

PRÁCTICAS DE REGULACIÓN AUTOMÁTICA (IEEF ETSIDI-UPM)

Carlos Platero

Práctica 4: Reguladores continuos

El objetivo fundamental de un regulador de control es mejorar el comportamiento dinámico de un sistema físico, de forma que se adecue a las especificaciones deseadas para su funcionamiento. En la figura 1 se observa cuál es la posición específica de un regulador clásico posicionado en cascada. Este elemento tiene por objeto actuar sobre la planta en función de la diferencia entre la consigna de entrada, $X(s)$, y la señal de salida del sistema, $Y(s)$.

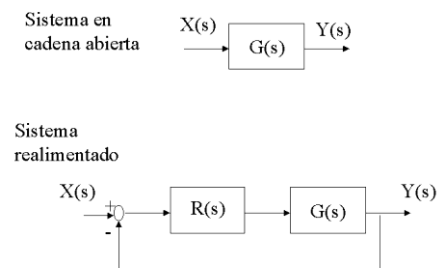


Figura 1 Sistemas en cadena abierta y realimentado

Esta práctica se centra en el estudio del regulador utilizando el método del lugar de las raíces (LDR). Para ello se deben tener en cuenta los puntos que se citan a continuación:

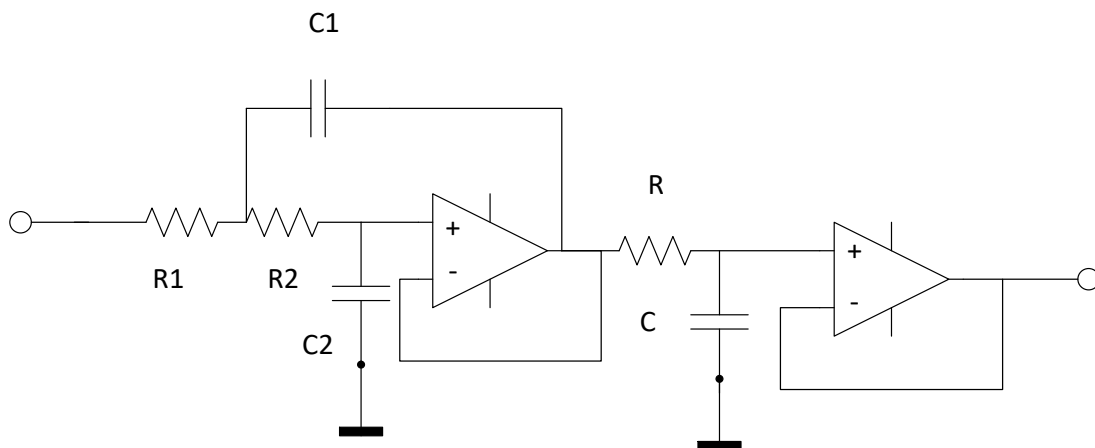
1. Experimentalmente se ha comprobado que, si un sistema tiene un buen comportamiento ante una entrada en escalón, puede estimarse que también responderá correctamente a otro tipo de entradas.
2. Que la respuesta dinámica de un sistema puede ser aproximada a uno equivalente de segundo orden. Y se considera que el sistema tiene una respuesta aceptable si su amortiguamiento está entre $\xi=0.4$ y $\xi=0.7$, que corresponde a sobreoscilaciones entre el 35% y el 5%. Factores de amortiguamientos menores implican elevadas oscilaciones, y amortiguamientos mayores dan lugar a sistemas excesivamente lentos.
3. Respecto al régimen permanente, el estudio del error e_p , e_v , y e_a , permite conocer el error que se produce con las entradas en escalón, rampa y parábola respectivamente.
4. Es importante comprobar que **no se producen saturaciones** en ninguna de las variables intermedias que hubiese entre la entrada y la salida del sistema. Por ejemplo, una saturación a la salida del regulador causaría un comportamiento no lineal de todo el sistema físico. Este efecto provocará en el sistema una respuesta distinta que la prevista.

Lugar de las raíces para el cálculo de un regulador

El trazado directo e inverso del LDR muestra la relación que existe entre la ganancia estática de un sistema, k , y su respuesta temporal. Así pues, dadas unas especificaciones temporales, como pudieran ser el tiempo de establecimiento, el tiempo de pico o el valor de pico, deberá reformarse dicho lugar de las raíces de manera que se alcancen las especificaciones deseadas. El regulador más simple es una acción proporcional, aunque generalmente no es suficiente para cumplir los requisitos de diseño. Cuando se precise modificar el lugar de las raíces puede

probarse la utilización de una red PID (Proporcional, Integral y Derivativa), con el fin de mejorar el comportamiento dinámico del sistema realimentado.

En esta práctica se va a repasar la actuación de los diferentes reguladores sobre dos sistemas. En la primera parte se utilizará un sistema que tiene un modelo matemático conocido. Se trata de un sistema de tercer orden con un polo real y un polo complejo y conjugado, pues éste es el modelo que representa el procesamiento de una red RC en serie con un filtro paso bajo de Sallen-Key.



En la segunda parte se utiliza el concepto de “caja negra”. Se analizará la relación dinámica entre la entrada y la salida aplicada a un servo motor con realimentación. *System Identification toolbox* de Matlab se empleará para tener una estimación del modelo dinámico en el dominio de Laplace.

Ya sea de la extracción del conjunto de ecuaciones algebro-diferenciales o mediante técnicas experimentales de identificación, un modelo de la planta es obtenido. Desde la FDT del sistema y utilizando las técnicas de compensación se mejorará sus prestaciones dinámicas, incluyendo las mejoras del tiempo de respuesta, precisión, aumento de la estabilidad e inmunidad a las perturbaciones.

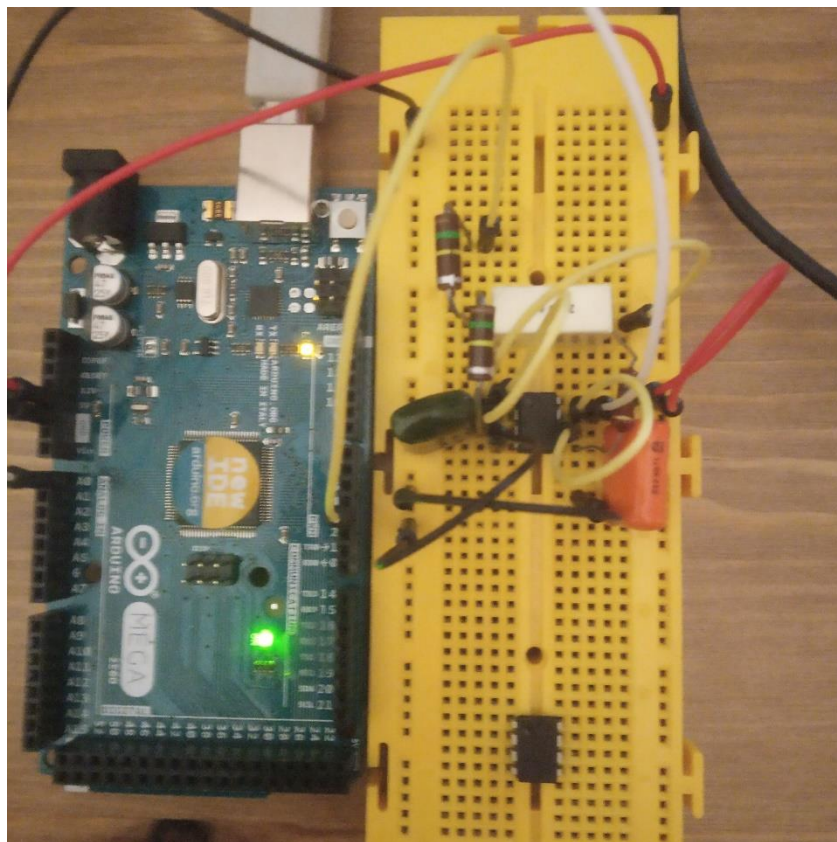
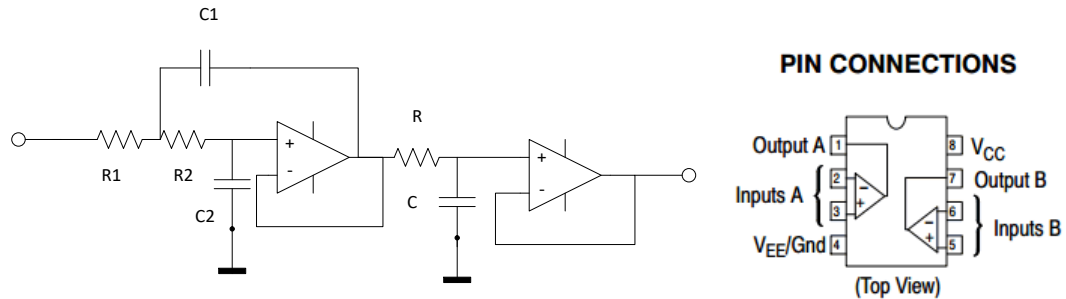
Cuestiones teóricas

- 1) Obtener la ganancia de tensión del circuito electrónico de tercer orden (figura 1) considerando los siguientes valores: $C_1 = 2.2 \mu\text{F}$, $C_2 = 180 \text{ nF}$ y $R_1 = R_2 = 150 \text{ k}\Omega$ para el filtro de Sallen-Key y en cuádrupolo RC $C = 1 \mu\text{F}$ y $R = 1 \text{ M}\Omega$
- 2) Partiendo del modelo dado en el apartado anterior, dibujar el trazado directo de su Lugar de Raíces (LDR). Calcular los polos dominantes y la respuesta del sistema ante una entrada en escalón para los casos de tener un regulador proporcional con ganancia de valor 1 y de valor 3 (representar los valores de error y las magnitudes más representativas del comportamiento dinámico).
- 3) Visto que el error en el régimen permanente es elevado, se procede a Introducir un regulador PI $G_C(s) = 3 \frac{1+0.3s}{s}$. Representar su LDR, calcular los polos dominantes y determinar la respuesta ante el escalón unitario del lazo cerrado, como siempre indicar los valores más significativos.

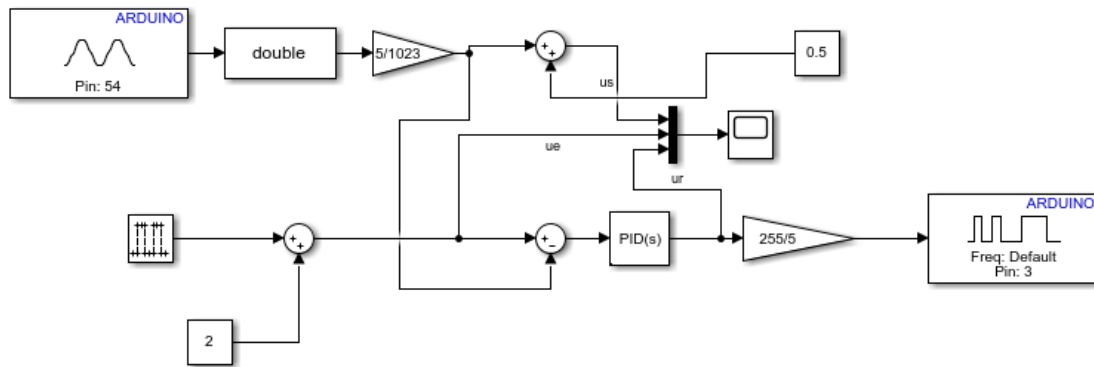
- 4) Prestaciones del servo motor con realimentación a utilizar en el trabajo experimental.
- 5) Funcionamiento de *System Identification Toolbox* de Matlab.

Cuestiones prácticas

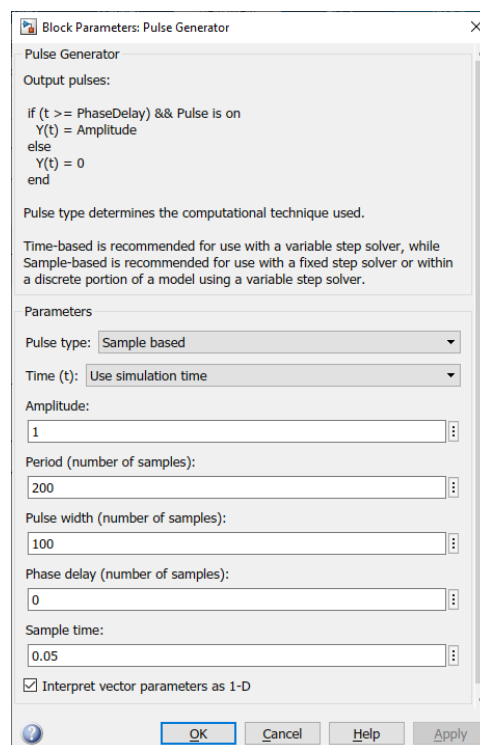
Montar el circuito de la figura utilizando los siguientes valores: $C = 1\ \mu\text{F}$, $C_1 = 2.2\ \mu\text{F}$, $C_2 = 180\ \text{nF}$, $R = 1\ \text{M}\Omega$ y $R_1 = R_2 = 150\ \text{k}\Omega$. Como siempre, se recomienda tener montado los circuitos electrónicos antes de iniciar las prácticas.



La señal de mando, el amplificador de error y el regulador estarán definidos en el modelo de Simulink.



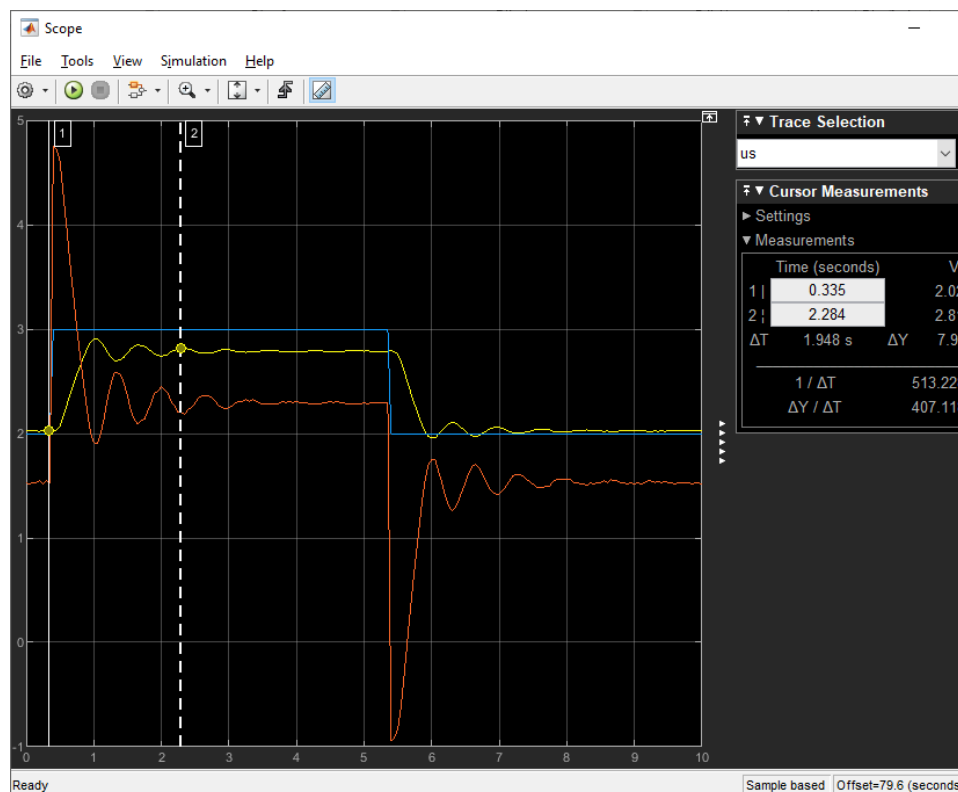
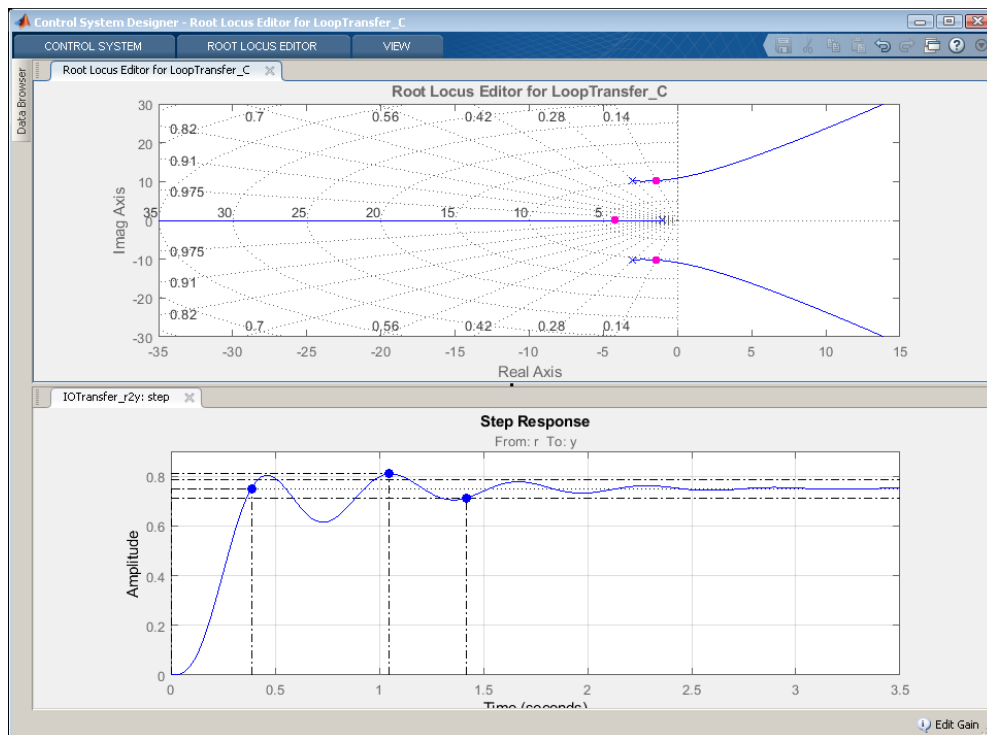
Utilizar un periodo de muestro de 0.05s, con un proceso de carga y descarga de 10 s en la señal de mando.



Para el ajuste del regulador se utilizará la herramienta de Matlab **rltool()**. Emplear el siguiente código:

```
R=150e3;
C1=2.2e-6;
C2=180e-9;
Av1=tf(1,[R*R*C1*C2 C2*2*R 1]);
%% RC
R=1e6;
C=1e-6;
Av2=tf(1,[R*C, 1]);
%% Analyzed tool
Av=Av1*Av2;
rltool(Av);
```

Ajustar el regulador P para 1 y 3. Analizar los resultados tanto en la herramienta rltool() como mediante Simulink+arduino. Para $k=3$, medir en ambas aplicaciones los tiempos de subida, de pico y de establecimiento junto con la sobreoscilación y el error al escalón.



Analizar el resultado que da rltool() empleando “PID Tuning” con un regulador proporcional.

PID Tuning

Compensator

C = 3.2216

Select Loop to Tune

LoopTransfer_C

Add new loop ...

Specifications

Tuning method: Classical design formulas

Controller Type: ☐ P ☒ PI ☐ PID ☐ PID with derivative filter

Formula: Approximate MIGO frequency response

Update Compensator Help

Ajustar mediante este procedimiento un regulador PI y aplicar los parámetros al montaje y a la simulación con `rttool()`. Medir en ambas aplicaciones los tiempos de subida, de pico y de establecimiento junto con la sobreoscilación y el error al escalón.

PID Tuning

Compensator

C = 3.0351 x $\frac{(1 + 0.34s)}{s}$

Select Loop to Tune

LoopTransfer_C

Add new loop ...

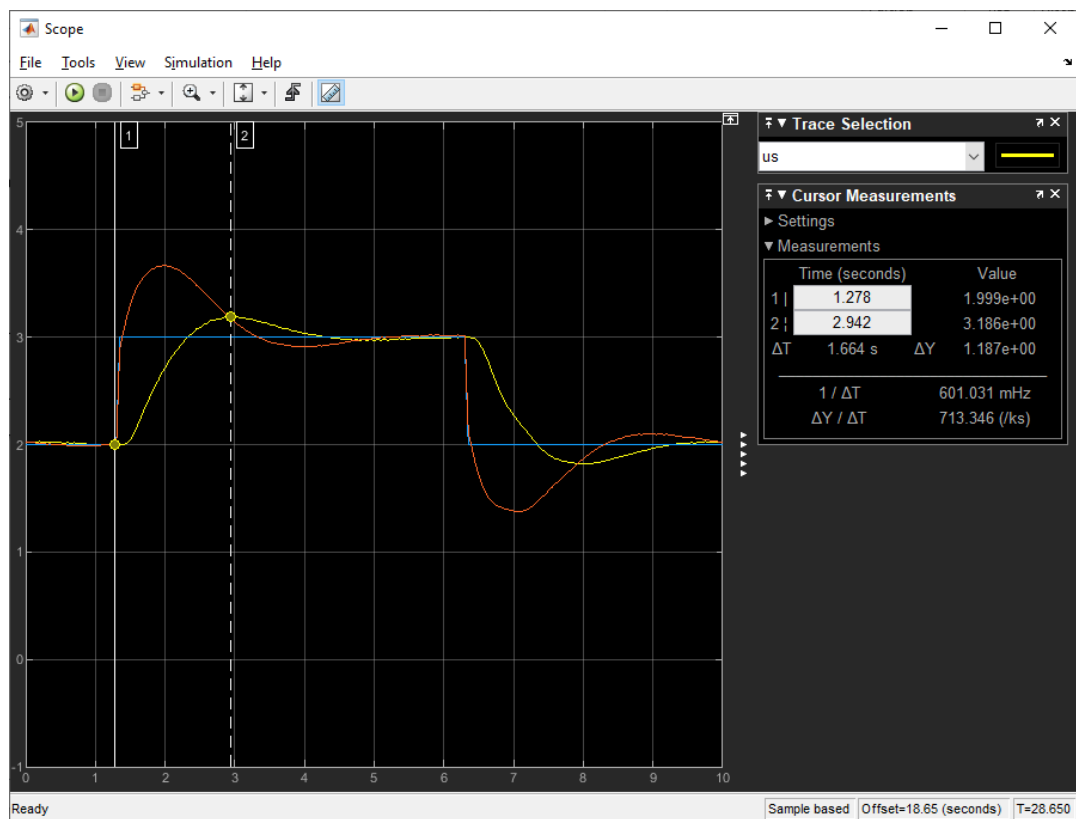
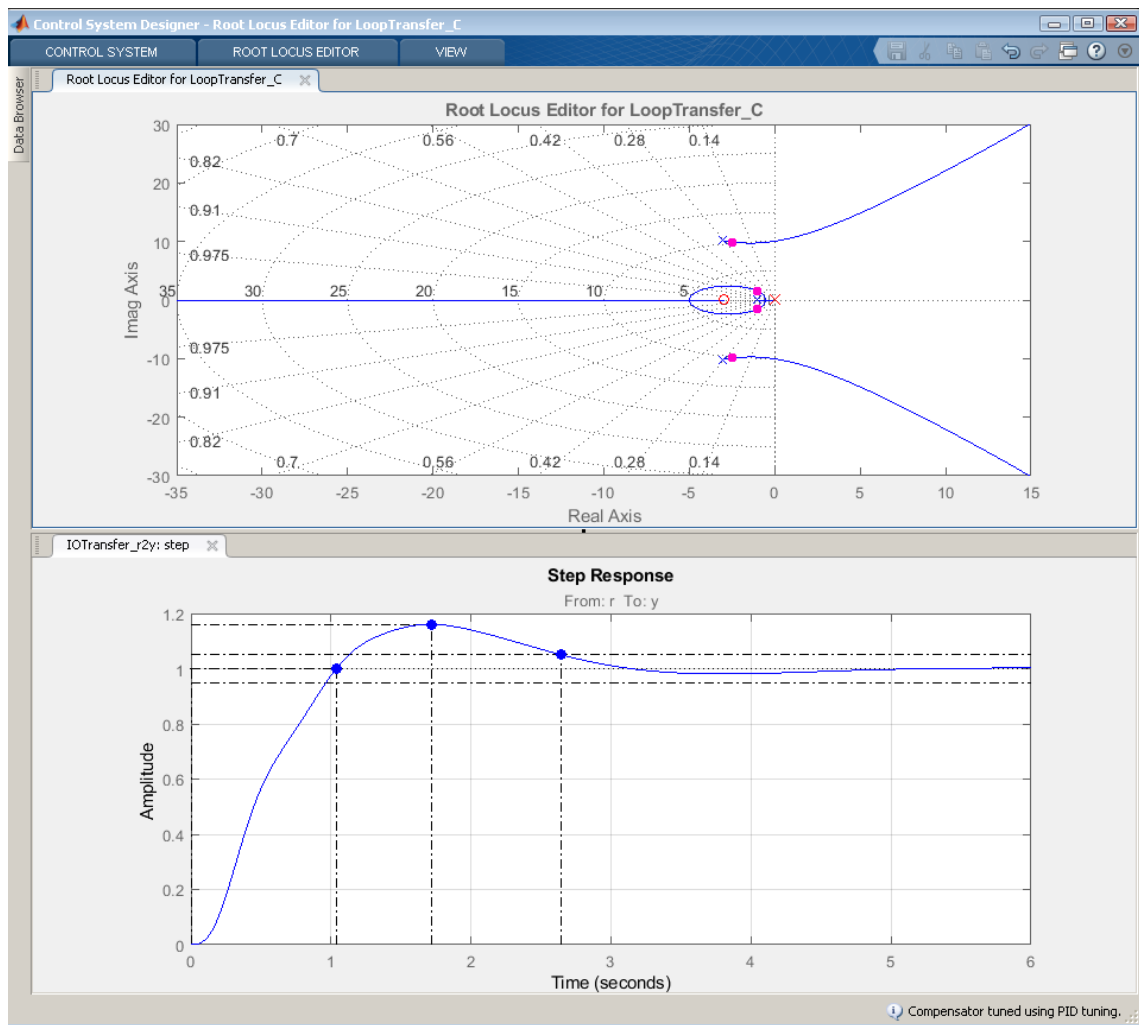
Specifications

Tuning method: Classical design formulas

Controller Type: ☐ P ☒ PI ☐ PID ☐ PID with derivative filter

Formula: Approximate MIGO frequency response

Update Compensator Help

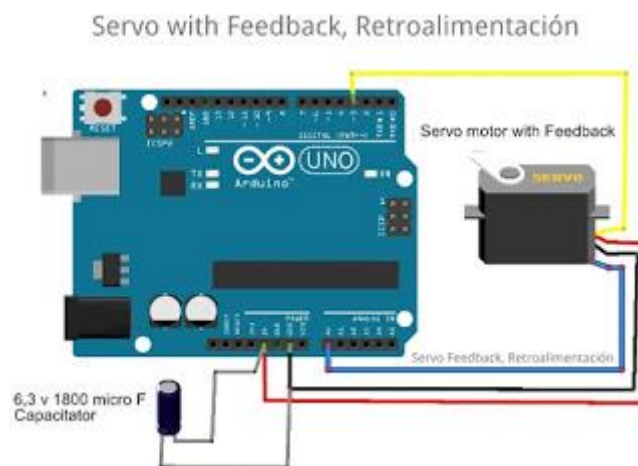


Control de posicionamiento angular utilizando un servo motor con realimentación

Conectar el servo con realimentación al arduino. En este experimento se ha utilizado el modelo FT1117M-FB con feedback.



Por ejemplo, la entrada de mando del motor se une al puerto PWM D3 y la salida del sensor de posicionamiento a la entrada analógica A0.



La siguiente función de Matlab utiliza la librería 'SERVO'. Se procede a girar de 0 a 180 grados. También se hace lectura del sensor.

```
function values=servo_pr4RA

%% Arduino objects
a = arduino();
s = servo(a, 'D3');

port_Analog = 'A0';
writePosition(s, 0);
pause;
i=1;

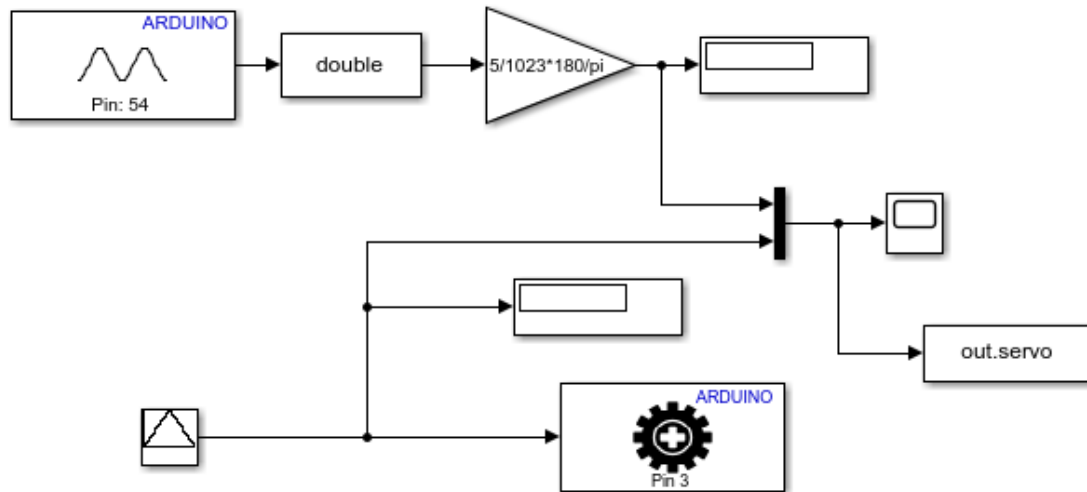
%% Move
angle=0:0.1:1;
values=zeros(length(angle),2);
for angle = 0:0.1:1
    writePosition(s, angle);

    time=10;
    while(time>0)
        current_pos = readPosition(s);
        current_pos = current_pos*180;
        voltage = readVoltage(a, port_Analog);
        values(i,:)= [current_pos,voltage*180/pi];
        i=i+1;
        fprintf('Current motor position is %d degrees %.2f volts\n',...
            current_pos,voltage);
        time=time-1;
    end
end

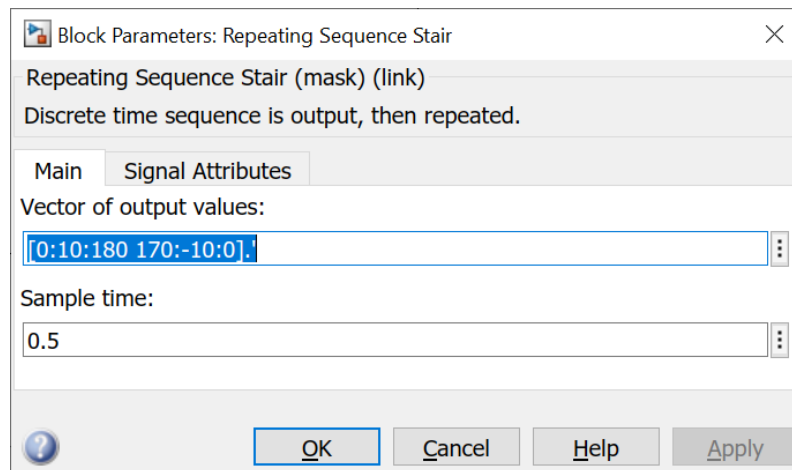
end

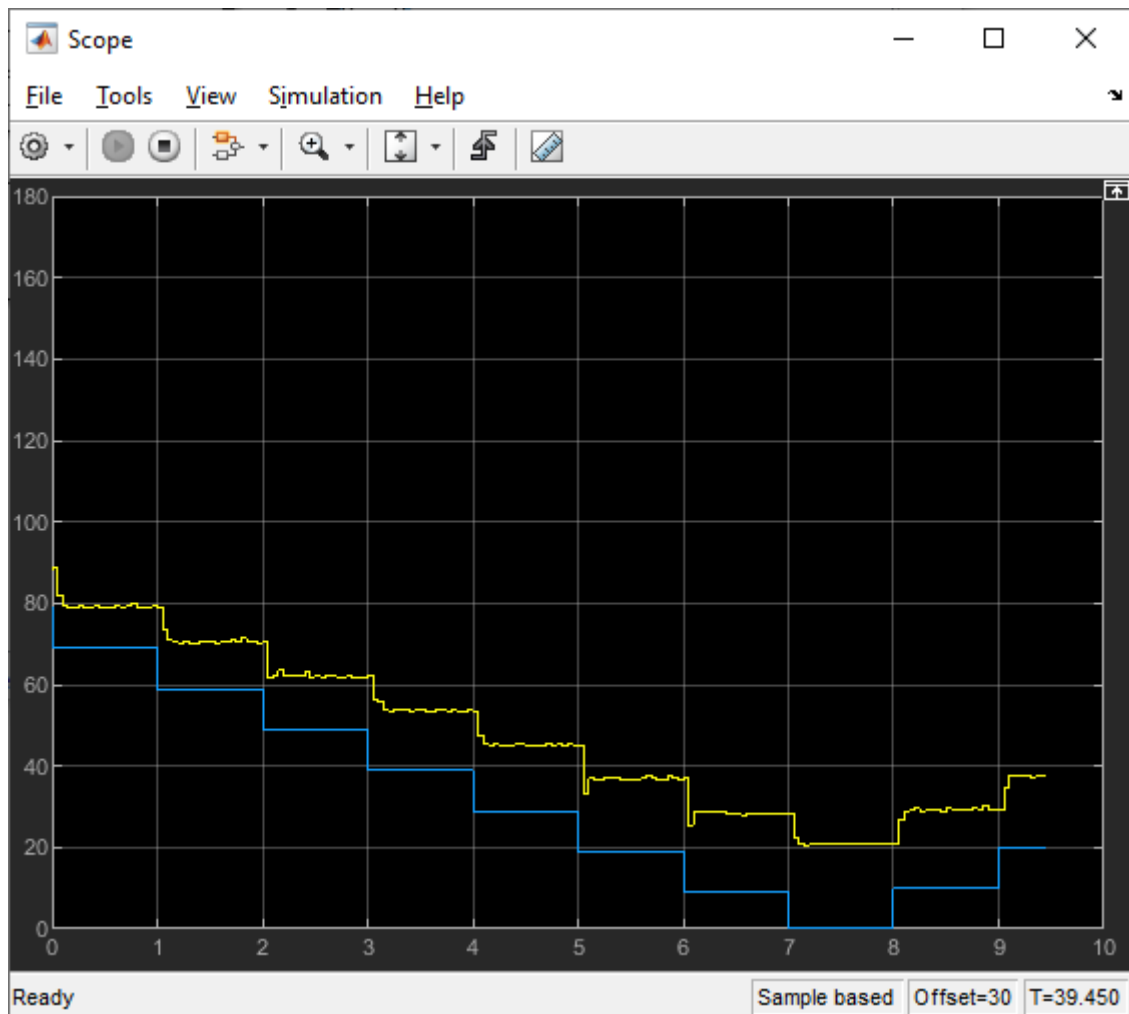
%% delete
clear a s
end
```

Es observable que existe diferencia entre la señal de mando y la lectura de posición. A continuación, se procede a construir el siguiente modelo de Simulink. El servo-motor es probado con una trayectoria rampa de 0 a 180 grados y de 180 a 0. También se hace lectura del sensor de posicionamiento angular.



Muestrear la señal del sensor con un periodo de 0.1s. Generar una señal de entrada en forma de rampa que vaya desde 0 a 180 grados y de 180 a 0 con saltos de 10 grados.



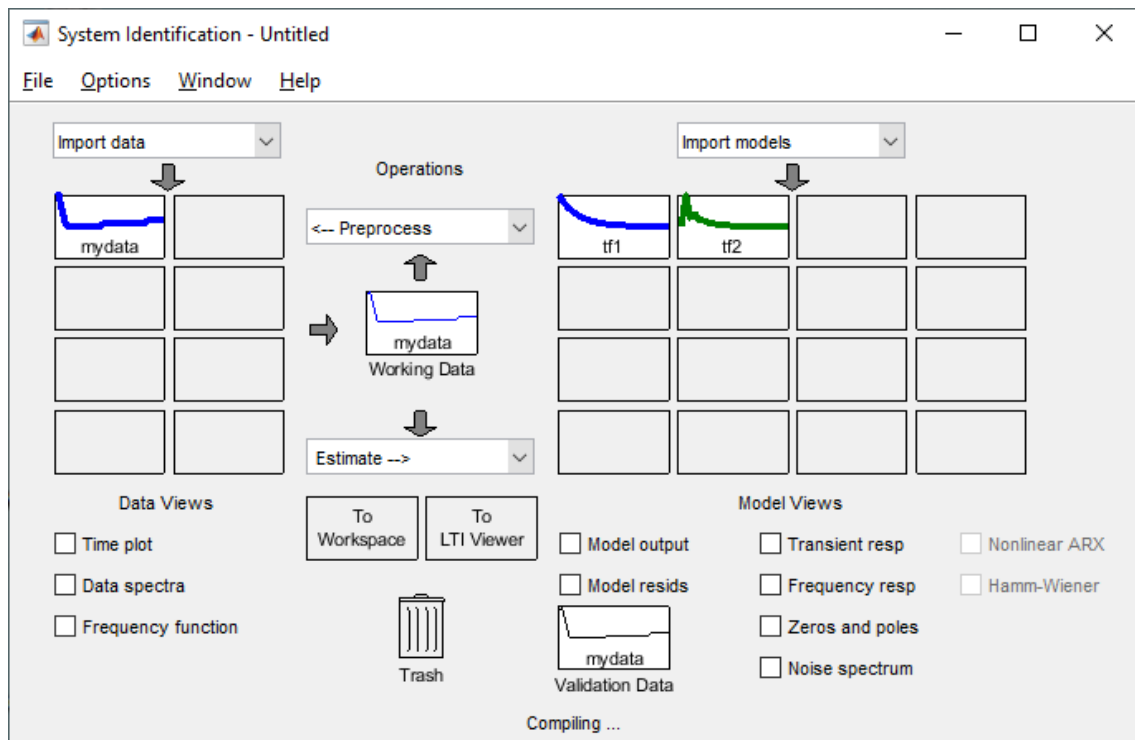


Estas trayectorias son enviadas al Workspace de Matlab. Realizar al menos tres perfiles de velocidades. El servo motor se analiza como una caja negra. Nótese que se tiene una entrada y una salida. Ambas señales son introducidas en el toolbox de System Identification.

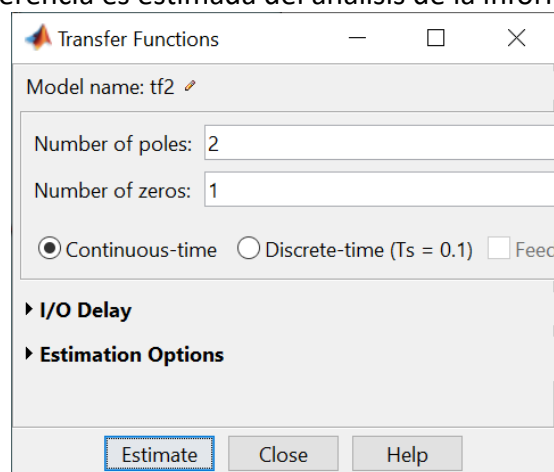
```
t=servo.time;
in=servo.signals.values(:,2);
out=servo.signals.values(:,1);
```

The Import Data dialog box is shown with the following settings:

- Data Format for Signals:** Time-Domain Signals (dropdown menu)
- Workspace Variable:**
 - Input: in
 - Output: out
- Data Information:**
 - Data name: mydata
 - Starting time: 1
 - Sample time: 0.1
 - More (button)
- Buttons:** Import, Reset, Close, Help



Una función de transferencia es estimada del análisis de la información aportada.

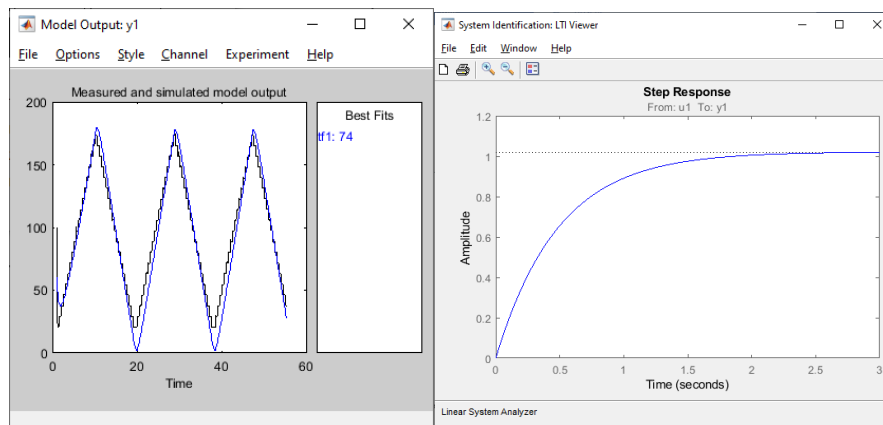


Exportar la FDT estimada:

g1 =

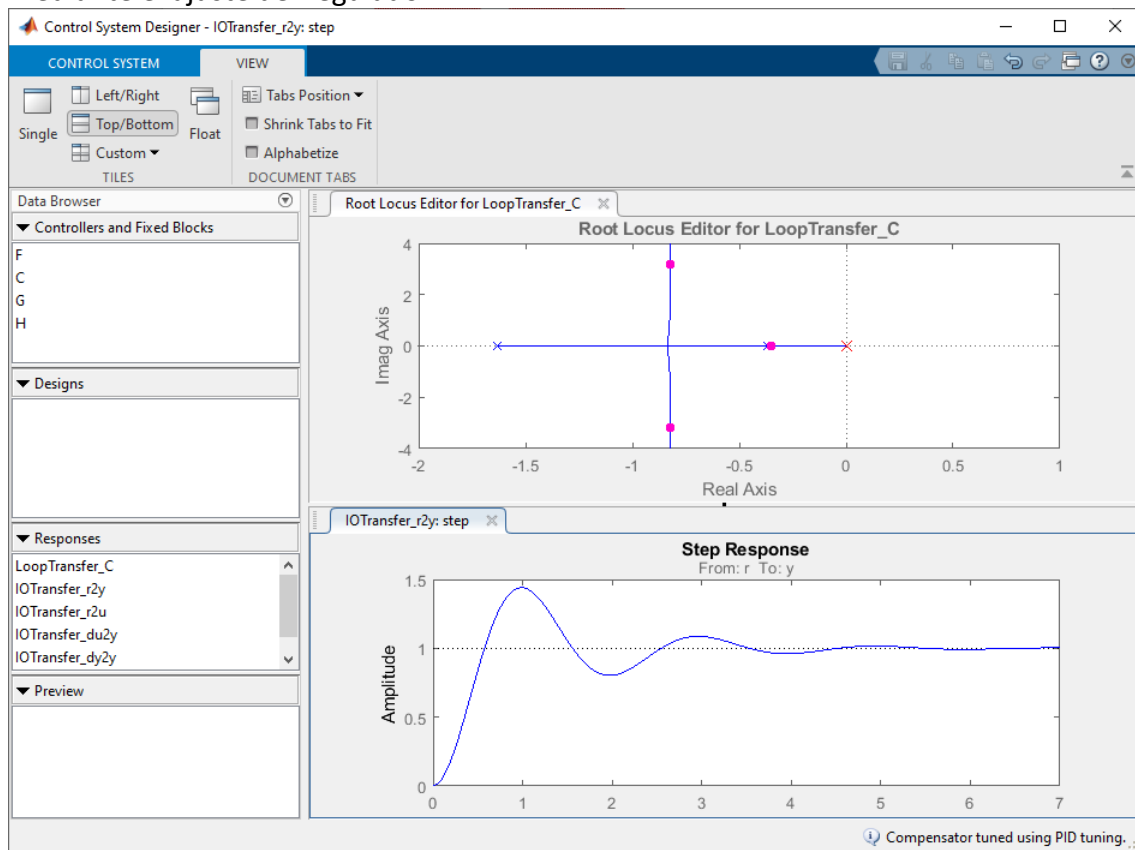
$$\frac{1.5 s + 0.53}{s^2 + 2 s + 0.6}$$

Simular el comportamiento de la FDT estimada mediante la opción “Model output” o en “Show in LTI viewer”



Si la FDT estimada no parece que se ajuste al comportamiento dinámico del sistema, repita de nuevo el experimento de dar una excitación en rampa al servomotor.

Definido la FDT del servomotor, introducir el modelo en rltool() y mejorar su dinámica mediante el ajuste del regulador.



PID Tuning

Compensator

C

= 7.2013 x $\frac{1}{s}$

▼ Select Loop to Tune

LoopTransfer_C

Add new loop ...

Specifications

Tuning method:

Robust response time

Controller Type:

☐ P

☐ I

☒ PI

☐ PD

☐ PID

☐ Design with first order derivative filter

Design mode:

Time

«

Slower

Response Time (seconds)

Faster

»

0.6473

▲▼

↺

Reset Parameters

Aggressive

Transient Behavior

Robust

0.6

▲▼

Update Compensator

Help

