Tarea 10 - Atrasada

Roberto Cadena Vega

12 de marzo de 2015

Análisis de estabilidad para sistema bajo realimentación integral

Ejemplo 1

Dado el sistema:

$$\dot{x}(t) = Ax(t) + Bu(t-h)$$

con la ley de control:

$$u(t) = k\left[x(t) + \int_{-h}^{0} e^{-A(\theta+h)} Bu(t+\theta) d\theta\right]$$

por lo que el sistema en lazo cerrado es:

$$\dot{x}(t) = \left(A + e^{-Ah}BK\right) x(t)$$

el sistema para el que haremos este desarrollo es:

$$\dot{x}(t) = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(t-h)$$

con $h = 1$.
Calculando $e^{-A(\theta+h)}$ tenemos:

In [2]:
```
from IPython.display import display

from sympy import var, sin, cos, Matrix, Integer, eye, Function, Rational, exp, Symbol, I
from sympy.physics.mechanics import mlatex, mechanics_printing
from sympy.integrals import laplace_transform
mechanics_printing()
```

In [3]:
```
var("t h θ s ω")
```

Out [3]:

$$(t, \quad h, \quad \theta, \quad s, \quad \omega)$$

In [4]:
```
A = Matrix([[0, 0], [1, 1]])
A
```

Out [4]:

$$\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$$

In [5]:
```
exp(-A*(θ+ h))
```

Out[5]:

$$\begin{bmatrix} 1 & 0 \\ e^{-h-\theta} - 1 & e^{-h-\theta} \end{bmatrix}$$

Sustituyendo A, B y $e^{-A(t+\theta)}$ en $u(t)$, tenemos:

$$\begin{aligned} u(t) &= \begin{pmatrix} k_1 & k_2 \end{pmatrix} x(t) + \begin{pmatrix} k_1 & k_2 \end{pmatrix} \int_{-h}^{0} e^{-A(\theta+h)} B u(t+\theta) d\theta \\ &= \begin{pmatrix} k_1 & k_2 \end{pmatrix} x(t) + \begin{pmatrix} k_1 & k_2 \end{pmatrix} \int_{-h}^{0} \begin{pmatrix} 1 & 0 \\ e^{-(\theta+h)} - 1 & e^{-(\theta+h)} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(t+\theta) d\theta \\ &= \begin{pmatrix} k_1 & k_2 \end{pmatrix} x(t) + \begin{pmatrix} k_1 & k_2 \end{pmatrix} \int_{-h}^{0} \begin{pmatrix} 1 \\ e^{-(\theta+h)} - 1 \end{pmatrix} u(t+\theta) d\theta \end{aligned}$$

Sustituyendo $k_1 = 1 - 4e^h$ y $k_2 = -4e^h$ tenemos:

$$u(t) = \begin{pmatrix} 1 - 4e^h & -4e^h \end{pmatrix} x(t) + \begin{pmatrix} 1 - 4e^h & -4e^h \end{pmatrix} \int_{-h}^{0} \begin{pmatrix} 1 \\ e^{-(\theta+h)} - 1 \end{pmatrix} u(t+\theta) d\theta$$

y podemos meter estas ganancias a la integral, para obtener:

$$\begin{aligned} u(t) &= \begin{pmatrix} 1 - 4e^h & -4e^h \end{pmatrix} x(t) + \int_{-h}^{0} \left(1 - 4e^{-\theta}\right) u(t+\theta) d\theta \\ &= \begin{pmatrix} 1 - 4e^h & -4e^h \end{pmatrix} x(t) + \int_{-h}^{0} u(t+\theta) d\theta - \int_{-h}^{0} 4e^{-\theta} u(t+\theta) d\theta \end{aligned}$$

```
In [6]: B = Matrix([[1], [0]])
        K = Matrix([[1 - 4*exp(h), -4*exp(h)]])
```

```
In [7]: (K*exp(-A*(θ + h))*B)[0].simplify()
```

Out[7]:

$$1 - 4e^{-\theta}$$

```
In [8]: x1 = Function("x1")(t)
        x2 = Function("x2")(t)

        X = Matrix([[x1], [x2]])

        u = Function("u")(t + θ)
```

```
In [9]: A*X
```

Out[9]:

$$\begin{bmatrix} 0 \\ x_1 + x_2 \end{bmatrix}$$

```
In [10]: ((K*exp(-A*(θ + h))*B)[0].simplify()*u).integrate((θ, -h, 0))
```

Out[10]:

$$\int_{-h}^{0} \left(e^{\theta} - 4\right) \mathrm{u}(t+\theta) e^{-\theta} \, d\theta$$

```
In [11]: (K*X)[0] + ((K*exp(-A*(θ + h))*B)[0].simplify()*u).integrate((θ, -h, 0))
```

Out[11]:

$$\left(-4e^{h}+1\right)x_{1}-4x_{2}e^{h}+\int_{-h}^{0}\left(e^{\theta}-4\right)\mathrm{u}\left(t+\theta\right)e^{-\theta}\,d\theta$$

Si aplicamos la transformada de Laplace a esto, obtendremos:

$$u(t)-\int_{-h}^{0}u(t+\theta)d\theta+\int_{-h}^{0}4e^{-\theta}u(t+\theta)d\theta=(1-4e^{h})x_1(t)-4e^{h}x_2(t)$$
$$\left[1-\frac{1-e^{-hs}}{s}+4\frac{1-e^{-h(s-1)}}{s-1}\right]u(s)=(1-4e^{h})x_1(s)-4e^{h}x_2(s)$$

Por lo que el polinomio caracteristico del controlador del sistema es:

$$1-\frac{1-e^{-hs}}{s}+4\frac{1-e^{-h(s-1)}}{s-1}=0$$

Si ahora introducimos los parametros $\alpha_1=k_1-k_2=1$ y $\alpha_2=e^{-h}k_2=-4$, este polinomio caracteristico queda de la forma:

$$1-\alpha_1\frac{1-e^{-hs}}{s}-\alpha_2\frac{1-e^{-h(s-1)}}{s-1}=0$$

Este polinomio caracteristico corresponde al controlador del sistema, por otro lado, tambien tenemos que analizar la estabilidad del sistema bajo esta realimentación, la ecuación para esto es:

$$\dot{x}(t)=\left(A+e^{-Ah}BK\right)x(t)$$

por lo que la función de transferencia del sistema realimentado será:

$$\det\left(sI-A-e^{-Ah}BK\right)$$

In [12]:
```
(s*eye(2) - A - exp(-h*A)*B*K).det()
```

Out[12]:

$$s^{2}+2s+1$$

Por lo que observamos que esta realimentación coloca dos polos en -1, sin embargo queremos analizar la estabilidad bajo los parametros que establecimos, por lo que notamos que este polinomio puede ser escrito como:

$$s^{2}-\left(1+\alpha_1+\alpha_2\right)s+\alpha_1$$

In [13]:
```
α1, α2 = K[0] - K[1], exp(-h)*K[1]
α1, α2
```

Out[13]:

$$\left(1,\quad -4\right)$$

In [14]:
```
s**2 - (1 + α1 + α2)*s + α1
```

Out[14]:

$$s^{2}+2s+1$$

Este polinomio caracteristico esta libre de retardos, por lo que podemos analizarlo con Routh-Hurwitz y obtener las siguientes condiciones:

$$\alpha_1 > 0$$
$$\alpha_1 < -1 - \alpha_2$$

Por otro lado, si hacemos un analisis de D-particiones, al sustituir $s = 0$ y $s = j\omega$ obtenemos que:

$$\alpha_1 = 0$$
$$\alpha_1 = -1 - \alpha_2$$

```
In [15]: var("α_1 α_2 ω")
```

Out[15]:

$$(\alpha_1, \quad \alpha_2, \quad \omega)$$

```
In [16]: (s**2 - (1 + α_1 + α_2)*s + α_1).subs(s, 0)
```

Out[16]:

$$\alpha_1$$

```
In [17]: (s**2 - (1 + α_1 + α_2)*s + α_1).subs(s, 1j*ω).coeff(-1j*ω)
```

Out[17]:

$$\alpha_1 + \alpha_2 + 1$$

Lo cual es consistente con los resultados de Routh-Hurwitz. Al graficar estas curvas limite de las D-particiones, obtenemos:

```
In [18]: from numpy import linspace, zeros
```

```
In [19]: %matplotlib inline
         from matplotlib.pyplot import plot, style, figure, legend, fill
         style.use("ggplot")
```

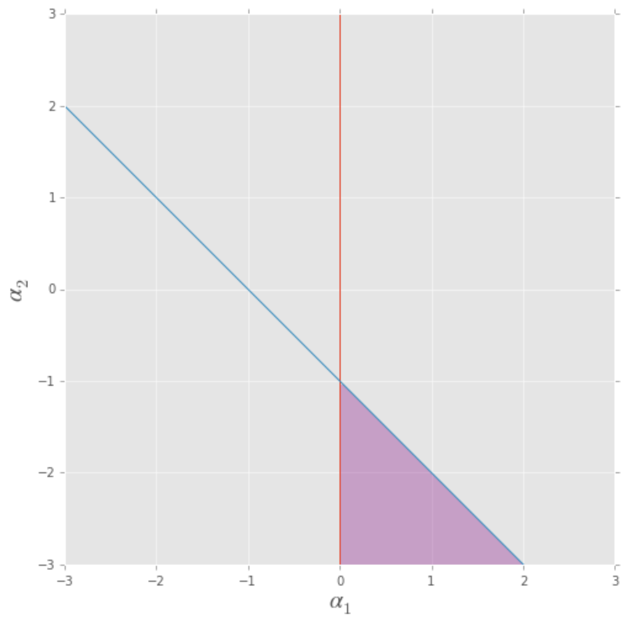
```
In [20]: x = linspace(-4, -1, 100)
         alpha1 = linspace(-4, 4, 100)
         alpha2 = -alpha1 - 1
```

```
In [21]: f = figure(figsize=(8, 8))
         plot(zeros(len(alpha1)), alpha1)
         plot(alpha1, alpha2)

         ax = f.gca()
         ax.set_xlim(-3, 3)
         ax.set_ylim(-3, 3)

         ax.fill_betweenx(alpha2, 0, alpha1, where=alpha1>0, alpha=0.3, facecolor='purple')

         ax.set_xlabel(r"$α_1$", fontsize=20)
         ax.set_ylabel(r"$α_2$", fontsize=20);
```



Ejemplo 2

Para el sistema:

$$\dot{x}(t) = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x(t) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(t-h)$$

con $h = 1$.

```
In [22]: var("k1 k2")
```

```
Out[22]:
```

$$(k_1, \quad k_2)$$

```
In [23]: A1 = Matrix([[0, 1], [0, 0]])
         B1 = Matrix([[0], [1]])
         K1 = Matrix([[k1, k2]])
```

Tiene un polinomio caracteristico:

```
In [24]: ((s*eye(2) - A1 - exp(-A1*h)*B1*K1).det()).collect(s)
```

```
Out[24]:
```

$$-k_1 + s^2 + s\left(hk_1 - k_2\right)$$

o bien:

$$s^2 + \left(hk_1 - k_2\right)s - k_1$$

Al cual podemos aplicar el criterio de estabilidad de Routh-Hurwitz y obtener:

$$k_1 < 0$$
$$k_2 < hk_1$$

Por lo que la gráfica de D-particiones se verá:

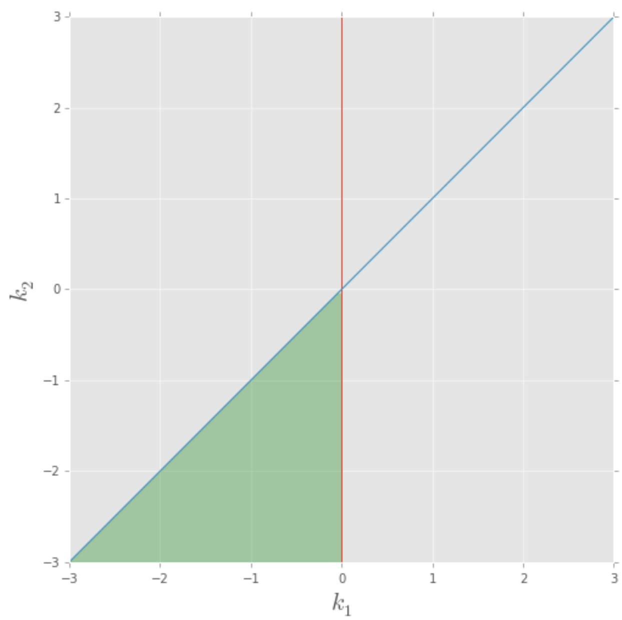
```
In [25]: h = 1
         x = linspace(-4, -1, 100)
         K_1 = linspace(-4, 4, 100)
         K_2 = h*K_1

In [26]: f = figure(figsize=(8, 8))
         plot(zeros(len(K_1)), K_1)
         plot(K_1, K_2)

         ax = f.gca()
         ax.set_xlim(-3, 3)
         ax.set_ylim(-3, 3)

         ax.fill_betweenx(K_2, 0, K_1, where=K_1<0, alpha=0.3, facecolor='green')

         ax.set_xlabel(r"$k_1$", fontsize=20)
         ax.set_ylabel(r"$k_2$", fontsize=20);
```



Por otro lado, para analizar el comportamiento del controlador, sustituimos los datos en la ecuación del controlador:

$$
\begin{aligned}
u(t) &= \begin{pmatrix} k_1 & k_2 \end{pmatrix} x(t) + \begin{pmatrix} k_1 & k_2 \end{pmatrix} \int_{-h}^{0} e^{-A(\theta+h)} B u(t+\theta) d\theta \\
&= \begin{pmatrix} k_1 & k_2 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \begin{pmatrix} k_1 & k_2 \end{pmatrix} \int_{-h}^{0} e^{-A(\theta+h)} B u(t+\theta) d\theta
\end{aligned}
$$

```
In [27]: exp(-A1*(θ + h))
```

Out[27]:

$$
\begin{bmatrix} 1 & -\theta - 1 \\ 0 & 1 \end{bmatrix}
$$

$$
\begin{aligned}
u(t) &= \begin{pmatrix} k_1 & k_2 \end{pmatrix} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + \int_{-h}^{0} \begin{pmatrix} k_1 & k_2 \end{pmatrix} \begin{pmatrix} 1 & -(\theta + h) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(t+\theta) d\theta \\
&= k_1 x_1(t) + k_2 x_2(t) - \int_{-h}^{0} k_1 \theta u(t+\theta) d\theta - \int_{-h}^{0} k_1 h u(t+\theta) d\theta + \int_{-h}^{0} k_2 u(t+\theta) d\theta
\end{aligned}
$$

y al aplicar la transformada de Laplace, tenemos:

$$
u(s) = k_1 x_1(s) + k_2 x_2(s) - k_1 \left(1 - e^{-hs}\right) u(s) - k_1 h \frac{1 - e^{-hs}}{s} u(s) + k_2 \frac{1 - e^{-hs}}{s} u(s)
$$

por lo que al pasar a un solo lado todos los terminos de $u(s)$:

$$
\left[1 + k_1 \left(1 - e^{-hs}\right) + k_1 h \frac{1 - e^{-hs}}{s} - k_2 \frac{1 - e^{-hs}}{s}\right] u(s) = k_1 x_1(s) + k_2 x_2(s)
$$

obtenemos el polinomio caracteristico de la ecuación de control:

$$
1 + k_1 \left(1 - e^{-hs}\right) + k_1 h \frac{1 - e^{-hs}}{s} - k_2 \frac{1 - e^{-hs}}{s}
$$

y al sustituir $s = j\omega$, obtendremos dos ecuaciones, correspondientes a la parte real e imaginaria:

$$
\begin{aligned}
k_1 \left[h(1 - \cos(\omega h)) - \omega \sin(\omega h)\right] - k_2 \left[1 - \cos(\omega h)\right] &= 0 \\
\omega + k_1 \left[\omega(1 - \cos(\omega h)) - h \sin(\omega h)\right] - k_2 \sin(\omega h) &= 0
\end{aligned}
$$

por lo que podemos despejar k_2 de ambas ecuaciones y obtener:

$$
k_2 = \frac{k_1 \left[h(1 - \cos(\omega h)) - \omega \sin(\omega h)\right]}{1 - \cos(\omega h)} = \frac{k_1 \left[\omega(1 - \cos(\omega h)) - h \sin(\omega h)\right] + \omega}{\sin(\omega h)}
$$

y haciendo un poco de algebra, podemos obtener:

$$
\begin{aligned}
\frac{k_1 \left[h(1 - \cos(\omega h)) - \omega \sin(\omega h)\right] \sin(\omega h)}{1 - \cos(\omega h)} - k_1 \left[\omega(1 - \cos(\omega h)) - h \sin(\omega h)\right] &= \omega \\
\frac{k_1 h (1 - \cos(\omega h)) \sin(\omega h)}{1 - \cos(\omega h)} - \frac{k_1 \omega \sin^2(\omega h)}{1 - \cos(\omega h)} - k_1 \omega (1 - \cos(\omega h)) - k_1 h \sin(\omega h) &= \omega \\
-\frac{k_1 \omega \sin^2(\omega h)}{1 - \cos(\omega h)} - k_1 \omega (1 - \cos(\omega h)) &= \omega \\
-k_1 \omega \frac{\sin^2(\omega h) + (1 - \cos(\omega h))^2}{1 - \cos(\omega h)} &= \omega \\
k_1 \frac{\sin^2(\omega h) + (1 - \cos(\omega h))^2}{1 - \cos(\omega h)} &= -1 \\
k_1 \frac{\sin^2(\omega h) + 1 + \cos^2(\omega h) - 2\cos(\omega h)}{1 - \cos(\omega h)} &= -1 \\
k_1 \frac{2 - 2\cos(\omega h)}{1 - \cos(\omega h)} &= -1 \\
k_1 &= -\frac{1}{2}
\end{aligned}
$$

por otro lado, podemos reducir aun mas la expresión para k_2:

$$
k_2 = \frac{k_1 \left[h(1 - \cos(\omega h)) - \omega \sin(\omega h)\right]}{1 - \cos(\omega h)} = k_1 \left[h - \frac{\omega \sin(\omega h)}{1 - \cos \omega h}\right]
$$

Si sustituimos un punto a la derecha de esta curva, $(k_1, k_2) = (0, 0)$, podemos ver que el polinomio caracteristico es trivialmente estable por el criterio de Routh-Hurwitz:

$$P(s) = 1$$

por lo que la gráfica de D-particiones para el controlador queda:

In [28]:
```
def par1(ω, h):
    from numpy import sin, cos
    num = -1.0
    den = 2.0
    return num/den

def par2(ω, h):
    from numpy import sin, cos
    num = h - ω*sin(ω*h)/(1 - cos(ω*h))
    den = 2.0
    return num/den
```

In [29]:
```
from numpy import pi
```

In [30]:
```
τ = 2*pi
k_1 = [par1(om, 1.0) for om in linspace(-2*τ, 2*τ, 100)]
k_2 = [par2(om, 1.0) for om in linspace(-2*τ, 2*τ, 100)]
```

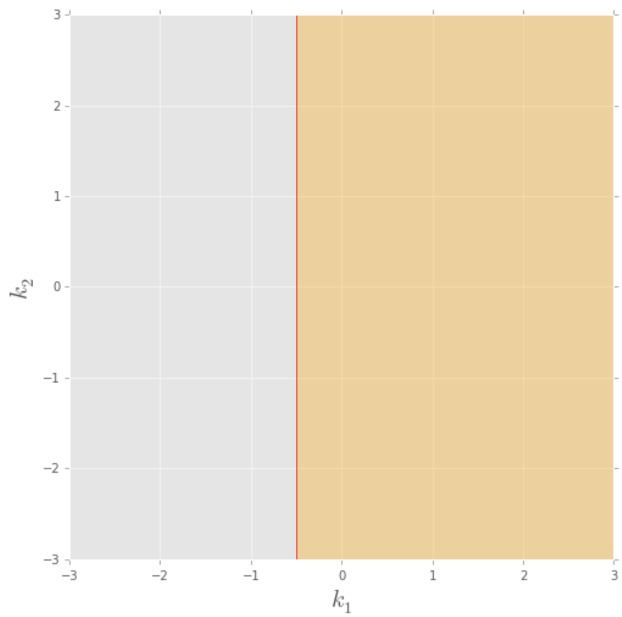
```
/Users/roberto/miniconda3/lib/python3.4/site-packages/IPython/kernel/__main__.py:9: RuntimeWarning: divi
```

In [31]:
```
f = figure(figsize=(8, 8))
plot(k_1, k_2)

ax = f.gca()
ax.set_xlim(-3, 3)
ax.set_ylim(-3, 3)

ax.axvspan(-0.5, 3, alpha=0.3, color='orange')

ax.set_xlabel(r"$k_1$", fontsize=20)
ax.set_ylabel(r"$k_2$", fontsize=20);
```



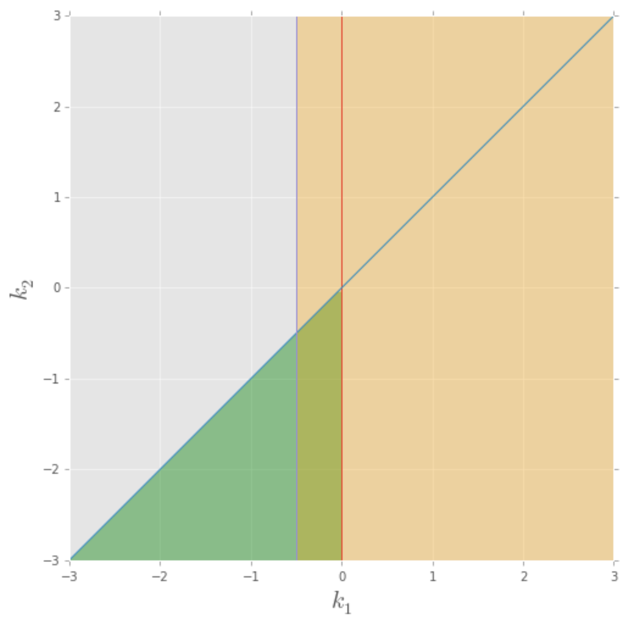
Y el sistema con este controlador será estable para los valores de k_1 y k_2 escogidos tal que se encuentren en la intersección de estas dos regiones:

```
In [32]: f = figure(figsize=(8, 8))
         plot(zeros(len(K_1)), K_1)
         plot(K_1, K_2)
         plot(k_1, k_2)

         ax = f.gca()
         ax.set_xlim(-3, 3)
         ax.set_ylim(-3, 3)

         ax.fill_betweenx(K_2, 0, K_1, where=K_1<0, alpha=0.4, facecolor='green')
         ax.axvspan(-0.5, 3, alpha=0.3, color='orange')

         ax.set_xlabel(r"$k_1$", fontsize=20)
         ax.set_ylabel(r"$k_2$", fontsize=20);
```



Puedes acceder a este notebook a traves de la página
http://bit.ly/1B705kd
o escaneando el siguiente código: