

# Práctica 4: Identificación de Sistemas

Nombre:

Paralelo:

Fecha:

## 4.1. Objetivos

### 4.1.1. Objetivo General

Al finalizar esta sesión el estudiante estará en capacidad de realizar la identificación de un sistema, utilizando MATLAB<sup>®</sup>, para la obtención de la función de transferencia de un proceso.

### 4.1.2. Objetivos Específicos

- Diseñar y probar diferentes filtros de primer orden en una señal.
- Realizar la identificación del sistema en estudio utilizando el toolbox **System identification** de MATLAB<sup>®</sup>.

## 4.2. Introducción

En el área de Ingeniería de Control es sumamente importante la identificación de sistemas debido a que este proceso permite obtener la representación matemática de una planta real con la que se podrá diseñar diferentes controladores. Para realizar una correcta identificación se deben diseñar apropiadamente las entradas a utilizar y escoger los puntos de operación en base a los requerimientos del usuario.

Además se debe acondicionar los datos adquiridos; recuerde que al identificar el sistema se trata de caracterizar las variaciones de la salida del sistema ante variaciones de la entrada del sistema alrededor de sus respectivos puntos de operación. Los datos de salida de sistemas reales presentan ruido, el cual puede afectar considerablemente la salida del sistema y por lo tanto el proceso de identificación. Por esta razón se vuelve importante el diseño y uso de filtros como paso previo a la identificación.

#### 4.2.1. Filtros

Los filtros son elementos que dejan pasar, atenúan o eliminan ciertas componentes de frecuencia de una señal. Las componentes que se ven afectadas depende del diseño y tipo del filtro. El ruido introducido a la salida del sistema por los sensores utilizados suele ser una señal de alta frecuencia. Se debe diseñar un filtro que rechace, en la medida de lo posible, las componentes de alta frecuencia. De esta manera, al filtrar los datos de salida que contienen ruido, se tendrá como resultado una señal con un nivel de ruido inferior a la señal original.

El filtro a utilizar para este fin es un **pasa bajo**; este filtro deja pasar las componentes de baja frecuencia y atenúa las componentes de mayor frecuencia. El filtro pasa bajo más simple se puede representar por la función de transferencia  $F(s) = \frac{p_f}{s + p_f}$  donde  $p_f$  corresponde a la frecuencia de corte del filtro. Su respuesta de frecuencia se muestra a continuación:

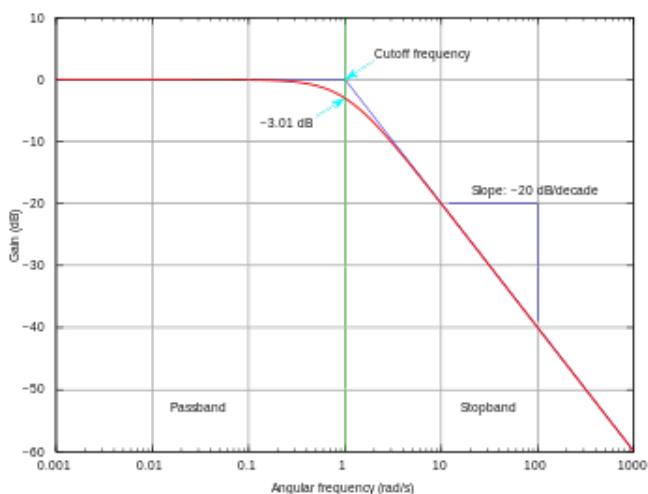


Figura 4.1: Respuesta en frecuencia de un filtro pasa bajo: Magnitud

Las componentes de frecuencia mayores a la frecuencia de corte (cutoff frequency) se atenúan significativamente mientras que las componentes de frecuencia menores a la frecuencia de corte pasan sin mayor alteración. Para la selección de  $p_f$  se debe tomar en cuenta que el filtro corresponderá a un bloque que se encuentra en serie con la planta. Por esta razón el polo del filtro aparecerá como parte de la función de transferencia de lazo abierto; sin embargo no se desea incluir a este polo como parte del resultado del proceso de identificación. Para filtrar los datos de salida pero a la vez asegurar que el filtro no afecte la dinámica real del sistema se utiliza el concepto de **dominancia** del sistema. Se asume que el filtro no afecta la dinámica del sistema si su polo se encuentra al menos 5 veces alejado del polo o polos dominantes de la planta; en la práctica se usa un factor mayor a 10 para la selección del polo del filtro. El polo o polos dominantes del sistema se determinan a partir de la prueba escalón y de los índices de desempeño estimados previamente.

Debido al ruido presente en la señal de salida, la estimación del polo dominante a partir del tiempo de estabilización del sistema no es precisa por lo que se debe aumentar el factor de 5 ( $p_f \geq 5p$ ) e ir visualizando la señal filtrada y sin filtrar para comprobar que la dinámica del sistema se mantenga. En las figuras 4.2 y 4.3 se observa la respuesta del sistema filtrado y sin filtrar para el caso de un filtro dimensionado de manera correcta e incorrecta respectivamente.

Para el filtro dimensionado correctamente, la señal filtrada se encuentra dentro de los límites mínimos y máximos de la señal sin filtrar manteniendo así la dinámica del sistema original.

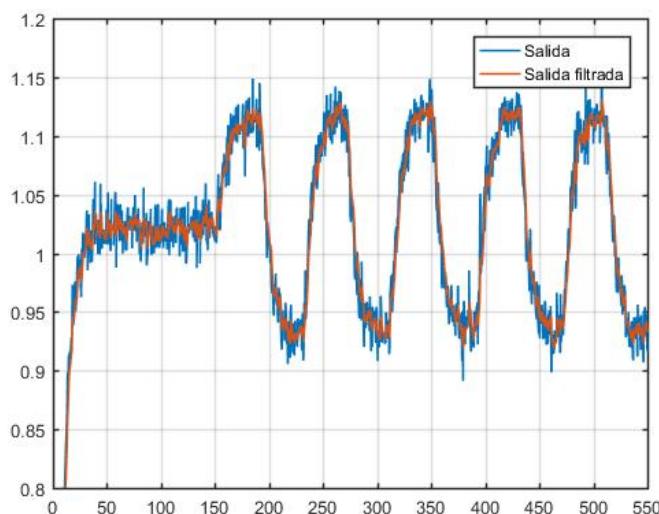


Figura 4.2: Señal filtrada y sin filtrar: Filtro dimensionado correctamente

En cambio, para el caso del filtro dimensionado de manera incorrecta, la señal filtrada se encuentra desplazada con respecto a la señal original cambiando la dinámica original del sistema. En este caso se debe simular nuevamente el filtro aumentando el factor de 5 de manera que el polo del filtro se aleje más del polo dominante estimado de la planta. Se debe tener cuidado al aumentar significativamente el factor debido a que mientras mayor sea este valor, se filtrará menos la señal. Además se debe diseñar el filtro de manera que la ganancia DC del mismo sea 1; es decir  $F(s = 0) = 1$ .

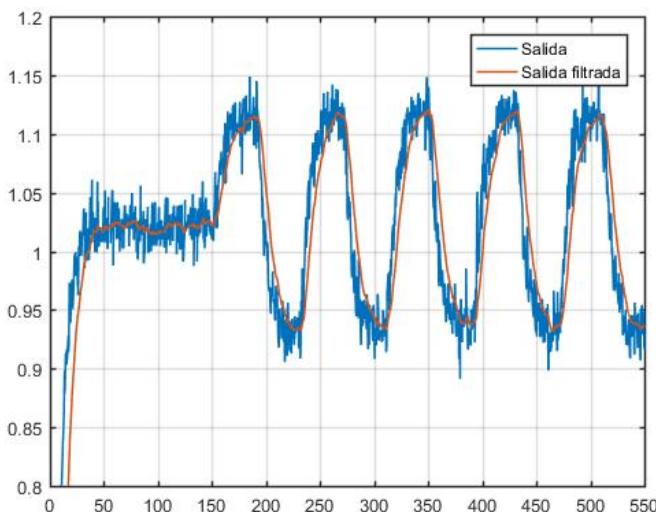


Figura 4.3: Señal filtrada y sin filtrar: Filtro mal dimensionado

### 4.3. Ejercicio

Utilizando como apoyo la **Guía de prácticas de Laboratorio de Sistemas de Control**, descargue el script **STA4**, los datos adjuntos y el formato del Trabajo Autónomo 4 y, desarrolle lo solicitado. Recuerde comentar las líneas de código que desarrolle y colocar al inicio del script su nombre y paralelo práctico; separe por secciones su código para una mejor organización de este.

- Coloque el script y datos descargados en un mismo directorio.
- En el script STA4 deberá añadir las líneas de código necesarias para realizar lo siguiente:
  1. Guardar los datos de entrada, salida y tiempo en las matrices **in**, **out** y **t** respectivamente.  
**Sugerencia:** Identifique a qué variable corresponde cada columna de la matriz **datos**.
  2. Graficar en dos figuras independientes la entrada y salida del sistema en función del tiempo. Recuerde incluir títulos, cuadrículas, etiquetas, etc.
  3. Estimar el valor del polo dominante a partir del tiempo de estabilización, sobrenivel porcentual o tiempo pico del sistema.
  4. Diseñar y simular un filtro de primer orden adecuado para la salida y guardar estos datos en la matriz **outF**.  
**Sugerencia:** Cree el filtro usando el comando **tf** o **zpk**; recuerde que la ganancia DC del filtro debe ser igual a 1. Simule el filtro con el comando **lsim**; consulte la ayuda del comando.

5. Comprobar que el filtro diseñado es adecuado. Grafique en una nueva figura la señal sin filtrar (out) y la señal filtrada (outF) versus tiempo. Recuerde que la señal filtrada debe mantener las características transientes del sistema. Diseñe y pruebe **al menos 3 filtros** y luego escoja el más adecuado.
  6. Identificar el punto de operación de la entrada y estimar el punto de operación de la salida a partir de las gráficas utilizando el comando **mean**. Registre estos datos como **in\_op** y **out\_op**.  
**Sugerencia:** El punto de operación de la entrada lo puede identificar directamente de la gráfica; utilice el comando **mean** para calcular el promedio del **grupo de datos** de la salida donde el sistema se encuentre en el punto de operación.
  7. Restar los puntos de operación a las señales de entrada y salida(después de filtrar) y guardar estos datos en las matrices **in\_Mop** y **out\_Mop** respectivamente. Verifique lo anterior graficando estas señales versus tiempo. Grafique ambas señales en una sola figura.
  8. Llamar a la herramienta System Identification de MATLAB® ingresando el comando **systemIdentification**. Para versiones anteriores de MATLAB® utilice el comando **ident**.
- Importe los datos en el dominio del tiempo. Para la configuración de la ventana **Import Data** necesita conocer el tiempo de muestreo (**Sample time**) con el que fueron adquiridos los datos que se usarán en la identificación. El tiempo de muestreo lo puede obtener a partir del vector de tiempo (t). Adicionalmente, coloque en el **Starting Time** el valor de 0 que por defecto aparece en 1 y haga click en Import. Recuerde que debe importar los datos sin los puntos de operación, es decir in\_Mop y out\_Mop.
  - Escoja del menú **Preprocess** la opción **Select Range** e importe al menos dos grupos de datos que contengan al menos un periodo completo de la señal cada uno; estos grupos escogidos deben ser disjuntos y lo más parecidos entre sí.
  - Seleccione uno de los rangos importados para usar como **Working Data** y otro para **Validation Data**; arrastre cada grupo de datos a su respectivo recuadro. El grupo de datos que se encuentre en Working Data, será el que use la herramienta para estimar la función de transferencia, mientras que el otro grupo que se encuentre en Validation Data será el que se utilice para comparar la salida de los sistemas estimados contra la respuesta real del sistema.
  - Seleccione del menú **Estimate** la opción **Process Models** para la estimación de la función de transferencia; escoja la cantidad de polos y ceros en base al comportamiento conocido del sistema. **No incluya delay o integradores**.
  - Usando la herramienta de identificación de MATLAB®, realice la estimación de al menos tres funciones de transferencia (con distinta estructura) que puedan representar al sistema y elija la que usted considere adecuada. Recuerde que se debe buscar obtener un buen porcentaje

de aproximación (recomendable superior al 70 %) entre la respuesta del modelo identificado y la respuesta real del sistema (**Model output**). Sin embargo, también se busca de que la función de transferencia identificada no sea tan compleja; por esta razón se evita el uso de retrasos o retardos (**Delay**) e integradores. Use la **Guía de prácticas de Laboratorio de Sistemas de Control** como referencia para la familiarización y uso de la herramienta.

**Nota:** Todas las gráficas registradas de las simulaciones en el formato de la práctica deben ser mostradas utilizando el comando **plot** y tener bien diferenciados los puntos de interés que utilizó para sus estimaciones. Recuerde que las gráficas se realizan versus tiempo y deben incluir títulos, cuadrículas y leyendas de ser necesarias.

#### 4.4. Conclusiones y Recomendaciones