

UNIVERSIDAD DE CONCEPCIÓN

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA



Informe Laboratorio de control e instrumentación

Laboratorio N°3: Técnicas de Control Adaptativo Gain Scheduler.

Jocelyn Matus Ancavil.
Rafael Burgos Yousuff.

Profesor Juan Pablo Segovia Vera.

Concepción, miércoles 09 de diciembre 2020.

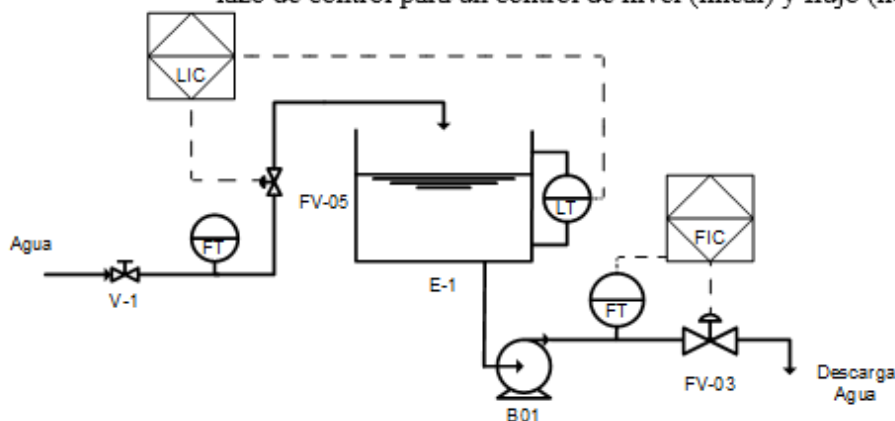
Enunciado

LABORATORIO DE CONTROL

Proyecto N°3: “Técnicas de Control Adaptativo Gain Scheduler”

Objetivos : Diseñar, Configurar y conectar un sistema de control PID adaptativo utilizando una estrategia Gain Scheduler.
Modelar, simular, contrastar y concluir acerca de técnica adaptativa Gain Scheduler aplicada a un lazo de control de flujo.

Desarrollo : Para el proceso descrito en la siguiente figura configurar y sintonizar un lazo de control para un control de nivel (lineal) y flujo (no lineal)



- Diseñar modelo no lineal que describa el comportamiento de válvula de descarga.
- Programar en PLC Control Logix usando ladder y texto estructurado la lógica de sintonización control de flujo de descarga mediante gain scheduler contra modelo.
- Desarrollar un set de pruebas para verificar la sintonía lineal versus técnicas adaptativa Gain Scheduler.

Informe :

- Programación y configuración PID-Gain scheduler sobre RSLogix 5000.(1.5 puntos)
- Factory Acceptance Test (FAT) para control de flujo, mediante conexión RS Logix 5000 vía OPC a simulink, utilización de modelos No lineal obtenidos en laboratorio N°1.(1.5 puntos)
- Contrastar para diversos métodos lineal versus no lineal.(2.0 pts)
- Conclusiones y resultados.(1 pto)

Fecha de entrega: diciembre 09 de 2020, hasta las 18:00 horas en Infoda.

Formato: <Informe_3_Grupo_##.rar o zip>

Diagramas de la planta

Diagrama P&ID (Piping And Instrument Diagram)

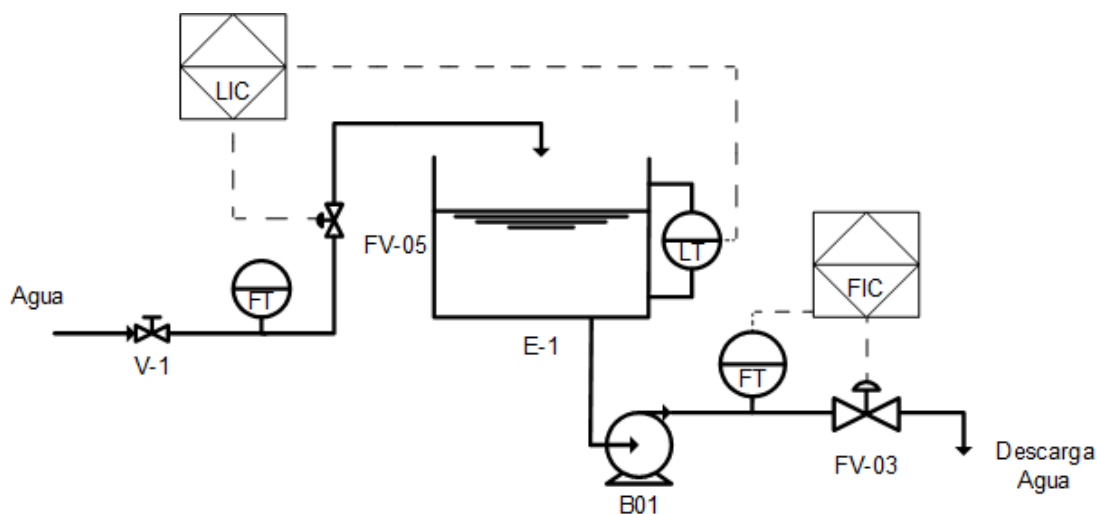


Figura N°1, Diagrama P&ID

Diagrama SAMA (Scientific Apparatus Manufacture Asociación)

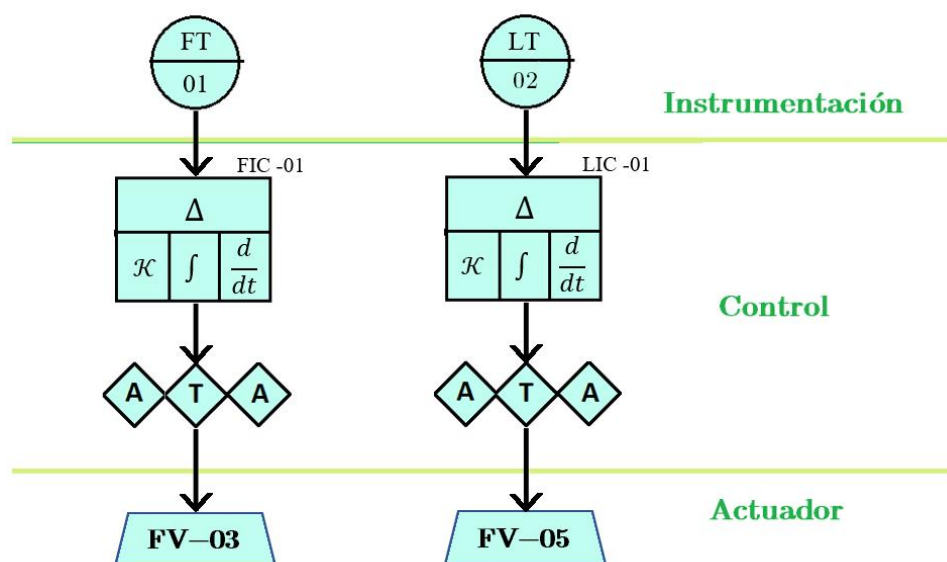


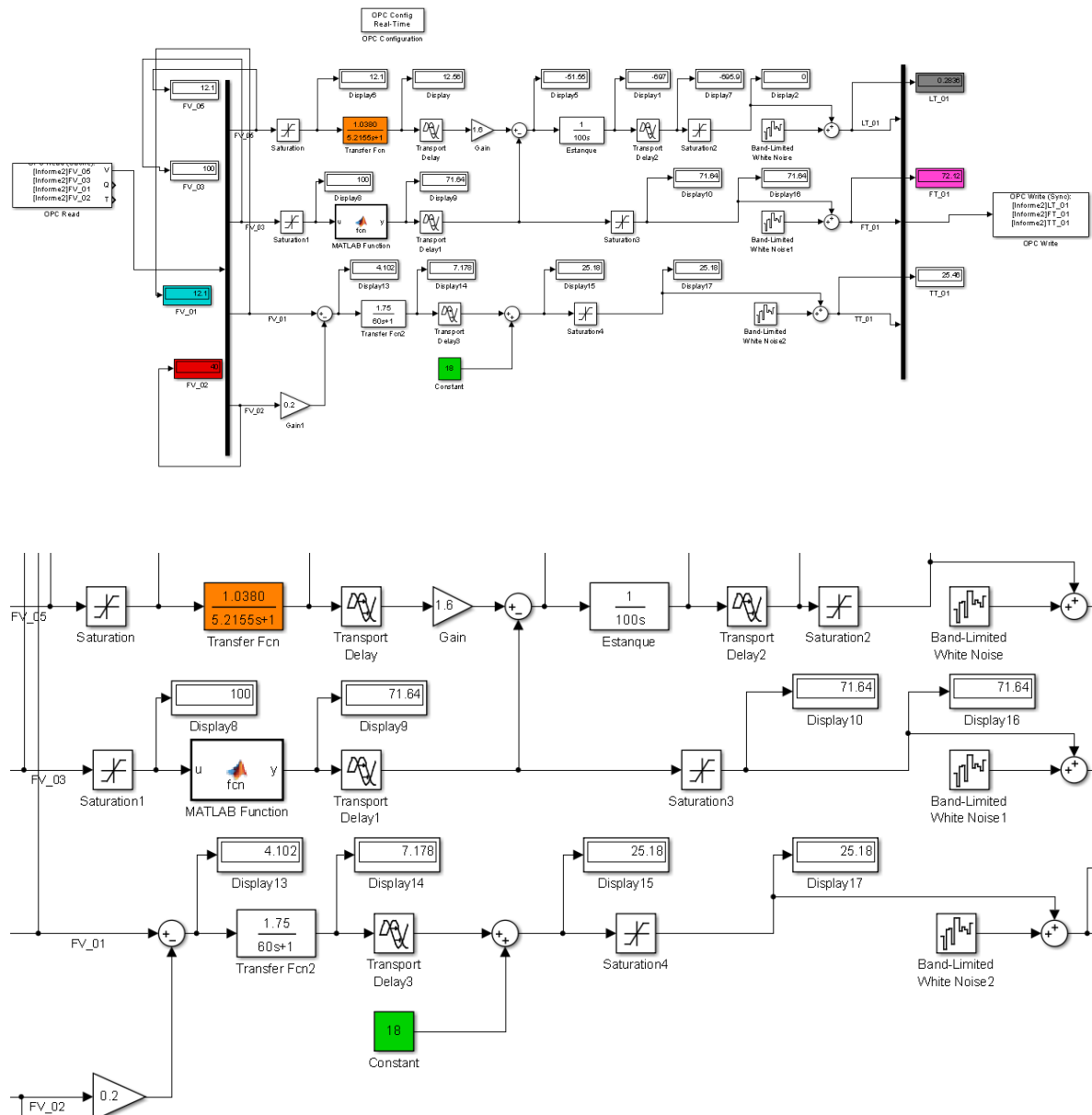
Figura N°2, Diagrama SAMA de la planta.

2. Parte a)

2.1 Diagramas

Diagrama de bloque de planta:

En lo que se refiere a la planta, se simulo la planta en Simulink de la siguiente manera:



2.2 Programación y configuración

En lo referente a la programación, se utilizó varios programas para poder realizar el método Gain Scheduler.

Para esto, para poder obtener los valores necesarios para poder realizar la interpolación, se tuvo que realizar el ajuste de mínimos cuadrados. Para esto, se utilizó el siguiente código entregado por el profesor:

```
close all
clear
clc
% Classify Text Data Using Deep Learning
filename = "testdata.xlsx";
opts = detectImportOptions(filename)
%preview(filename,opts)% Preview 8 rows of the data
data = readtable(filename,'TextType','char');
head(data)
w=table2array(data);
tf=max(w(:,1));
figure
t=linspace(1,tf);
subplot(2,1,1),plot(w(:,1),w(:,2),'r'),
hold,plot(w(:,1),w(:,3),'b'),plot(w(:,1),w(:,4),'g'),ylabel('sp, h y cv01
(%)'),xlabel('t(s)'),legend(['ref' 'nivel' 'cv01']);
subplot(2,1,2),plot(w(:,1),w(:,6),'r'),hold,plot(w(:,1),w(:,5),'b'), title('qsal
y cv02 (%)'),xlabel('time [sec]'),ylabel('ft01 y cv02 [%]'),legend(['qsal';'
cv02']);
figure

subplot(2,1,1)

%plot(w(:,1),w(:,6),'r'),hold,stairs(w(:,1),w(:,5),'b'), title('qsal y cv02
(%)'),xlabel('time [sec]'),ylabel('ft01 y cv02 [%]'),legend(['qsal';' cv02']);
```

1. Análisis de linealidad de válvula de descarga

```
pd=diff(w(:,5)); % busca delta señal a válvula de flujo descarga
subplot(2,1,2),plot(w(2:length(w),1),pd),hold
ptros=find(pd>=1 | pd<=-1); %punteros
ptros=[1; ptros; length(w)];
```

2. Calcula el valor medio entre cambios de u(k)

```
t0=0.1; % tiempo de muestreo (s)
delay=15/t0;
for i=1:length(ptros)-1
    lapso=ptros(i)+delay:1:ptros(i+1)-1;
    %subplot(2,1,2),plot(lapso*t0,pd(lapso),'r'),
    % % calcula la media de uk e yk
    muk(i)=mean(w(lapso,5)); % media uk
    myk(i)=mean(w(lapso,6)); % media yk
    %subplot(2,1,1),plot(lapso*t0,myk(i)*ones(1,length(lapso)),'g'),
    lapso2(i)=sum(lapso)/length(lapso);
end
%
```

```

figure
%subplot(1,2,1),plot(muk,myk,'.-'),xlabel('variación uk[%]'),ylabel('variación
yk[%]'),title('relación estática qs vs uk')
dy=diff(myk);
du=diff(muk);
kv=dy./du;
subplot(1,2,2),plot(muk(2:length(lapso2)),kv,'.'),hold,xlabel('x:variación
uk[%]'),ylabel('y:Ganacia (0/1)')
% % ajuste polinómico y=f(x)
x=muk(2:length(lapso2))'; % se define valor de variable x
y=kv'; % datos de la ganacia válvula
% M=[x,ones(size(x))]; % y=p(1)*x+p(2) modelo (1)
M=[x.^2,x,ones(size(x))]; % y=p(1)*x^2+p(2)*x+p(3) modelo (2)
p=M\y %coeficientes del polinomio mediante minimos cuadrados
grid on
title('Polinomio aproximador')
u=0:1:100;
% Calculo de la ganacia estimada
% ke=p(1).*u+p(2); %modelo (1)
ke=p(1)*u.^2+p(2).*u+p(3); % Modelo (2)
%plot(u,ke,'r')
hold off

x2= muk(2:length(lapso2)-28)';
M2= [x2.^4, x2.^3, x2.^2, x2, ones(size(x2))];
yTau= [2.4; 2.4; 2.34; 2.406; 2.21197; 2.088; 1.881523; 2.0408785; 2.1276133];
pTau=M2\yTau
ktau= pTau(1)*u.^4 + pTau(2)*u.^3 + pTau(3)*u.^2 + pTau(4)*u + pTau(5);

figure
%plot(u, ktau, [20 30 40 50 60 70 80 90 100], yTau(:), 'or')
hold on

```

La mayoría del código no está modificado, a excepción de lo siguiente:

```

x2= muk(2:length(lapso2)-28)';

M2= [x2.^4, x2.^3, x2.^2, x2, ones(size(x2))];

yTau= [2.4; 2.4; 2.34; 2.406; 2.21197; 2.088; 1.881523; 2.0408785; 2.1276133];

pTau=M2\yTau

ktau= pTau(1)*u.^4 + pTau(2)*u.^3 + pTau(3)*u.^2 + pTau(4)*u + pTau(5);

figure

%plot(u, ktau, [20 30 40 50 60 70 80 90 100], yTau(:), 'or')
hold on

```

Antes de explicar la sección de código antes mencionada, se realizó, anteriormente, un ajuste de mínimos cuadrados para obtener los parámetros del polinomio que se ajustara mejor al comportamiento de la ganancia de la planta, que ya se encontraba en el código entregado.

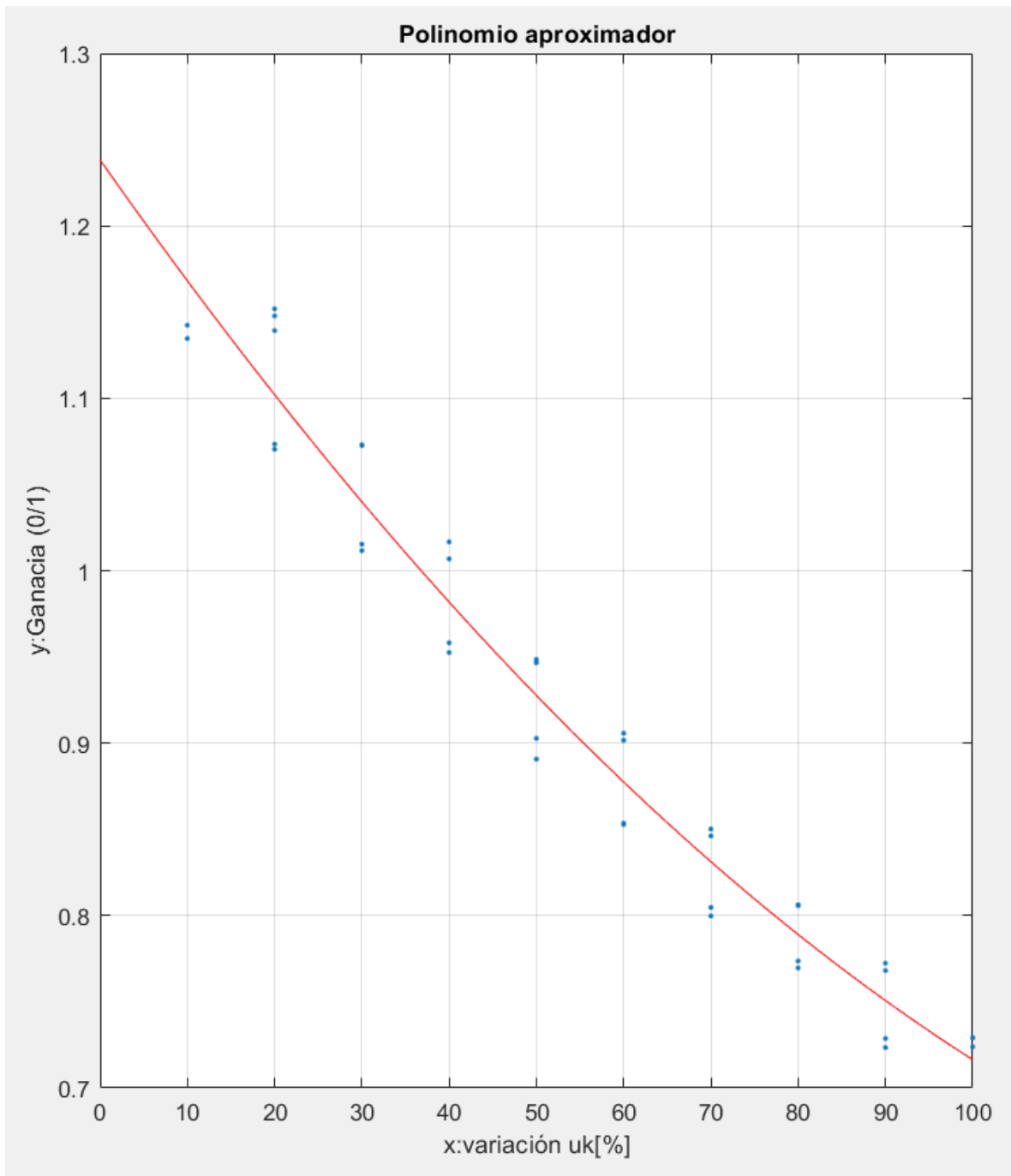
De la misma manera, se buscó un polinomio que se ajustara al comportamiento del cambio de constante de tiempo que tiene la planta.

Se calculo los τ para cada respuesta a cambio escalón del sistema para tener un polinomio que se ajustara más a la realidad para realizar el ajuste de mínimos cuadrados. Con esto, se calcularon los coeficientes y se encontraron los siguientes polinomios

Para el polinomio que se ajusta al comportamiento de la ganancia:

$$f(u) = 1.986 \cdot 10^{-5} \cdot u^2 - 0.0072 \cdot u + 1.2382$$

Se configuro el código para que el polinomio que se ajustara sea de segundo grado. En el siguiente grafico se contrasta gráficamente el polinomio con los valores de ganancia:

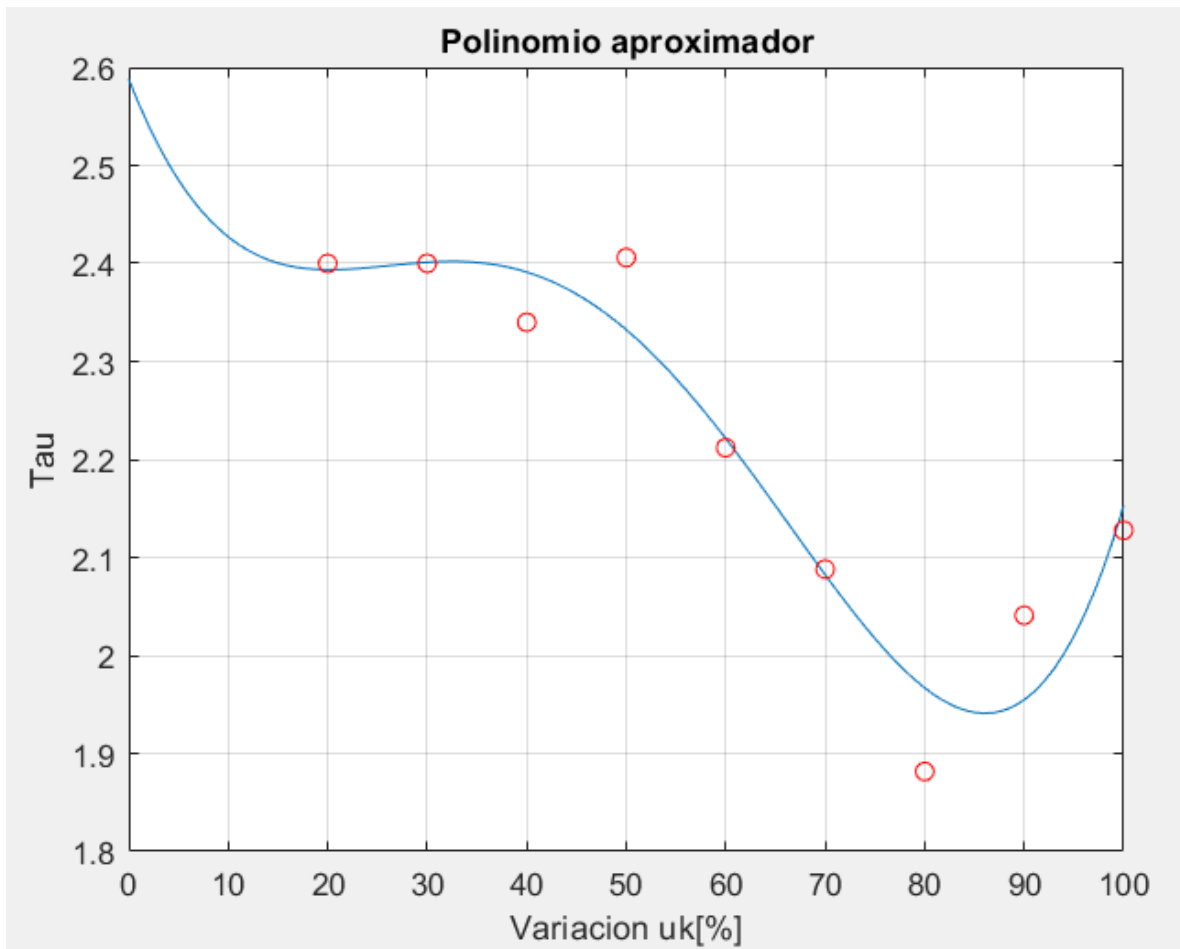


Para el polinomio que se ajuste al comportamiento de la constante de tiempo:

$$f(u) = 1.1517 \cdot 10^{-7} \cdot u^4 - 2.131008 \cdot 10^{-5} \cdot u^3 + 0.00119 \cdot u^2 - 0.02604 \cdot u + 2.587$$

Para este caso, se quiso ajustar los datos a un polinomio de cuarto grado

En el siguiente grafico se contrasta gráficamente el polinomio con los valores de constante de tiempo obtenidos:



Con esto ya obtenido, se tiene que, para poder representar el comportamiento no lineal de la planta, con la ayuda de los polinomios anteriormente calculados, se utilizó el siguiente código, el cual está dentro del bloque Matlab function de la planta:

```
function y = fcn(u)
%#codegen
persistent yk1 uk1 uk2 uk3 ukN1

if isempty(yk1)
    yk1= 0;
    uk1= 0;
    uk2= 0;
    uk3= 0;
    ukN1= 0;
end

kv = (1.986864628632069e-05*u^2) - (0.007205165697606*u) + 1.238208466872341;
tau = (1.151761276223747e-07*u^4) - (2.131008401320826e-05*u^3) +
(0.001196127667541*u^2) - (0.026040342735040*u) + 2.587971649999962;
theta= 0.6;
t0 =0.1;
N = theta/t0;
alpha= exp(-t0/tau);
beta= kv*(1-alpha);

y = alpha*yk1 + beta* ukN1;
```

```
yk1= y;  
uk1= u;  
uk2= uk1;  
uk3= uk2;  
ukN1= uk3;
```

Para poder representar el comportamiento no lineal de la válvula, se ajustó a uno de sistema de primer orden, en el cual los parámetros son cambiantes.

Ahora, en lo que se refiere a la programación del controlador, en RSLogixx, se tiene el siguiente texto estructurado para el funcionamiento del controlador PID de la válvula FV_03:

```
if sw_m_al then sp_Flujo:= uk1_n + sign_n*abs(duk_n); end_if;  
  
//Selección de alpha  
  
if SW_GS= 0 then alpha:= FV_03;  
elseif SW_GS= 1 then alpha:= sp_Flujo;  
elseif SW_GS= 2 then alpha:= FT_01;  
end_if;  
  
alpha:= alpha;  
  
//jsr(GS_Kc_f, 2, alpha, metodo, Kc_f);  
//jsr(GS_Ti_f, 2, alpha, metodo, Ti_f);  
  
//Parametros del modelo PID discreto  
  
q0_f:= Kc_f*(1+(Tm_f/(2*Ti_f)))+(Td_f/Tm_f)*sw_d_f; //Los q estan relacionados con la derivada del error  
q1_f:= -Kc_f*(1-(Tm_f/(2*Ti_f)))+(2*Td_f/Tm_f)*sw_d_f;  
q2_f:= (Kc_f*Td_f/Tm_f)*sw_d_f;  
  
p0_f:= -(1-sw_d_f)*(Kc_f*Td_f/Tm_f); //Los p estan relacionados con la derivada de PV  
p1_f:= (1-sw_d_f)*(2*Kc_f*Td_f/Tm_f);  
p2_f:= -(1-sw_d_f)*(Kc_f*Td_f/Tm_f);  
  
//Valores actuales de error y el PV(y(t))  
  
yk_f:= FT_01; //Guarda lo que entrega el sensor de flujo  
ek_f:= sp_Flujo-yk_f;
```

```

// Banda de proteccion contra el ruido
if ABS(ek_f) < nb_f then ek_f:= 0; end_if;

//Calculo variacion del accionamiento
duk_f:= q0_f*ek_f + q1_f*ek1_f + q2_f*ek2_f + p0_f*yk_f + p1_f*yk1_f + p2_f*yk2_f;

// Calculo del signo de duk
if duk_f>0 then sign_f:= 1;
ELSIF duk_f<0 then sign_f:= -1;
ELSE sign_f:= 0; end_if;

//Limitador de variacion (Proteccion del actuador)
If abs(duk_f) >dumax_f then duk_f:= dumax_f; end_if;

//Calculo del uk_f
if sw_am_f then uk_f:= uk1_f + sign_f*abs(duk_f); end_if; //Si esta en automatico, entonces el uk_f se actualiza al valor pasado de uk_f mas el duk_f

// Limitador maximo y minimo del actuador (Proteccion del actuador)
if uk_f < umin_f then uk_f:= umin_f; end_if;
if uk_f >umax_f then uk_f:= umax_f; end_if;

//Enviar resultado de accionamiento a la valvula correspondiente

if sw_am_f then FV_03:= uk_f; //Aqui, si esta el switch en modo automatico, entonces
// se lleva a la valvula lo que se calculo en uk_f
ELSE FV_03:= FV_03; end_if; //Si esta en manual, nosotros elegimos
// A que valor estara la valvula, y
//el set point requerido para esto
//se actualizara con
//el sensor correspondiente

//Memoria Actuador
uk1_f:= uk_f;

//Memoria error
ek2_f:= ek1_f;
ek1_f:= ek_f;

// Memoria PV
yk2_f:= yk1_f;
yk1_f:= yk_f;

```

En la primera imagen del texto estructurado mostrado, se ve la configuración para poder trabajar en modo Gain Scheduler. El α puede ser seleccionado para seguir a 3 referencias: al set point de flujo, al valor de FV_03, o al valor del sensor de transmisión FT_01.

Después, un poco mas abajo, se encuentra unos comandos de JSR, los cuales llaman a las funciones (textos estructurados) GS_Kc_f y GS_Ti_f para el cálculo de los parámetros Kc y Ti, que cambian constantemente según el valor de α actual.

Entonces, los códigos de estas funciones son las siguientes:

GS_Kc_f:

```

SBR(alphaKc, metodo1Kc);

listKc[0, 0] := 0;
listKc[1, 0] := 40;
listKc[2, 0] := 80;
listKc[3, 0] := 100;

listKc[0, 1] := 3.4;
listKc[1, 1] := 3.5;
listKc[2, 1] := 3.5;
listKc[3, 1] := 6;

nf:= 4;

if metodo1Kc then //Si metodo1Kc = 1, entonces se realiza por tabla. Si metodo1Kc = 0, entonces se realiza interpolacion
    if alphaKc < listKc[1,0] then
        outputKc:= listKc[0,1];

    elseif alphaKc>= listKc[1,0] & alphaKc< listKc[2,0] then
        outputKc:= listKc[1,1];

    else outputKc:= listKc[2,1];

    end_if;
else
    for idKc:= 0 to nf-2 by 1 do
        if alphaKc>= listKc[idKc, 0] & alphaKc<= listKc[idKc+1, 0] then
            phKc[idKc]:= (alphaKc - listKc[idKc+1, 0])/(listKc[idKc, 0] - listKc[idKc+1, 0]);
            phKc[idKc+1]:= (alphaKc - listKc[idKc, 0])/(listKc[idKc+1, 0] - listKc[idKc, 0]);
        else phKc[idKc+1]:= 0;
        end_if;
    end_for;

    outputKc:= 0.0;

    for j:= 0 to nf-1 by 1 do
        outputKc:= outputKc + phKc[j]*listKc[j, 1];
    end_for;

    if alphaKc<= listKc[0, 0] then
        outputKc:= listKc[0, 1];
    end_if;

    if alphaKc>= listKc[nf-1, 0] then
        outputKc:= listKc[nf-1, 1];
    end_if;

end_if;

RET(outputKc);

```

GS_Ti_f:

```

//Lista de datos
SBR(alphaTi, metodo1Ti);

listTi[0, 0] := 0;
listTi[1, 0] := 40;
listTi[2, 0] := 80;
listTi[3, 0] := 100;

listTi[0, 1] := 5.9;
listTi[1, 1] := 5.7;
listTi[2, 1] := 5.5;
listTi[3, 1] := 4.5;

nf:= 4;

if metodo1Ti then //Si metodo1Ti = 1, entonces se realiza por tabla. Si metodo1Ti = 0, entonces se realiza interpolacion
    if alphaTi < listTi[1,0] then
        outputTi:= listTi[0,1];

    elseif alphaTi>= listTi[1,0] & alphaTi< listTi[2,0] then
        outputTi:= listTi[1,1];

    else outputTi:= listTi[2,1];
    end_if;
else
    for idTi:= 0 to nf-2 by 1 do
        if alphaTi>= listTi[idTi, 0] & alphaTi<= listTi[idTi+1, 0] then
            phTi[idTi]:= (alphaTi - listTi[idTi+1, 0])/(listTi[idTi, 0] - listTi[idTi+1, 0]);
            phTi[idTi+1]:= (alphaTi - listTi[idTi, 0])/(listTi[idTi+1, 0] - listTi[idTi, 0]);
        else phTi[idTi+1]:=0;
        end_if;
    end_for;
    outputTi:= 0.0;

    outputTi:= 0.0;

    for j:= 0 to nf-1 by 1 do
        outputTi:= outputTi + phTi[j]*listTi[j, 1];
    end_for;

    if alphaTi<= listTi[0, 0] then
        outputTi:= listTi[0, 1];
    end_if;

    if alphaTi>= listTi[nf-1, 0] then
        outputTi:= listTi[nf-1, 1];
    end_if;
end_if;

RET(outputTi);

```

Este texto estructurado esta, basados en el código que fue entregado por el profesor, el código de Matlab ueval, el cual, matemáticamente, tiene dos modos de trabajo:

1) Si se quiere trabajar el Gain Scheduler a través de intervalos de operación (tabla); o sea, desde un valor de α hasta otro valor de α , se tengan ciertos valores de Kc y Ti, se define el valor de metodo1 a 1, y el texto estructurado se trabaja de esa manera.

2) Si se quiere trabajar como interpolación, el código realiza esta interpolación con los valores de Kc y Ti que se trabajaron en el modo anterior, para que los valores de Kc y Ti cambien durante todos los cambios de α , se tiene que definir el valor de metodo1 a 0.

Antes de que se pueda trabajar el método, se tiene que encontrar los valores de Kc y Ti que corresponderían a los que se trabajan en el modo de tabla.

En este modo, se tiene que dividir el comportamiento no lineal de la válvula en zonas lineales. Dentro de estas zonas lineales, los valores de Kc y Ti son fijos, y cambian de un paso de zona a otro. Estas zonas fueron divididas según el porcentaje de apertura de la válvula FV_05.

Para encontrar estos valores, se tuvo que realizar a través del método Ultimate la sintonización del controlador dentro de esas áreas.

En nuestro caso, dividimos en 3 las zonas de trabajo: desde 0% hasta 40% de apertura, de 40% a 80% de apertura, y de 80 a 100% de apertura.

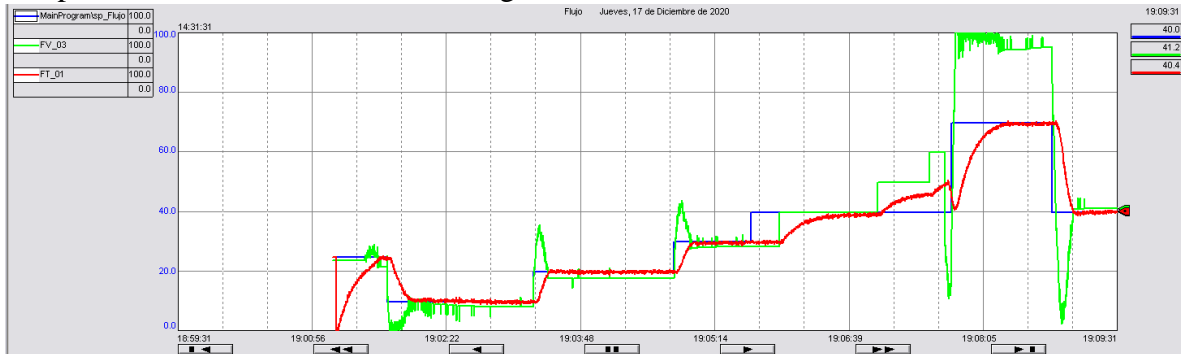
Estos fueron los valores que fueron encontrados para cada zona

	Kc	Ti
0% a 40%	3.5	5.7
40% a 80%	3.5	5.5
80% a 100%	6	4.5

Parte b) Factory acceptance test (FAT)

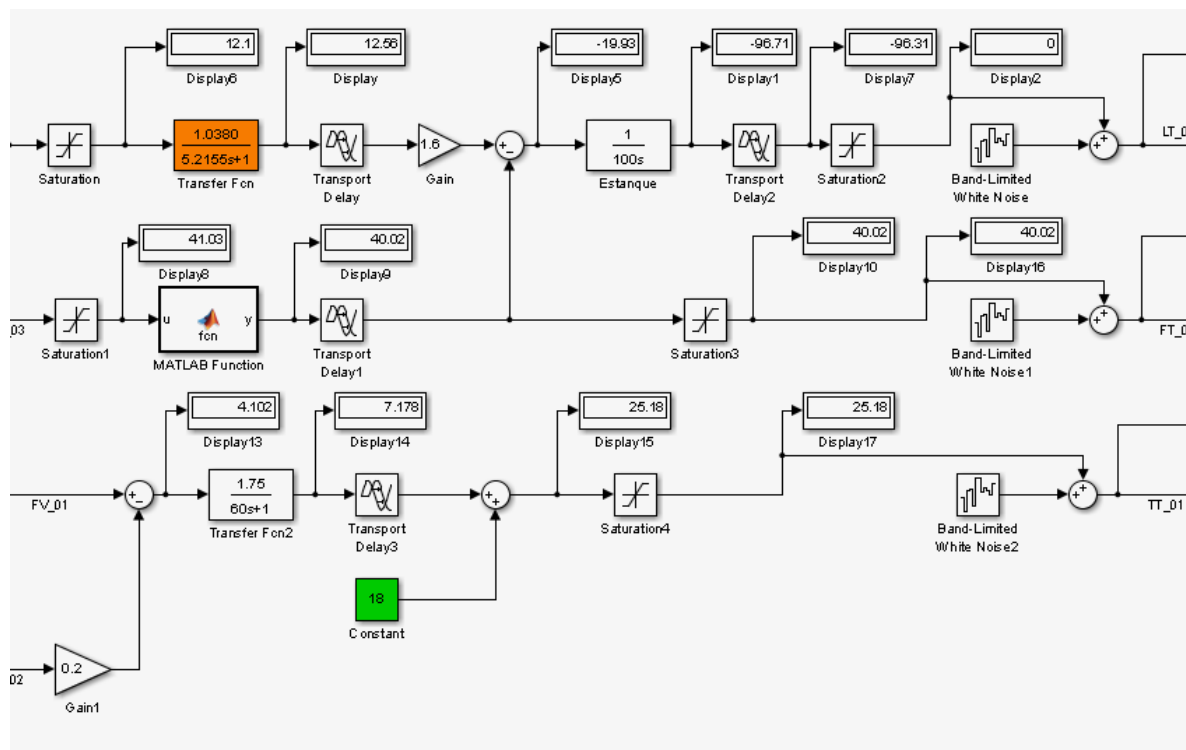
Se tiene los siguientes resultados donde se pueda confirma que la planta funciona correctamente:

Por parte del controlador, se tiene lo siguiente:



Se realizó distintas modificaciones al set point, mientras se cambiaba que el Gain Scheduler se trabajara a través de tabla o interpolación, y además, entre medio, se selecciono el controlador en modo manual para ver la respuesta en lazo abierto, y después, de nuevo, en modo automático para que trabajara el controlador.

Y se tiene una imagen de la planta simulada en Simulink:

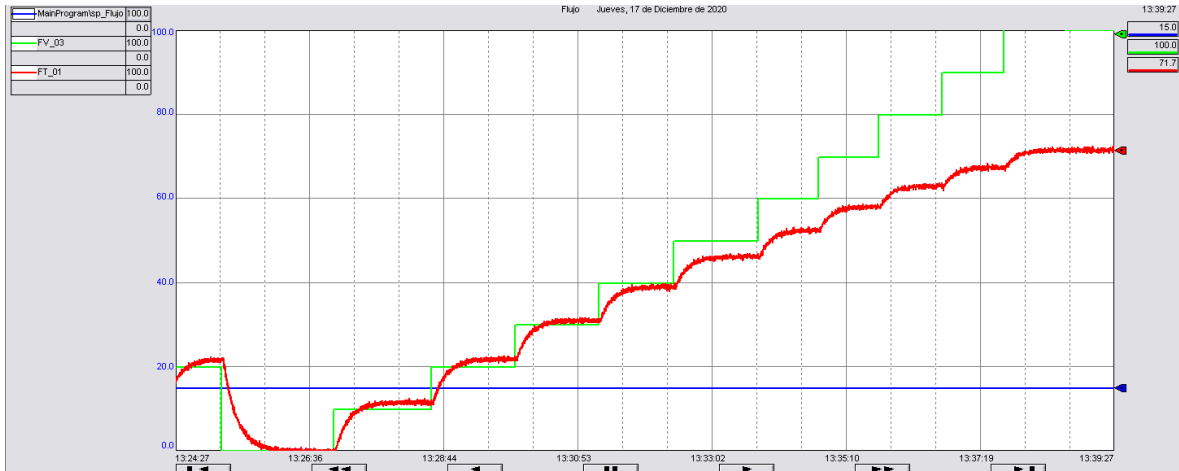


Se puede ver que está funcionando, ya que se muestran valores a través del display que corresponden a los valores actuales del controlador, lo que entrega y recibe.

Parte c) contrastación métodos

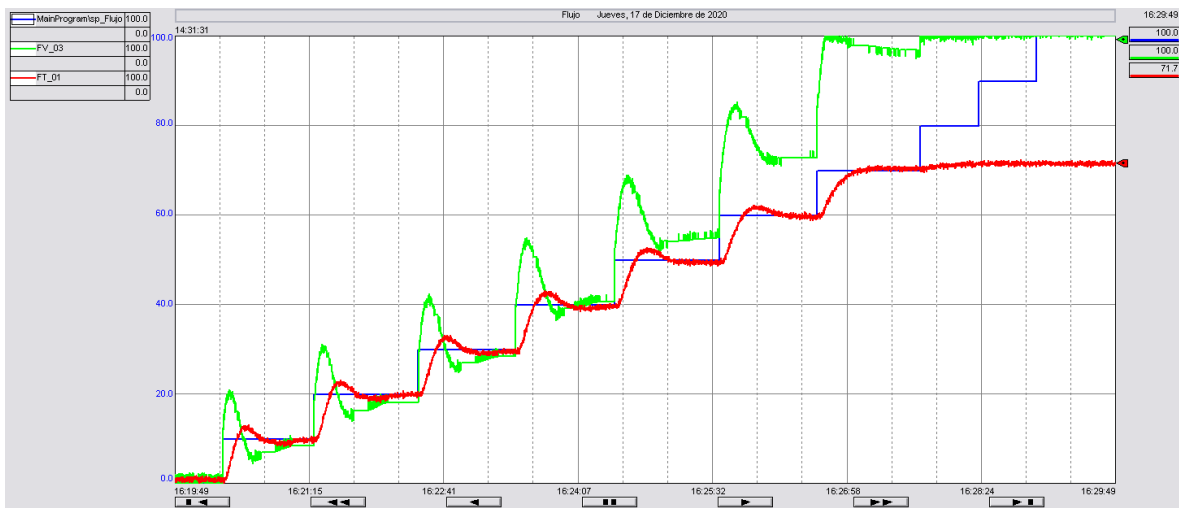
Para la contrastación de métodos, se obtuvo los siguientes gráficos, en los cuales se puede ver la aplicación de control Gain Scheduler.

Primero que nada, en la siguiente imagen se puede ver la respuesta de lazo abierto de la planta a respuestas escalo de 10% de apertura de la válvula:

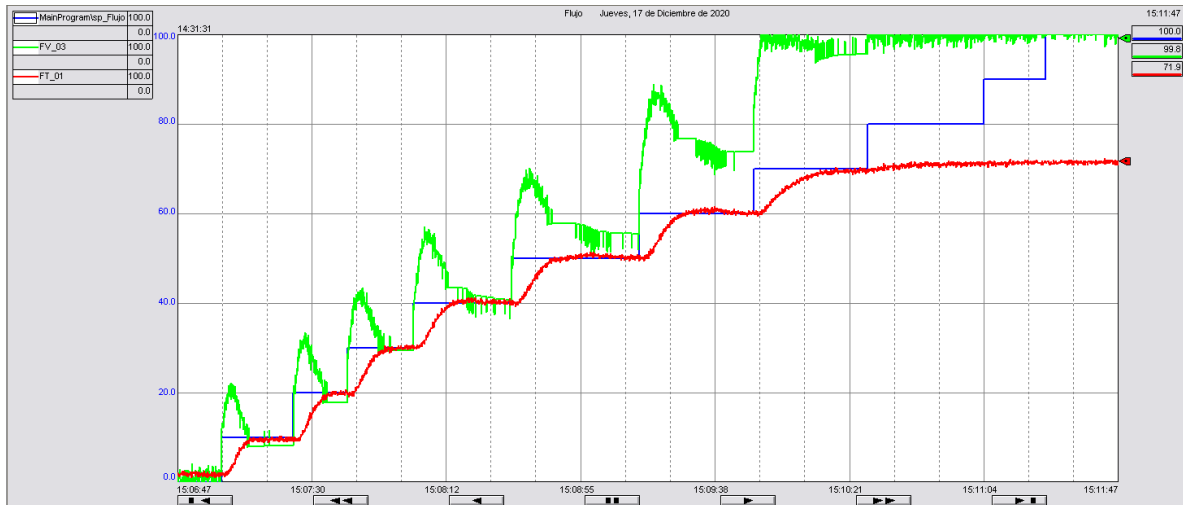


Entonces, con esto en perspectiva, se tienen los siguientes gráficos:

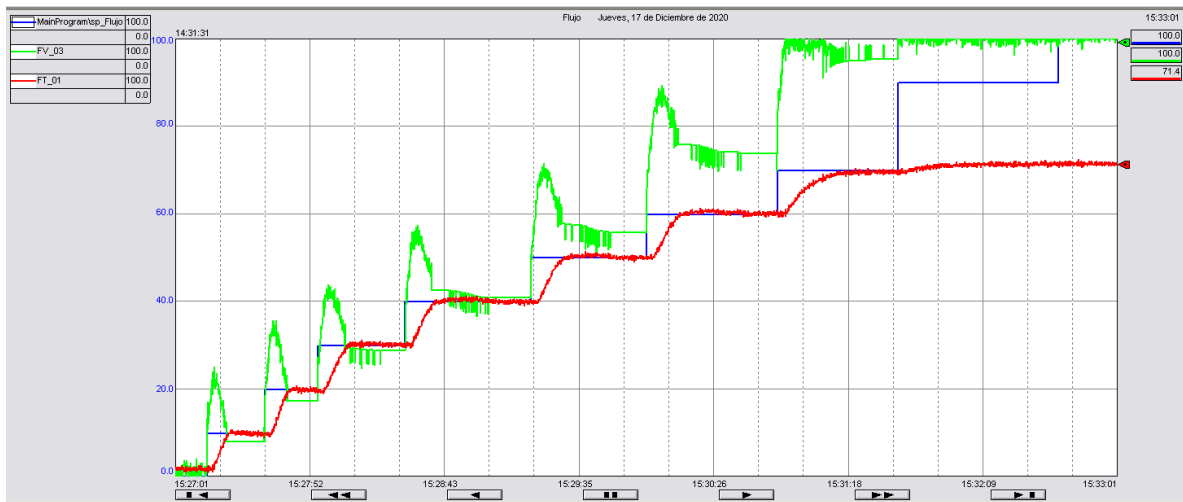
Respuesta con valores fijos de K_c y T_i (control simple):



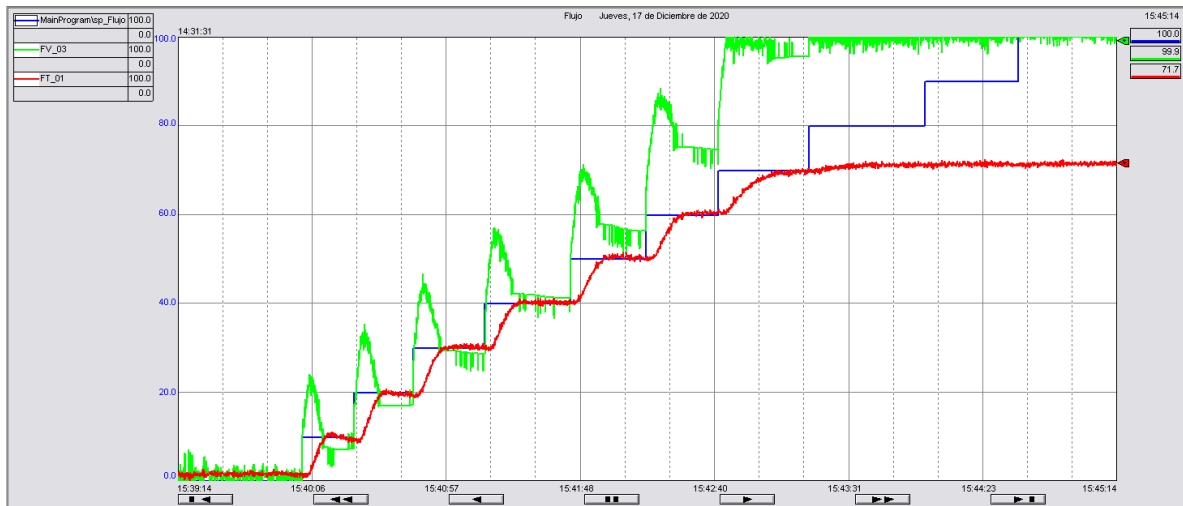
- Respuestas con Gain Scheduler:
- Respuestas con método de tabla:
- α siguiendo a FV_05:



α siguiendo a set point de flujo:

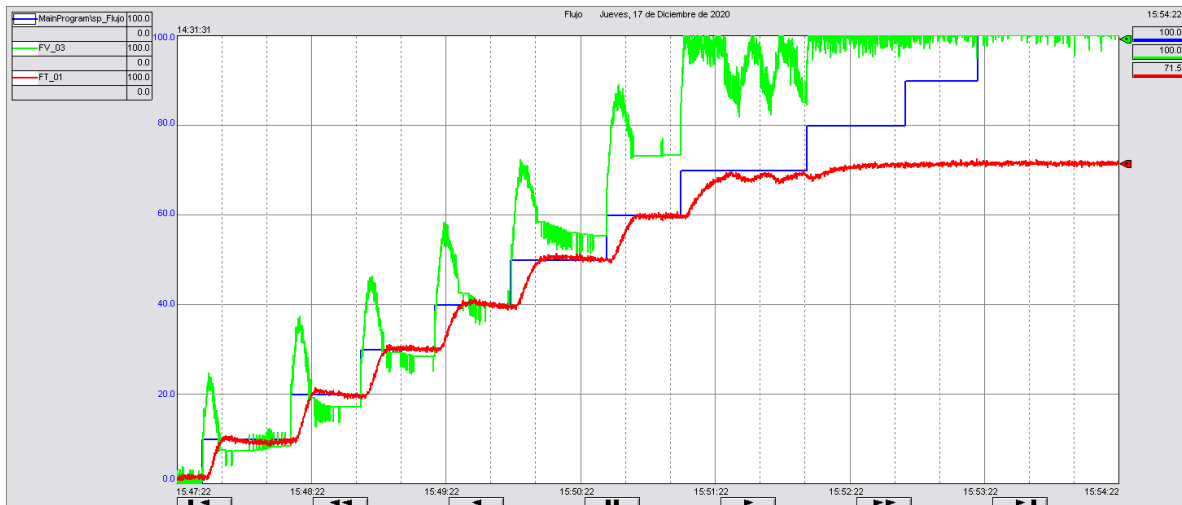


α siguiendo a FT_01:

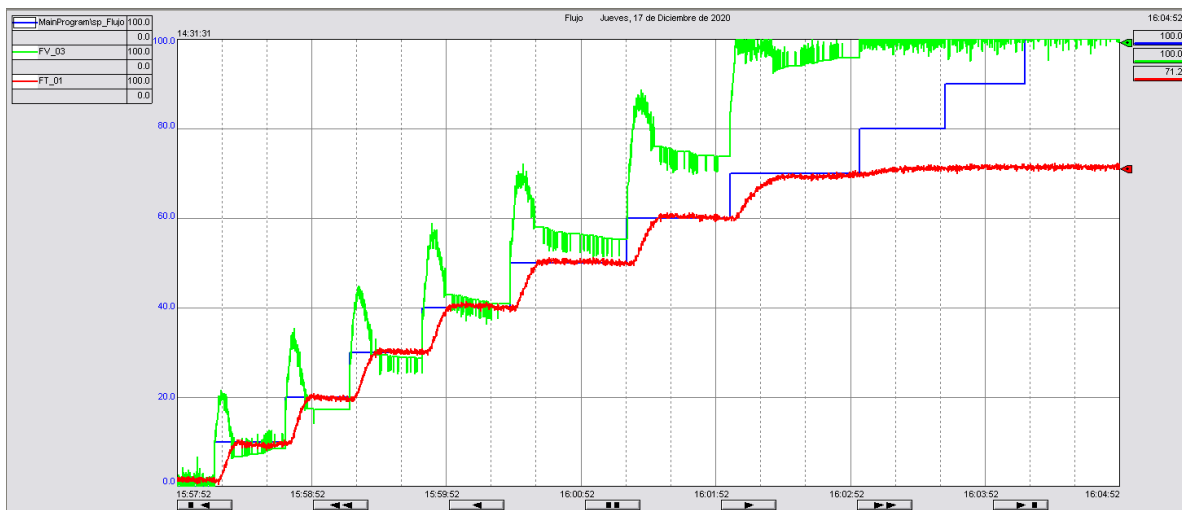


Respuestas con método de interpolación:

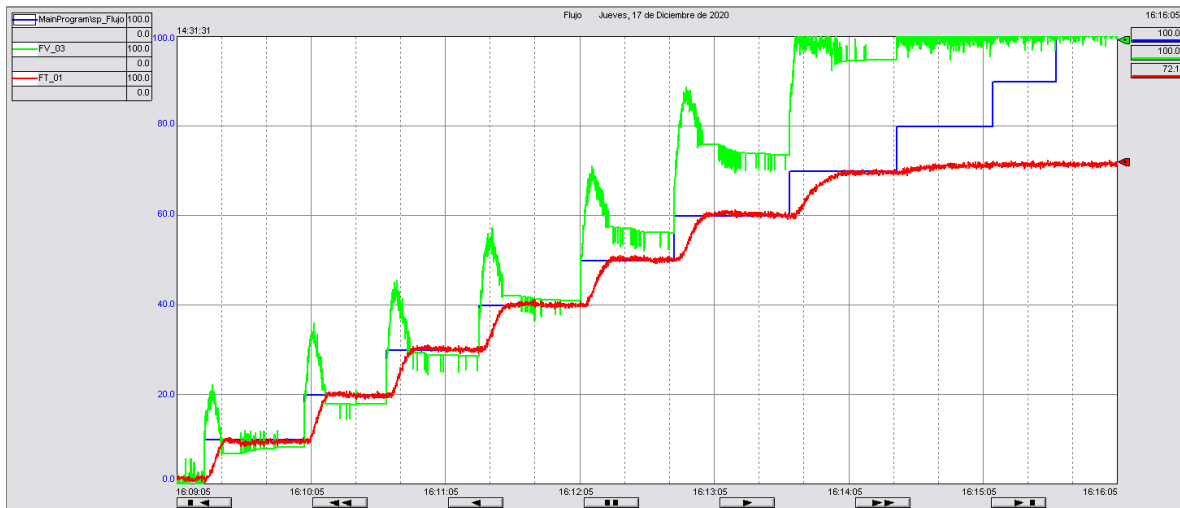
α siguiendo a FV_05:



α siguiendo a set point de flujo:



α siguiendo a FT_01:



Se puede realizar varias comparaciones entre las gráficas:

- Entre los métodos de tabla y de ganancia, la respuesta alcanza el estado estacionario mucho más pronto en el método de interpolación que en el método de tabla, aunque esto significa la válvula FV_05 se abre mucho más arriba en comparación con el método de tabla.
- Comparando entre los distintos α , no hay mucho cambio entre cual es la guía de α en el control, pero, en general, el α igual a FV_05 es mas lento, y, en el caso de usar el método de tabla, no puede llegar a ser tan estable.
- Comparándolo con el control simple, el método de Gain Scheduler es mucho mejor en poder adaptarse y poder tener una respuesta más satisfactoria.

5. Parte d) Resultados y conclusiones

Se puede llegar a distintas conclusiones:

- El uso de Gain Scheduler es mucho más ventajoso cuando la planta es no lineal, ya que, con este método, se puede linealizar por tramos la planta y, de esta manera, se puede utilizar un control que pueda manejar de mejor manera los cambios de parámetros que pueda tener la planta.
- Si se pidiera usar un α en específico, sería el de set point de flujo, ya que se parecería que fuera el más estable y rápido entre los tres α .
- También, se puede ver que, sin importar que tipo de control de usa, la respuesta de la planta se satura a llegar cercano a 71%. Esto se da por la característica no lineal de la válvula. Y, por esto, la respuesta siempre va a ser ineficiente.
- Si se pidiera elegir usar entre el método por tabla o por interpolación, se preferiría usar el método de interpolación, ya que, si se usara el método por tabla, cuando se pasara de una zona a otra, el cambio de K_c y T_i sería muy brusco, y no se podría tener una respuesta acorde a lo que se quiere tener, mientras que el método por interpolación no sucede eso, ya que siempre se esta adaptando según el α , y, por esto, trabaja mucho más optimo, pudiendo llegar a una respuesta estacionaria mucho más rápida.

Anexo A: Matlab

Anexo B: Gráficos Simulink

Anexo C: RSLogix
