Trabajo Práctico N°1.

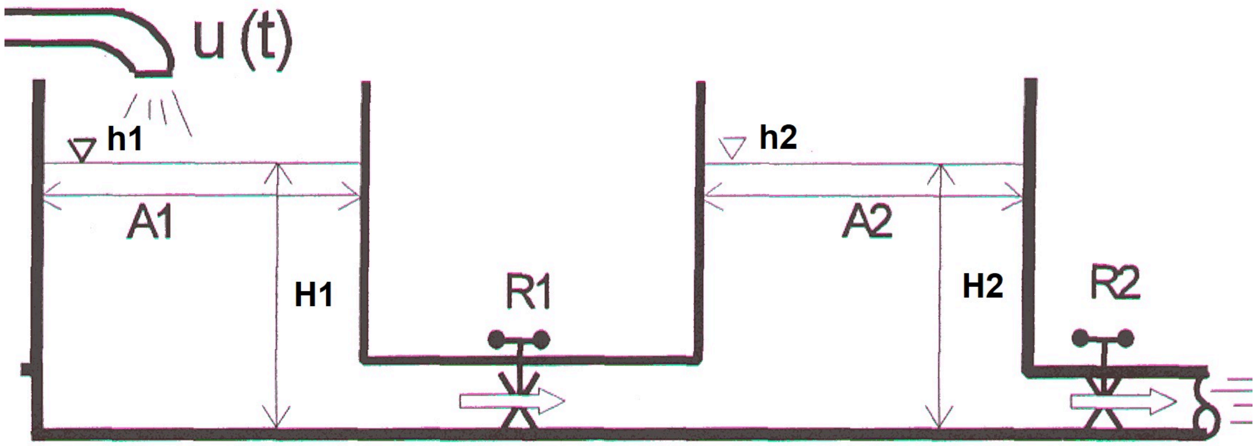
Controlador PID – Asignación de polos - LQR

Nieva Miguel Pedro Valentin | M.U:00932

Código: https://github.com/valkur5/Control-Optimo/tree/main/TP1

Problema N°1: Planta Hidráulica

Se nos plantea crear un sistema de control lineal para la siguiente planta hidráulica:



Donde las ecuaciones que representan este sistema son:

$$A_1 \frac{dh_1}{dt} = \frac{1}{R_1}(h_2 - h_1) + u(t) \qquad (1)$$

$$A_2 \frac{dh_2}{dt} = -\frac{1}{R_1}(h_2 - h_1) - \frac{1}{R_2}h_2 \quad (2)$$

Donde *u(t)* es el caudal del líquido que entra en el primer tanque.
Las constantes son $A_1 = A_2 = 1$; $R_1 = 1/2$; $R_2 = 1/3$.
El objetivo es poder mantener la altura del segundo tanque (h_2), siga una referencia constante manipulando el caudal que entra al primer tanque.
Se nos pide realizar este control utilizando diferentes controladores:

- Controlador PID.
- Controlador por asignación de polos.
- LQR en tiempo continuo, para referencia no nula igual a 1.
- DLQR con tiempo de evolución fija, es decir, un controlador variante en el tiempo.

Antes de comenzar el desarrollo de nuestros controladores, debemos analizar las ecuaciones 1 y 2, donde tenemos que obtener las matrices para el análisis en el espacio de estados y a partir de allí crearemos los controladores correspondientes.

Matrices del espacio de estados:

Nuestras variables de estado las definiremos como:

$$X'_1 = \frac{dh1}{dt}$$

$$X'_2 = \frac{dh2}{dt}$$

De esta forma, si despejamos de las ecuaciones 1 y 2 nuestras variables de estado, nos queda que:

$$X'_1 = \frac{h2}{A1*R1} - \frac{h1}{A1*R1} + \frac{u(t)}{A1}$$

$$X'_2 = -\frac{h2}{A2*R1} + \frac{h1}{A2*R1} + \frac{h2}{A2*R2}$$

De estas expresiones podemos determinar entonces:

$$A = \begin{bmatrix} -\frac{1}{A_1R_1} & \frac{1}{A_1R_1} \\ \frac{1}{A_2R_1} & -\frac{1}{A_2R_1} - \frac{1}{A_2R_2} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{A1} \\ 0 \end{bmatrix};\ C = \begin{bmatrix} 0 & 1 \end{bmatrix};\ D = 0$$

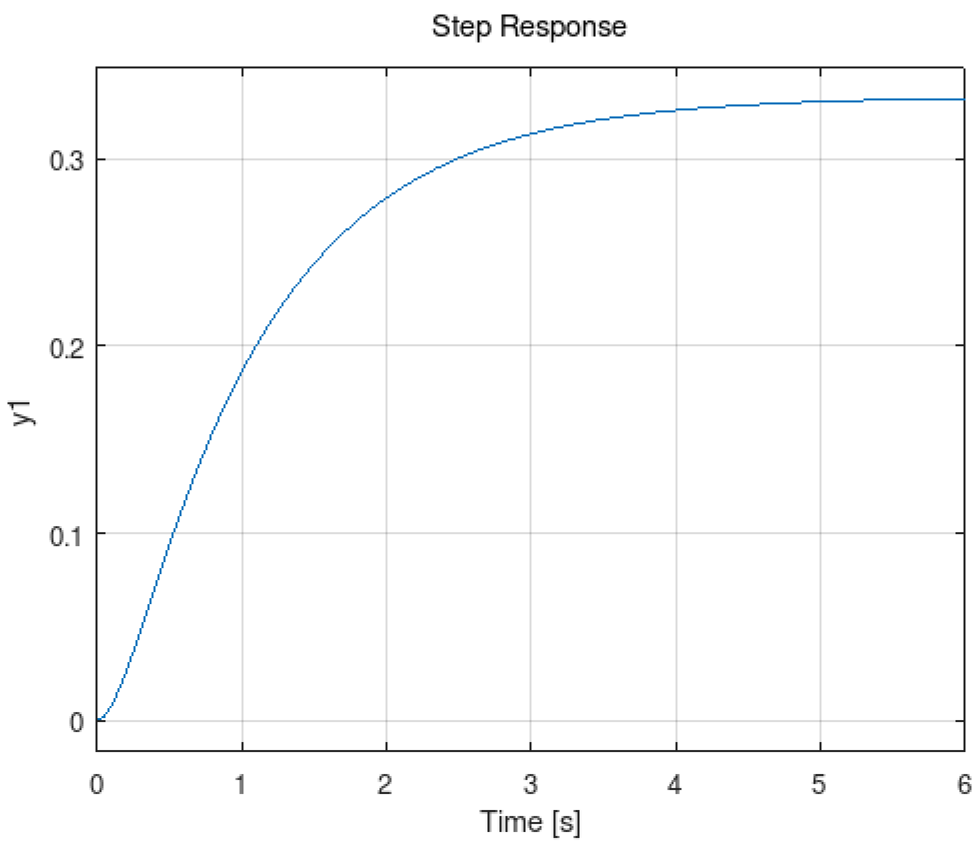
(Imagen con las matrices generada con www.mathcha.io)

Una vez definidas nuestras matrices de estado, podemos empezar con nuestros controladores.

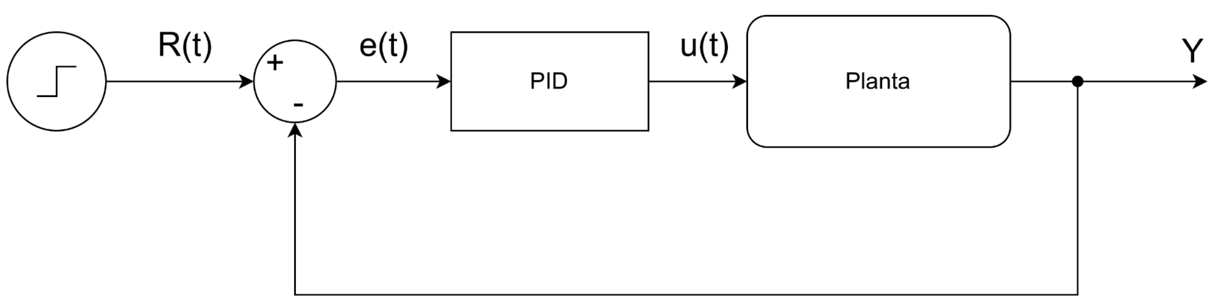
Controlador PID:

Una vez definido nuestro sistema, analizaremos primero su respuesta ante una entrada escalón, esto lo hacemos primero creando nuestro sistema con el comando *ss(A,B,Ct,D)*, lo que nos devuelve nuestro sistema formado por las matrices A, B, C y D respectivamente.

Luego utilizamos el comando ***step***, para simular la respuesta de este sistema ante una entrada escalón, dándonos por resultado la siguiente gráfica:



Tenemos entonces un sistema que es lento (tarda unos 5.5 segundos en llegar a régimen permanente), tenemos un error de estado estacionario de aproximadamente 0.66 (ya que el valor en régimen es 0.33 y debería ser 1).
Con nuestro controlador PID queremos que este sistema responda más rápido y con un error en estado estacionario menor.
Nuestro diagrama de bloques de nuestro sistema, tendría la siguiente forma:



Donde vemos que alimentamos nuestro controlador PID con el error *e(t)*.
Lo primero que debemos hacer determinar nuestras constantes del controlador PID, de momento dejaremos las 3 constantes con valor unitario, esto es:

$$Kp = 1;\ Ki = 1;\ Kd = 1$$

Debido a la forma que nosotros realizamos esta simulación, es necesario "discretizar" nuestro controlador para poder implementarlo. No estamos haciendo nuestro sistema discreto, solamente lo hacemos para que el controlador pueda ser usado en la forma que se está simulando.
Esto lo hacemos como:

$$A_1 = \frac{(2*Kp*h) + \left(Ki*h^2\right) + (2*Kd)}{2*h}$$

$$B_1 = \frac{(-2*Kp*h) + \left(Ki*h^2\right) - (4*Kd)}{2*h}$$

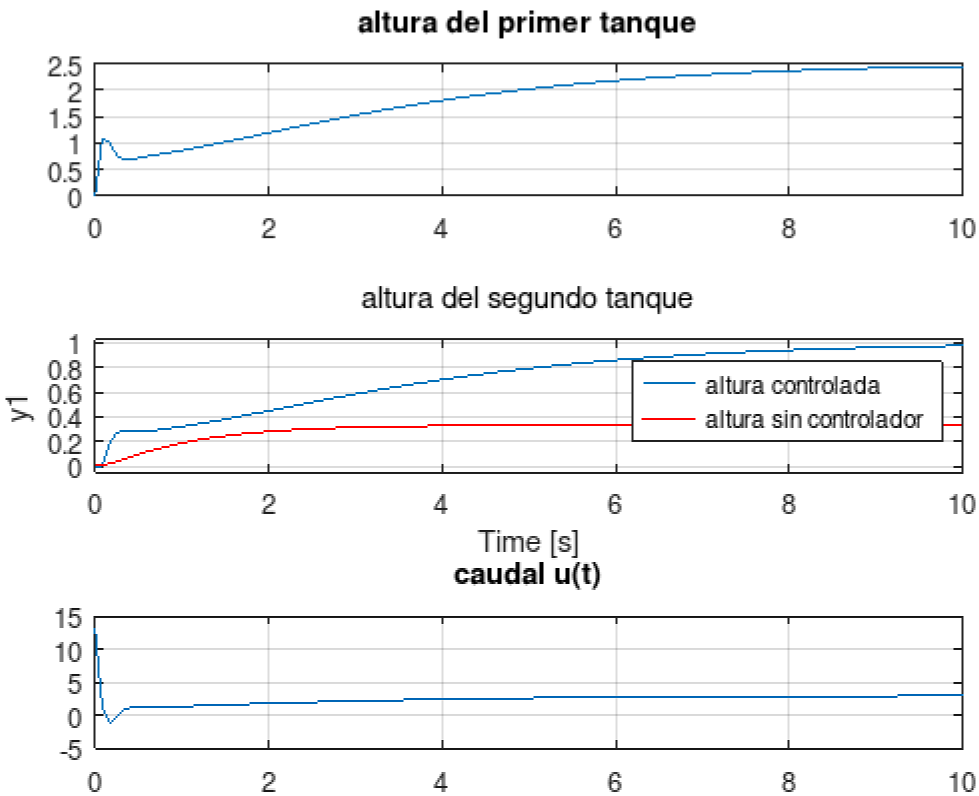
$$C_1 = \frac{Kd}{h}$$

Siendo h el tiempo entre cálculos de iterados, es decir, nuestro Δt.
Luego, la acción de control la calcularemos como:

$$u = [u + A_1*error(k) + B_1*error(k-1) + C_1*error(k-2)]$$

Donde el error se calcula como: $error = referencia - h_2$, donde lo que hacemos es calcular la diferencia entre la referencia que debe seguir la altura del segundo estanque con la altura actual del segundo estanque.
Esto lo hacemos entre cada iteración, utilizando la integración de euler. (Ver el código Anexo)

Ahora nos queda ver cómo responde nuestro sistema con las constantes del PID seleccionadas. Dando por resultado la siguiente simulación:



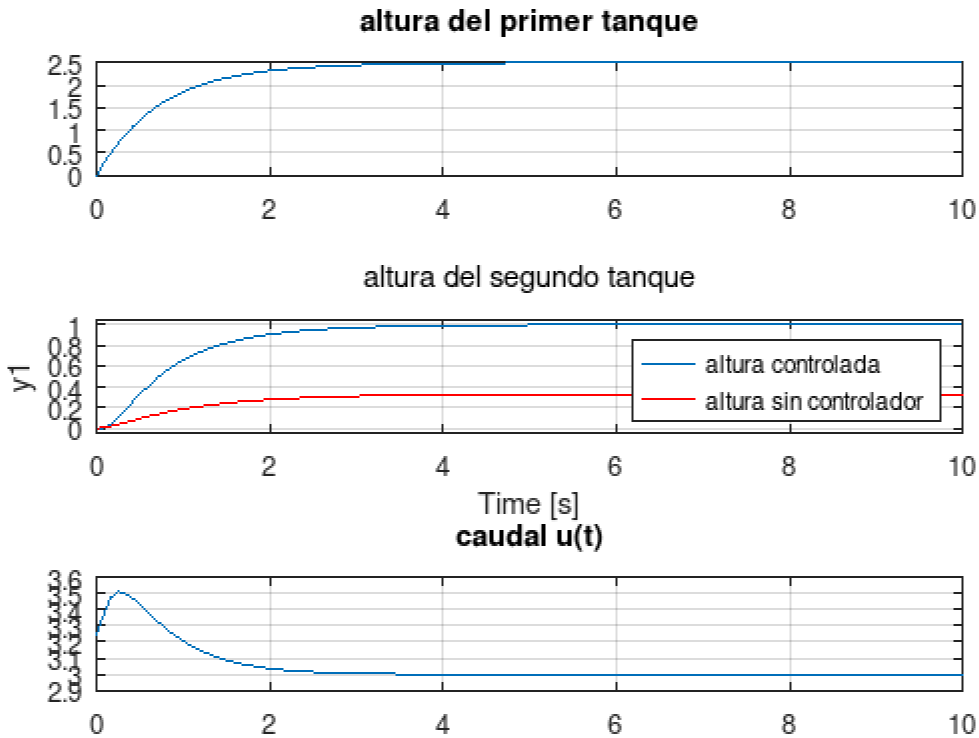
Vemos que con las constantes del PID unitarias, el sistema intenta seguir una referencia, solamente que lo hace de una manera lenta. Para mejorar esto debemos cambiar los valores del PID, y esto lo haremos siguiendo el siguiente criterio:

- **Aumentar la constante proporcional** Kp: Reduce el tiempo de error pero aumenta la sobrecarga y las oscilaciones.
- **Aumentar la constante Integral** Ki: Reduce el error estacionario pero aumenta la velocidad de respuesta.
- **Aumentar la constante derivativa** Kd: Reduce el tiempo de estabilización y la sobrecarga, pero puede aumentar la sensibilidad al ruido

Con estos criterios y mediante prueba y error, llegamos a la conclusión de que los mejores valores para nuestro controlador (sin resultar en un excesivo caudal *u(t)*), son:

$$Kp = 3\ ;\ Ki = 3\ ;\ Kd = 0.01$$

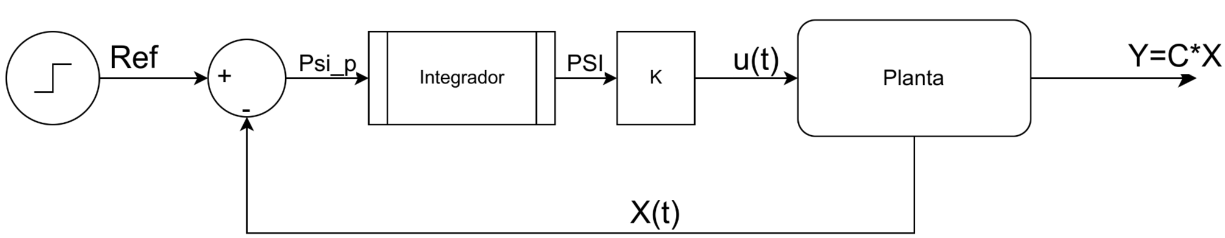
De esta forma, obtenemos como resultado:



Vemos que tenemos una altura constante en el segundo estanque y con un caudal no mayor a $3.5\ m^3/s$.

Controlador mediante asignación de polos:

Este tipo de controlador consiste en realizar una realimentación de nuestro sistema mediante una ganancia de lazo “K”, que hará que el sistema tenga los polos donde nosotros deseemos para obtener una mejor respuesta. Nuestro diagrama en bloques será:



Lo primero que tenemos que hacer antes de realizar este tipo de controladores es conocer los polos de nuestro sistema, esto lo hacemos conociendo los autovalores de nuestra matriz de estado “A”, en *octave* podemos obtener este valor rápidamente con el comando *eig*.

```
>> eig(A)
ans =

  -6
  -1
```

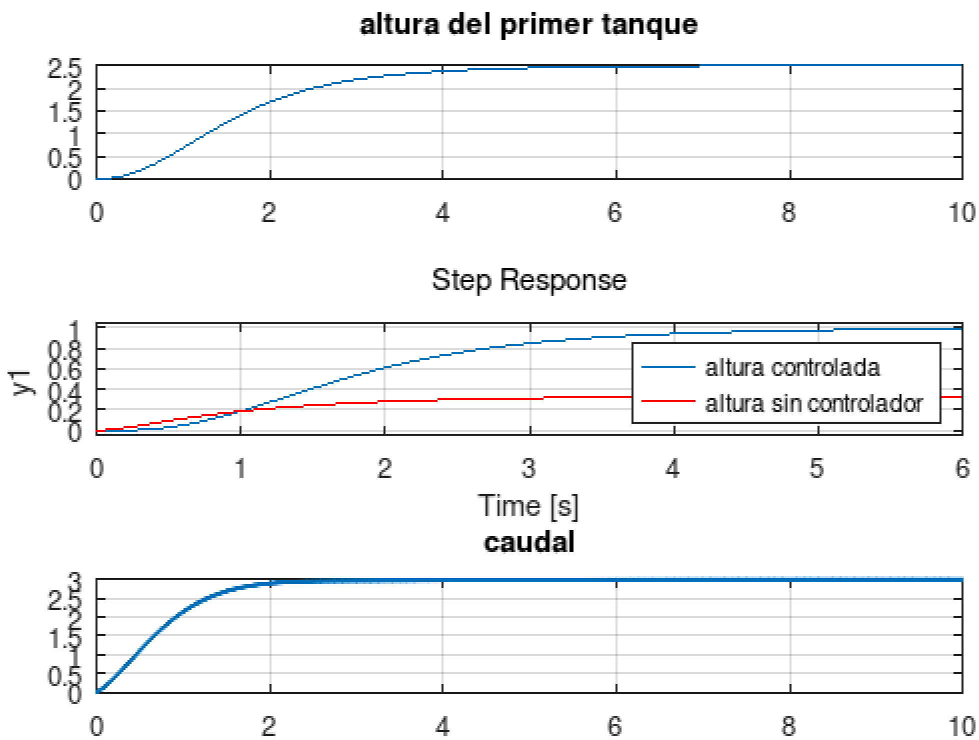
Vemos que nuestro sistema tiene 2 polos, uno en -6 y otro en -1.
Si bien este sistema es estable, podemos seleccionar polos que nos otorguen una mejor respuesta de control.
Para esto tenemos que tener en cuenta lo siguiente:

- Los polos cerca del origen nos darán respuestas rápidas, si estas se encuentran sobre el eje real esta respuesta será amortiguada.
- Los polos cerca del origen serán los polos dominantes del sistema. Los polos alejados del origen tendrán poca influencia en la evolución de nuestro sistema.
- Un par de polos complejos conjugados cerca del origen nos dará una respuesta rápida y con ciertas oscilaciones, así también como una sobreelongación máxima. Si este par está muy cerca del origen puede llevar a grandes oscilaciones y mayores tiempos de asentamiento.

Para nuestro controlador, además de un posicionamiento de polos, añadiremos un integrador de error, esto con el fin de obtener una respuesta y una acción de control más estable debido a que tenemos una referencia distinta de cero.
Además, para poder obtener el controlador K necesario para realizar esta acción de control con el integrador de error, es necesario trabajar con matrices de estado ampliadas.
Nuestro sistema ampliado nos queda entonces:

$$\begin{bmatrix} \dot{\mathbf{x}}_t \\ \dot{\xi}_t \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \cdot \begin{bmatrix} x_t \\ \xi_t \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_t + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r_t .$$

En nuestro caso, para nuestro controlador asignaremos 3 polos (debido a que nuestro sistema es ampliado), ubicados en $-1.5 \pm j$ y -1. Estos puntos generan una respuesta más rápida y que sigue mejor la referencia deseada, como podemos ver en la imagen.



Dando por resultado un sistema que tarda aproximadamente unos $5.5\ [s]$ en llegar a régimen permanente, con un caudal constante de $3\ [m^2/s]$.
Con esto damos por concluido nuestro controlador mediante asignación de polos.

Controlador mediante LQR:

Si bien el control mediante la asignación de polos, es bastante eficiente, para sistemas más complejos, no es tan sencillo determinar cuál es la mejor ubicación de los polos para obtener una respuesta deseada. Es por esto que se usa el LQR (Regulador Lineal Cuadrático en español), que nos permite determinar K, eligiendo qué tanto queremos que varíen nuestros valores de nuestro sistema. Para realizar esto nosotros trabajaremos nuevamente con las matrices ampliadas, y utilizaremos la función ***lqr*** presente en *octave* para realizar este cálculo.
Primero debemos determinar nuestras matrices **Q** y **R**.

La matriz Q se encarga de penalizar las desviaciones de nuestras variables de estado, mientras que la matriz R se encarga de penalizar la acción de control (Ideal cuando no queremos cambios bruscos o exagerados).
En nuestro caso, primero comenzaremos con todas las constantes iguales a 1, para a partir de ahí ir variando sus valores hasta obtener la respuesta deseada, es decir:

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

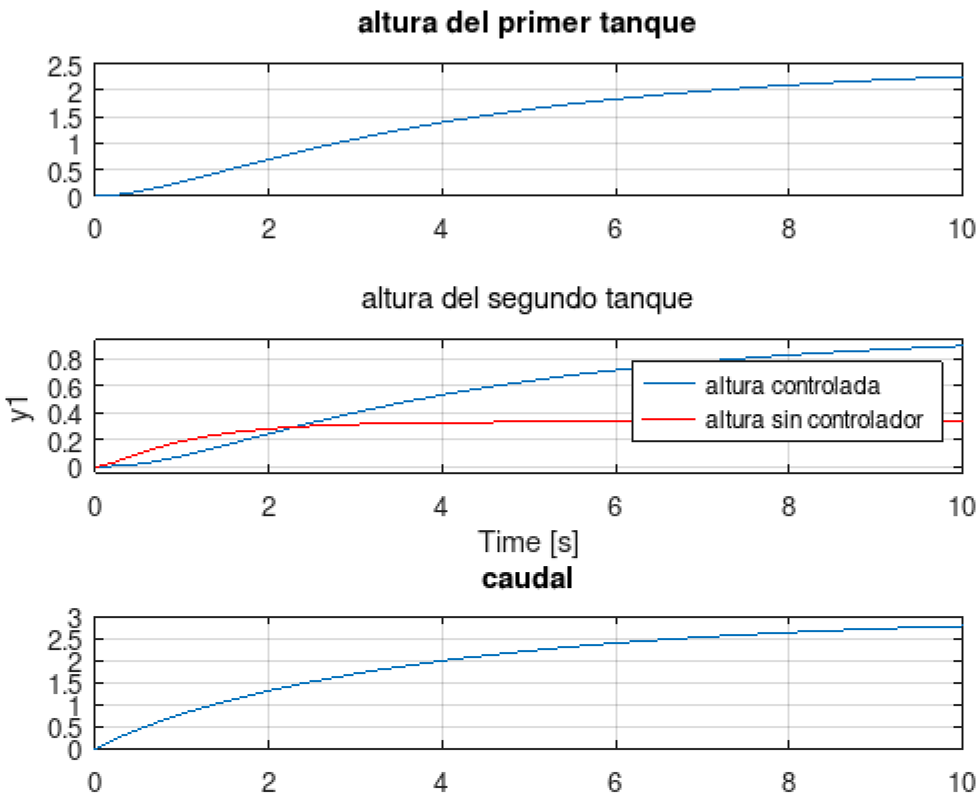
$$R = \begin{bmatrix} 1 \end{bmatrix}$$

Utilizando estas matrices, y las matrices ampliadas de nuestro sistema, obtenemos:

```
>> K=lqr(A_amp,B_amp,Q,R)
K =

   0.5847   0.4202  -1.0000
```

Luego, el proceso es exactamente el mismo que el método de asignación de polos, solamente que utilizando este K calculado. Esto da por resultado:



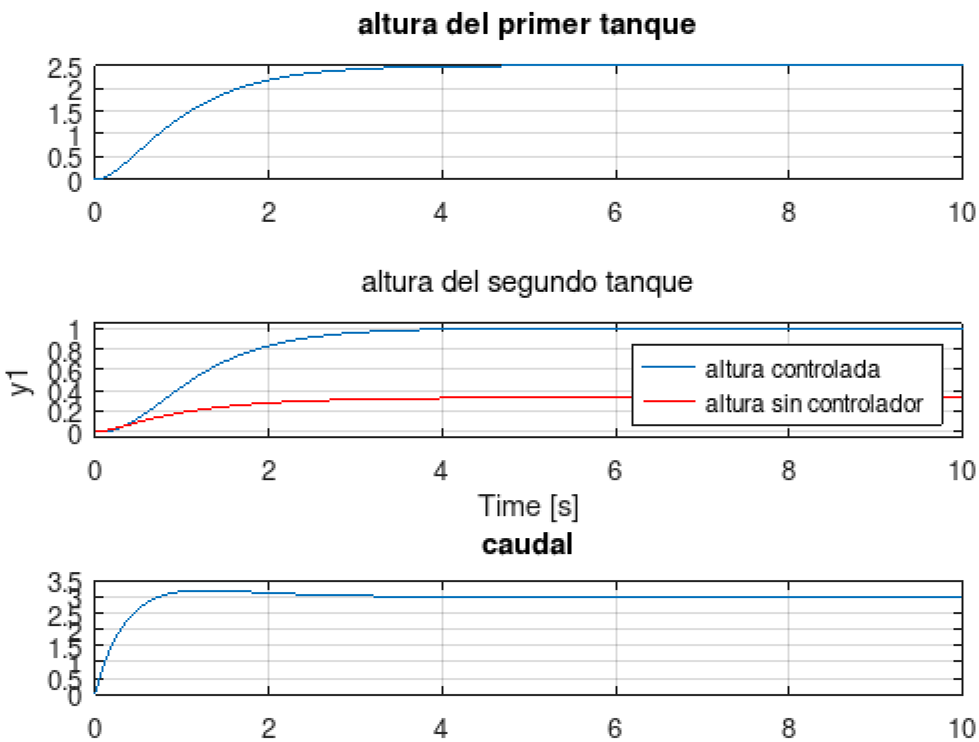
Vemos que la altura del segundo estanque parece querer seguir la referencia, solamente que lo hace muy lento.
Variando los valores de las matrices Q y R, concluimos que una buena respuesta se obtiene con:

$$Q = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 100 \end{bmatrix}$$

$$R = \begin{bmatrix} 1 \end{bmatrix}$$

La matriz Q lo que hace es que permite que h1 (posición 1x1) pueda variar más y también el error (posición 3x3). Haciendo que la variable que queremos controlar, que es h2 (posición 2x2), se mantenga lo más estable posible.

La matriz R no provee ninguna penalización a la entrada u.
Esto nos dá por resultado:



Vemos que el sistema es estable, rápido, y sin error en estado estacionario, siguiendo perfectamente la referencia de entrada.

Controlador mediante DLQR:

El **DLQR** o Regulador Lineal Cuadrático Discreto, es el equivalente al LQR pero en tiempo discreto. Ideal para situaciones en las cuales queremos evaluar la evolución de nuestro sistema en tiempo contínuo con un microcontrolador u otro sistema inteligente que genere la acción de control.
En este caso, el controlador se implementa de la misma forma que el LQR, solamente que se calcula diferente y cambia la forma en la que simulamos la evolución del sistema.

Lo primero que hacemos es discretizar nuestro sistema representado en espacio de estados, esto lo hacemos con la función de octave "*c2d*", como vemos a continuación:

```
sys_D=c2d(sys,Ts,'zoh');
Ad  = sys_D.a;
Bd  = sys_D.b;
Ctd = sys_D.c;
Dd  = sys_D.d;
```

De esta forma tenemos rápidamente discretizado nuestro sistema (*sys* en el código), siendo *Ts* nuestro periodo de muestreo. Este periodo de muestreo se calculó teniendo en cuenta una frecuencia de muestreo mayor al doble de la frecuencia más rápida del sistema (teniendo en cuenta los polos del sistema).

Luego que tenemos nuestro sistema en espacio de estados discretos hace falta calcular las matrices ampliadas de nuestro sistema, esto se hace siguiendo

$$Ad_{Amp} = \begin{bmatrix} Ad & 0 \\ -Cd{*}Ad & 1 \end{bmatrix}$$

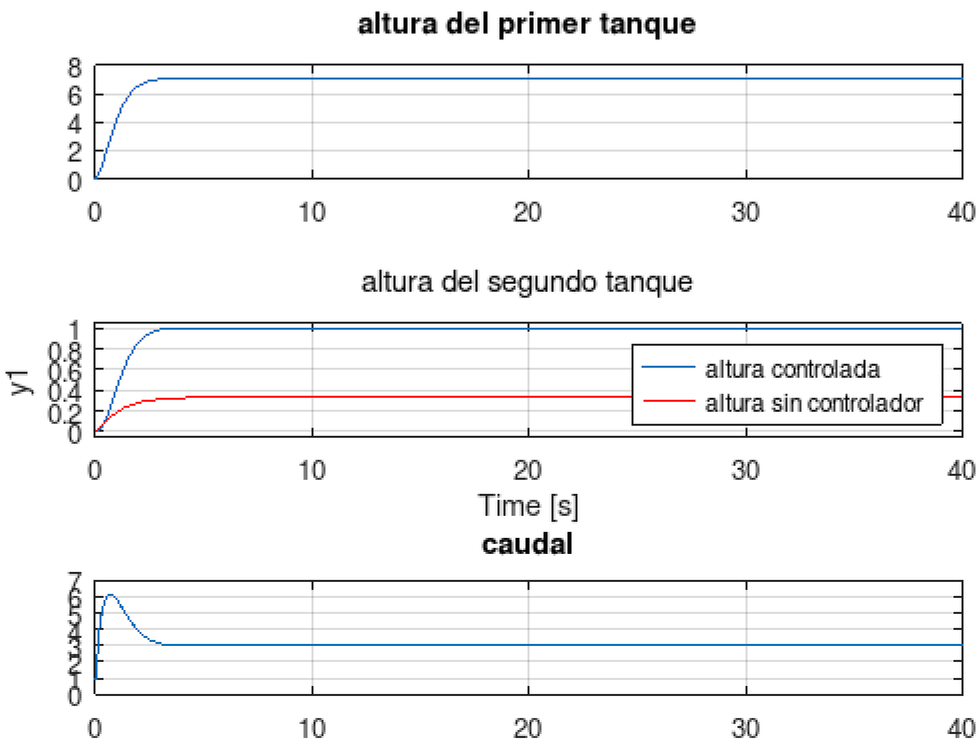
$$Bd_{Amp} = \begin{bmatrix} Bd \\ -Cd{*}Bd \end{bmatrix}$$

Luego debemos calcular las matrices Q y R para nuestro controlador con este sistema ampliado de matrices discretas. Para empezar haremos Q como la matriz identidad 3x3 y R un valor unitario.
Luego utilizaremos la función de octave *dlqr*, y obtendremos lo siguiente:

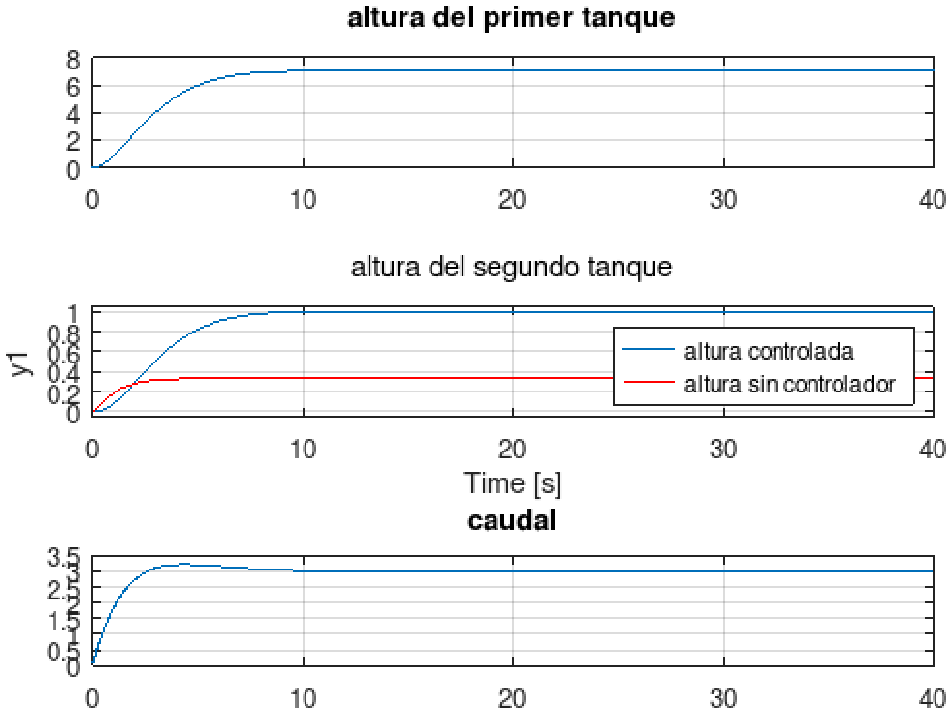
```
>> Kd=dlqr(A_ampd,B_ampd,Qd,Rd)
Kd =

   3.0750   4.7677  -0.9382
```

Con esto realizaremos nuestra simulación en tiempo discreto (Ver el código en el repositorio anexado), obteniendo el siguiente resultado.



Vemos que obtenemos una altura en el segundo estanque llega a rápidamente a la referencia deseada, sin embargo, la acción de control es algo excesiva ($6\ [m^3/s]$), por lo que haciendo más grande el valor de R, tendremos una acción de control no tan brusca, como podemos ver a continuación



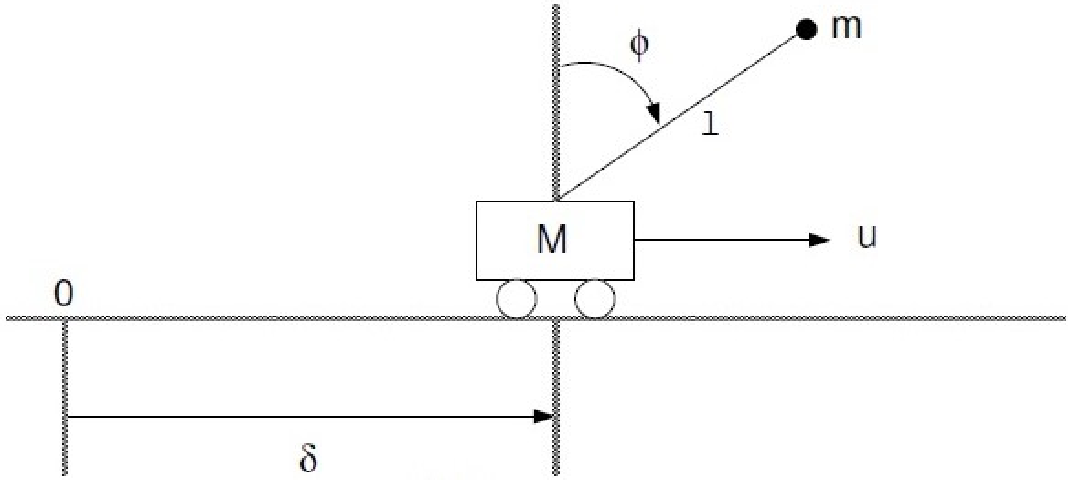
DLQR con R=100

La altura del segundo estanque sigue un poco más la referencia un poco más lento que el caso anterior, pero tenemos una acción de control de caudal más suave y con valores más realistas.

Con eso damos por concluido el controlador DLQR y finalizado el ejercicio 1.

Problema N°2: Péndulo Invertido

Se nos plantea el siguiente caso de estudio, donde debemos controlar el movimiento de un carro del que cuelga un péndulo invertido, y se desea que parta de una posición inicial a otra posición final, analizando desde qué posición inicial de ángulo distinto de cero se puede controlar.
El caso es el siguiente:



El sistema está representado por las siguientes matrices de estado:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\dfrac{F}{M} & -\dfrac{m*g}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{F}{l*M} & \dfrac{g(m+M)}{l*M} & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 1/M \\ 0 \\ -\dfrac{1}{l*M} \end{bmatrix}$$

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

$$D = 0$$

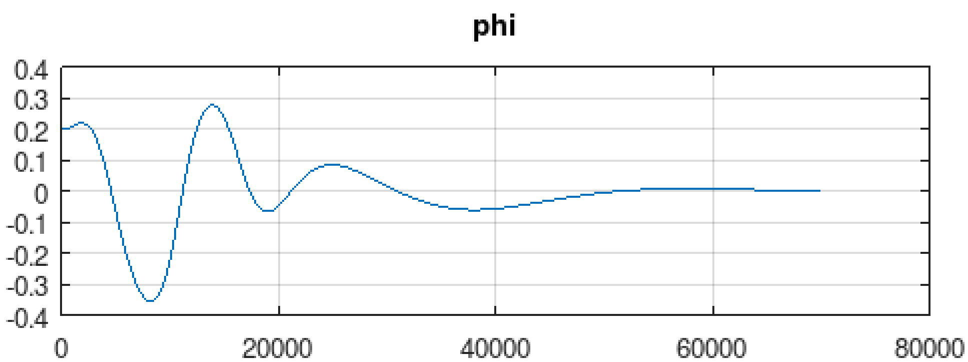
Para realizar la acción de control en este sistema es necesario crear un observador para controlar nuestras variables de estado.

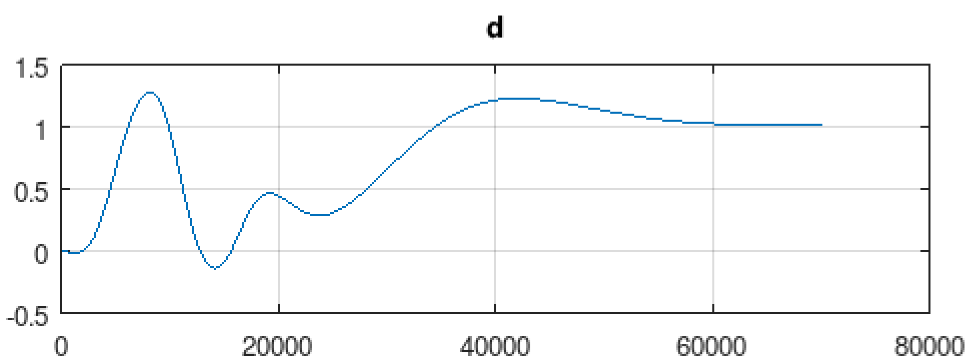
Se nos pide que realicemos dos casos de estudio, uno donde nuestro observador recibe solo una variable de salida, y otra donde nuestro observador recibe dos variables de salida.

Caso de estudio: Una variable de salida

Este caso es el representado por las matrices de estado anteriormente mostradas, donde tenemos una sola variable de salida de nuestro sistema. Lo que haremos es crear un observador y un *DLQR* para la acción de control.

Se nos pide que con este controlador y observador, determinemos desde qué condición inicial de ángulo $\phi \neq 0$ se puede controlar.
Mediante múltiples pruebas y cambios en los valores de muestreo de simulación, determinamos que para un a referencia de posición $d = 1$ y para un ángulo $\phi = 0.2\ [radianes]$, podemos obtener la siguiente respuesta estable.





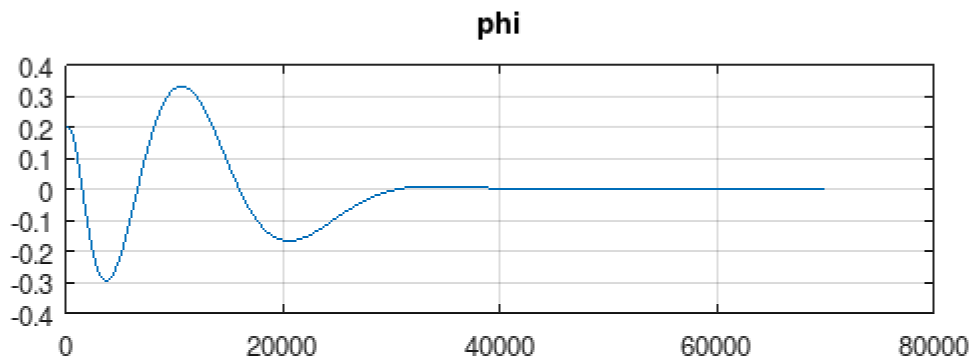
para ángulos $\phi > 0.2\ [radianes]$ el sistema se vuelve inestable.

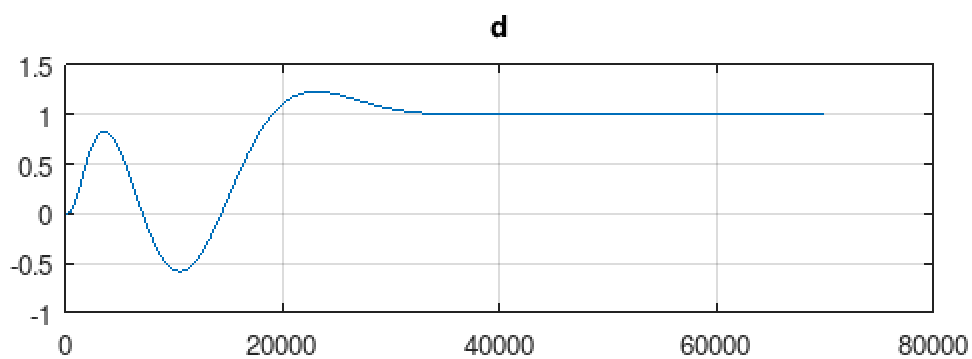
Caso de estudio: Dos variables de salida

En este caso cambiamos nuestra matriz de salida C por una de dos filas, donde obtendremos así dos variables de salida. Esto es beneficioso a la hora de control, debido a que nuestro observador posee más información de nuestro sistema original, por lo que puede controlarlo mejor.

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

Podemos observar que en nuestro caso no pudimos mejorar las condiciones iniciales desde las cuales el sistema puede estabilizarse en la referencia, sin embargo, nuestra respuesta si cambia para la misma condición inicial que el caso anterior





Obtenemos curvas más suaves y que se estabilizan más rápido que en el caso anterior, con lo cual podemos afirmar que este sistema responde mejor al observar más de una variable de salida del sistema.