Trabajo Practico Sistemas de ecuaciones diferenciales y Estabilidad

Objetivos

- Comprender el concepto de sistema de ecuaciones diferenciales y su importancia en la modelización de fenómenos dinámicos en ingeniería.
- Aplicar métodos analíticos y numéricos —como la Transformada de Laplace y el análisis de autovalores/autovectores— para estudiar la estabilidad de sistemas lineales.
- Desarrollar habilidades en el uso de herramientas simbólicas y numéricas de Python (SymPy, NumPy, SciPy y Matplotlib) para resolver y visualizar sistemas de ecuaciones diferenciales.
- Analizar el comportamiento dinámico de sistemas reales mediante la interpretación de campos vectoriales, planos de fase y condiciones de estabilidad.
- Integrar conocimientos matemáticos y computacionales en la resolución de un caso aplicado de control automático, reflexionando sobre las limitaciones y mejoras del modelo.
- Fomentar la autonomía, el pensamiento crítico y el uso responsable de la inteligencia artificial como apoyo al aprendizaje y a la resolución de problemas complejos.
- Comunicar de manera efectiva los resultados del trabajo, utilizando recursos visuales y audiovisuales para explicar procesos matemáticos y su aplicación en contextos de ingeniería.

Etapa 1: Investigación del marco conceptual

Formato de presentación:
Documento en Word o PDF titulado 'GrupoX_TPsistemas_Conceptos' (donde X corresponde al número del grupo). Debe incluir nombres completos de los integrantes y bibliografía consultada. Si se utiliza IA, citar las herramientas empleadas y verificar las fuentes.
Propósito: Construir una base teórica sólida que fundamente las etapas siguientes del trabajo práctico.

Consigna

1. Explicá con tus palabras la función principal de la Transformada de Laplace en el análisis de sistemas dinámicos.
2. Definí qué es un punto de equilibrio en un sistema de EDOs y cómo se determina.
3. Analizá cómo los autovalores y autovectores permiten caracterizar soluciones y estabilidad en sistemas lineales homogéneos.
4. Compará las soluciones de un sistema lineal homogéneo con uno no homogéneo.
5. Elaborá un diagrama de flujo del algoritmo de resolución de sistemas homogéneos y no homogéneos (utilizando Laplace).
6. Diseñá un esquema visual que clasifique los tipos de puntos de equilibrio y la estabilidad asociada.

Etapa 2: Investigación Librerías de Python en el Cálculo Numérico y Simbólico

Formato de presentación:
Cuaderno en Jupyter o Google Colab titulado 'GrupoX_TPsistemas_Librerias'. Debe incluir los nombres del grupo al inicio. Se permite el uso de IA para explorar ejemplos, siempre que el grupo comprenda el código y pueda comprobar los resultados.
Propósito: Conocer y dominar las herramientas de Python necesarias para resolver y analizar sistemas diferenciales.

Consigna

1) Investiga la librería **NumPy** y explica cómo se realizan las siguientes operaciones con matrices: Multiplicación de matrices, Cálculo de la inversa de una matriz, Cálculo de valores propios y vectores propios de una matriz. Proporciona ejemplos de código en Python utilizando NumPy.

2) ¿Qué es **SymPy** y en qué áreas del cálculo diferencial se utiliza? Investiga cómo esta librería facilita la resolución simbólica de ecuaciones diferenciales y qué ventajas tiene sobre métodos puramente numéricos.

3) Describe brevemente las funciones principales de la librería **SciPy** para resolver ecuaciones diferenciales. En particular, investiga las funciones odeint y solve_ivp. ¿En qué situaciones es mejor utilizar una u otra?

4) ¿Cómo se utiliza **Matplotlib** para graficar campos vectoriales? Realiza una búsqueda sobre la función quiver. Explica qué parámetros necesita y cómo se puede representar un campo vectorial.

5) Toma una ecuación diferencial de la guía de trabajos prácticos y resuélvela usando SymPy y SciPy respectivamente. Compara los enfoques simbólico y numérico. ¿En qué casos preferirías usar uno u otro método?

6) Investiga cómo realizar la Transformada de Laplace utilizando SymPy. Explica el uso de las funciones laplace_transform e inverse_laplace_transform. ¿En qué aplicaciones prácticas crees que esta herramienta es útil?

7) Explora la librería SymPy y describe cómo se construyen y manipulan matrices simbólicas. Proporciona ejemplos en los que las entradas de una matriz no sean números, sino variables simbólicas.

8) Investiga cómo SciPy puede utilizarse para resolver sistemas de ecuaciones diferenciales. Busca ejemplos en los que se resuelvan sistemas acoplados de ecuaciones utilizando solve_ivp. Explica brevemente los pasos clave.

9) Explora una visualización interactiva de funciones o campos vectoriales usando Plotly o Mayavi. Busca ejemplos de gráficos 3D y describe las herramientas necesarias para realizar estos gráficos en Python.

Etapa 3: Actividades de aplicación

Formato de presentación
Cuaderno en Jupyter o Google Colab titulado 'GrupoX_TPsistemas_Actividades'. No se permite el uso de IA para resolver los problemas; el desarrollo debe ser propio. Si se utiliza IA como apoyo técnico (por ejemplo, para depurar código), deben anexarse los prompts utilizados.

Resolución simbólica con SymPy

1) Con el siguiente sistema:

$$x'(t) = 2x(t) + y(t)$$
$$y'(t) = -x(t) + y(t)$$

Desarrolle el código necesario para resolver las siguientes cuestiones:

a) Calcular los autovalores y auto vectores de la matriz asociada.

b) Clasificar el punto crítico (0,0).
c) Graficar el campo vectorial.
d) Representar algunas soluciones para diferentes condiciones iniciales.

2) Dado el siguiente sistema de ecuaciones

$$\begin{pmatrix} x_1'(t) \\ x_2'(t) \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 \\ -1 & -k \end{pmatrix}}_{A} \begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix}$$

Desarrolle el código necesario para analizar la estabilidad para los siguientes valores de k

a. $k > 2$
b. $k = 2$
c. $0 < k < 2$
d. $k = 0$
e. $-2 < k < 0$
f. $k = -2$
g. $k < -2$

3) Problema de aplicación
Utilice las herramientas matemáticas y de software estudiados para la resolución del siguiente problema.
En esta parte, es importante que no utilices la IA, ya que se espera que puedas comprender y aplicar los conceptos matemáticos y las herramientas de programación por ti mismo.

Problemas con el piloto automático

Consideremos un mecanismo que modela el piloto automático de un vehículo. Este mecanismo aplica un momento de torsión (movimiento rotacional) al eje de dirección, de modo que el vehículo seguirá un curso establecido con anterioridad (servomecanismo).
Si $y(t)$ es la dirección real (el ángulo) del vehículo en el instante t y $g(t)$ es la dirección deseada en el mismo instante, entonces el error entre la dirección deseada y la real viene dado por $e(t) = y(t) - g(t)$.
Supongamos ahora que el mecanismo puede medir el error, $e(t)$, y retroalimentar al eje de dirección mediante un componente del momento de torsión proporcional a $e(t)$, pero opuesto en signo. Ahora bien, sabemos que la segunda ley de Newton, expresada en términos de momentos de torsión, establece que:

$$(\text{ momento de inercia }) \times (\text{ aceleración angular }) = (\text{ momento de torsión total })$$

Por lo que, si denotamos $I \in \mathbb{R}$ como el momento de inercia del eje de dirección y consideramos $k > 0$ constante, entonces la ecuación que modela la dirección que sigue el vehículo a lo largo del tiempo viene dada por:

$$Iy'' = -ke$$

Consideramos que el eje de dirección está inicialmente en reposo en la dirección cero y la dirección deseada está dada por $g(t) = at$, donde $a \in \mathbb{R}$.

Comprensión del modelo

1. Explicar qué representan las siguientes variables y constantes:
   - y(t): …………………………………………….
   - g(t)= at: ……………………………………….
   - e(t)= g(t)−y(t): ………………………………………………..
   - I: ……………………………………………………….
   - k: ………………………………………………………………..

2. ¿Por qué se dice que el sistema funciona como un servomecanismo?

Planteo matemático

3. Escribir la ecuación diferencial que modela el sistema con sus condiciones iniciales.
4. ¿Qué tipo de sistema es? ¿Lineal/homogéneo?

Resolución analítica con transformada de Laplace

5. Aplicar la transformada de Laplace a la ecuación diferencial.
6. Resolver para E(s).
7. Calcular la transformada inversa y obtener e(t).

Análisis de estabilidad

8. Analizar cómo depende el comportamiento del error de los valores de k y I (casos: k > 0, k = 0, k < 0).
9. ¿El sistema es estable, asintóticamente estable o inestable? Justificar con autovalores.
10. Relacionar la clasificación con nodo, foco, centro o punto de silla.

Resolución numérica y gráfica en Python

11. Resolver el problema con SymPy y SciPy.solve_ivp.
12. Graficar e(t) para distintos valores de k y J.
13. Representar el plano de fases (e(t) vs e'(t)).

Reflexión

14. ¿Por qué aparecen oscilaciones en el error?
15. ¿Qué sucede cuando se aumenta k?
    ¿Qué limitaciones presenta el modelo? ¿Cómo podría mejorarse para lograr mayor estabilidad?

Seguimos analizando

Consideremos que se añade sobre el eje de dirección un componente adicional del momento de torsión proporcional a $e'(t)$ y opuesto en signo, para así poder controlar las oscilaciones. Así, si consideramos $\mu > 0$, la ecuación que modela ahora la dirección que sigue el vehículo a lo largo del tiempo viene dada por:

$$Iy'' = -ke - \mu e'$$

Vamos a calcular de nuevo el error $e(t)$ para el piloto automático en el caso particular en el que el eje de dirección está inicialmente en reposo en la dirección cero y la dirección deseada está dada por $g(t) = at$, donde $a \in \mathbb{R}$. Consideraremos además amortiguamiento ligero $(\mu < 2\sqrt{Ik})$. Para dicho cálculo, tenemos entonces que resolver el siguiente problema de valores iniciales:

$$\begin{cases} Iy''(t) = -k(y(t) - at) - \mu(y'(t) - a) \\ y(0) = 0 \\ y'(0) = 0 \end{cases}$$

16. Encuentre Y(t).
17. Recuerde que $E(t) = Y(t) - \frac{a}{t^2}$ encuentre la fuencion del error e(t) y analice sus parámetros para estabilizar el error.

Conclusiones

18. Elabore conclusiones de los dos análisis.

Presentación integradora final (video)

Formato de entrega: Video explicativo de 5 a 8 minutos (subido al canal de YouTube indicado por la catedra).
Título sugerido: 'GrupoX_TPsistemas_VideoPresentacion'.
Debe incluir a todos los integrantes del grupo tanto en los créditos como en la descripción del video.
Contenido esperado del video:

- Presentación breve del problema trabajado.
- Explicación del modelo matemático y del método de resolución.
- Descripción de las herramientas de Python utilizadas.
- Visualización de resultados numéricos y gráficos.
- Conclusiones y reflexiones sobre la estabilidad del sistema y el proceso de aprendizaje.

Criterios de evaluación:

- Claridad conceptual y argumentativa.
- Coherencia entre teoría, código y resultados.
- Creatividad y originalidad en la exposición.
- Correcta interpretación de los gráficos y conclusiones.
- Participación equitativa del grupo.