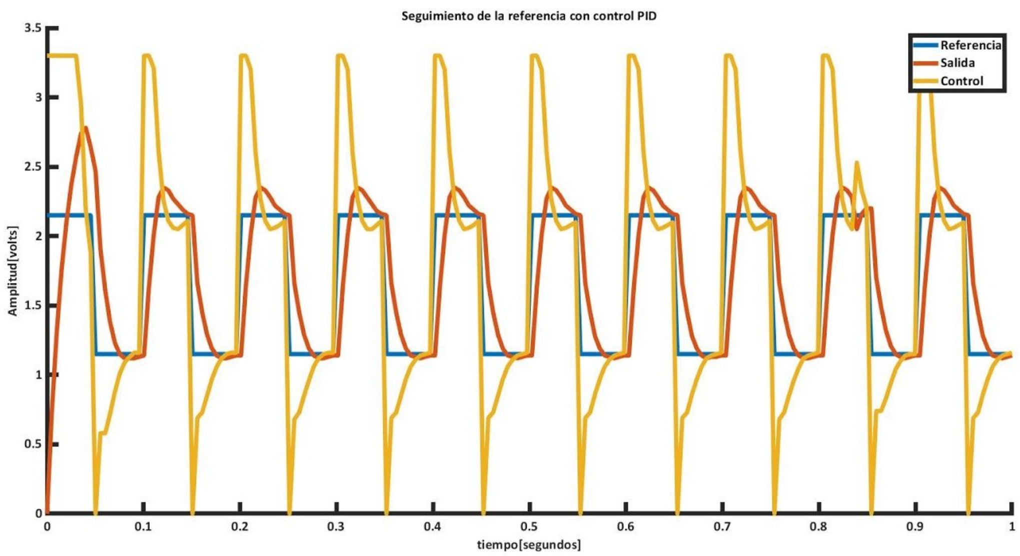
Implementar en el microcontrolador, un control PID para modificar la respuesta al escalón del circuito RC (debe ser una modificación visible respecto a la respuesta al escalón).

Se adaptó el pid_controller.c de la biblioteca de la EDU-CIAA con los siguientes parámetros:

- h = 3 msegs
- K_p = 2.80
- K_i = 366.67
- K_d = 0.00027
- b = 1
- N = 200 muestras

Aplicar la acción de control al circuito RC. Obtener el gráfico de la respuesta del circuito RC más el control PID.

El resultado obtenido fue el siguiente:



Ejercicio 6

Obtener matemáticamente un control por pole-placement del circuito RC. Debe quedar en función de R, C y K.

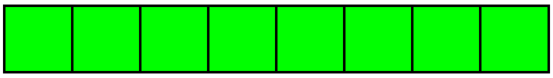
Utilizando MatLab para los cálculos, se analizó la controlabilidad (*ctrb(A,B)*) del sistema discretizado calculando el rango (rank(Mc)) de la matriz de controlabilidad.

- *Mc = ctrb(a,b);*
- *Rango = rank(Mc);*

Debe ser del mismo orden de la planta para que sea controlable. Como Rango = 1, entonces el sistema es controlable.

El polinomio característico del sistema podemos calcularlo como: $|zI - [a]| = 0$ y se obtiene:

$$z - e^{-\frac{h}{RC}} = 0$$



Aplicando la fórmula de Ackerman: $K = {M_c}^{-1}\ \Phi([a])$ eligiendo un polo a lazo cerrado de valor 0.5: $K = \frac{[a]-0.5}{[b]} = \frac{e^{-\frac{h}{RC}}-0.5}{C(1-e^{-\frac{h}{RC}})} = \frac{e^{-\frac{3\,ms}{20\,ms}}-0.5}{1\ \mu F(1-e^{-\frac{3\,ms}{10\,ms}})} = 929148$

Aplicando el teorema del valor final:

$$K_0 = \frac{1-[a]+[b]*K}{[c][b]} = \frac{1-e^{-\frac{3\,ms}{20\,ms}}+1\ \mu F(1-e^{-\frac{3\,ms}{10\,ms}})*929148}{\frac{1}{1\mu F}1\ \mu F(1-e^{-\frac{3\,ms}{10\,ms}})} = 1.929148$$

Finalmente: $u[k] = K_0 * r[k] - K * \frac{y[k]}{[c]}$

Ejercicio 7

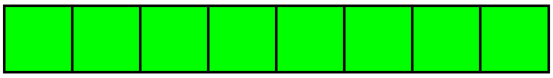
Implementar en el microcontrolador un control por pole placement. Ubicar el polo de manera tal que se produzca un cambio visible en la respuesta al escalón del circuito RC.

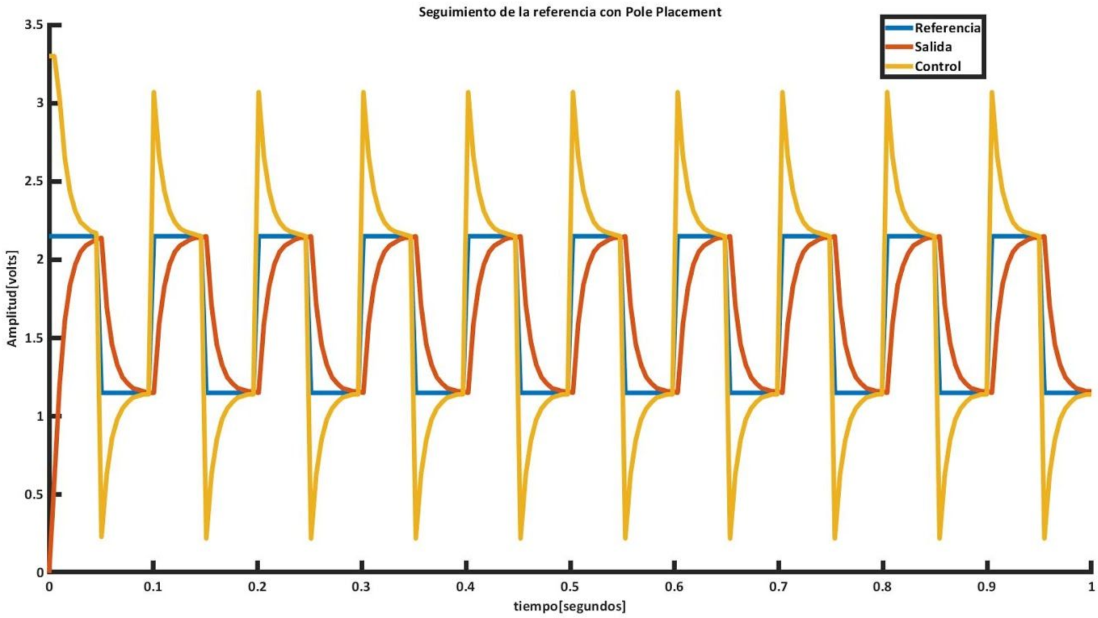
Se adaptó el código de la biblioteca sAPI para que la salida sea ahora tal como se definió en el ejercicio anterior, con los parámetros calculados.

```
#if (MODO == POLE)
        K  = 929148;
        K0 = 1.929148;
        Capacitor = CAPACITOR;

                // x[k] = y[k]*c // Variable de estado x
        float x = y * Capacitor;

        /*----- Calculando salida del controlador (U)----------------------------------*/

        // u[k] = K0*r[k] - K*x[k]
        float U = K0 * r  - K * x;
#endif
```

Obtener el gráfico de la respuesta del circuito RC más el control Pole-Placement.

Los valores obtenidos fueron graficados en MatLab para obtener la siguiente figura:





Ejercicio 8

Implementar en el microcontrolador el algoritmo de identificación iterativo. Imprimir por consola el valor de R estimado tomando C como un valor conocido.

Se inyectó ruido aleatorio como señal de entrada a la planta, se mide la salida y tomando el valor de C como conocido ($1\mu F$) se procede a estimar el valor de la resistencia (cercano a 10 KΩ). El algoritmo generado itera N=1000 veces en cada ejecución hasta alcanzar un valor de convergencia (utilizando estados iniciales nulos) . Se obtuvieron los siguientes resultados:

- $\Theta = [0.772; 0.003; 0.226]$
- $a_0 = 0.772\ (estimado)$
- $R = 11609.480\Omega = 11.6K\Omega\ (estimado)$