Tarea 4 Laboret

Se dispondrá para cada alumno una tabla con valores de masa (m), longitud (1), coeficiente de rozamiento (b), constante gravitatoria g (10) y ángulo de referencia en grados (delta) de un péndulo simple con la ecuación

$$ml^2\ddot{\theta} + b\dot{\theta} + mglsen(\theta) = T$$

Se desea que el péndulo se estabilice en el ángulo δ dado

Tomando como estados, entrada y salida respectivamente (nótese que se ha desplazado el punto de equilibrio al origen tomando el error como salida)

$$x_1 = \theta - \delta = e$$
$$x_2 = \dot{\theta}$$
$$u = T$$
$$y = e$$

- Hallar el sistema dinámico en VE

$$\dot{x} = f(x,u)$$
$$y = h(x)$$

- Hallar el torque estático necesario u_f para que el sistema tenga como punto de equilibrio el origen, es decir $f(0,u_f) = 0$
- Linealizar el sistema mediante la jacobiana

$$\dot{x} = Ax + Bu$$
$$y = Cx$$

En el punto

$$\begin{bmatrix} x_1 \\ x_2 \\ u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ u_f \end{bmatrix}$$

- Hallar los autovalores de A y determinar estabilidad por el método indirecto de Lyapunov
- Comparar los resultados obtenidos con los de la linealizacion por Matlab y Simulink

```
m=
b=
delta=  %en grados
l=1,G=10
[A,B,C,D]=linmod('pendulo_mod_tarea',delta*pi/180)
eig(A)
rank(ctrb(A,B))
```

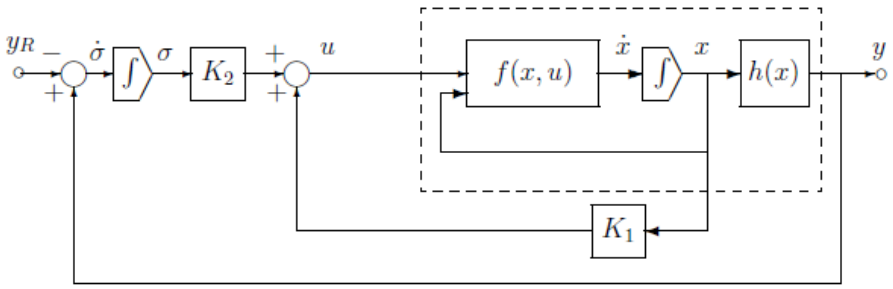
Se desea diseñar para el péndulo linealizado en el punto de operación dado un controlador con acción integral como se muestra

$$u = k_1(\theta - \delta) + k_2\dot{\theta} + k_3\sigma$$
$$\dot{\sigma} = \theta - \delta$$

Que no es otra cosa que un PID en la forma PI+D, ya que el torque vale

$$T = k_1 e + k_3 \int_0^t e(\tau)d\tau + k_2\dot{\theta}$$

Donde el error se definio en el sentido tradicional como $e = \delta - \theta$



Control por realimentación de estados y salidas con acción integral

- Encontrar las matrices del sistema ampliado

$$A_A = \begin{bmatrix} A & 0 \\ C & 0 \end{bmatrix} \quad B_A = \begin{bmatrix} B \\ 0 \end{bmatrix}$$

```
Aa=[[A;C] zeros(3,1)]
Ba=[B;0]
eig(Aa)
rank(ctrb(Aa,Ba))
```

- Verificar autovalores, estabilidad y controlabilidad del nuevo par

- Diseñar por asignación de polos un controlador con la orden *acker()* de matlab

$$u = -\begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \sigma \end{bmatrix}$$

Para ubicar un polo triple en p (dato) lo cual daría una respuesta sin sobrepaso (si el sistema fuera lineal y no tuviera ceros de lazo cerrado) y el tiempo 2% seria $t_{ss} \simeq \dfrac{7.5}{-p}$

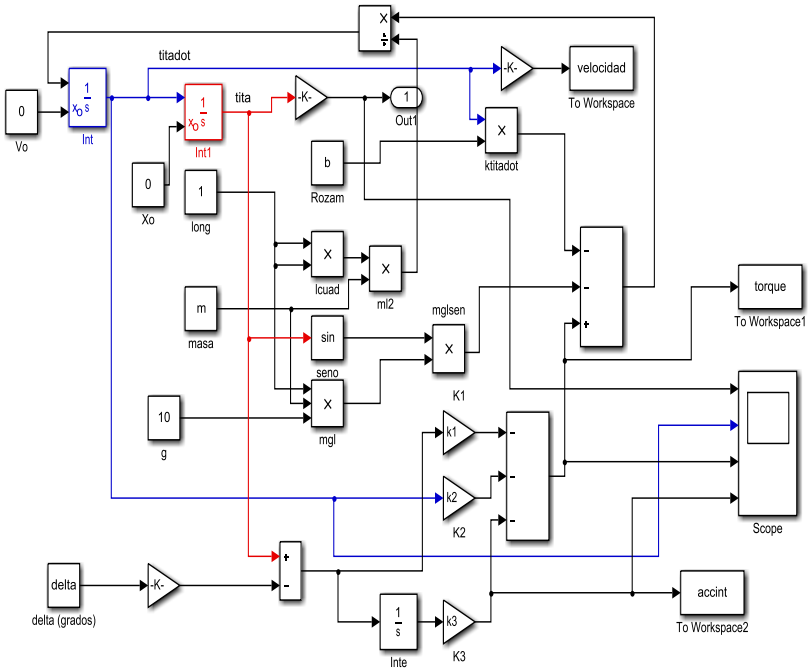
```
p=  %dato
K=acker(Aa,Ba,[p p p])
k1=K(1)
k2=K(2)
k3=K(3)
eig(Aa-Ba*K) % polos lazo cerrado
tscalc=7.5/(-p) % tiempo de respuesta calculado
```

- Simular el péndulo con PID como se muestra en la figura partiendo del origen con velocidad nula y referencia δ (dato)

- Dibujar la salida, el plano de fases, el torque total y la acción integral, comparar con el valor de u_f calculado antes y verificar sobrepaso y tiempo de establecimiento real versus el calculado

➢ En todos los casos ajustar las escalas para una visualización correcta

Se adjunta el archivo de simulación pendulo_PID_tarea.slx



Esquema simulación péndulo con PID

```
sim('pendulo_PID_tarea')
figure(1), plot(tout,yout)
grid on, title('Salida')
figure(2), plot(yout,velocidad) %plano de fase
grid on, title('Plano de fases')
figure(3), plot(tout,torque) % torque total
grid on, title('Torque')
figure(4), plot(tout,-accint) % acción integral
grid on,  title('Accion integral')
ymax=max(yout) % máximo valor de salida
S=(ymax-delta)/delta*100 % sobrepaso en %
erel=(delta-yout)/delta; %error relativo
efinal=erel(end) % error final, debe ser cero
ind=find(abs(erel)>.02); % índice elementos con error relativo absoluto menor a 2%
tss=tout(ind(end)) % tiempo de establecimiento (ultimo valor del vector)
yte=yout(ind(end)) % salida al tiempo ts
uf=torque(end) % torque final
Intf=-accint(end) % acción integral final
```

- Analizar la robustez variando la masa del péndulo en mas y menos 10% analizar los nuevos valores de sobrepaso, tiempo de establecimiento y acción de control final, elaborar una tabla con los resultados para los distintos valores de masa

```
% para analizar robustez
mnom=m % masa nominal
m=0.9*mnom % correr de nuevo el código de simulación, dibujo y analisis
……….
m=1.1*mnom % ídem
```