ANEXOS

Anexo 1: Códigos empleados en MATLAB

A continuación se realiza una lista de los códigos más relevantes empleados en el software para la realización de este trabajo especial de grado:

display(' ')	Muestra en la pantalla de comando lo que se encuentre dentro de los símbolos ' '
x=input(' ');	Comando empleado para la introducción de un valor por medio del teclado, además tiene la misma función que display entre los ' '
Hold	La gráfica anterior a este comando queda almacenada mientras este comando este activo y permite la superposición de gráficos
zpk(T)	Expresa la función de trasformación (T), de forma que es fácil identificar la ganancia, los polos y los ceros.
Conv(a,b)	Multiplica las variables a y b, las cuales pueden ser polinomios, o matrices (ciertas condiciones se aplican)
poly(A)	Determina la ecuación característica de la Matriz A
eig(A)	Determina los autovalores y el auto vector de la matriz A
[n,d]=ord2(z,wn)	Almacena en las variables n y d, el numerador y el denominador respectivamente, a un función de transformación de 2do orden con un factor (z) de amortiguación y un (wn) de frecuencia
rlocus(T)	Gráfica el lugar geométrico de la función de transferencia T
sgrid(z,wn)	Realiza dentro de la gráfica del lugar geométrico, las líneas correspondientes a un coeficiente de amortiguación (z) y a una frecuencia (wn) Puede realizarse en un rango de coeficientes y de frecuencias.
[k,p]=rlocfind(T)	De forma gráfica, mediante el lugar geométrico al seleccionar un punto con el ratón se determina la ganancia (k) y el polo correspondiente
Bode(num,den)	Gráfica el diagrama de Bode para la función de transformación con numerador (num) y denominador (den)
k=place(A,B,p)	Permite la determinación de los coeficientes de realimentación (k), al insertar las matrices (A) y (B) del sistema, ubicando los polos en el vector (P)
[P,L,K]=care(A,B,Q,R)	Determina la matriz (P), por el método de Riccati, los autovalores en lazo cerrado (L), y los coeficientes de realimentación (K), para las matrices (A) y (B) del sistema, con las matrices (Q) y (R), de ponderación
B=series(T,k);	Almacena en la variable B, la multiplicación en serie de la función de trasformación T y K
SYS=feedback(B,H);	Almacena en la variable SYS, la realimentación de la función de trasformación B, con la función H referente al instrumento de medición
ltiview(SYS);	Ventana interactiva en donde se pueden obtener las característica de la respuesta, también se puede ingresar distintos tipos de entrada, entre otras.

Todas las palabras que estén luego de % representan comentarios para el programa

Anexo 2: Expresión algebraica de la ganancia integral para realimentación de variables.

A continuación, se muestra el desarrollo matemático para la determinación de las expresiones algebraicas para la determinación de la ganancia integral en la realimentación de variables de estado.

La expresión que se obtiene al realimentar variables de estado es la siguiente:

$$\begin{bmatrix} \dot{x} \\ \dot{x_n} \end{bmatrix} = \begin{bmatrix} (A - BK & B\,K_e \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r$$

$$y = [C\ 0\,] \begin{bmatrix} x \\ x_n \end{bmatrix}$$

En donde x, representa el vector de estado del modelo del sistema, x_n es una variable que se ha añadido para expresar la integral del error, K es la matriz de los del sistema, k_e es la ganancia de realimentación de variables de estado y las matrices A, B, C son las correspondientes en el modelado en variables de estado.

El diagrama de bloque que representa el sistema anteriormente descrito es el siguiente.

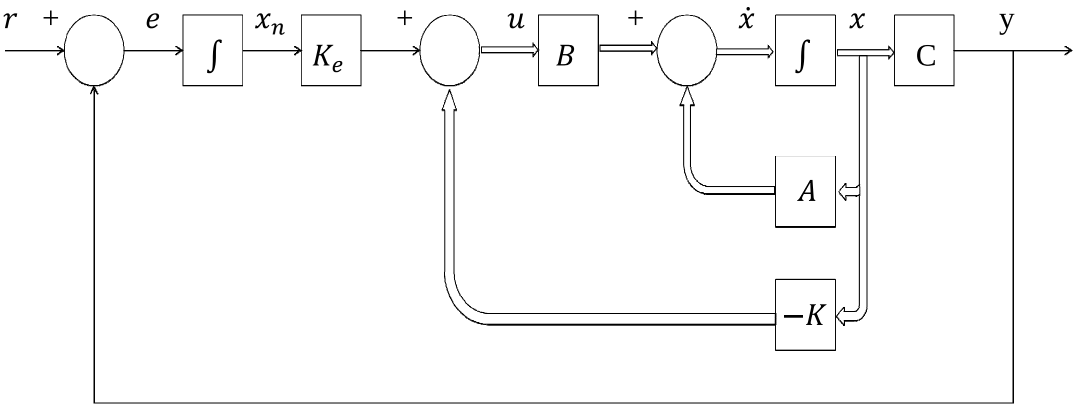


Diagrama de bloque de la representación general de realimentación de variables con una acción integral.

En el diagrama expresa que los coeficientes de realimentación de variables de estado (K), son una matriz de al menos dos filas, lo que significa que se está representando un sistema que puede tener múltiples entradas o salidas.

Como se ha mencionado a lo largo del trabajo especial de grado, las raíces de la ecuación características para el caso de variables de estado se determinan a partir de los auto valores de la matriz del proceso, como se está planteando un sistema con una acción integral, la matriz de este proceso sería:

$$A' = \begin{bmatrix} (A - BK & B\,K_e \\ -C & 0 \end{bmatrix}$$

Y como es conocido las raíces se determinan mediante. $\det(Is - A')$

Esto es para el caso de un sistema con una sola salida y una sola entrada, con las matrices:

$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}; B = \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix}; C^t = [c_{11} \quad c_{12}]\,; D = 0$, considerando que se realimentara ambas variables de estado por medio de $K_e = \begin{bmatrix} K_1 \\ K_2 \end{bmatrix}$

Se obtiene la expresión

$$\det(Is - A') = \begin{bmatrix} s - a_{11} + b_{11}K_1 & -a_{12} + b_{11}K_2 & -b_{11}K_e \\ -a_{21} + b_{21}K_1 & s - a_{22} + b_{21}K_2 & -b_{21}K_e \\ +c_{11} & +c_{12} & s \end{bmatrix}$$

En donde luego de realizar las correspondientes operaciones para agrupar términos y simplificarse obtiene:

$$
\begin{aligned}
s^3 + s^2(-a_{22} + b_{21}K_2 - a_{11} + b_{11}K_1) \\
+ s(K_e b_{21} C_{12} + a_{11}a_{22} - a_{11}b_{21}K_2 - b_{11}K_1 a_{22} - a_{21}a_{12} \\
- a_{21}b_{11}K_2 + b_{21}K_1 a_{12} + c_{11}K_e b_{11}) \\
+ (-a_{21}K_e b_{21}C_{12} + a_{21}C_{12}K_e b_{11} + C_{11}K_e b_{21}a_{12} - C_{11}K_e b_{11}a_{22})
\end{aligned}
$$

Ahora se debe partir de una ecuación deseada, en donde previamente se hayan calculado las raíces que cumplan con cierta especificación, por lo tanto se tiene una ecuación deseada de forma

$$s^2 + \beta_1 s + \beta_2 = 0$$

Ahora estas ecuaciones deben igualarse de acuerdo a los coeficientes de cada grado en termino de S, puede observarse también que el orden de la ecuación deseada es 2, mientras que la ecuación producto de la realimentación de variables de estada puede ser de orden superior a tres (de acuerdo al número de entradas y salidas el orden del sistema aumentara).

Por lo tanto se debe agregar un polo en la ecuación característica deseada que no intervenga con las especificaciones seleccionadas, es aquí en donde se hace uso de del criterio de los polos dominantes, es decir, las raíces que se añadirán deben estar por lo menos tres o cuatro veces a la izquierda de los polos dominantes, para asegurar que no se modifique las características deseadas.

Los polos no pueden seleccionarse demasiado alejados puesto que de hacerlo así, al realizar las operaciones correspondientes se requeriría de mayores decimales para obtener una buen resultado.

Finalmente la ecuación deseada con el polo añadido para este caso resulta de la siguiente forma:

$$s^3 + \alpha_1 s^2 + \alpha_2 s + \alpha_3 = 0$$

Igualando los coeficientes de las ecuaciones, agrupando términos convenientemente, sustituyendo y despejando se obtienen las siguientes expresiones de los coeficientes de realimentación para este caso planteado :

$$K_e = \frac{\alpha_3}{-a_{11}b_{21}C_{12} + a_{21}C_{21}b_{11} + C_{11}b_{21}a_{21} - C_{11}b_{11}a_{22}}$$

$$K_1 = -\frac{-\alpha_2 b_{21} - a_{21}a_{12}b_{21} + K_e b_{21}^2 c_{12} + k_e b_{21} c_{11} b_{11} - \alpha_1 a_{11} b_{21} + a_{21} \alpha_1 b_{11} + a_{21} b_{11} a_{22} - a_{11}^2 b_{21} + a_{21} a_{11} b_{11}}{-b_{21}b_{11}a_{22} + b_{21}^2 a_{12} + b_{11}a_{11}b_{21} - a_{21}b_{11}^2}$$

$$K_2 = \frac{(\alpha_1 + a_{22} + a_{11} - b_{11}K_1)}{b_{21}}$$

Y con estas expresiones se determinan los coeficientes de realimentación para un sistema SISO, si se procede a calcular los coeficientes de realimentación por otro método y luego se quiere emplear la expresión para determinar la ganancia de realimentación K_e, la misma solo puede tener una valor entre cercanos a 0 y 1, ya que de ser mayor a 1, de acuerdo al caso estudiado, el sistema se hace inestable.

Ahora para el caso de la aplicación para este trabajo especial de grado, cuyas matrices son:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}; B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}; C^T = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}; D = 0; K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$$

Para este caso es más complicado el algebra, pues se obtiene expresiones más complejas, las cuales resultan las siguientes, luego se realiza el mismo procedimiento anterior y se añaden dos polos a la ecuación característica deseada:

$s^4 \rightarrow 1 = 1$

$s^3 \rightarrow \mathrm{b}_{21}\mathrm{K}_{12} - \mathrm{a}_{22} - \mathrm{a}_{11} + \mathrm{b}_{12}\mathrm{K}_{21} + \mathrm{b}_{22}\mathrm{K}_{22} + \mathrm{b}_{12}\mathrm{K}_{11} = \alpha_1$

$$
\begin{aligned}
s^2 \rightarrow\ & b_{12}K_{21}b_{21}K_{12} - b_{21}K_{11}b_{12}K_{22} - b_{22}K_{21}b_{12}K_{12} + a_{21}b_{12}K_{12} \\
& + a_{21}b_{12}K_{22}C_{21}K_e b_{12} + b_{22}K_{21}a_{12} + C_{12}K_e b_{21} + C_{11}K_e b_{12} \\
& - b_{12}K_{11}a_{22} + a_{11}a_{22} - a_{21}a_{12} + C_{22}K_e b_{22} + b_{21}K_{11}a_{12} \\
& - b_{12}K_{21}a_{22} - a_{11}b_{21}K_{12} - a_{11}b_{22}K_{22} + b_{12}K_{11}b_{22}K_{22} = \alpha_2
\end{aligned}
$$

$$
\begin{aligned}
s \rightarrow\ & -a_{11}C_{12}K_e b_{21} - C_{11}K_e b_{21}b_{12}K_{22} + a_{21}C_{12}K_e b_{12} - a_{11}C_{22}K_e b_{22} \\
& - C_{21}K_e b_{22}b_{12}K_{12} - C_{21}K_e b_{12}a_{22} + b_{12}K_{21}C_{12}K_e b_{21} \\
& - b_{22}K_{21}C_{12}K_e b_{12} + C_{11}K_e b - 21\ a_{12} - b_{21}K_{11}C_{22}K_e b_{12} \\
& - C_{11}K_e b_{12}a_{22} + C_{11}K_e b_{12}b_{22}K_{22} + C_{21}K_e b_{21}b_{12}k_{12} \\
& + C_{21}K_e b_{21}b_{12}K_{12} + C_{21}K_e b_{22}a_{12} + a_{21}C_{22}K_e b_{12} + b_{12}K_{11}C_{22}K_e b_{22} \\
& = \alpha_3
\end{aligned}
$$

$$
\begin{aligned}
0 \rightarrow\ & C_{11}C_{22}K_e^2 b_{12}b_{22} + C_{21}C_{12}K_e^2 b_{21}b_{12} - C_{21}C_{12}K_e^2 b_{12}b_{22} - C_{11}C_{22}K_e^2 b_{21}b_{12} \\
& = \alpha_4
\end{aligned}
$$

Si se analiza minuciosamente las ecuaciones anteriores, se puede observar que se tiene cuatro ecuaciones y cinco incógnitas, es allí en donde se debe seleccionar alguna relación entre la matiz de realimentación o se puede seleccionar una ganancia integral convenientemente a partir de conocer los coeficientes por otro método, como por ejemplo mediante MATLAB.

En el caso de la aplicación se determinó por medio de MATLAB, los coeficientes de la matriz de realimentación, y se seleccionó convenientemente la ganancia integral que se acercara más a un máximo pico de 20%, en el menor tiempo de establecimiento posible.

Anexo 3: Expresión algebraica de los coeficientes de la matriz P para el método de Riccati

Partiendo del marco teórico y de los comentarios allí realizado, se procede a determinar algebraicamente las expresión es para determinar los coeficientes de la matriz P, utilizada para la solución de la ecuación de control óptimo.

Suponiendo que se tiene las siguientes matrices:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}; B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}; C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix}; D = 0; Q = \begin{bmatrix} q_1 & 0 \\ 0 & q_2 \end{bmatrix}$$

$$R = \begin{bmatrix} r_1 & 0 \\ 0 & r_2 \end{bmatrix} \; y \; P = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix}$$

Las cuales representan de forma algebraica las matrices del sistema modelado y se añaden las matrices Q y R de ponderación de las variables de estado.

La ecuación de Riccati, es la siguiente:

$$PA + A^t P + Q - PBR^{-1}B^T P = 0$$

A partir de la cual se obtiene la matriz P, que se debe sustituir en la ecuación que acontinuación se muestra para determinar los coeficientes de realimentación

$$K = R^{-1}B^T P$$

Ahora sustituyendo las matrices en la ecuación de Riccati, se deben agrupar términos y obtener cuatro ecuaciones escalares en donde se tiene tres incógnitas, es decir se tiene una ecuación que es redundante, las ecuaciones al realizar las operaciones correspondientes son las siguientes:

$$-\left(\frac{(p_{11}b_{11}+p_{12}b_{21})b_{11}}{r_1}+\frac{(p_{11}b_{12}+p_{12}b_{22})b_{12}}{r_2}\right)p_{11}$$
$$-\left(\frac{(p_{11}b_{11}+p_{12}b_{21})b_{21}}{r_1}+\frac{(p_{11}b_{12}+p_{12}b_{22})b_{22}}{r_2}\right)p_{21}+q_1$$
$$+2p_{11}a_{11}+a_{21}p_{21}+p_{12}a_{21}=0$$

$$-\left(\frac{(p_{11}b_{11}+p_{12}b_{21})b_{11}}{r_1}+\frac{(p_{11}b_{12}+p_{12}b_{22})b_{12}}{r_2}\right)p_{12}$$
$$-\left(\frac{(p_{11}b_{11}+p_{12}b_{21})b_{21}}{r_1}+(\frac{(p_{11}b_{12}+p_{12}b_{22})b_{22}}{r_2}\right)p_{22}+p_{11}a_{12}$$
$$+p_{12}a_{22}+a_{11}p_{12}+p_{22}a_{21}=0$$

$$-\left(\frac{(p_{12}b_{11}+p_{22}b_{21})b_{11}}{r_1}+\frac{(p_{12}b_{12}+p_{22}b_{22})b_{12}}{r_2}\right)p_{12}$$
$$-\left(\frac{(p_{12}b_{11}+p_{22}b_{11})b_{21}}{r_1}+\frac{(p_{12}b_{12}+p_{22}b_{22})b_{22}}{r_2}\right)p_{22}+q_2$$
$$+2p_{22}a_{22}+a_{12}p_{12}+p_{12}a_{12}=0$$

Ahora para obtener las expresiones algebraicas a partir de estas ecuaciones es muy complicado, por lo tanto se deben sustituir los valores para obtener expresiones que pueden ser fácilmente manipulables.

Por lo tanto se sustituyen los valores de las matrices del modelo a simular (mezclador):

$$A=\begin{bmatrix}-0{,}01 & 0\\ 0 & -0{,}02\end{bmatrix};B=\begin{bmatrix}1 & 1\\ -0{,}25 & 0{,}75\end{bmatrix};C=\begin{bmatrix}0{,}01 & 0\\ 0 & 1\end{bmatrix}$$

Obteniendo las siguientes ecuaciones:

$$-\left(\frac{p_{11}-0{,}25p_{12}}{r_1}+\frac{p_{11}+0{,}75p_{12}}{r_2}\right)p_{11}$$
$$-\left(\frac{-0{,}25\ (p_{11}-0{,}25p_{12})}{r_1}+\frac{0{,}75(p_{11}+0{,}75p_{12})}{r_2}\right)p_{12}+q_1$$
$$-0{,}02p_{11}=0$$

$$-\left(\frac{p_{11}-0{,}25p_{12}}{r_1}+\frac{p_{11}+0{,}75}{r_2}\right)p_{12}$$
$$-\left(\frac{-0{,}25\ p_{11}-0{,}25\ p_{12}}{r_1}+\frac{0{,}75(p_{11}+0{,}75\ p_{12})}{r_2}\right)p_{22}-0{,}03p_{12}$$
$$=0$$

$$-\left(\frac{p_{12}-0{,}25\ p_{22}}{r_1}+\frac{p_{12}+0{,}75p_{22}}{r_2}\right)p_{12}$$
$$-\left(\frac{-0{,}25(p_{12}-0{,}25\ p_{22})}{r_1}+\frac{0{,}75\ (p_{12}+0{,}75p_{22})}{r_2}\right)p_{22}-0{,}04p_{22}$$
$$+q_2=0$$

Ahora con estas ecuaciones se procede a sustituir los valores de las matrices de ponderación (Q y R) y obtener de esta manera ecuaciones con las cuales se pueden realizar las operaciones matemáticas necesarias, para obtener los coeficientes de la matriz P.

Finalmente se sustituyen estos valores para determinar los coeficientes de realimentación.

Anexo 4: Programación en MATLAB; Control Proporcional

Para el control proporcional de la concentración se emplea el siguiente código en MATLAB

```
%% Constantes
w1=0.015;                % Caudal de canal 1 m^3/s
w2=0.005;                % Caudal de canal 2 m^3/s
w0=0.02;                 % Caudal de salida m^3/s
c1= 1;                   % Concentración de canal 1 Kmol/m^3
c2=2;                    % Concentración de canal 2 Kmol/m^3
c0=1.25;                 % Concentración de Salida Kmol/s
v0=1;                    % Volumen del tanque en m^3

tao=v0/w0;               % Constante de Retención
kx1=(c1-c0)/w0;          % Constante de Linealización  1
kx2=(c2-c0)/w0;          % Constante de Linealización  2
num=(-kx1+kx2);

%Función de transferencia de la planta
den=[tao  1 ];
T=tf(num,den);            % Función de transferencia en lazo abierto
rlocus(num,den);
z=0:0.1:1;
 wn=10;
 sgrid(z,wn);
display('seleccionar un punto en el grafico para seleccionar la
ganancia'  );
x=1;
while (x==1)
    [k,p]=rlocfind(T)
    display(' Comprobar otro valor');
    display('la respuesta debe ser binaria)=  ');
    x=input('  ');

end

%Acción del controlador a la función de transferencia
B=series(T,k);

%realimentación con H=1
SYS=feedback(B,1);

%señal de tipo escalón
ltiview(SYS);
```

Para el caso de la ecuación de caudal, solo se cambia las ecuaciones de la planta.

Anexo 5: Programación en MATLAB; Control Integral

A continuación, se escribe el código de programación para la ecuación de concentración del modelo simulado.

```
clear all;
close all;
%% Constantes
w1=0.015;                % Caudal de canal 1 m^3/s
w2=0.005;                % Caudal de canal 2 m^3/s
w0=0.02;                 % Caudal de salida m^3/s
c1= 1;                   % Concentración de canal 1 Kmol/m^3
c2=2;                    % Concentración de canal 2 Kmol/m^3
c0=1.25;                 % Concentración de Salida Kmol/s
v0=1;                    % Volumen del tanque en m^3

tao=v0/w0;               % Constante de Retención
kx1=(c1-c0)/w0;          % Constante de Linealización  1
kx2=(c2-c0)/w0;          % Constante de Linealización  2
num=(-kx1+kx2);
%% función de transferencia de la planta

den=[tao 1];
T=tf(num,den);           % Función de transferencia en lazo abierto
rlocus(num,den);         % lugar geométrico de las raíces
  z=0:0.1:1;
  wn=100;
 sgrid(z,wn);
%% Procedimiento iterativo para la selección del cero
% selección del valor del cero
 x=0
 display(' Seleccionar la ubicación del cero y la ganancia');
 while (x==0)
     a=input(' ubicación de cero=  ');
  %control PI)
      A=[1 -a];
      B=[1 0];
      TI=tf(A,B);           % Función de transferencia (PI)
      C=series(T,TI);       % Compensación en serie
 % grafica del lugar geométrico de las raíces con la adición del
% cero  y el polo en el origen de coordenadas
      rlocus(C);
      z=0:0.1:1;
      wn=10;
      sgrid(z,wn);
      k=input(' ganancia=  ');
      display(' Confirmación...');
```

```
    x=input(' La respuesta debe ser binario (1=si, 0=no) ');
end
% Fin del proceso iterativo
%% Planteamiento de la realimentación

D=series(C,k);              % Adición de la ganancia al sistema
display(' valor de la ganancia integral ');
ki=-a*k

SYS=feedback(D,1);
ltiview(SYS)
```

Para el caso de la ecuación de caudal se cambia el planteamiento del sistema por el siguiente:

```
%% función de transferencia de la planta
num=(c2-c1);
den=(c2-c0);
T=tf(num,den);            % Función de transferencia en lazo abierto
rlocus(num,den);          % grafica del lugar geométrico de las
raíces
```

Los demás códigos para la selección de la ganancia y de la ubicación del cero permanecen igual.

Anexo 6: Códigos de programación en MATLAB; Control derivativo

Para la ecuación de concentración se utilizaron los siguientes códigos de programación:

```
clear all;
close all;

%% Constantes

w1=0.015;                % Caudal de canal 1 m^3/s
w2=0.005;                % Caudal de canal 2 m^3/s
w0=0.02;                 % Caudal de salida m^3/s
c1= 1;                   % Concentración de canal 1 Kmol/m^3
c2=2;                    % Concentración de canal 2 Kmol/m^3
c0=1.25;                 % Concentración de Salida Kmol/s
v0=1;                    % Volumen del tanque en m^3

tao=v0/w0;               % Constante de Retención
kx1=(c1-c0)/w0;          % Constante de Linealización  1
kx2=(c2-c0)/w0;          % Constante de Linealización  2
num=(-kx1+kx2);

%% Función de transformación de planta
  den=[tao 1];
  T=tf(num,den);          % función de trasferencia en lazo abierto
  rlocus(T);
  z=0:0.1:1;
  wn=10;
  sgrid(z,wn);
%% Procedimiento iterativo para la selección del cero
% selección de la ubicación  del cero
 x=0
 display(' Seleccionar la ubicación del cero y la ganancia');
 while (x==0)
     a=input(' ubicación de cero=  ');
     %control PD
     A=[(1/-a) 1];
     B=[1];
     TD=tf(A,B);           % Función de trasferencia del controlador
     C=series(T,TD);       % Compensación en serie

 % grafica del lugar geométrico de las raíces con la adición del
%cero
      rlocus(C);
      z=0:0.1:1;
      wn=10;
```

```
    sgrid(z,wn);
    k=input(' ganancia=  ');
    display(' Confirmación...');
    x=input(' La respuesta debe ser binario (1=si, 0=no) ');
end
% fin del procedimiento iterativo

D=series(C,k);
display(' valor de la ganancia derivativa ');
kd=k/-a
SYS=feedback(D,1);     % realimentación con H=1
ltiview(SYS)
```

Para el caso de la ecuación de caudal, como se ha mencionado anteriormente no se le puede aplicar esta estrategia.

Anexo 7: Código de programación MATLAB: Control PID

Los siguientes códigos son los empleados para la ecuación de concentración:

```
clear all
close all
%% Constantes

w1=0.015;                % Caudal de canal 1 m^3/s
w2=0.005;                % Caudal de canal 2 m^3/s
w0=0.02;                 % Caudal de salida m^3/s
c1= 1;                   % Concentración de canal 1 Kmol/m^3
c2=2;                    % Concentración de canal 2 Kmol/m^3
c0=1.25;                 % Concentración de Salida Kmol/s
v0=1;                    % Volumen del tanque en m^3

tao=v0/w0;               % Constante de Retención
kx1=(c1-c0)/w0;          % Constante de Linealización  1
kx2=(c2-c0)/w0;          % Constante de Linealización  2
num=(-kx1+kx2);

%% Función de transferencia de la planta
 den=[tao 1];
 T=tf(num,den);          % función de trasferencia en lazo abierto
 rlocus(T);
 z=0:0.1:1;
 wn=10;
 sgrid(z,wn);            % Grafica en el rlocus las líneas de z y wn
 hold

%% Procedimiento iterativo para la selección de los ceros
display('  Selección de los ceros de los controles derivativos y
proporcional')

% Control derivativo

   x=0;
       while  (x==0)
       a=input(' Ubicación del cero en PD =  ' );
       A=[(1/-a) 1];               % Ubicación del cero derivativo
       B=[1];
       TD=tf(A,B);                 % Función de trasferencia para PD
       C=series(TD,T);             % Multiplicación del PD con la Gp
       rlocus(C);
       sgrid(z,wn);
       display('Confirmación..');
       x=input(' si=1; no=0  ');   % revisión de lugar geométrico
       hold
    end
```

```
% Control integral
   y=0;
   while (y==0)

       b=input(' Ubicación del cero en PI =  ');
       D=[1 -b];                  % Ubicación del cero del C integral
       E=[1 0];
       TI=tf(D,E);                % acción integral
       F=series(TI,C);            % compensación en serie
       rlocus(F);
       sgrid(z,wn);
       display('Selección de la ganancia');
       k=input('Valor de la ganancia=  ');
       display(' Confirmación');
       y=input(' si=1; no=0  ');
   end

% Acción de los controladores a la planta

G=series(F,k)
display(' valor de la ganancia derivativa ');
kd=k/-a
display(' valor de la ganancia integral ');
ki=-b*k

%realimentación con H=1

SYS=feedback(G,1);
ltiview(SYS);
```

Para el caso de la ecuación de caudal, se cambia la función de transformación de la planta por las siguientes líneas de programación:

```
%% Función de transferencia de la planta
num=(c2-c1);
den=(c2-c0);
T=tf(num,den);             % Función de transferencia en lazo abierto
rlocus(num,den);           % grafica del lugar geométrico
```

Anexo 8: Código de programación en MATLAB: Control LAG o LEAD

A continuación se presentan los códigos de programación para el control en adelanto o en atraso para la ecuación de concentración. Para este caso en partícula, de acuerdo a la selección del polo o el cero el control es en adelanto o en atraso.

```
clear all; close all;
%% Constantes
w1=0.015;                % Caudal de canal 1 m^3/s
w2=0.005;                % Caudal de canal 2 m^3/s
w0=0.02;                 % Caudal de salida m^3/s
c1= 1;                   % Concentración de canal 1 Kmol/m^3
c2=2;                    % Concentración de canal 2 Kmol/m^3
c0=1.25;                 % Concentración de Salida Kmol/s
v0=1;                    % Volumen del tanque en m^3

tao=v0/w0;               % Constante de Retención
kx1=(c1-c0)/w0;          % Constante de Linealización  1
kx2=(c2-c0)/w0;          % Constante de Linealización  2
num=(-kx1+kx2);

%% Función de transferencia de la planta
den=[ tao 1];
T=tf(num,den);           % Función de trasferencia en lazo abierto
figure(1);
grid
bode(num,den);
hold on
figure(2);
rlocus(num,den);         % lugar geométrico de las raíces
 z=0:0.1:1;
 wn=100;
 sgrid(z,wn);
 grid
 hold on

%% Acción del controlador a la función de transferencia
% selección del Cero y el polo
x=0;
 display(' Seleccionar la ubicación del cero, el polo y la
ganancia');
   while (x==0)
     zc=input(' ubicación de cero=  ');
     pc=input(' ubicación del polo= ');
     A=[1 -zc];
     B=[1 -pc];
```

```
        if abs(pc)>abs(zc)
             display('Compensador en adelanto')
        end
        if  abs(pc)<abs(zc)
             display('Compensador en atraso')
        end
     TC=tf(A,B);             % Controlador
     C=series(TC,T);
     figure(2);
     rlocus(C);
     z=0:0.1:1;
     wn=100;
     figure(1);
     bode(C);
     display(' Confirmación...');
     x=input(' La respuesta debe ser binario (1=si, 0=no) ');
    end

y=0;
 while (y==0)
     display(' Seleccionar la ganancia');
     k=input(' Ganancia  ');
     D=series(k,C);
     figure(1);
     bode(D);
     y=input(' Confirmación s1=1, no=0   ');
 end

%% Realimentación

SYS=feedback(D,1);
ltiview(SYS);
```

De igual manera para la ecuación de caudal, se modifica la función de transformación de planta por:

```
%% Función transformación de la planta
num=(c2-c1);
den=(c2-c0);
T=tf(num,den);            % Función de transferencia en lazo abierto
```

Anexo 9: Código de programación en MATLAB: Control LAG-LEAD

Las siguientes líneas de programación son para la ecuación de concentración, para la estrategia de compensación de atraso y adelanto.

```
clear all ;
close all;

%% Constantes
w1=0.015;                 % Caudal de canal 1 m^3/s
w2=0.005;                 % Caudal de canal 2 m^3/s
w0=0.02;                  % Caudal de salida m^3/s
c1= 1;                    % Concentración de canal 1 Kmol/m^3
c2=2;                     % Concentración de canal 2 Kmol/m^3
c0=1.25;                  % Concentración de Salida Kmol/s
v0=1;                     % Volumen del tanque en m^3

tao=v0/w0;                % Constante de Retención
kx1=(c1-c0)/w0;           % Constante de Linealización  1
kx2=(c2-c0)/w0;           % Constante de Linealización  2
num=(-kx1+kx2);

%% Función de transferencia de la planta
den=[ tao 1];
T=tf(num,den);           % Función de trasferencia en lazo abierto
figure(1);
grid
bode(num,den);
hold on
figure(2);
rlocus(num,den);          % lugar geométrico de las raíces
 z=0:0.1:1;
 wn=100;
 sgrid(z,wn);
 grid
 hold on

%% Acción del controlador a la función de transferencia
% selección del Cero y el polo para el adelanto

x=0;
 display('      Adelanto');
 display(' Seleccionar la ubicación del cero y el polo (el polo debe
ir a la izquierda)');
   while (x==0)
     zc=input(' ubicación de cero=  ');
     pc=input(' ubicación del polo= ');
```

```
      A=[1 -zc];
      B=[1 -pc];

      TCLE=tf(A,B);        % Función de trasferencia del controlador
      C=series(TCLE,T);
      figure(2);
      rlocus(C);
      z=0:0.1:1;
      wn=100;
      figure(1);
      bode(C);
      display(' Confirmación...');
      x=input(' La respuesta debe ser binario (1=si, 0=no) ');
    end

     z=0;
 display('      Atraso');
 display(' Seleccionar la ubicación del cero y el polo (el polo debe
ir a la Derecha)');

  while (z==0)
      zc2=input(' ubicación de cero=  ');
      pc2=input(' ubicación del polo= ');
      A2=[1 -zc2];
      B2=[1 -pc2];

      TCLA=tf(A2,B2);      % Función de trasferencia del controlador
      C2=series(TCLA,C);
      figure(2);
      rlocus(C2);
      z=0:0.1:1;
      wn=100;
      figure(1);
      bode(C2);
      display(' Confirmación...');
      z=input(' La respuesta debe ser binario (1=si, 0=no) ');
     end

y=0;
 while (y==0)
      display(' Seleccionar la ganancia');
      k=input(' Ganancia  ');
      D=series(k,C2);
      figure(1);
      bode(D);
      y=input(' Confirmación s1=1, no=0   ');
     end


%% Realimentación he implementación del control
%realimentación con H=1
```

```
SYS=feedback(D,1);
ltiview(SYS);
```

Y para la ecuación de transformación del caudal se cambia la función de transformación por:

```
%% Función de transferencia de la planta
num=(c2-c1);
den=(c2-c0);
T=tf(num,den);
```

Anexo 10: Códigos de programación realimentación de variables por asignación de polos

A través de los siguientes códigos de programación, fue posible el estudio de la realimentación de variables de estado por el método de asignación de polos.

```
%% Constantes
os=20;                   % Porcentaje de sobre respuesta
ts=20;                   % Tiempo de establecimiento
w1=0.015;                % Caudal de canal 1 m^3/s
w2=0.005;                % Caudal de canal 2 m^3/s
w0=0.02;                 % Caudal de salida m^3/s
c1= 1;                   % Concentración de canal 1 Kmol/m^3
c2=2;                    % Concentración de canal 2 Kmol/m^3
c0=1.25;                 % Concentración de Salida Kmol/s
v0=1;                    % Volumen del tanque en m^3

teta=v0/w0;

%% Sistema en variables de estado
a=[(-1/(2*teta)) 0; 0 (-1/teta)];
b=[1 1 ; ((c1-c0)/v0) ((c2-c0)/v0)];
c=[(1/(2*teta)) 0; 0 1]
d=[0 0;0 0];

%% funciones

z=(-log(os/100))/(sqrt((pi^2)+(log(os/100)^2)));  % cálculo del
%factor de amortiguación
wn=4 /(z*ts);                           % frecuencia
[num2,den2]=ord2(wn,z);                 % Ecuación del sistema deseado
display(' Polos del sistema deseado');
p=roots(den2)
k=place(a,b,p)                       % Generador de matriz de ganancia
A=a-(b*k);                           % Matriz del sistema modificada

e=input(' Polo (sin interferencia en los polos dominantes)=  ');
den1=conv([1 -e],[1 -e]);     % polo para la igualdad de ecuaciones
display('Ecuación característica del sistema= ');
den3=conv(den1,den2)    % Ecu característica para el uso de la
% ecuación de realimentación anexo 2)
Ke=input('Ganancia Integral=  ');  % Se determina mediante anexo 3
B=b*Ke;
A2=[A(1) A(3) B(1) B(3); A(2) A(4) B(2) B(4) ;-c(1) -c(3) 0 0; -c(2)
-c(4) 0];          % Matriz del sistema realimentado con integración
```

```
B2=[0; 0; 1 ;1];
C=[ 1 0 0 0; 0 1 0 0];
D=0;

sys=ss(A2,B2,C,D);

ltiview(sys);
```

De acuerdo a la selección de la ganancia integral, se obtienen diferentes respuestas, esta ganancia debe estar en un rango aproximado de 0 a 1, ya que de ser un valor muy elevado a este rango, el sistema se hace inestable.

Anexo 11: Códigos de MATLAB: Control óptimo

A continuación se muestran los códigos empleados en el software para la realimentación de variables por el método del control óptimo, como se ha mencionado anteriormente, las ponderaciones de las matrices de rendimiento y de costo, se fueron variando de acuerdo al caso:

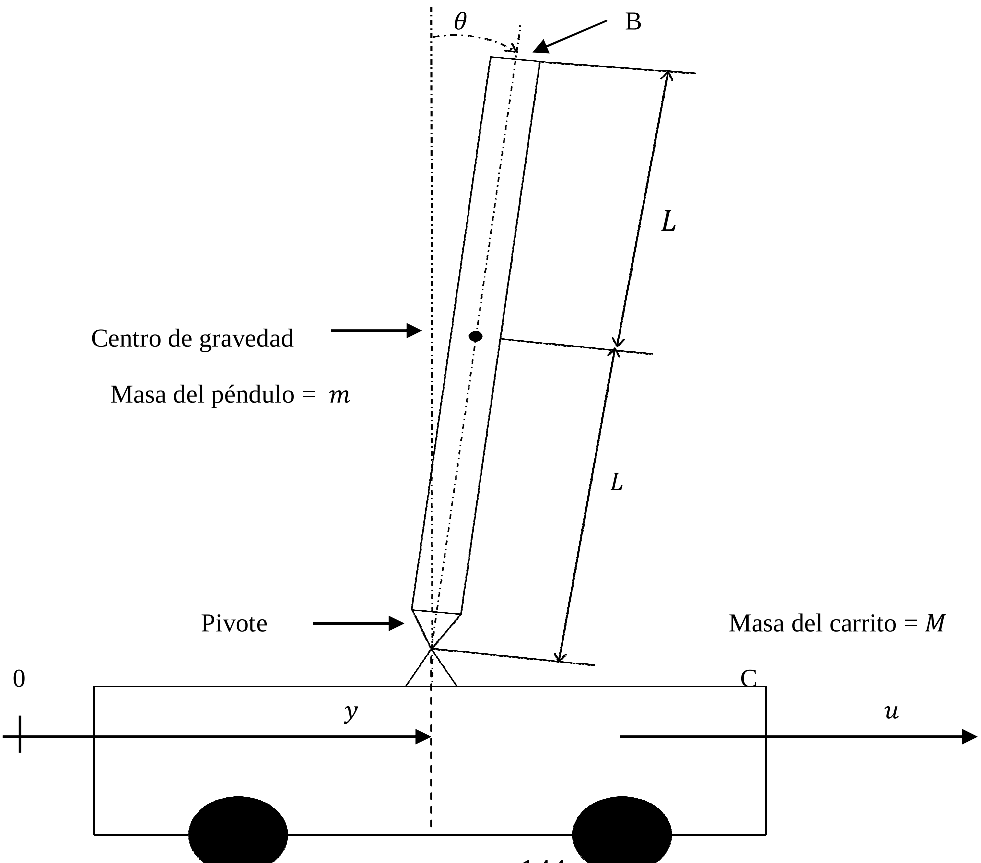
```
%% Constantes

w1=0.015;                % Caudal de canal 1 m^3/s
w2=0.005;                % Caudal de canal 2 m^3/s
w0=0.02;                 % Caudal de salida m^3/s
c1= 1;                   % Concentracion de canal 1 Kmol/m^3
c2=2;                    % Concentracion de canal 2 Kmol/m^3
c0=1.25;                 % Concentracion de Salida Kmol/s
v0=1;                    % Volumen del tanque en m^3
teta=v0/w0;

%% Sistema
a=[(-1/(2*teta)) 0; 0 (-1/teta)];
b=[1 1 ; ((c1-c0)/v0) ((c2-c0)/v0)];
c=[(1/(2*teta)) 0; 0 1];
d=[0 0;0 0];

%% Riccati
Q=[1 0; 0 2];             % Matriz de Rendimiento
R=[2 0; 0 1];             % matriz de  Costo

[x,l,g]=care(a,b,Q,R)     % Sol de la ecu de Riccati
k=g;
A=(a-b*k);

Ke=input('Ganancia Integral=  ');

B=b*Ke;

A2=[A(1) A(3) B(1) B(3); A(2) A(4) B(2) B(4) ;-c(1) -c(3) 0 0; -c(2)
-c(4) 0 0];
B2=[0; 0; 1 ;1];
C=[ 1 0 0 0; 0 1 0 0];
D=0;

sys=ss(A2,B2,C,D);
ltiview(sys);
```

Anexo 12: Péndulo invertido

La figura mostrada, representa el esquema de un péndulo invertido (B), el cual rota sobre el eje perpendicular al plano del papel en el punto de pivote, por medio del movimiento del carrito (C) solo sobre el eje horizontal. Esto es similar a controlar un bastón con la punta del dedo.

Se ha restringido el estudio del modelo a una dirección del carrito y solo al ángulo θ, pues el objetivo principal es la ilustración de la realimentación de variables de estado y sus beneficios.

La idea es equilibrar el péndulo sobre el eje vertical, mediante la aplicación de la fuerza u, la posición horizontal del carrito con respecto al sistema de coordenada horizontal está determinada por y.



Construcción del modelo

Para realizar el modelado del sistema, se realiza un análisis de fuerzas y momentos de forma adecuada, mediante la realización de los diagramas de cuerpo libre de los cuerpos (B) y (C),

Donde resalta la iteración en el pivote, la cual se denominará H(reacción horizontal) y V (reacción vertical), también es importante determinar la posición del sistema mediante el sistema de coordenadas. Se sugiere consultar la bibliografía (Elgerd), en donde se detallan las fuerzas involucradas en el sistema

Finalmente se obtienen las siguientes ecuaciones del modelado del sistema:

Posición del horizontal del carrito= y

Posición horizontal del péndulo, (según el centro de masa)= $y + L\ sen(\theta)$

Posición vertical del péndulo, (según el centro de masa)= $L \cos{(\theta)}$

$$\curvearrowright \quad \sum M_A \qquad I\ \frac{d^2\theta}{dt^2} = V * L * sen(\theta) - H * L * \cos{(\theta)} \qquad [1]$$

$$\uparrow \quad \sum F^B{}_{Ver} \qquad V - mg = m\frac{d^2}{dt^2}(L * \cos(\theta)) \qquad [2]$$

$$\longrightarrow \quad \sum F^B{}_{Hrz} \qquad H = m\frac{d^2}{dt^2}(y + L * sen(\theta)) \qquad [3]$$

$$\longrightarrow \quad \sum F^C{}_{Hrz} \qquad u - H = M\frac{d^2y}{dt^2} \qquad [4]$$

Como se ha mencionado H y V, representan las reacciones del pivote en sentido horizontal y vertical, respectivamente. I Representan la inercia del péndulo con respecto al punto A (ubicación del pivote), el cual por representa mediante la siguiente expresión $I = \frac{1}{3}\ m * L^2$

Como puede apreciarse el sistema no es lineal, por lo tanto debe linealizarse para obtener expresiones más fáciles de manejar, por ello se toma como punto de estudio el ángulo $\theta = 0$, y se realizaran pequeñas variaciones alrededor de él, por lo tanto se pueden realizar las siguientes aproximaciones:

$$sen(\theta) \approx \theta \qquad [5]$$

$$\cos(\theta) \approx 1 \qquad [6]$$

Por lo ello las ecuaciones de trabajo son las siguientes:

$$I\ddot{\theta} \approx VL\theta - HL \qquad [7]$$

$$V - mg \approx 0 \qquad [8]$$

$$H \approx m\ddot{y} + mL\ddot{\theta} \qquad [9]$$

$$u - H \approx M\ddot{y} \qquad [10]$$

De las cuatro ecuaciones anteriores, mediante algebra se puede obtener dos expresiones en donde se eliminan las reacciones H y V:

$$(I + mL^2)\ddot{\theta} + mL\ddot{y} - mgL\theta \approx 0 \qquad [11]$$

$$mL\ddot{\theta} + (m + M)\ddot{y} = 0 \qquad [12]$$

Ahora como se desea expresar en forma de variable de estado, se deben encontrar las expresiones en donde se expresen las derivadas más altas de cada ecuación, es decir de debe despejar la aceleración angular del péndulo y la aceleración del carrito.

Esto es:

$$\ddot{\theta} = \frac{g(m+M)mL}{I(m+M)+mML^2}\ \theta - \frac{mL}{I(m+M)+mML^2}\ u \qquad [13]$$

$$\ddot{y} = \frac{I+mL^2}{I(m+M)+mML^2}\ u - \frac{gm^2L^2}{I(m+M)+mML^2}\ \ \theta \qquad [14]$$

Por lo tanto se tiene un sistema de 4to orden, en donde se tienen como variables de estado $x_1 = \theta; x_2 = \dot{\theta}; x_3 = y;\ x_4 = \dot{y}$.

Por lo tanto las matrices del sistema son las siguientes:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \dfrac{g(m+M)\ mL}{\mathrm{I} + mML^2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -\dfrac{gm^2\ L^2}{\mathrm{I} + mML^2} & 0 & 0 & 0 \end{bmatrix} \qquad [15]$$

$$B = \begin{bmatrix} 0 \\ -\dfrac{mL}{\mathrm{I} + mML^2} \\ 0 \\ \dfrac{\mathrm{I} + mL^2}{\mathrm{I} + mML^2} \end{bmatrix} \qquad [16]$$

$$C^T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad [17]$$

Con las matrices anteriores se puede simular mediante MATLAB, el sistema, suponiendo los valores de masas y longitud del carrito y el péndulo.

El código de programación será el siguiente:

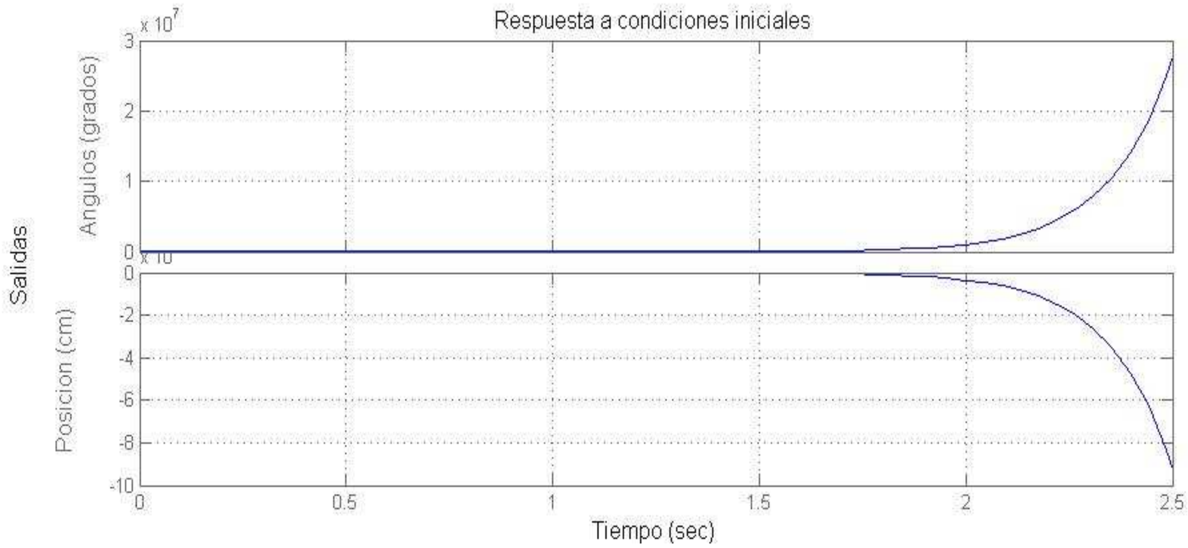
```
%% Planteamiento del sistema
% Constantes del sistema
m=0.2;                                    % masa del péndulo (Kg)
M=1;                                      % masa del carrito  (Kg)
L=0.2;                                    % longitud del péndulo (m)
g=9.8;                                    % Gravedad (m/s^2)
I=(1/3)*m*L*L;                            % Inercia del péndulo

% Matrices del sistema

A=[ 0 1 0 0; ((g*m*L*(m+M)/(I*(m+M)+(m*M*L*L)))) 0 0 0; 0 0 0 1; ((-
g*m*m*L*L)/(I*(m+M)+m*M*L*L)) 0 0 0];
B=[ 0; ((-m*L)/(I*(m+M)+m*M*L*L)); 0;
((I+(m*L*L))/(I*(m+M)+m*M*L*L))];
C=[ 1 0 0 0; 0 0 1 0];
D=0;

%%
display(' Raíces del sistema (autovalores de la matriz A)');
eig(A)                              % Auto valores de A

%% Realimentación de variables de estado
% localización de los polos
p=[-1-2.5i -1+2.5i  -3 -4];        % Polos del sistema
display(' Ganancias del realimentación');
k=place(A,B,p)                      % Coeficientes de realimentación

% Matriz realimentada
a=A-B*k;

%% Sistema realimentado

SYS2=ss(a,B,C,D);                      % Modelo de estado
ltiview(SYS2);                         % Visualizador de respuestas
```

Como se han mencionado anteriormente, las palabras que estén luego del símbolo (%) representan comentarios que no influyen en los códigos de programación, y el símbolo (%%), representa un bloque de programación para el programa.

Mediante el comando `(eig(A)),` se aprecian los siguientes valores:

```
eig(A)
ans =
         0
         0
    6.4807
   -6.4807
```

Los cuales representan los auto valores del sistema y por lo tanto las raíces del polinomio característico. Puede apreciarse que el sistema es inestable debido a la raíz positiva lo que hace que el sistema no sea controlable por medio de las estrategias clásicas de control.



En la figura anterior se representan las respuestas del ángulo del péndulo (superior) y la posición del carrito (inferior), a la condición inicial de un ángulo de cinco grados en el sistema sin realimentación que se puede observar como las respuestas con inestables.

El vector p, representa la ubicación de las nuevas raíces al realizar la realimentación de variables, como se ha descrito en la teoría la realimentación de variables mediante la adición de los coeficientes de realimentación aseguran la ubicación de los nuevos polos.

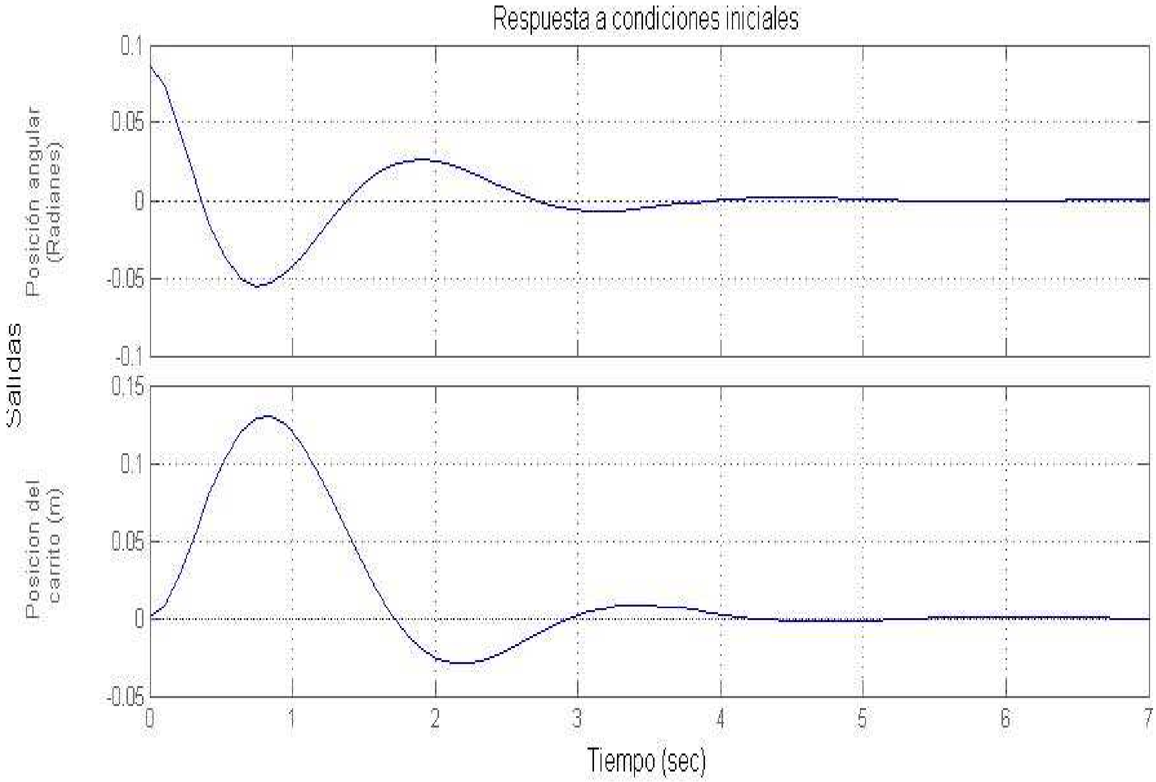
Para el caso mostrado se han seleccionado 2 polos reales los cuales se encuentran alejados de los polos dominantes, los cuales son números imaginarios.

La selección de las raíces de igual manera determina la forma de la respuesta, es decir al tener raíces imaginarias el sistema es más oscilatorio así se tienen raíces puras, de igual manera la selección de las raíces para este problema es irrelevante pues solo se busca ilustrar la importancia de la realimentación de variables.

Mediante el comando `k=place(A,B,p);` se determinan los coeficientes de realimentación convenientes para la ubicación de los polos deseados:

```
eig(a)

ans =

  -1.0000 + 2.5000i
  -1.0000 - 2.5000i
  -4.0000
  -3.0000
```

Por lo tanto las repuestas del sistema a realimentar, al considerar como condición inicial un ángulo de cinco grados (0.087 radianes), son las siguientes:



Puede observarse que tanto el ángulo del péndulo como el carrito regresan a la posición deseada en el origen de coordenadas y en equilibrio.

Cabe destacar que si se selecciona una raíz en el origen, el péndulo regresa al equilibrio, pero el carrito no regresa a la posición inicial en el origen de coordenadas.

Se ha seleccionado esta condición inicial del ángulo basándose en el rango de la aplicación para la linealización, una manera de determinar el rango como se mencionado es a través de la evaluación de las ecuaciones diferenciales, en algunos casos de acuerdo a las ecuaciones involucradas es complicado determinar este rango.

Para este caso, se ha evaluado por comparación simple la linealización considerada para el seno y el coseno.

A continuación se muestra una tabla en donde se pueden determinar hasta que punto es posible realizar la linealización empleada.

Angulo (radianes)	Angulo (Grados)	f(x)=cos(x)	Error relativo	f(x)=sen(x)	Error Relativo
0,000	0,0000	1,0000	0,0000	0,0000	0,0000
0,010	0,5730	1,0000	0,0050	0,0100	0,0017
0,020	1,1459	0,9998	0,0200	0,0200	0,0067
0,030	1,7189	0,9996	0,0450	0,0300	0,0150
0,040	2,2918	0,9992	0,0801	0,0400	0,0267
0,050	2,8648	0,9988	0,1251	0,0500	0,0417
0,060	3,4377	0,9982	0,1803	0,0600	0,0600
0,070	4,0107	0,9976	0,2455	0,0699	0,0817
0,080	4,5837	0,9968	0,3209	0,0799	0,1067
0,090	5,1566	0,9960	0,4064	0,0899	0,1351
0,100	5,7296	0,9950	0,5021	0,0998	0,1669
0,110	6,3025	0,9940	0,6081	0,1098	0,2020
0,120	6,8755	0,9928	0,7243	0,1197	0,2404
0,130	7,4485	0,9916	0,8510	0,1296	0,2822
0,140	8,0214	0,9902	0,9881	0,1395	0,3274
0,150	8,5944	0,9888	1,1356	0,1494	0,3760
0,160	9,1673	0,9872	1,2938	0,1593	0,4279
0,170	9,7403	0,9856	1,4626	0,1692	0,4833
0,180	10,3132	0,9838	1,6422	0,1790	0,5420
0,190	10,8862	0,9820	1,8326	0,1889	0,6042
0,200	11,4592	0,9801	2,0339	0,1987	0,6698
0,210	12,0321	0,9780	2,2463	0,2085	0,7388
0,220	12,6051	0,9759	2,4698	0,2182	0,8112
0,230	13,1780	0,9737	2,7046	0,2280	0,8871
0,240	13,7510	0,9713	2,9508	0,2377	0,9665
0,250	14,3239	0,9689	3,2085	0,2474	1,0493
0,260	14,8969	0,9664	3,4779	0,2571	1,1356
0,270	15,4699	0,9638	3,7591	0,2667	1,2254
0,280	16,0428	0,9611	4,0523	0,2764	1,3187
0,290	16,6158	0,9582	4,3576	0,2860	1,4155
0,300	17,1887	0,9553	4,6752	0,2955	1,5159
0,310	17,7617	0,9523	5,0052	0,3051	1,6198
0,320	18,3346	0,9492	5,3479	0,3146	1,7273

0,330	18,9076	0,9460	5,7035	0,3240	1,8383
0,340	19,4806	0,9428	6,0721	0,3335	1,9530
0,350	20,0535	0,9394	6,4540	0,3429	2,0712

Como puede apreciarse en la función $f(x) = \cos(x)$, se puede evaluar hasta 0.19 radianes obteniendo un error relativo menor al 2 %; este sería el límite para el modelo linealizado del péndulo invertido.

Así como también para todos los modelos que incluyan esta linealización en donde se incluyan ambas funciones trigonométricas ya que como se puede apreciar para el caso de la función $f(x) = sen(x)$, el error relativo mayor a 2 % se obtiene luego de evaluar 0.34 radianes.