

# Comprensión "*intuitiva*" del comportamiento de un sistema dinámico lineal: cálculo de tren de escalones para comportamiento deseado, NO funciona con mucha "inercia" (orden>1)

© 2023, Antonio Sala Piqueras, Universitat Politècnica de València. Todos los derechos reservados.

Presentación en vídeo en <http://personales.upv.es/asala/YT/V/linregla3ord2.html>

Este código ejecutó sin errores en Matlab R2022b (Linux)

**Objetivo:** comprender intuitivamente el concepto de sistema (dinámico) lineal invariante en el tiempo, y cómo usar la respuesta temporal para calcular perfiles de entrada aproximados que consigan ciertos objetivos. Comprender que en procesos de orden mayor que 1 **no funciona tan bien** como en el caso de primer orden.

## Tabla de Contenidos

Ensayo escalón del que obtener datos.....	1
Resolución de problemas de cálculo de escalones de entrada.....	2
Valor final prefijado.....	2
Valor final y tiempo de establecimiento prefijado.....	3
Apéndice funciones auxiliares.....	5

## Ensayo escalón del que obtener datos

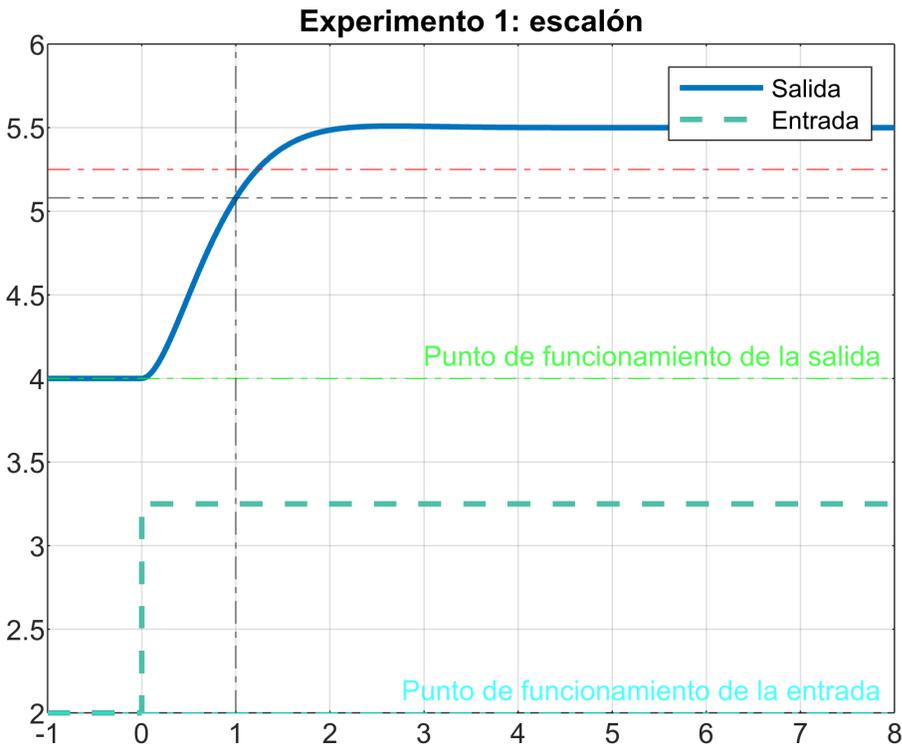
Consideremos un cierto sistema desconocido, del que sólo sabemos que es lineal (o aproximadamente) alrededor de un punto de operación  $u=2$ ,  $y=4$ ;

Simulemos su respuesta ante escalón de 0.5 (incremental)

```
u_pf=2; y_pf=4;
inc_u=1.25;
u=@(t) u_pf+inc_u*(t>=0);
Y=simulasistema(u);
yline(y_pf, '-.g', Label="Punto de funcionamiento de la salida")
yline(u_pf, '-.c', Label="Punto de funcionamiento de la entrada")
```

Marcamos en la gráfica ciertas líneas que necesitaremos en los ejemplos posteriores

```
t_est_deseado=1.0;
xline(t_est_deseado, '-.') %lo necesitamos luego
yline(5.25, '-.r') %también lo necesitamos luego
yline(5.08, '-.') %también lo necesitamos luego
legend("Salida", "Entrada"), title("Experimento 1: escalón")
```



```
valfin=Y(end) %valor final
```

```
valfin = 5.5000
```

## Resolución de problemas de cálculo de escalones de entrada

Con sólo esa información del ensayo ante escalón, podemos contestar a varias preguntas.

### Valor final prefijado

1.) ¿Qué entrada será necesaria para subir a **5.25** unidades la salida?

Por "proporcionalidad"

```
incsalida=valfin-y_pf
```

```
incsalida = 1.5000
```

```
gananciaestatica=incsalida/inc_u %incremento por unidad de entrada
```

```
gananciaestatica = 1.2000
```

```
incsalida_deseado=5.25-y_pf
```

```
incsalida_deseado = 1.2500
```

```
inc_entrada_calculado=incsalida_deseado/gananciaestatica
```

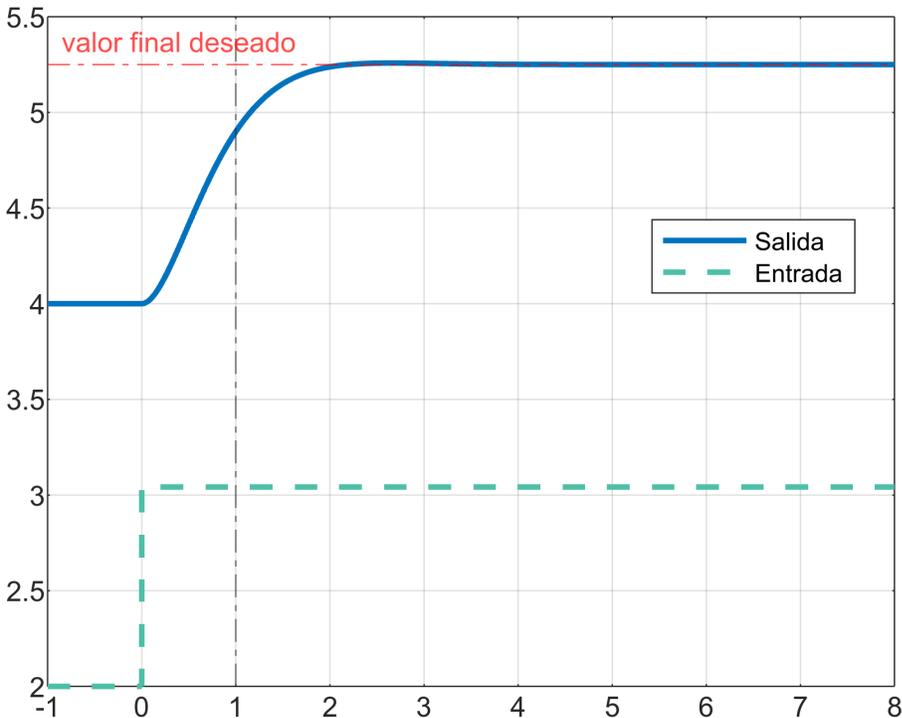
```
inc_entrada_calculado = 1.0417
```

Por tanto, el valor "absoluto" que en equilibrio conseguirá la salida deseada será:

```
u_calculado=u_pf+inc_entrada_calculado
```

```
u_calculado = 3.0417
```

```
u=@(t) u_pf+inc_entrada_calculado*(t>=0);  
simulasistema(u);  
xline(t_est_deseado,'-.')  
yline(5.25,'-.r',Label="valor final deseado",LabelHorizontalAlignment="left")  
legend("Salida","Entrada",Location="best")
```



## Valor final y tiempo de establecimiento prefijado

1.) ¿Qué entrada será necesaria para subir a **5.25** unidades la salida **en 1.4 segundos**?

Por "proporcionalidad", si con incremento 1.25 sube 0.35 unidades en 1.4 segundos, para subir a 5.25:

```
incsalida=5.08-y_pf
```

```
incsalida = 1.0800
```

```
ganancia_en_1segundo=incsalida/inc_u %incremento por unidad de entrada
```

```
ganancia_en_1segundo = 0.8640
```

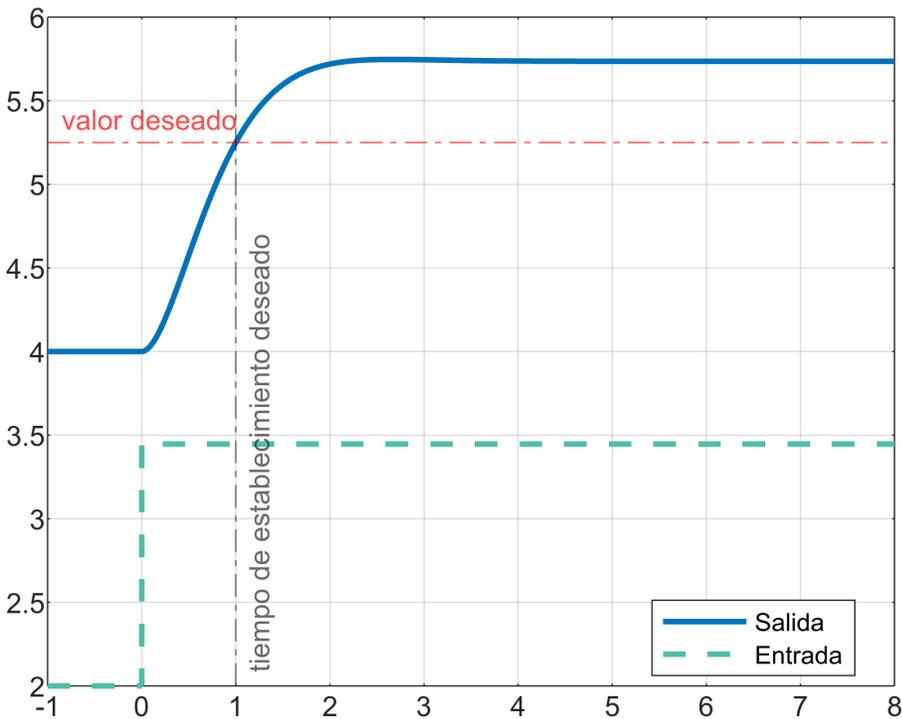
```
incsalida_deseado=5.25-y_pf
```

```
incsalida_deseado = 1.2500
```

```
inc_entrada_calculado2=incsalida_deseado/ganancia_en_1segundo
```

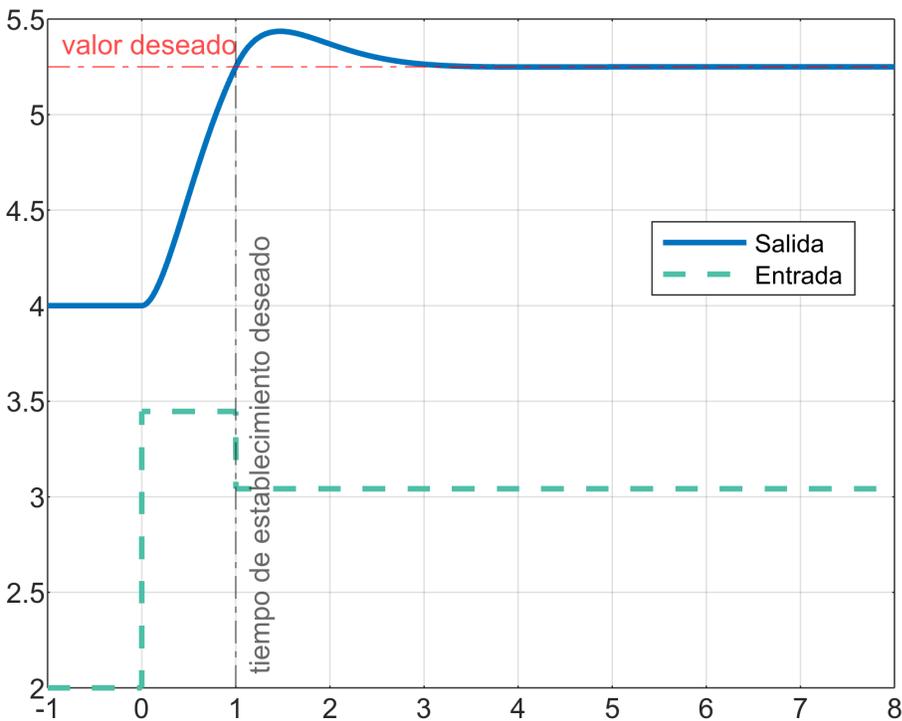
```
inc_entrada_calculado2 = 1.4468
```

```
u=@(t) u_pf+inc_entrada_calculado2*(t>=0);  
simulasistema(u);  
yline(5.25, '-.r', Label="valor deseado", LabelHorizontalAlignment="left")  
xline(t_est_deseado, '-.', Label="tiempo de establecimiento deseado", LabelVerticalAlignment="bottom")  
legend("Salida", "Entrada", Location="best")
```



Para que no "se pase" habrá que conmutar al valor de entrada que mantiene el valor deseado en equilibrio una vez se alcanza:

```
switchfact=1;  
u=@(t) u_pf+inc_entrada_calculado2*(t>=0).*(t<=t_est_deseado*switchfact)+inc_entrada_calculado2*(t>t_est_deseado*switchfact);  
simulasistema(u);  
yline(5.25, '-.r', Label="valor deseado", LabelHorizontalAlignment="left")  
xline(t_est_deseado, '-.', Label="tiempo de establecimiento deseado", LabelVerticalAlignment="bottom")  
legend("Salida", "Entrada", Location="best")
```



Esto es, hemos diseñado un perfil de 2 escalones: un "boost" inicial para que suba más rápido y luego un valor "permanente" para mantenerse en el punto deseado.

**NOTA:** los cálculos realizados sólo son "**exactos**" en **sistemas lineales de primer orden**; en *sistemas lineales de orden superior* hay cierta "*inercia*" que hará que aunque se baje la entrada hasta el punto de equilibrio calculado, haya un cierto "sobrepasamiento transitorio".

La generalización de esto lleva a cosas que se llaman "control óptimo bang-bang, bang-off-bang, control deadbeat"...

## Apéndice funciones auxiliares

Este código es "*secreto*": no hace falta examinarlo para realizar los cálculos objetivo de este material.

```
function dxdt=modelo1(x,u)
A=[0 1;-5 -3.8]; B=[0;6];
dxdt=A*x+B*u+[0;8];
end

function Y=simulasistema(u)
opts=odeset(RelTol=1e-5,AbsTol=1e-5);
[T,X]=ode45(@(t,x) modelo1(x,u(t)),[-1 8],[4;0],opts);
Y=X(:,1);
```

```
plot(T,Y,LineWidth=2), grid on
hold on
plot(T,u(T),'--',LineWidth=2,Color=[.3 .75 .65])
hold off
end
```