Partiendo de las ecuaciones que describen el comportamiento del sistema no lineal

Ahora se procede a calcular las derivadas parciales para estas ecuaciones y posteriormente evaluar en el punto de equilibrio z ̅ = [00]T

Ahora se reemplazan los valores del punto de equilibrio para obtener el jacobiano

Finalmente se obtiene el espacio de estados linealizado alrededor de un punto de la siguiente manera

Ahora se procede a calcular la función de transferencia del sistema a partir de la siguiente igualdad

Inicialmente se resuelve lo que está al interior del paréntesis

Esta es la función de transferencia de la abierto, ahora se calcula función de transferencia de lazo cerrado, a partir de la siguiente ecuación

Finalmente tenemos que H(s) es la función de transferencia de lazo cerrado del sistema

Dado que

Donde y OJO EL DENOMINADOR DEBE SER POSITIVO O SI NO EL COMPORTAMIENTO ESTARA EL 4 CUANDRANTE, yo cambie el signo y reemplace épsilon.

Luego de realizar la sustitución de

primero necesitamos determinar los polos deseados del sistema cerrado. En este caso, queremos un tiempo de establecimiento mínimo de Ts = 2 seg, un overshoot máximo de Mp = 10%, y una precisión estática de estado cero (ess) igual a cero.

Dado que se menciona que los bloques filtro tienen un valor de 1, podemos omitirlos en el diseño del controlador algebraico. Por lo tanto, la función de transferencia del sistema se simplifica a:

G(s) = 1/(s^2 + s)

Para lograr el tiempo de establecimiento mínimo y el overshoot máximo, utilizaremos una técnica de asignación de polos. Los polos deseados se seleccionan en función de los requisitos de desempeño. En este caso, queremos un tiempo de establecimiento mínimo de Ts = 2 seg y un overshoot máximo de Mp = 10%.

El polinomio característico deseado se puede escribir como:

P\_desired(s) = (s + zeta \* omega\_n) ^ 2

Donde zeta es el factor de amortiguamiento y omega\_n es la frecuencia natural no amortiguada.

Para un overshoot máximo de Mp = 10%, el factor de amortiguamiento zeta se puede calcular como:

zeta = sqrt((log(Mp/100)^2) / (pi^2 + log(Mp/100)^2))

Para un tiempo de establecimiento mínimo de Ts = 2 seg, la frecuencia natural no amortiguada omega\_n se puede calcular como:

omega\_n = 4 / (zeta \* Ts)

Sustituyendo los valores, tenemos:

zeta = sqrt((log(10/100)^2) / (pi^2 + log(10/100)^2)) = 0.5916

omega\_n = 4 / (0.5916 \* 2) = 3.387

El polinomio característico deseado se convierte en:

P\_desired(s) = (s + 0.5916 \* 3.387)^2 = s^2 + 4.019s + 4.02

Ahora podemos encontrar el controlador algebraico utilizando la técnica de asignación de polos. El controlador algebraico se define como:

C(s) = (P\_desired(s) - G(s)) / F(s)

Donde F(s) es el filtro que está presente en el lazo de control.

Reemplazando los valores, tenemos:

C(s) = (s^2 + 4.019s + 4.02 - 1/(s^2 + s)) / 1

Simplificando:

C(s) = (s^4 + 4.019s^3 + 4.02s^2 - 1) / (s^2 + s)

A picture containing text, line, plot, diagram

Description automatically generated

Fig. Respuesta al escalón unitario con el controlador diseñado por asignación de polos

**Realizar el diseño e implementacion matematica de una ley de control por realimentacion a la salida, al sistema de oscilador de**

**van der pool donde su funcion de transferencia linealizada esta dada por:**

**(1)/(s^2+s). diseñando una retroalimentacion de estados y un estimador de luenberger. Obtenga las ganancias del controlador**

**y observador considerando el polinomio del bloque controlador dado por :C(s) = (s^4 + 4.019s^3 + 4.02s^2 - 1) / (s^2 + s).**

Dada la función de transferencia linealizada del sistema:

G(s) = 1/(s^2 + s)

Para obtener la forma canónica controlable del sistema con la función de transferencia linealizada G(s) = 1/(s^2 + s) , primero necesitamos escribir la ecuación de espacio de estado correspondiente.

Dada la función de transferencia G(s), podemos obtener la ecuación de espacio de estado en la forma general:

x˙ =Ax+Bu

y=Cx+Du

Donde:

x es el vector de estados.

˙x˙ es la derivada del vector de estados con respecto al tiempo.

A es la matriz de coeficientes de estado.

B es la matriz de coeficientes de entrada.

C es la matriz de coeficientes de salida.

D es la matriz de coeficientes de realimentación directa.

En este caso, dado que G(s) es de segundo orden, necesitamos introducir una variable de estado adicional para obtener una forma canónica controlable completa. Entonces, el vector de estados

x se define como:

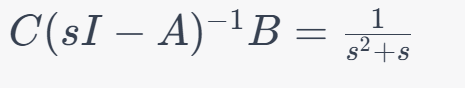
x=[x1 x2 ] T

La función de transferencia

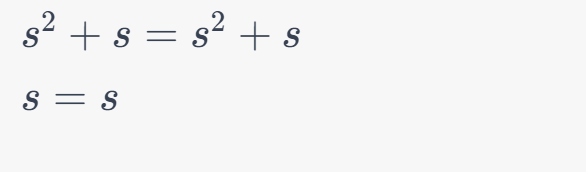
G(s) se puede expresar en términos de las matrices de estado y de entrada de la siguiente manera:



Comparando esto con la función de transferencia dada, obtenemos:

​

Comparando coeficientes, tenemos:



Esto nos dice que la matriz A es:

A=[ 0 -1

0 0]

La matriz B se puede obtener sustituyendo s=0 en la ecuación de transferencia:

B=[ 1

0 ]

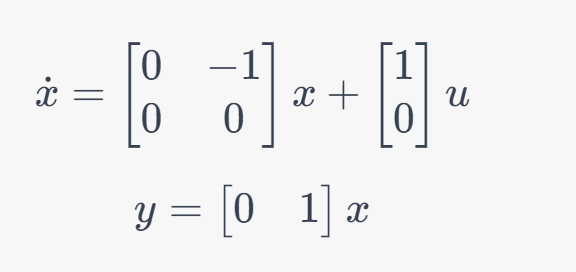
La matriz C se puede obtener sustituyendo s=0 en la ecuación de transferencia:

C= [ 0 1]

Finalmente, la matriz D se puede obtener sustituyendo s=0 en la ecuación de transferencia:

D=0

Entonces, la ecuación de espacio de estado en forma canónica controlable para el sistema dado es:



El controlador por realimentación de estados se define como:

u = -Kx

Donde K es una matriz de ganancias a determinar. Para encontrar la matriz de ganancias K, podemos utilizar el método de la asignación de polos.

El polinomio del bloque controlador se da por:

C(s) = (s^4 + 4.019s^3 + 4.02s^2 - 1) / (s^2 + s)

Dado que estamos utilizando una retroalimentación de estados, la función de transferencia de lazo cerrado del sistema se puede expresar como:

Gc(s) = (C(s)G(s)) / (1 + C(s)G(s))

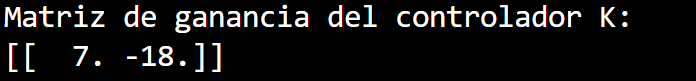
Reemplazando los valores de C(s) y G(s) en esta expresión, podemos obtener la función de transferencia de lazo cerrado del sistema:

Gc(s) = (s^4 + 4.019s^3 + 4.02s^2 - 1) / (s^4 + 4.019s^3 + 4.02s^2)

El polinomio característico deseado se puede obtener a partir del denominador de Gc(s). En este caso, el polinomio característico deseado es:

p\_desired = [s^4 + 4.019s^3 + 4.02s^2]

Usando el método de la asignación de polos, podemos calcular la matriz de ganancias K para que los polos del sistema en lazo cerrado coincidan con los polos deseados. En este caso, se deben asignar los polos del sistema a los polos del polinomio característico deseado.



Ahora, vamos a calcular las ganancias del estimador utilizando la técnica de colocación de polos. La ganancia del estimador se puede calcular como:

L = [l1

l2]

Donde l1 y l2 son las ganancias para determinar.

A black background with white text

Description automatically generated with low confidence

Aplicando la fórmula de la ganancia del estimador, obtenemos:

L = [λ3 + λ4 -1 0]

Finalmente, el estimador de Luenberger es:

x'(t) = Ax(t) + Bu(t) + L(y(t) - Cx(t))

A graph with blue and orange lines

Description automatically generated with low confidence

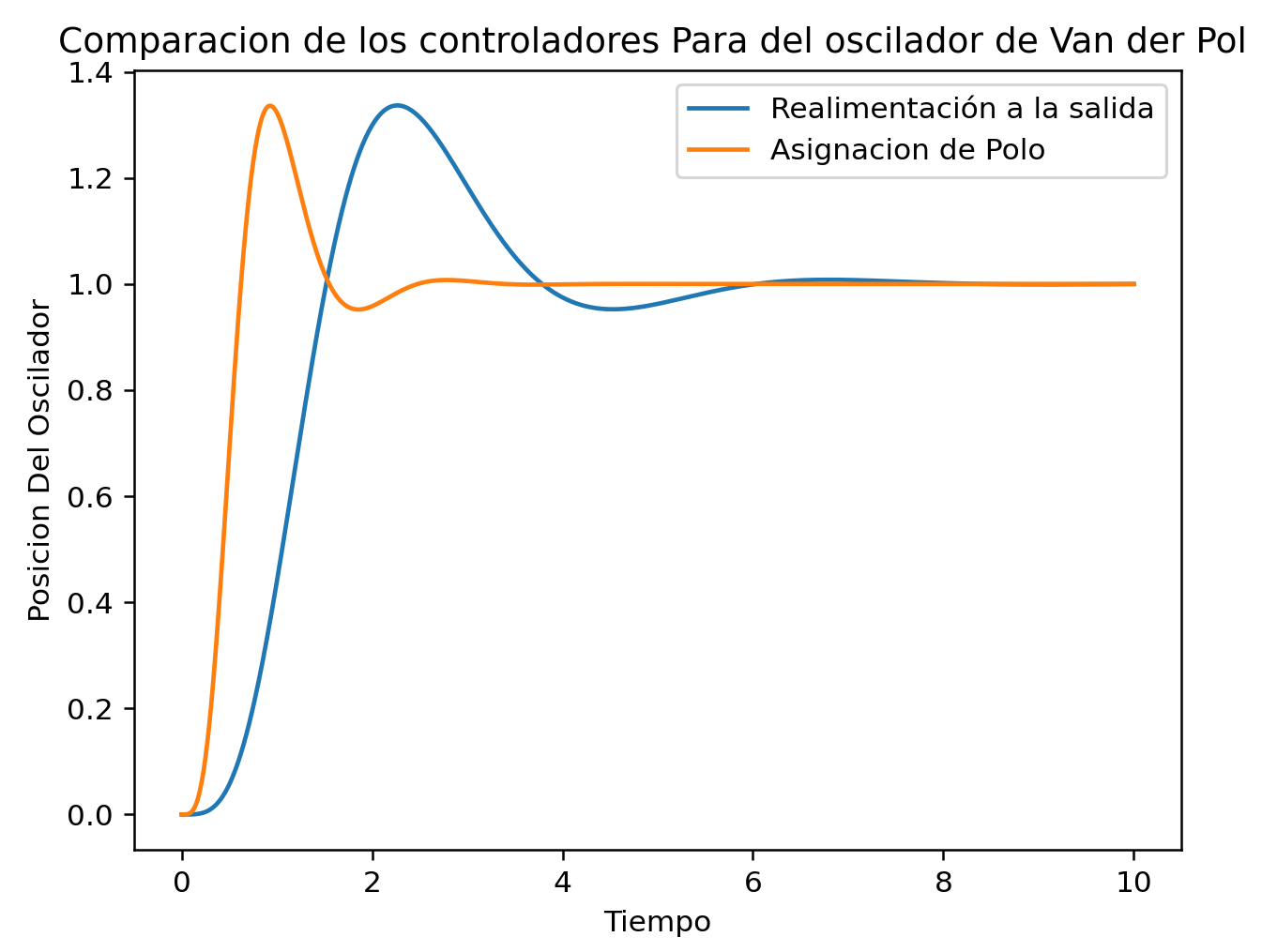
Fig. Respuesta al escalón unitario con el controlador diseñado la realimentación dinámica a la salida

Comparación por energía y robustes de los controladores

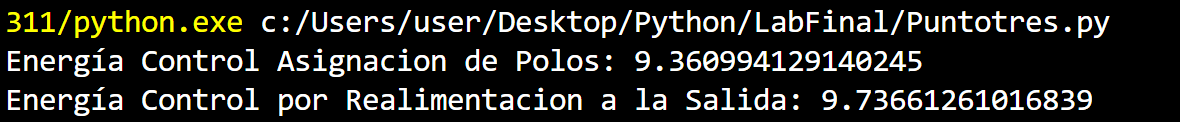
Para calcular la energía de un sistema representado por su respuesta al escalón, puedes utilizar la siguiente fórmula:

Energía = ∫(y(t))^2 dt

Donde y(t) es la respuesta al escalón unitario del sistema y la integral se realiza en el intervalo de tiempo deseado.



Comparación de los 2 controladores.



Realimentación a la salida

Tiempo de establecimiento: El sistema 1 parece estabilizarse alrededor de 5 segundos después del escalón unitario, aunque hay una pequeña fluctuación después de ese tiempo. El tiempo de establecimiento podría considerarse aceptable en este caso.

Overshoot: En el sistema 1, se observa un pequeño overshoot en la respuesta al escalón. Sin embargo, el nivel de sobrepaso es relativamente bajo y la respuesta parece suave y controlada.

Oscilaciones: Después de estabilizarse, el sistema 1 muestra algunas oscilaciones pequeñas, pero estas oscilaciones son limitadas y no parecen ser significativas.

Precisión: La respuesta del sistema 1 se ajusta bastante bien a la referencia después de estabilizarse. Hay una ligera diferencia, pero en general, la precisión en el seguimiento de la referencia es aceptable.

Asignacion de Polo:

Tiempo de establecimiento: El sistema 2 parece estabilizarse alrededor de 3 segundos después del escalón unitario. El tiempo de establecimiento es relativamente rápido en comparación con el sistema 1.

Overshoot: En el sistema 2, se observa un mayor overshoot en la respuesta al escalón en comparación con el sistema 1. Esto indica una respuesta menos controlada y puede ser considerado un aspecto menos deseable en términos de estabilidad.

Oscilaciones: Después de estabilizarse, el sistema 2 muestra algunas oscilaciones más pronunciadas en comparación con el sistema 1. Estas oscilaciones podrían indicar una respuesta menos estable.

Precisión: La respuesta del sistema 2 se ajusta razonablemente bien a la referencia después de estabilizarse. Sin embargo, la diferencia entre la respuesta y la referencia es ligeramente mayor en comparación con el sistema 1.

Conclusiones.

La exploración, análisis e implementación de sistemas de control en Python para sistemas dinámicos lineales y no lineales ha demostrado ser efectiva y versátil. Python ofrece una amplia gama de bibliotecas y herramientas, como NumPy, SciPy y control, que facilitan el diseño y la implementación de controladores para diferentes tipos de sistemas. Esto permite a los ingenieros y científicos trabajar de manera eficiente en el desarrollo de soluciones de control.

Los algoritmos numéricos desarrollados en Python para el análisis de sistemas dinámicos continuos son de gran utilidad. Mediante el uso de métodos numéricos, como la integración numérica y los esquemas de discretización, es posible estudiar y comprender el comportamiento de los sistemas dinámicos continuos. Python proporciona las herramientas necesarias para implementar estos algoritmos de manera eficiente y precisa.

La implementación de rutinas de programación en Python para la simulación de un sistema de control en el dominio de la frecuencia y el tiempo ha permitido evaluar el rendimiento del sistema y del controlador diseñado. Python ofrece bibliotecas como matplotlib y control, que facilitan la visualización de las respuestas en el dominio de la frecuencia y el tiempo. Estas rutinas de programación han sido fundamentales para evaluar el desempeño del controlador y ajustarlo si es necesario.

La implementación y validación de sistemas dinámicos lineales y no lineales utilizando Python ha demostrado ser exitosa. Python proporciona las herramientas necesarias para modelar y simular sistemas dinámicos de manera precisa y eficiente. La validación del controlador diseñado con Python ha permitido comprobar su desempeño y realizar ajustes si es necesario. Esta capacidad de implementación y validación en Python brinda confianza en la efectividad y eficacia de los sistemas de control desarrollados.