

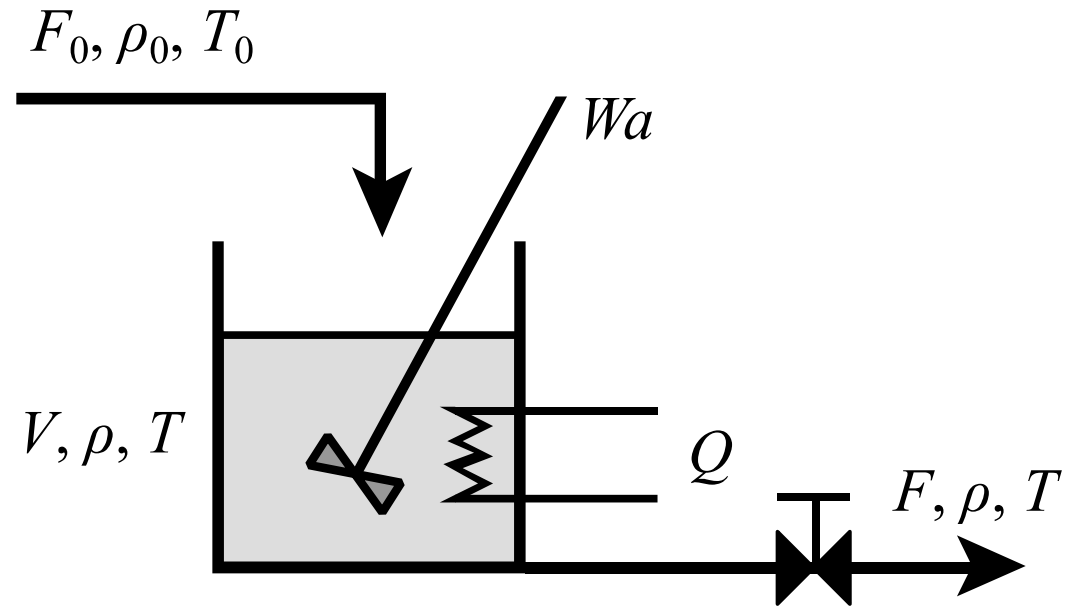
Modelado Parte V

Enrique E. Tarifa, Facultad de Ingeniería, UNJu

Tanque calefaccionado

Tanque calefaccionado

- Tanque calefaccionado
- Estado estacionario anterior
- Serpentín en la base
- Vapor saturado a 3 kgf/cm²
- $W_a = 2000 \text{ W}$, $T_0 = 25 \text{ °C}$, $T_{\text{ini}} = 60 \text{ °C}$
- $x = 0.25$, $\dot{T}$ cuando rebalse?
- Aumento de 65 % en F_0 ($x = 0.5$), $\dot{T}$ cuando rebalse?



Simulación dinámica

$$\frac{dL}{dt} = \frac{F_0 - F}{A}$$

$$\frac{dT}{dt} = \frac{F_0 \rho C_p (T_0 - T) + Q + W_a}{A L \rho C_p}$$

$$F = C_v x \sqrt{\rho g L}$$

$$Q = U A_s (T_v - T)$$

Verificar estado estacionario inicial.

Ver Tanque calefaccionado.mmd

```
{Tanque calefaccionado}
```

```
METHOD RK4
```

```
STARTTIME = 0
```

```
STOPTIME = 1100
```

```
DT = 10
```

```
; Inicialización
```

```
INIT L = 1
```

```
INIT T = 60
```

```
; Sistema ODEs
```

```
L' = (F0-F)/A
```

```
T' = (F0*rho*Cp*(T0-T)+Q+Wa)/(A*L*rho*Cp)
```

```
; Sistema AEs
```

```
F = Cv*x*sqrt(rho*g*L)
```

```
Q = UAs*(Tv-T)
```

```
; Datos
```

```
F0 = 2E-3
```

```
A = 0.785
```

```
Cv = 4.039E-5
```

```
rho = 1000
```

```
g = 9.81
```

```
x = 0.5
```

```
Cp = 4.187E3
```

```
UAs = 4.04E3
```

```
T0 = 25
```

```
Tv = 132
```

```
Wa = 2000
```

tanque_calefaccionado.m

```
% Tanque calefaccionado
% En X están las variables de estado.
% En Y deben ir las variables que se requieren en
las ODEs o que se quieren graficar.

clear all; close all; clc;

%===== Modelo =====
```

```

% ODEs
function dX = ODEs(t,X)
    % En dX devuelve el vector columna de derivadas

    % Recupera variables X
    [L T] = num2cell(X') {1, :};

    % Recupera variables Y
    Y = AEs(t,X);
    [A F0 F rho Cp T0 Q Wa] = num2cell(Y) {1, :};

    % Ecuaciones diferenciales
    dL = (F0 - F)/A;
    dT = (F0*rho*Cp*(T0-T)+Q+Wa) / (A*L*rho*Cp);

    dX = [dL dT]'; % vector columna
endfunction % ODEs

```

```

% AEs
function Y = AEs(t,X)
    % En Y devuelve el vector fila de variables requeridas por ODEs o a graficar.

    % Recupera variables X
    [L T] = num2cell(X') {1, :};

    % Parámetros
    F0 = 2E-3; A = 0.785; Cv = 4.039E-5; rho = 1000; g = 9.81;
    Cp = 4.187E3; UAs = 4.04E3; T0 = 25; Tv = 132; Wa = 2000; % Sistema mks

    % Ecuaciones algebraicas
    if t < 0
        x = 0.5;
    else
        x = 0.25;
    endif

    F = Cv*x*sqrt(rho*g*L);
    Q = UAs*(Tv-T);

    Y = [A F0 F rho Cp T0 Q Wa];
endfunction % AEs

```

```
% Inicialización
function [tfin dt Xini LX LY] = inicializacion
    % Inicializa la simulación

    % Parámetros de simulación
    tfin = 1100; % tiempo final
    dt = 10; % paso temporal

    % Inicialización
    Lini = 1; % m
    Tini = 60; % °C
    Xini = [Lini Tini]; % Inicializa las variables de estado

    % Leyendas
    LX = {'L' 'T'}; % Leyendas de las variables X
    LY = {'A' 'F0' 'F' 'rho' 'Cp' 'T0' 'Q' 'Wa'}; % Leyendas de las variables Y
endfunction % inicializar
```



```

% Análisis
function analizar(LX,LY,tpts,X,Y)
    % Análisis de resultados.
    exportar('resultados.csv');

    graficar({'L'}, 'Nivel vs. tiempo', 's', 'm', [0 3]);
    graficar({'F0' 'F'}, 'Caudales vs. tiempo', 's', 'm^3/s', [0 4E-3]);
    graficar({'T'}, 'Temperatura vs. tiempo', 's', '°C', [0 120]);

    % Control de rebalse
    Lmax = 2; %m Altura del tanque
    Lt = vector('L'); % Recupera el vector de niveles.
    if Lt(end) <= Lmax % Verifica el último nivel porque es el mayor.
        disp('El tanque no rebalsó.');
```

```

    else
        tr = interp1(Lt, tpts, Lmax); % Se puede usar interp1 porque Lt es creciente.
        disp(['El tanque rebalsó en el tiempo igual a ' num2str(tr) ' s.']);

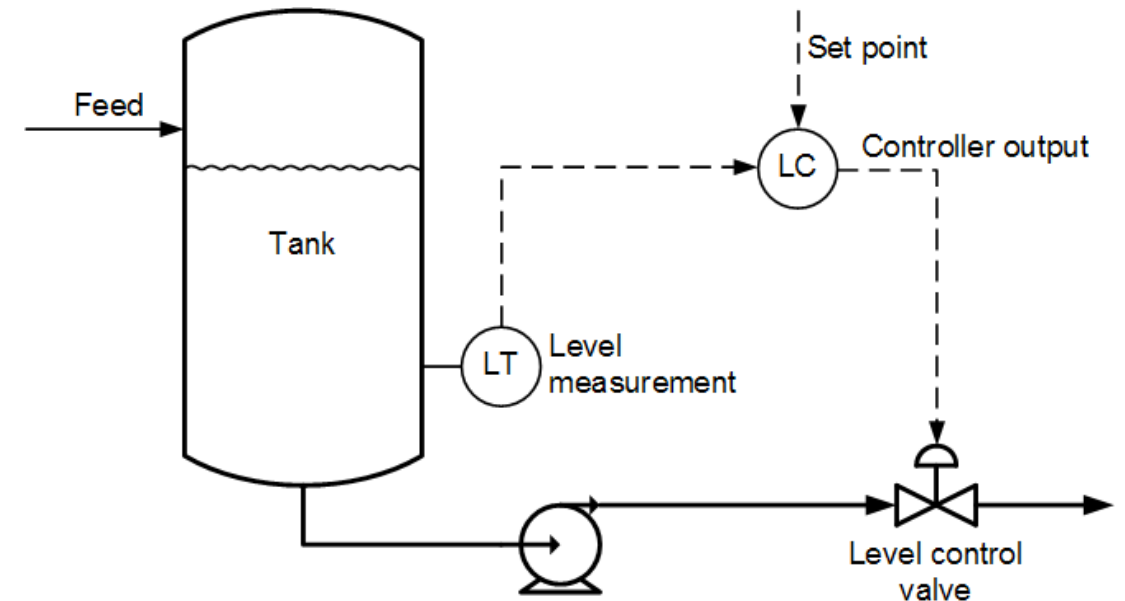
        Tt = vector('T'); % Recupera el vector de temperaturas.
        Tr = interp1(tpts, Tt, tr); % Interpola la temperatura para tr.
        disp(['La temperatura del tanque en ese tiempo fue ' num2str(Tr) ' °C.']);
    endif
endfunction % analizar

```

Controlador de nivel

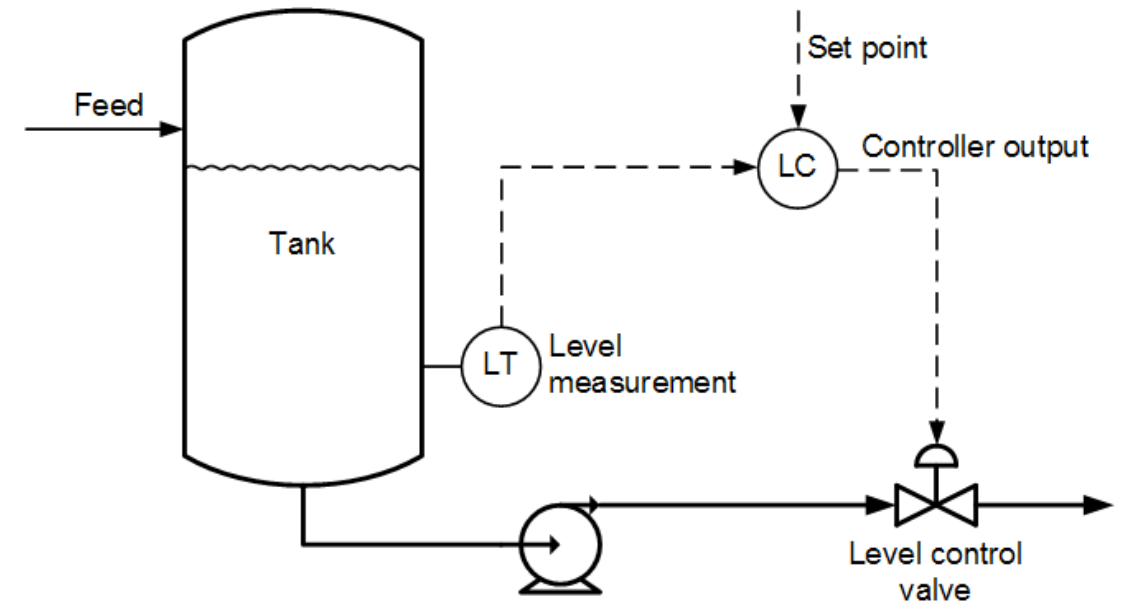
Selección de acción en modo servo

1. Acción inversa y válvula NC
2. Setpoint $L_{sp}(+)$
3. $L_{sp}(+) \rightarrow e(+) \rightarrow Ac(+) \rightarrow x(+) \rightarrow F(+) \rightarrow L(-)$
4. Si $L(+)$, aceptar la acción.



Selección de acción en modo regulador

1. Acción inversa y válvula NC
2. $L(+)$
3. $L(+) \rightarrow e(-) \rightarrow Ac(-) \rightarrow x(-) \rightarrow F(-) \rightarrow L(+)$
4. Si $L(-)$, aceptar la acción.



Prompt para determinar el tipo de acción

Actúa como un experto en Ingeniería Química. Determina el tipo de acción, directa o inversa, que deberá tener un controlador de nivel de un tanque cuando la válvula de control es del tipo normalmente cerrada y está instalada en la corriente de descarga.

Actúa como un experto en Ingeniería Química. Determina el tipo de acción, directa o inversa, que deberá tener un controlador de nivel de un tanque cuando la válvula de control es del tipo normalmente cerrada y está instalada en la corriente de descarga.

Resultado

Ítem	Respuesta
Tipo de acción del controlador	Inversa

Resumen lógico final

Nivel del tanque sube → queremos **abrir más la válvula** → **aumentar la señal** → error baja → **controlador debe ser inverso** para que ante error descendente la salida suba.

Convenciones

Variable usada	Condición	Acción del controlador
Error $e = SP - PV$	error $\uparrow \rightarrow$ salida $\uparrow$	Acción directa
Variable medida (PV)	PV $\uparrow \rightarrow$ salida $\uparrow$	Acción directa

Usamos la convención que considera PV.

Prompt para determinar el tipo de acción

Actúa como un experto en Ingeniería Química. Determina el tipo de acción, directa o inversa, que deberá tener un controlador de nivel de un tanque cuando la válvula de control es del tipo normalmente cerrada y está instalada en la corriente de descarga. Para definir el tipo de acción, considera la convención que analiza la relación entre la salida del controlador y la variable de proceso, no con el error.

Actúa como un experto en Ingeniería Química. Determina el tipo de acción, directa o inversa, que deberá tener un controlador de nivel de un tanque cuando la válvula de control es del tipo normalmente cerrada y está instalada en la corriente de descarga. Para definir el tipo de acción, considera la convención que analiza la relación entre la salida del controlador y la variable de proceso, no con el error.

Relación entre PV (variable de proceso) y salida del controlador

- Si PV sube (nivel sube), la salida del controlador debe aumentar para abrir más la válvula.
- Si PV baja (nivel baja), la salida del controlador debe disminuir para cerrar más la válvula.

Entonces:

- $PV \uparrow \rightarrow salida \uparrow$
- $PV \downarrow \rightarrow salida \downarrow$

Esta es una relación directa.

Conclusión

Bajo la convención basada en la relación entre PV y la salida del controlador:

Acción directa

Instalación de un control PI para el nivel

Modelo original

$$Ac = Ab + Kp \left(e + \frac{1}{\tau i} Ai \right)$$

$$e = L - L_{sp}$$

$$Ai = \int_0^t e dt$$

$$x = \begin{cases} 0 & Ac < 0 \\ 1 & Ac > 1 \\ Ac & \text{en otro caso} \end{cases}$$

Modelo simplificado

$$\frac{dAi}{dt} = e$$

$$e = L - L_{sp}$$

$$Ac = Ab + Kp \left(e + \frac{1}{\tau i} Ai \right)$$

$$x = \begin{cases} 0 & Ac < 0 \\ 1 & Ac > 1 \\ Ac & \text{en otro caso} \end{cases}$$

Simulación dinámica

Sistema ODEs

$$\frac{dL}{dt} = \frac{F_0 - F}{A}$$

$$\frac{dT}{dt} = \frac{F_0 \rho C_p (T_0 - T) + Q + W_a}{A L \rho C_p}$$

$$\frac{dA_i}{dt} = e$$

Sistema AEs

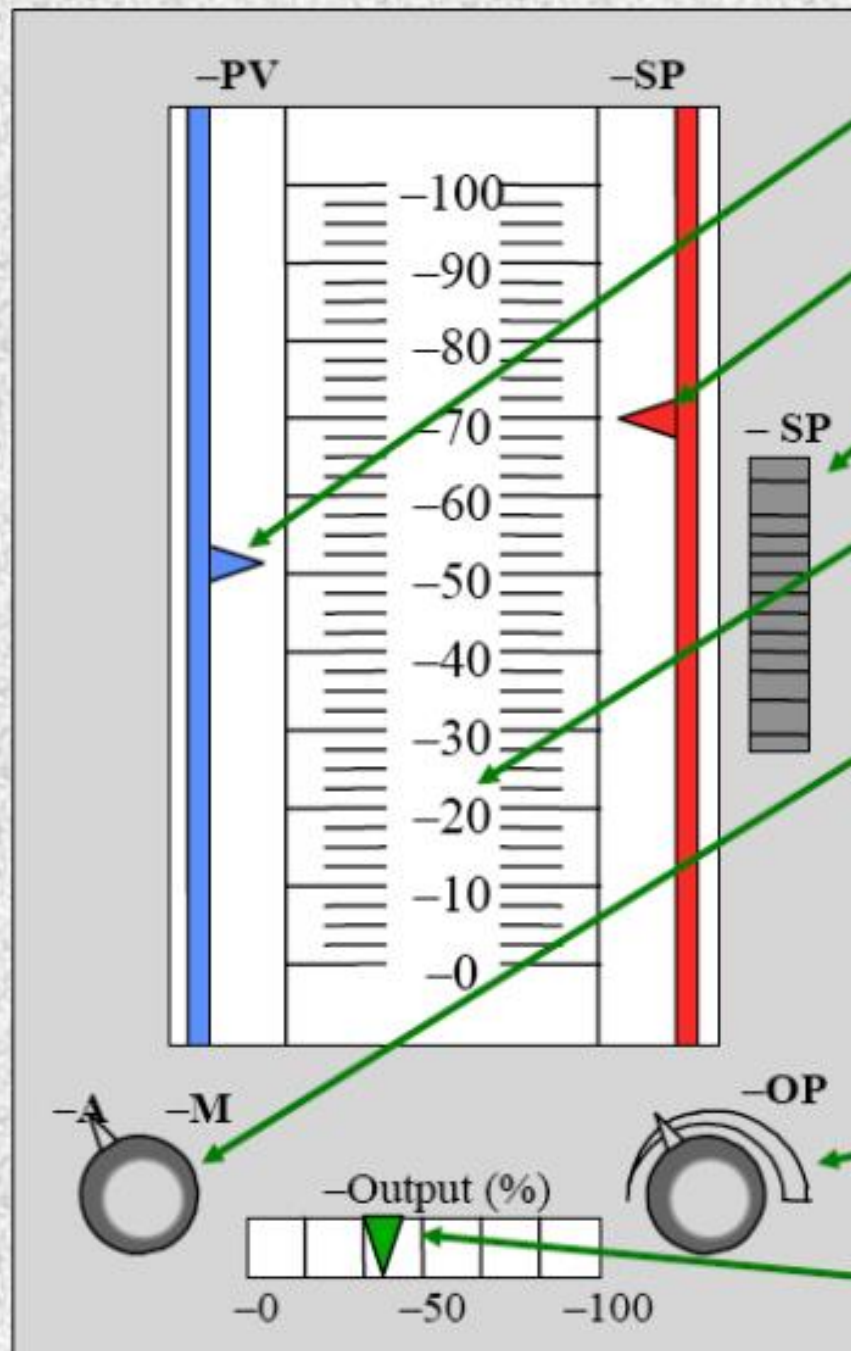
$$F = C_v \times \sqrt{\rho g L}$$

$$Q = U A_s (T_v - T)$$

$$e = L - L_{sp}$$

$$A_c = A_b + K_p \left(e + \frac{1}{\tau_i} A_i \right)$$

$$x = \begin{cases} 0 & A_c < 0 \\ 1 & A_c > 1 \\ A_c & \text{en otro caso} \end{cases}$$



Indicador valor de la variable controlada

Indicador del valor del Set Point

Modificador del Set Point

Escala porcentual para Set point y variable controlada

Conmutador AUTOMÁTICO/MANUAL
– **AUTOMÁTICO**: el controlador decide el valor de la variable manipulada
– **MANUAL**: el valor de la variable manipulada se fija de forma manual

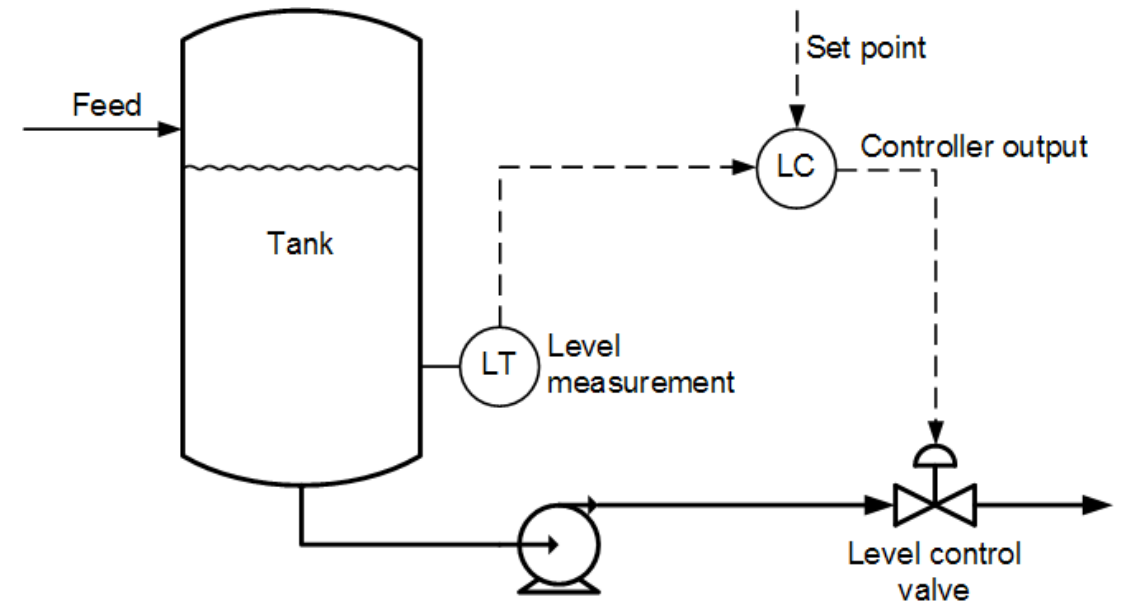
Modificador de la variable manipulada

Indicador de la variable manipulada

Reset manual en un controlador real

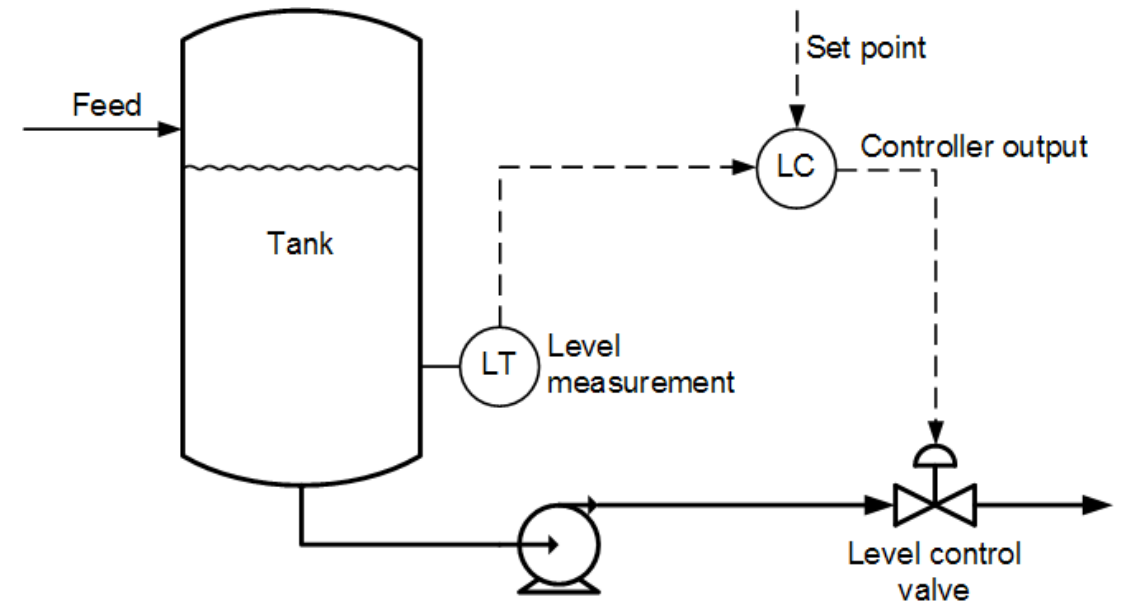
1. Colocar el controlador en manual.
2. Elegir un punto de operación L_{sp} .
3. Variar A_c manualmente para lograr que $L = L_{sp}$.
4. Hacer $A_b \leftarrow A_c$.

El *reset* automático hace esto solo.



Reset manual en un controlador P

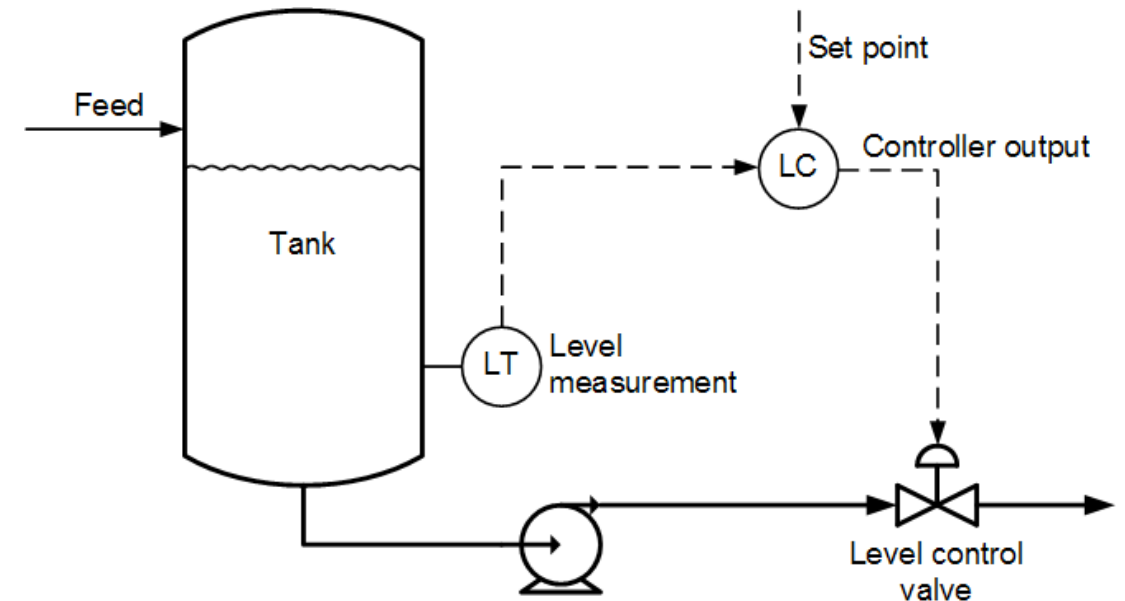
1. Elegir un punto de operación L_{sp} .
2. Variar Ab manualmente para lograr que $L = L_{sp}$.



Ese punto de operación será el único libre de *offset*.

Reset manual en un controlador PI o PID

1. Hacer $Ab = 0$.
2. Hacer $Ai(0) = 0$.
3. Elegir un punto de operación L_{sp} .
4. Esperar hasta que $L = L_{sp}$.
5. Hacer $Ab \leftarrow Ac$.



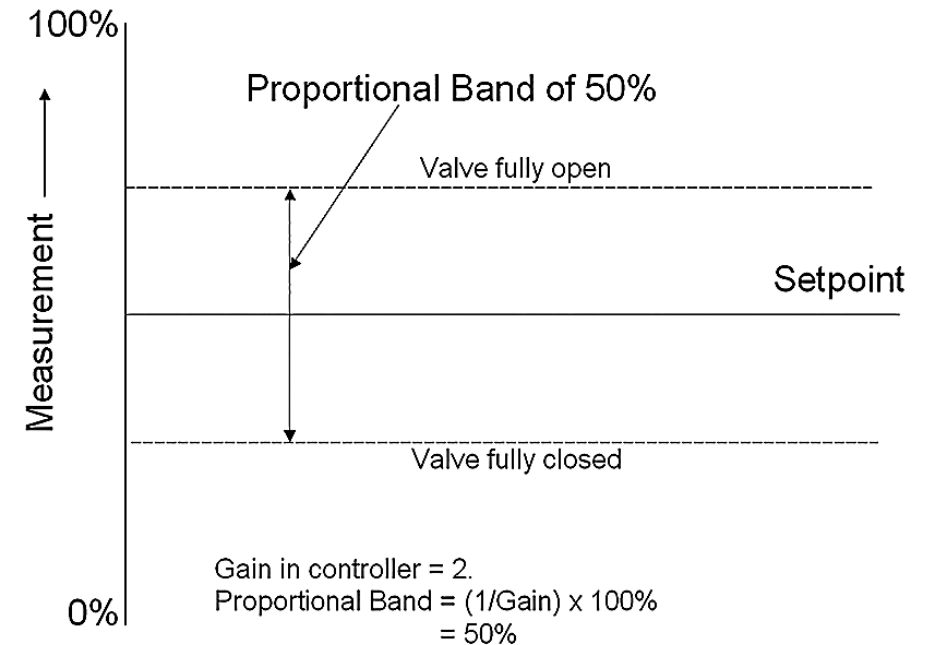
[Simulador online control PID](#)

Reset manual

- $Ab = 0.5$ valor ideal
- Para una válvula NC:
 - Si la válvula es grande, C_v grande, $Ab < 0.5$.
 - Si la válvula es chica, C_v chico, $Ab > 0.5$.
- Para una válvula NA:
 - Si la válvula es grande, C_v grande, $Ab > 0.5$.
 - Si la válvula es chica, C_v chico, $Ab < 0.5$.

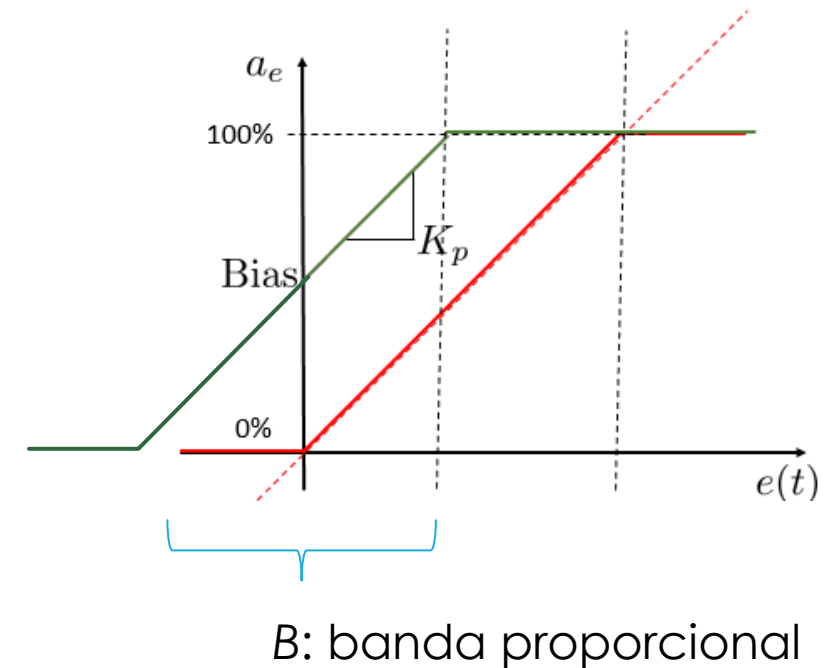
Banda proporcional

- Valores ideales:
 - $Ab = 0.5$
 - $Ai(0) = 0$
- Pb : Banda proporcional en % (la base es el rango de medición)
- B : Banda proporcional con unidades.
- $B = Pb \ y_{sp}/100$
- $Kp = 1/B$



Valores iniciales

- P_b define una “zona” alrededor del *setpoint*.
- En ella, el controlador modula suavemente su salida de mínimo a máximo.
- Fuera de ella, el controlador satura (salida 0 % o 100 %).



Selección de acciones

- Flujo: P o PI. No D por el ruido.

- $Kp = 0.4 - 0.65$

- $\tau i = 6 \text{ s}$

- Nivel: P o PI. No D por el ruido.

- $Kp = 2 - 20$

- $\tau i = 1 - 5 \text{ min}$

- Presión:

- $Kp = 0.5 - 2 \text{ (líq.)}, 2 - 10 \text{ (gas)}$

- $\tau i = 6 - 15 \text{ s (líq.)}, 2 - 10 \text{ min (gas)}$

- Temperatura:

- $Kp = 2 - 10$

- $\tau i = 2 - 10 \text{ min}$

- $\tau d = 0 - 5 \text{ min}$

Sintonía de un controlador PID

Métodos

- Heurísticos
- Basados en reglas:
 - Ziegler-Nichols, Chien, Hrones y Reswick, Cohen-Coon, Kappa-tau y Lambda.
- Basados en modelos:
 - Modelo
 - Especificaciones

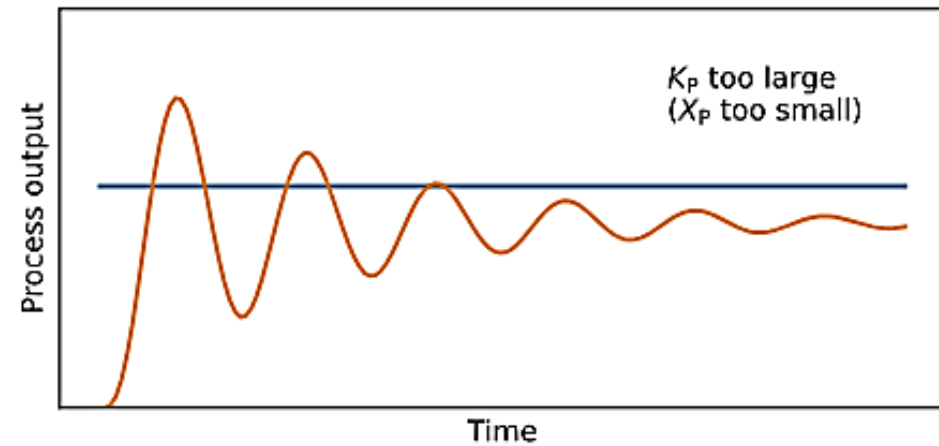
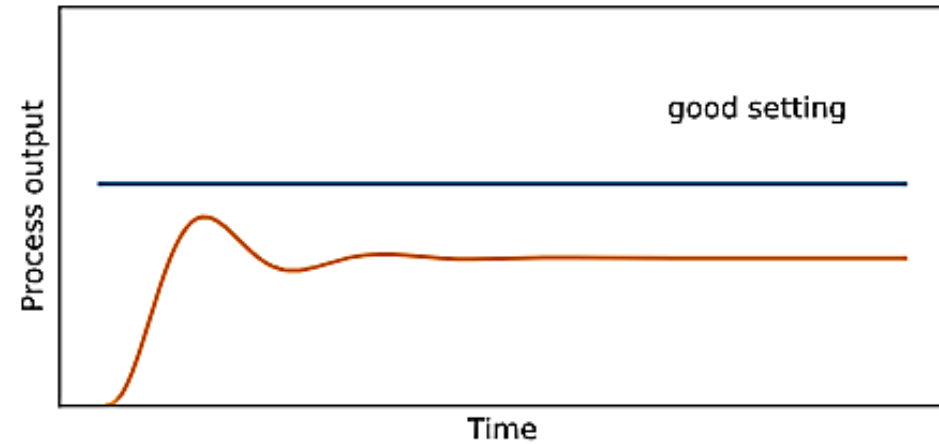
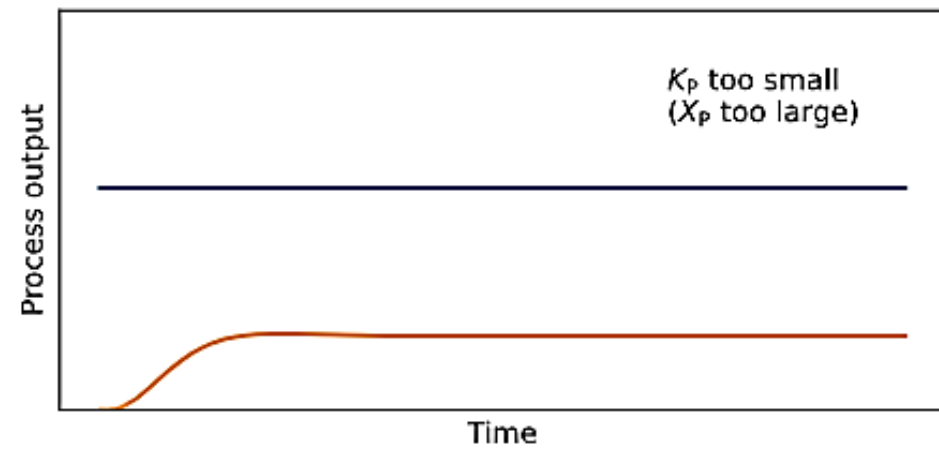
Heurístico

- Seleccionar el tipo de acción.
- Ajustar A_b .
- Anular P, I y D.
- Provocar un escalón en y_{sp} .
- Aumentar P hasta que oscile.
- Aumentar I hasta que oscile.
- Aumentar D hasta que oscile.

Sintonía de un controlador PID

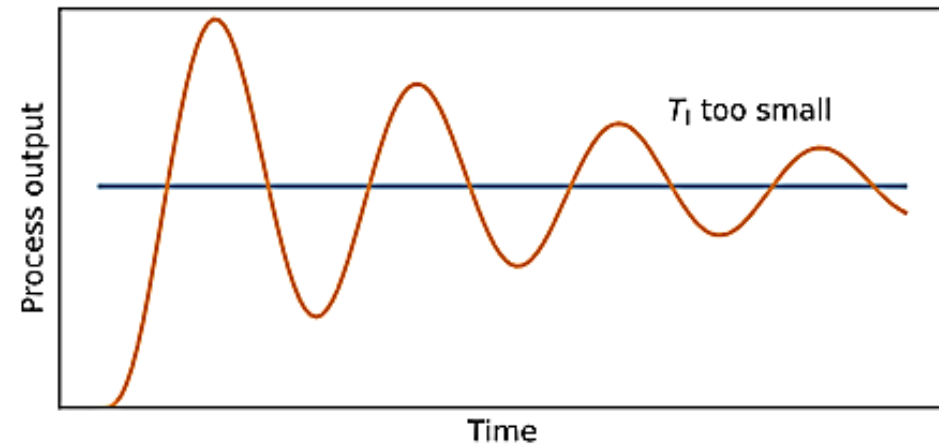
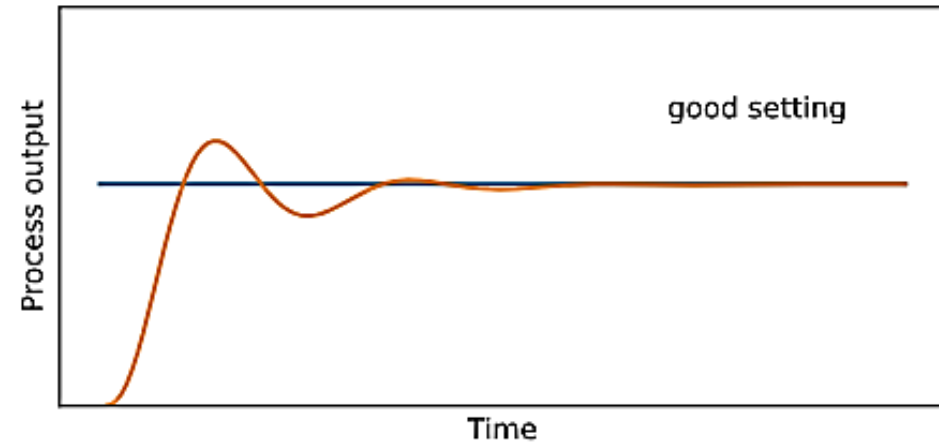
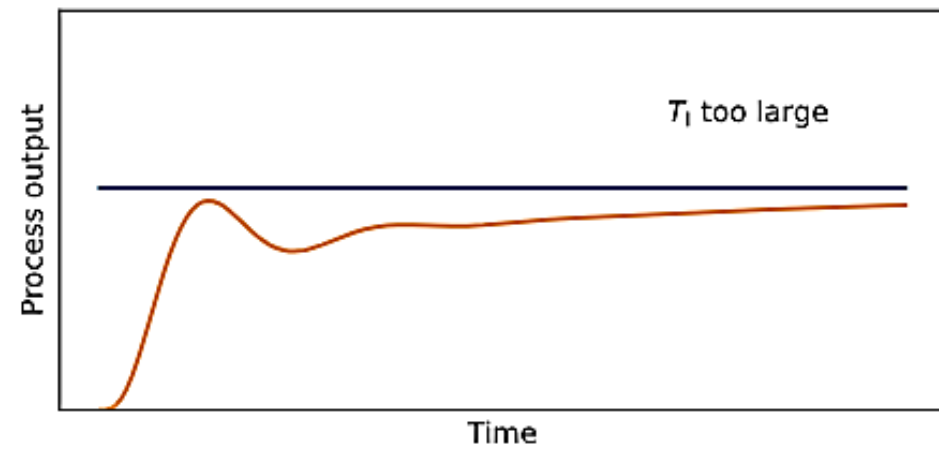
Ajuste de K_p

Sintonía de un controlador PID



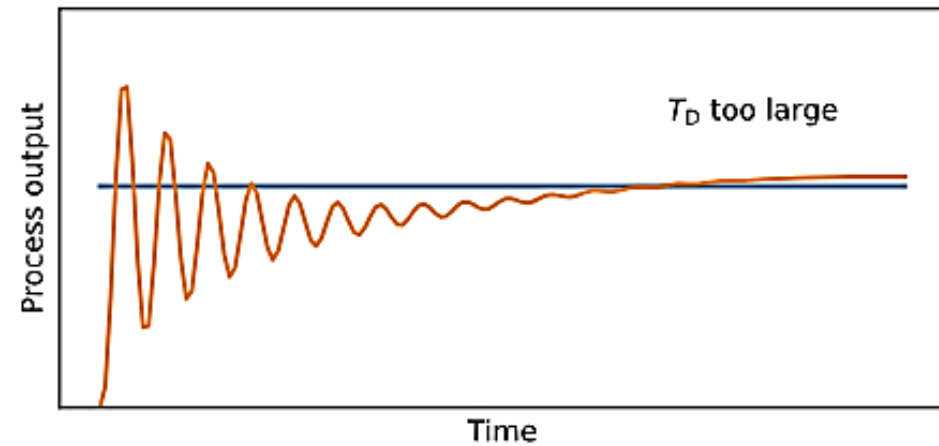
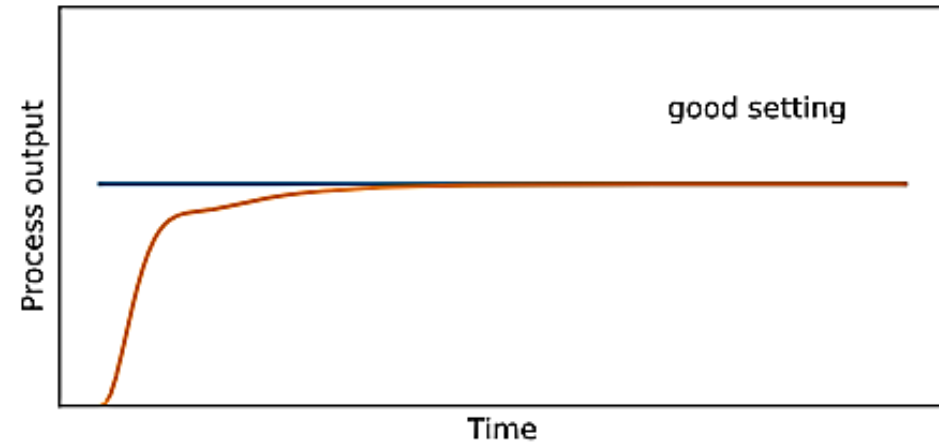
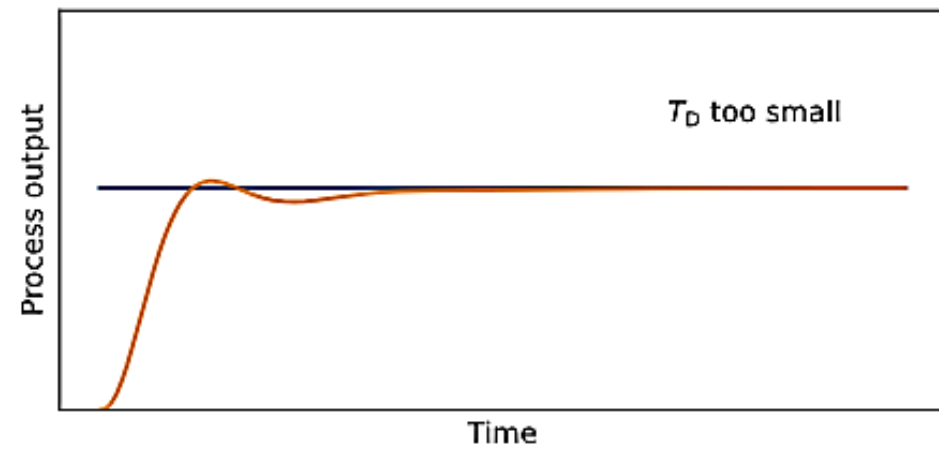
Ajuste de τ_i

Sintonía de un controlador PID



Ajuste de τ_d

Sintonía de un controlador PID



Reset manual en un controlador P

- $K_p = 2 \text{ m}^{-1}$
- $\tau_i = 100000 \text{ s}$
- $A_i(0) = 0$
- $L_{sp} = 1 \text{ m}$
- Probar valores de Ab comenzando con 0.5.

Listado en Berkeley Madonna

```
{Tanque calefaccionado con CL}

METHOD RK4

STARTTIME = 0
STOPTIME = 2000
DT = 10

; Inicialización
INIT L = 1
INIT T = 60
INIT Ai = 0

; Sistema ODEs
L' = (F0-F)/A
T' = (F0*rho*Cp*(T0-T)+Q+Wa)/(A*L*rho*Cp)
Ai' = e
```

```
; Sistema AEs
e = L-Lsp
Ac = Ab+Kp*(e+Ai/taui)
x = Ac
LIMIT x >= 0
LIMIT x <= 1

F = Cv*x*sqrt(rho*g*L)
Q = UAs*(Tv-T)
```

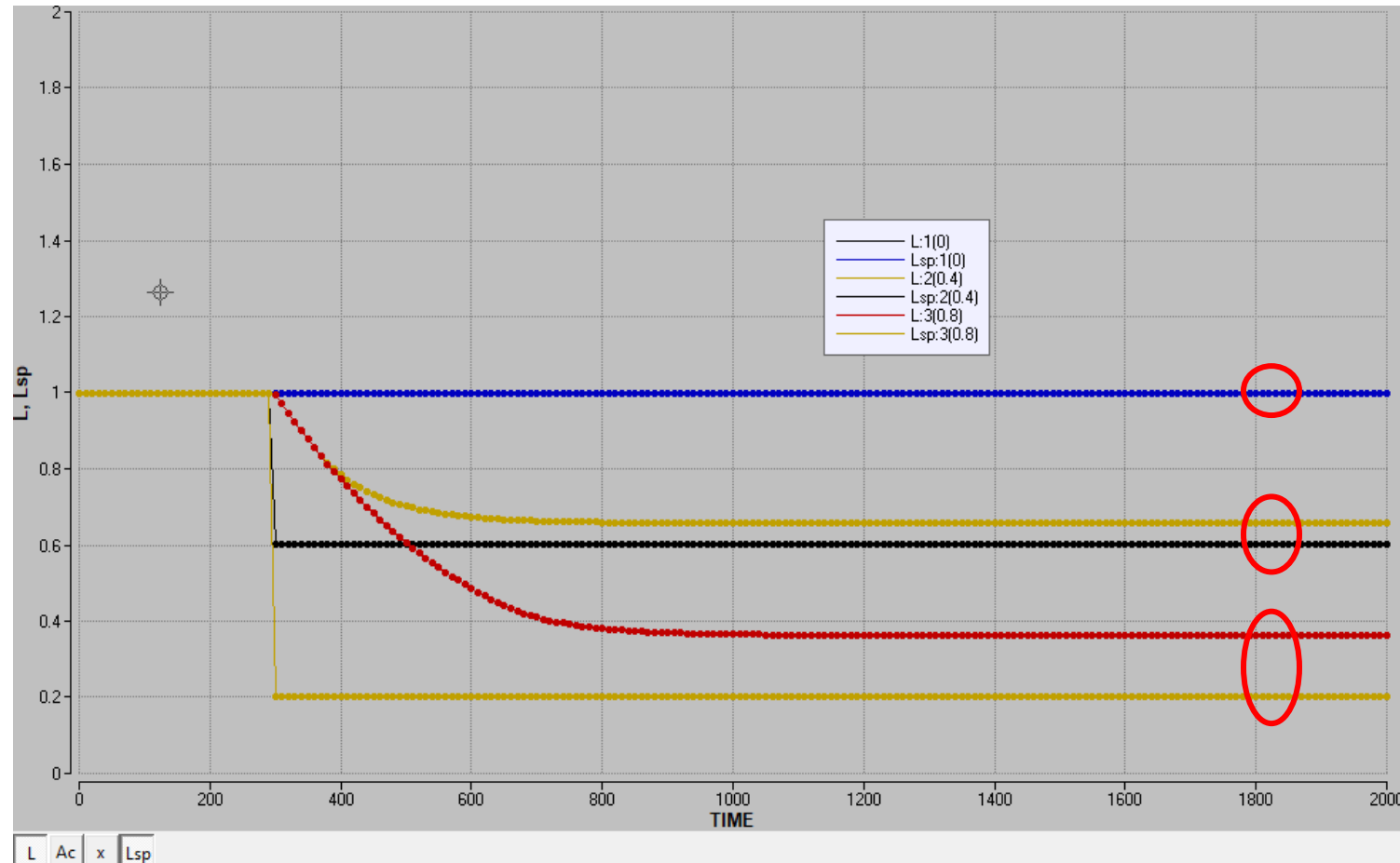
```
; Datos
F0 = 2E-3
A = 0.785
Cv = 4.039E-5
rho = 1000
g = 9.81

Cp = 4.187E3
UAs = 4.04E3
T0 = 25
Tv = 132
Wa = 2000

Ab = 0.5
Kp = 2
taui = 100000
Lsp = 1
```

Ver Tanque calefaccionado con CL.mmd

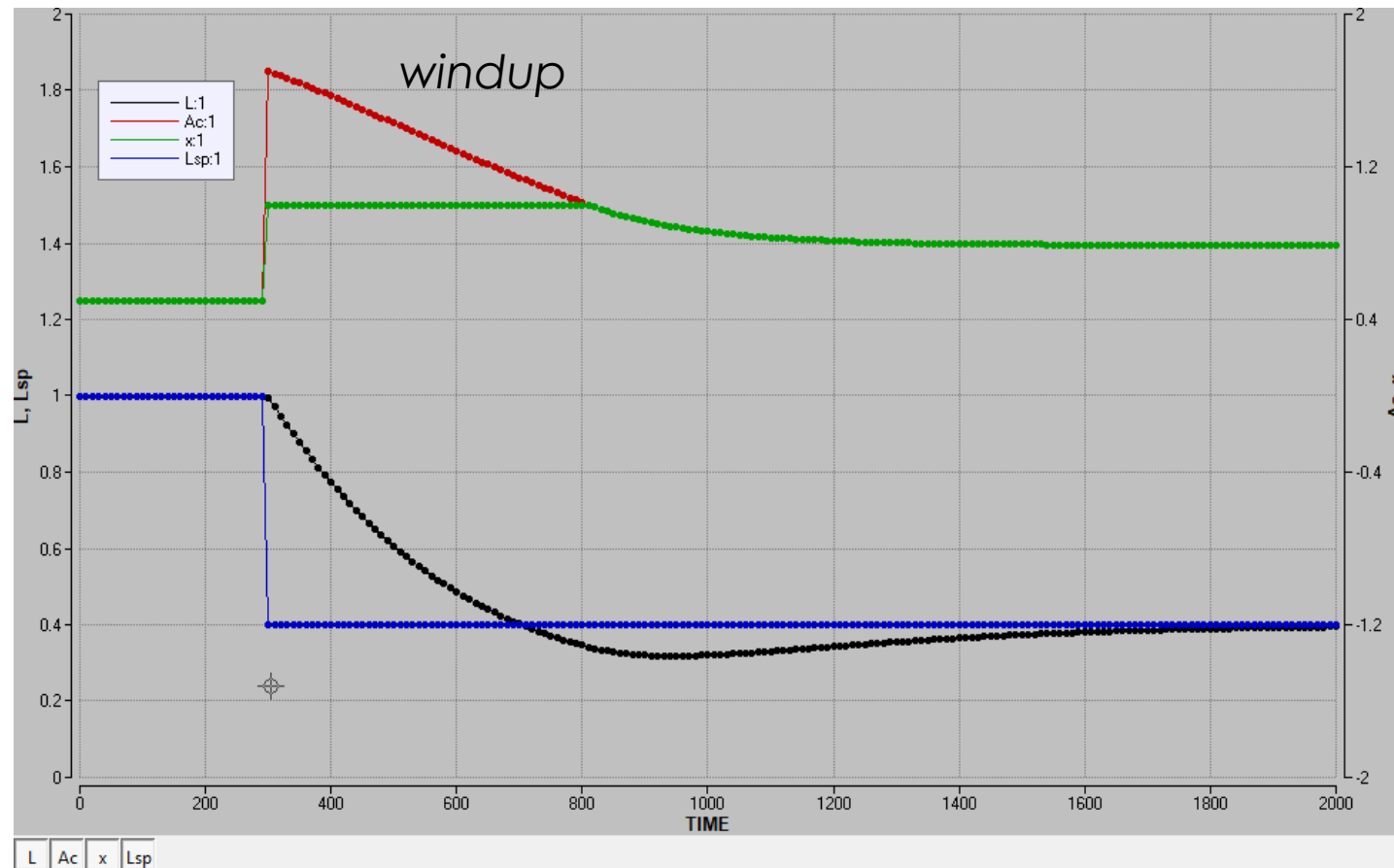
Offset variable ante escalones en L_{sp}



Sintonización del controlador PI

- $K_p = 2 \text{ m}^{-1}$
- $\tau_i = 300 \text{ s}$
- $A_i = 0$
- $L_{sp} = 1 \text{ m}$
- Verificar estado estacionario inicial.
- $A_i = 0$ valor ideal
 - Si el Ab es bajo por error, $A_i > 0$.
 - Si el Ab es alto por error, $A_i < 0$.

Saturación del CL ante un escalón en L_{sp}



tanque_calefaccionado_CL.m

```
% Tanque calefaccionado con CL
% En X están las variables de estado.
% En Y deben ir las variables que se requieren en
las ODEs o que se quieren graficar.

clear all; close all; clc;

%===== Modelo =====
```

```

% ODEs
function dX = ODEs(t,X)
    % En dX devuelve el vector columna de derivadas

    % Recupera variables X
    [L T Ai] = num2cell(X') {1, :};

    % Recupera variables Y
    Y = AEs(t,X);
    [A F0 F rho Cp T0 Q Wa e Lsp] = num2cell(Y) {1, :};

    % Ecuaciones diferenciales
    dL = (F0 - F)/A;
    dT = (F0*rho*Cp*(T0-T)+Q+Wa) / (A*L*rho*Cp);
    dAi = e;

    dX = [dL dT dAi]'; % vector columna
endfunction % ODEs

```

```
% AEs
function Y = AEs(t,X)
    % En Y devuelve el vector fila de variables requeridas por ODEs o a graficar.

    % Recupera variables X
    [L T Ai] = num2cell(X') {1,:};

    % Parámetros
    F0 = 2E-3; A = 0.785; Cv = 4.039E-5; rho = 1000; g = 9.81;
    Cp = 4.187E3; UAs = 4.04E3; T0 = 25; Tv = 132; Wa = 2000;
    Ab = 0.5; Kp = 2; tau1 = 300; Lsp = 1; % Sistema mks

    % Ecuaciones algebraicas
    if t > 300
        Lsp = 0.4;
    endif
```

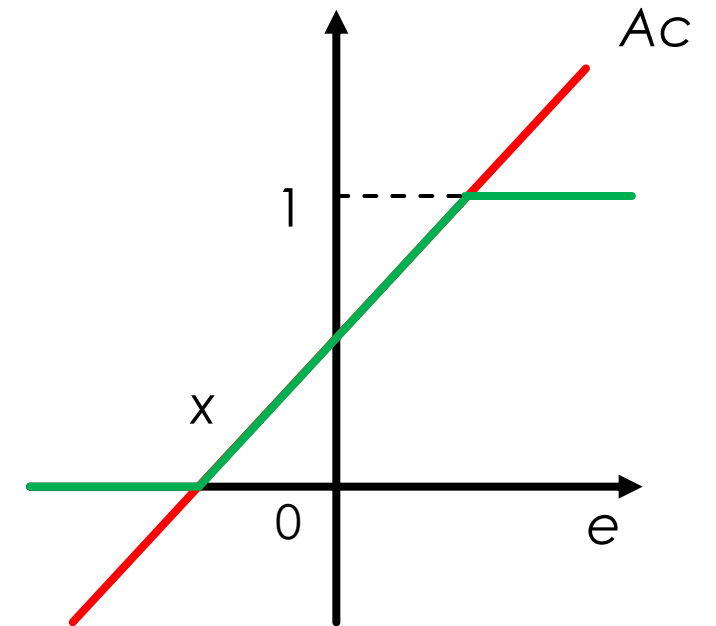
```

% Controlador
e = L-Lsp;
Ac = Ab+Kp*(e+Ai/taui);
x = max(0,Ac); % Acota x entre 0 y 1
x = min(x,1);

F = Cv*x*sqrt(rho*g*L);
Q = UAs*(Tv-T);

Y = [A F0 F rho Cp T0 Q Wa e Lsp];
endfunction % AEs

```




```
% Inicialización
function [tfin dt Xini LX LY] = inicializacion
    % Inicializa la simulación

    % Parámetros de simulación
    tfin = 2000; % tiempo final
    dt = 10; % paso temporal

    % Inicialización
    Lini = 1; % m
    Tini = 60; % °C
    Aiini = 0;
    Xini = [Lini Tini Aiini]; % Inicializa las variables de estado

    % Leyendas
    LX = {'L' 'T' 'Ai'}; % Leyendas de las variables X
    LY = {'A' 'F0' 'F' 'rho' 'Cp' 'T0' 'Q' 'Wa' 'e' 'Lsp'}; % Leyendas de las variables Y
endfunction % inicializar
```

```

% Análisis
function analizar(LX,LY,tpts,X,Y)
    % Análisis de resultados. Funciones disponibles:
    % exportar('resultados.csv',[{leyendas}])
    % graficar({leyendas}, 'título', 'rótulo x', 'rótulo y', [limitesy])
    % vector(leyenda)
    exportar('resultados.csv', {'L' 'Lsp'});

    graficar({'L' 'Lsp'}, 'Nivel vs. tiempo', 's', 'm', [0 3]);
    graficar({'F0' 'F'}, 'Caudales vs. tiempo', 's', 'm^3/s', [0 4E-3]);
    graficar({'T'}, 'Temperatura vs. tiempo', 's', '°C', [0 120]);

    % Determina el valle de L y el tiempo correspondiente.
    [min_valor, indice] = min(vector('L')); % Encuentra el valor máximo y su índice.
    disp('Datos del valle de nivel');
    disp(['El valor mínimo del nivel es ' num2str(min_valor) ' m.']);
    dt = tpts(2); % Es el delta t.
    disp(['Ese valor se alcanza en ' num2str(tpts(indice)) ' ± ' num2str(dt) ' s.']);
endfunction % analizar

```

```

% Análisis
function analizar(LX,LY,tpts,X,Y)
    % Análisis de resultados. Funciones disponibles:
    % exportar('resultados.csv',[{leyendas}])
    % graficar({leyendas}, 'título', 'rótulo x', 'rótulo y', [limitesy])
    % vector(leyenda)
    exportar('resultados.csv', {'L' 'Lsp'});

    graficar({'L' 'Lsp'}, 'Nivel vs. tiempo', 's', 'm', [0 3]);
    graficar({'F0' 'F'}, 'Caudales vs. tiempo', 's', 'm^3/s', [0 4E-3]);
    graficar({'T'}, 'Temperatura vs. tiempo', 's', '°C', [0 120]);

    % Determina el valle de L y el tiempo correspondiente.
    [min_valor, indice] = min(vector('L')); % Encuentra el valor máximo y su índice.
    disp('Datos del valle de nivel');
    disp(['El valor mínimo del nivel es ' num2str(min_valor) ' m.']);
    dt = tpts(2); % Es el delta t.
    disp(['Ese valor se alcanza en ' num2str(tpts(indice)) ' ± ' num2str(dt) ' s.']);
endfunction % analizar

```

Escalón en Lsp de 1 a 0.4 m

Resolvedor v01, 2025

Resolviendo el modelo...

Archivo exportado como "resultados.csv" en el directorio de trabajo.

Datos del valle de nivel

El valor mínimo del nivel es 0.31792 m.

Ese valor se alcanza en 950 ± 10 s.

Simulación finalizada.

>>

Punto mínimo

