|  |  |
| --- | --- |
|  |  |
| TECNICATURA SUPERIOR EN  **Telecomunicaciones** |  |
| Espacio: Sistemas de Control y Servicios - 2024 | |
| Alumno: Dario Arriola  Profesor: Gonzalo Vera | |
| Objetivos:  • Comprender los conceptos de sistema dinámico, lazo abierto y lazo cerrado.  • Analizar las ventajas y desventajas de los sistemas de control dinámico.  • Conocer los sistemas de control estáticos y su comparación con los  dinámicos.  • Familiarizarse con los sistemas lineales invariantes en el tiempo y sus  propiedades matemáticas.  • Comprender cómo estas propiedades permiten una mejor comprensión y  control de los sistemas. | |

# TRABAJO PRÁCTICO N° 3

## Crear un programa en Python utilizando la biblioteca NumPy para simular y analizar un sistema de control dinámico en lazo abierto. Realizar una comparación con el mismo sistema en lazo cerrado:

import numpy as np

import matplotlib.pyplot as plt

# Parámetros del sistema

capacidad\_bateria = 1000  # Capacidad de la batería en mAh

tiempo\_carga\_horas = 12  # Tiempo de carga deseado en horas

corriente\_max = capacidad\_bateria / tiempo\_carga\_horas  # Corriente máxima de carga en mA

tiempo\_simulacion = 24  # Tiempo de simulación en horas (más largo para visualización)

dt = 0.1  # Paso de tiempo en horas

error\_anterior = 0  # Inicializar el error anterior

def controlador\_p(error):

    kp = 0.1  # Ganancia proporcional del controlador P

    correccion = kp \* error

    return correccion

# Inicializar variables para lazo cerrado

tiempo = np.arange(0, tiempo\_simulacion, dt)

carga\_bateria\_cerrado = np.zeros\_like(tiempo)

carga\_actual\_cerrado = 0

# Simulación en lazo cerrado

for t in range(len(tiempo)):

    error = capacidad\_bateria - carga\_actual\_cerrado

    correccion = controlador\_p(error)

    corriente\_carga = min(corriente\_max, correccion)  # Limitar la corriente máxima

    carga\_actual\_cerrado += corriente\_carga \* dt

    carga\_bateria\_cerrado[t] = carga\_actual\_cerrado

# Inicializar variables para lazo abierto

carga\_bateria\_abierto = np.zeros\_like(tiempo)

carga\_actual\_abierto = 0

# Simulación en lazo abierto

for t in range(len(tiempo)):

    corriente\_carga\_abierto = corriente\_max  # Utilizar corriente máxima en lazo abierto

    carga\_actual\_abierto += corriente\_carga\_abierto \* dt

    carga\_bateria\_abierto[t] = carga\_actual\_abierto

# Visualización de resultados

plt.figure(figsize=(10, 6))

plt.plot(tiempo, carga\_bateria\_abierto, label='Lazo Abierto')

plt.plot(tiempo, carga\_bateria\_cerrado, label='Lazo Cerrado')

plt.axhline(capacidad\_bateria, color='r', linestyle='--', label='Capacidad máxima')

plt.xlabel('Tiempo (horas)')

plt.ylabel('Carga de la batería (mAh)')

plt.title('Comparación entre Lazo Abierto y Lazo Cerrado para Carga de Batería')

plt.legend()

plt.grid(True)

plt.show()

## 