**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**

FACULTAD DE CIENCIAS E INGENIERÍA

SECCIÓN DE ELECTRICIDAD Y ELECTRÓNICA

TEORÍA DE CONTROL 2

Laboratorio N°4



**Laboratorio de simulación de sistemas de control digital**

**Salvador Yábar**

**20200408**

**H0821**

2024-1

1. **Objetivos**

* Determinar un periodo de muestreo apropiado para el sistema
* Discretizar el controlador en Matlab
* Simular el controlador discreto

1. **Desarrollo**
2. **Simular el controlador continuo**

Se implementó el controlador continuo mediante el bloque *transfer function* en Simulink. Se realizó el esquema que se muestra en la figura 1, colocando como referencia 0.1 radianes, como plantea la guía.

A drawing of a boat

Description automatically generated

A partir de la simulación, se obtuvo la siguiente gráfica en la que se observan las 3 salidas:

A screen shot of a computer

Description automatically generated

Fig 2. Gráfica de las variables de estado del sistema

Se observa que el menor tiempo de subida lo tiene la salida que corresponde al ángulo de las aletas.

A continuación, se analiza el tiempo de subida para cada salida.

Para el ángulo del buque, se tiene un máximo de 1.176e-01. Se colocan los cursores en los valores correspondientes al 10 y 90%, y se obtiene un tiempo de subida de 306.164 ms.

A screen shot of a black screen

Description automatically generated

Fig 3. Gráfica de x1

De forma similar, para x2 se tiene un máximo de 3.992e-01. Se obtiene un tiempo de subida de 60.19 ms.

A graph on a black background

Description automatically generated

Fig 4. Gráfica de x2

Finalmente, para x3 se tiene un máximo de 9.304, y se obtiene un tiempo de subida de 20.298 ms.

A screen shot of a graph

Description automatically generated

Fig 5. Gráfica de x3

A partir de esto, se confirma que el tiempo de subida menor es el de x3. Se emplea este valor para obtener el periodo de muestreo, dividiendo el tiempo de subida entre 10.

**b) Discretizar el controlador**

Se determina el periodo de muestreo, y mediante la función c2d se obtiene el controlador discreto Gz.

A white background with black text

Description automatically generated

Fig 6. Código para obtener controlador discreto

A black and white text with numbers

Description automatically generated

Fig 7. Controlador discreto

Luego, se emplea el bloque de función de transferencia discreta para implementar el controlador discreto en Simulink, como se muestra en la figura 8.

A drawing of a ship

Description automatically generated

Fig 8. Simulink con controlador discreto

Se presenta la gráfica correspondiente a las variables de estado del sistema, empleando el controlador discreto.

A graph with lines and numbers

Description automatically generated with medium confidence

Fig 9. Gráfica de variables de estado

**c) Simulación en Matlab**

Primero, se discretiza la planta mediante el comando c2d.

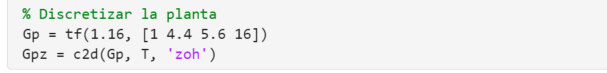
****

Fig 10. Código para discretizar la planta

A math equations and numbers

Description automatically generated with medium confidence

Fig 11. Planta discretizada

Se obtiene la función de transferencia en lazo cerrado mediante el comando feedback.



Fig 12. Lazo cerrado

A close-up of numbers

Description automatically generated

Fig 13. Función de transferencia en lazo cerrado

Se obtiene la ecuación en diferencias en base a la función de transferencia

q = 7898\*r - 3.928e04\*r\_1 + 7.815e04\*r\_2 - 7.774e04\*r\_3 + 3.866e04\*r\_4 - 7691\*r\_5 + 4.798\*q\_1 - 9.203\*q\_2 + 8.822\*q\_3 + 4.225\*q\_4 - 0.809\*q\_5;

Se muestra el código para obtener los valores de q(k) a partir de la ecuación en diferencias.

A white screen with black text

Description automatically generated

Fig 14. For loop para la ecuación en diferencias

Sin embargo, la gráfica obtenida no es la esperada. Esta debería asemejarse al resultado obtenido en el inciso b, ya que se trata del mismo controlador discreto y la misma planta.

**A graph with numbers and lines

Description automatically generated**

Fig 15. Gráfica de q

1. **Conclusiones**

En conclusión, se pudo determinar un periodo de muestreo apropiado para el sistema, en base al tiempo de subida de la variable más rápida. Al comparar las 3 variables de estado, se determinó que x3 era la más rápida, y se empleó un periodo de muestreo de la décima parte de su tiempo de subida.

Empleando el comando c2d en Matlab se pudo obtener el controlador discreto y simularlo con la planta en Simulink. Se obtuvo un resultado similar al de la planta con controlador continuo.

Finalmente, se discretizó la planta empleando el comando c2d, pero al realizar la gráfica se obtuvo un resultado no esperado.