



coba bikin state space dps dari persamaan

$$m \dot{v} + cv + dv = \tau + \tau_{\text{env}}$$

dengan state nya adalah v dan \dot{v}



Kita dapat merepresentasikan persamaan yang diberikan dalam bentuk **state-space representation**. Langkah-langkahnya adalah mendefinisikan state variabel terlebih dahulu, kemudian menuliskan persamaan dalam bentuk matriks state-space.

Persamaan:

$$m\dot{v} + cv + dv = \tau + \tau_{\text{env}}$$

Dimisalkan:

1. $\mathbf{x}_1 = v$ (kecepatan)
2. $\mathbf{x}_2 = \dot{v}$ (percepatan)

Sehingga, state vector adalah:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} v \\ \dot{v} \end{bmatrix}$$

Persamaan \dot{v} dapat ditulis ulang sebagai:

$$m\dot{x}_2 + cx_1 + dx_2 = \tau + \tau_{\text{env}}$$

$$\dot{x}_2 = \frac{1}{m}(\tau + \tau_{\text{env}} - cx_1 - dx_2)$$

Kemudian, representasi state-space:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{1}{m}(\tau + \tau_{\text{env}} - cx_1 - dx_2) \end{bmatrix}$$

Atau dalam bentuk matriks:

$$\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$$

dengan $\mathbf{u} = \tau + \tau_{\text{env}}$, dan:

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{c}{m} & -\frac{d}{m} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}$$

Persamaan Output (Jika Outputnya adalah $v = x_1$):

$$\mathbf{y} = C\mathbf{x} + D\mathbf{u}$$

dengan:

$$C = [1 \quad 0], \quad D = [0]$$

Hasil Akhir

State-space representation:

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -\frac{c}{m} & -\frac{d}{m} \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} (\tau + \tau_{\text{env}})$$

$$\mathbf{y} = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}$$

coba codingan python nya dengan simulasi step



Created document

Saya telah membuat kode Python untuk simulasi step response dari sistem yang didefinisikan dalam bentuk state-space. Anda bisa menyalin kode tersebut dan menjalankannya untuk melihat hasil grafik respons step. Jika ada kebutuhan modifikasi, beri tahu saya!

m c d nya 3x3 matriksnya



Edited

Kode telah diperbarui untuk menangani sistem dengan matriks m , c , dan d berukuran 3×3 . Simulasi step response sekarang memperhitungkan sistem dengan matriks yang lebih besar. Anda bisa mencobanya, dan beri tahu saya jika ada yang perlu disesuaikan!

apakah tau nya tau x, tau y, dan tau N



Ya, dalam sistem dengan matriks m , c , dan d berukuran 3×3 , gaya input (τ) akan menjadi vektor berukuran 3, yang dapat direpresentasikan sebagai:

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_N \end{bmatrix}$$

Di mana:

- τ_x adalah gaya atau torsi pada arah x ,
- τ_y adalah gaya atau torsi pada arah y ,
- τ_N adalah gaya atau torsi pada arah rotasi (atau torsi netral, bergantung pada sistem).

Dalam kode simulasi yang telah diperbarui, τ diwakili oleh input u , yang merupakan vektor waktu berukuran 3 (untuk setiap komponen τ_x , τ_y , dan τ_N). Sistem akan mensimulasikan respons dari setiap komponen gaya input terhadap kecepatan (v_x , v_y , v_N).

Jika Anda ingin mendefinisikan nilai khusus untuk τ_x , τ_y , dan τ_N , Anda dapat memodifikasi matriks input u di bagian berikut:



python

Copy Edit

```
u = np.ones((len(time), 3)) # Step input: vektor ukuran 3 untuk setiap waktu
```

Misalnya, jika Anda ingin hanya memberikan gaya pada arah x (τ_x):

python

 Copy  Edit

```
u = np.zeros((len(time), 3)) u[:, 0] = 1 # Step input hanya pada \(\tau_x\)
```

apakah ini valid



Ya, pendekatan ini **valid**, dengan asumsi bahwa sistem m , c , d , dan gaya input $\tau = [\tau_x, \tau_y, \tau_N]^T$ memang memiliki hubungan yang linier seperti yang dimodelkan dalam **state-space representation**.

Validasi:

1. Dimensi Konsisten:

- m , c , dan d adalah matriks 3×3 .
- Input τ adalah vektor 3×1 .
- Kecepatan (v_x, v_y, v_N) dan percepatan ($\dot{v}_x, \dot{v}_y, \dot{v}_N$) juga berukuran 3×1 .
- Matriks state-space dibangun dengan dimensi konsisten:
 - $A : 6 \times 6$,
 - $B : 6 \times 3$,
 - $C : 3 \times 6$,
 - $D : 3 \times 3$.

2. Prinsip Sistem Linier:

- Model ini mengasumsikan hubungan linier antara gaya (τ) dan respons sistem (v dan \dot{v}).
- Ini sesuai untuk banyak sistem mekanik dengan massa, redaman, dan gesekan yang dijelaskan oleh persamaan diferensial linier.

3. Input Multidimensional:

- Representasi input τ_x, τ_y, τ_N dalam u sebagai vektor 3 dimensi valid untuk menangkap gaya/torque dalam arah berbeda.

Apa yang Perlu Dicek?

1. Sifat Fisik Matriks m :

- Matriks m harus positif-definit, yang berarti semua nilai eigen harus positif.
- Jika tidak, sistem akan memiliki properti fisik yang tidak masuk akal (misalnya, massa negatif).

2. Kondisi Sistem:

- Pastikan sistem stabil untuk parameter m , c , dan d yang dipilih. Matriks A dalam state-space akan menentukan sifat dinamika sistem.

3. Pengaruh Lingkungan (τ_{env}):

- Jika ada gangguan eksternal (seperti lingkungan), ini perlu dimodelkan secara eksplisit.

Contoh Pengujian

Untuk memvalidasi, Anda bisa mencoba input tertentu (misalnya, step input hanya pada τ_x) dan memeriksa apakah respons sistem (v_x , v_y , v_N) sesuai dengan yang diharapkan secara fisik. Jika sistem tidak stabil atau respons tidak masuk akal, mungkin ada kesalahan dalam parameter atau pemodelan.

Jika Anda memiliki kekhawatiran spesifik, saya bisa membantu menelusuri atau menyempurnakan modelnya. 😊