Universidad de Ingeniería y Tecnología

Departamento de Ingeniería Mecatrónica

Control no lineal

Laboratorio 1

Puntos de equilibrio

Instructor de laboratorio:

Jhon Charaja

Asistente de laboratorio:

Ricardo Terreros

Estudiantes:

Nombre Apellido

Nombre Apellido

Lima - Perú

2022 - 2

Laboratorio 1
Puntos de equilibrio

1 Objetivos

- Aprender a conectar el QUBE-Servo 2
- Entender el funcionamiento de los bloques de Quanser para recibir la posición del motor y enviar la señal de control.
- Obtener los puntos de equilibrio de un modelo no lineal.
- Entender el efecto de las condiciones iniciales en el comportamiento temporal de un modelo no lineal.
- Aproximar ecuaciones diferenciales no lineales con la serie de Taylor.

2 Desarrollo del laboratorio

La adecuada presentación del reporte de laboratorio debe considerar lo siguiente:

- El reporte debe contener el procedimiento que justifique la respuesta de cada actividad desarrollada.
- El reporte debe indicar de manera clara que actividad e inciso se está desarrollando.
- El reporte debe contener gráficas que indiquen con una etiqueta la variable que representa cada eje.
- Se trabajará en grupos de 2 personas y solo es necesario que un integrante de cada grupo presente el reporte, indicando el nombre de los integrantes del grupo.
- El reporte debe ser presentado en formato pdf a través del buzón de Canvas.
- La presentación del reporte se realizará hasta el término de la sesión de laboratorio. En caso de no presentar a la hora indicada se puede entregar hasta 10 minutos después con una penalización de 5 puntos en la nota del laboratorio.

Para el desarrollo de este laboratorio, se van a usar los archivos `pd_position_control_disk.xls`, `pd_control_inverted_pedulum.xls` y `simulink_act_1.xls`. Estos archivos contienen tres bloques de subsistema: (i) `ley_de_control`, (ii) `modelo_dinámico` y (iii) `guardar_datos`. Por un lado, el bloque `ley_de_control` contiene los valores de cada parámetro necesario para calcular la señal de control. Por otro lado, el bloque `modelo_dinámico` contiene el bloque de escritura de Quanser para enviar la señal de control y el bloque de lectura para recibir la posición del motor . Finalmente, el bloque `guardar_datos` guarda los estados del sistema y el tiempo de simulación. A fin de evitar problemas de ejecución, coloque todos los archivos en una sola carpeta.

2.1 Modelo no lineal

La mayoría de sistemas dinámicos presentan relaciones no lineales entre las variables que lo conforman. El comportamiento temporal de estos sistemas se puede representar con un número finito (n) de ecuaciones diferenciales de primer orden

$$\dot{\mathbf{x}} = f(\mathbf{x}), \tag{1}$$

donde $\mathbf{x} = [\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_\mathrm{n}]^\mathrm{T}$ representa el vector de los estados del sistema.

2.2 Punto de equilibrio

El punto de equilibrio ($\mathbf{x_e}$) representa la ubicación donde los estados de un sistema permanecen constantes de manera indefinida. Por este motivo, los puntos de equilibrio de un modelo no lineal, como de (1), se obtienen resolviendo $f(\mathbf{x}) = \mathbf{0}$. La característica de cada punto de equilibrio ($\mathbf{x_e}$), junto con las condiciones iniciales ($\mathbf{x}(0)$), define el comportamiento temporal de cada modelo no lineal.

2.3 Aproximación lineal con serie de Taylor

La serie de Taylor aproxima el resultado de una función para un valor establecido de cada variable de entrada. El proceso de aproximación consiste en la suma infinita de derivadas de cada variable de la función. En este laboratorio solo es necesario considerar hasta la primera derivada de cada variable de la función. De este modo, la aproximación lineal de (1), con ($\mathbf{x_e}$) como punto de equilibrio, se calcula como

$$\dot{\mathbf{x}} \approx f(\mathbf{x_e}) + \left.\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x_e}} (\mathbf{x} - \mathbf{x_e}), \tag{2}$$

donde $\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ representa la matriz jacobiana.

El proceso de aproximación se puede realizar de forma computacional usando la función `jacobian()` de MATLAB. De esta forma, la aproximación lineal se calcula como

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} = \texttt{jacobian}(f(\mathbf{x}), \mathbf{x}),$$

donde el primer término es la función no lineal y el segundo las variables a derivar.

2.4 Estabilidad mediante linealización

El comportamiento temporal del modelo no lineal (1) y su aproximación lineal (2) es similar cerca del punto de equilibrio ($\mathbf{x_e}$). Por consiguiente, ambos modelos comparten el mismo tipo de estabilidad cerca del punto de equilibrio. De ahí, la estabilidad del modelo no lineal se puede determinar a través de su aproximación lineal.

Los valores propios (λ) de la matriz jacobiana determinan la estabilidad del modelo lineal (2). La aproximación lineal es estable si la parte real de los valores propios (λ) es menor a cero, de otro modo, el sistema es inestable. Los valores propios se obtienen resolviendo

$$\det\left(A - \lambda \mathrm{I}\right), \tag{3}$$

donde $\det(\cdot)$ denota la función determinante, $A = \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$ es la matriz jacobiana de $f(\mathbf{x})$ y I la matriz identidad.

Los valores propios se pueden obtener de forma computacional usando la función `eig()` de MATLAB. De esta forma, los valores propios se calculan como

$$\lambda = \mathtt{eig}\left(\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}\right),$$

donde la variable de entrada es la matriz jacobiana de $f(\mathbf{x})$.