Model stanowy układu z zad 6

5 września 2025

Filip David 193204 Piotr Marwitz 193606

Opis układu

Rozważany jest układ dwóch bezwładności obrotowych J_1 i J_2 połączonych idealną (sztywną) przekładnią o przełożeniu $n_1 : n_2$. Po stronie wałów występują liniowe tłumienia do podłoża b_1 i b_2. Wejściem jest moment napędowy τ_m po stronie J_1.

Relacje kinematyczne przekładni sztywnej

Dla przekładni idealnej zachodzi

$$n_1\,\theta_1 = n_2\,\theta_2 \quad \Rightarrow \quad \theta_2 = \frac{n_1}{n_2}\,\theta_1, \qquad \omega_2 = \dot{\theta}_2 = \frac{n_1}{n_2}\,\omega_1. \tag{1}$$

Parametry zastępcze

Redukując do jednej osi przez to że przekładnia jest sztywna (po stronie 1) otrzymujemy:

$$J_{\text{eq}} = J_1 + J_2\left(\frac{n_1}{n_2}\right)^2, \qquad b_{\text{eq}} = b_1 + b_2\left(\frac{n_1}{n_2}\right)^2. \tag{2}$$

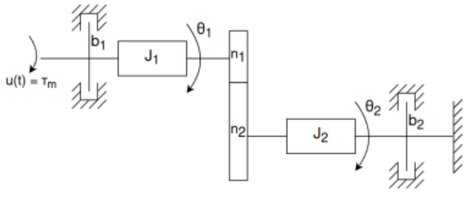
Równania ruchu

$$J_{\text{eq}}\,\dot{\omega}_1 + b_{\text{eq}}\,\omega_1 = \tau_m, \qquad \dot{\theta}_1 = \omega_1. \tag{3}$$

Model w przestrzeni stanów

Wybieramy stan i wejście:

$$x = \begin{bmatrix}\theta_1\\ \omega_1\end{bmatrix}, \qquad u = \tau_m. \tag{4}$$



Rysunek 1: schemat z zadania 6

Model:

$$\dot{x} = A\,x + B\,u, \qquad y = C\,x + D\,u, \tag{5}$$

gdzie (dla wyjścia $y = [\,\theta_2 \ \ \omega_2\,]^\top$):

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -\dfrac{b_{\mathrm{eq}}}{J_{\mathrm{eq}}} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \dfrac{1}{J_{\mathrm{eq}}} \end{bmatrix}, \quad C = \begin{bmatrix} \dfrac{n_1}{n_2} & 0 \\ 0 & \dfrac{n_1}{n_2} \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{6}$$

Schematy całkowania numerycznego (Euler i RK4)

Rozważmy ogólnie równanie stanu

$$\dot{x}(t) = f(t, x(t), u(t)), \qquad x(t_k) = x_k, \quad h = t_{k+1} - t_k. \tag{7}$$

Metoda Eulera.

$$x_{k+1} = x_k + h\,f(t_k, x_k, u_k). \tag{8}$$

Dla układu liniowego $f(t, x, u) = Ax + Bu$:

$$x_{k+1} = (I + hA)\,x_k + h\,B\,u_k. \tag{9}$$

Runge–Kutta rzędu 4 (RK4).

$$k_1 = f(t_k,\ x_k,\ u_k), \tag{10}$$

$$k_2 = f\Big(t_k + \tfrac{h}{2},\ x_k + \tfrac{h}{2}k_1,\ u_k\Big), \tag{11}$$

$$k_3 = f\Big(t_k + \tfrac{h}{2},\ x_k + \tfrac{h}{2}k_2,\ u_k\Big), \tag{12}$$

$$k_4 = f(t_k + h,\ x_k + h\,k_3,\ u_k), \tag{13}$$

$$x_{k+1} = x_k + \frac{h}{6}\,(k_1 + 2k_2 + 2k_3 + k_4). \tag{14}$$

Dla układu liniowego $f = Ax + Bu$ (przy stałym u_k na kroku):

$$k_1 = Ax_k + Bu_k, \tag{15}$$

$$k_2 = A\Big(x_k + \tfrac{h}{2}k_1\Big) + Bu_k, \tag{16}$$

$$k_3 = A\Big(x_k + \tfrac{h}{2}k_2\Big) + Bu_k, \tag{17}$$

$$k_4 = A(x_k + h\,k_3) + Bu_k, \tag{18}$$

$$x_{k+1} = x_k + \frac{h}{6}\,(k_1 + 2k_2 + 2k_3 + k_4). \tag{19}$$

Wyjście dyskretne.

$$y_k = C\,x_k + D\,u_k. \tag{20}$$

Działanie programu (implementacja w Python)

Program realizuje symulację układu dwumasowego z przekładnią sztywną w oparciu o model stanowy z rozdziału wcześniejszego. Aplikacja została zbudowana w Pythonie z użyciem: `tkinter` (GUI), `NumPy` (obliczenia), `Matplotlib` (wykresy) oraz `Pillow` (wczytanie grafiki układu).

Struktura i główne komponenty

- **Funkcje modelu:** $f(t, x, u, A, B) = Ax + Bu$, $output(x, u, C, D) = Cx + Du$ (równanie wyjścia).
- **Integratory numeryczne:** *euler_step* (Euler) oraz *rk4_step* (Runge–Kutta rzędu 4) odwzorowujące odpowiednio wzory z sekcji *Schematy całkowania numerycznego*.
- **Generatory sygnału wejściowego** $u(t)$: sinusoidalny, prostokątny i trójkątny.

Interfejs użytkownika

W klasie `SimulatorApp` tworzony jest panel sterujący (lewa kolumna) i obszar wykresów (prawa strona):

- **Parametry układu:** $J_1, J_2, b_1, b_2, n_1, n_2$.
- **Parametry sygnału** (w zależności od typu): amplituda, okres (T) lub częstotliwość (f), faza, wypełnienie (dla przebiegu prostokątnego).
- **Parametry symulacji:** czas początkowy t_0, czas końcowy t_f, krok całkowania h, warunki początkowe $\theta_1(0)$ (w stopniach, wewnętrznie konwertowane do radianów) oraz $\omega_1(0)$ (rad/s).
- **Wizualizacja:** osadzony rysunek układu, trzy wykresy Matplotlib (sygnał wejściowy, kąt θ_2, prędkość ω_2), pasek narzędzi (zoom, zapis, przesuw).

Budowa modelu stanowego na podstawie danych

Po naciśnięciu *Symuluj* wywoływana jest metoda `run_simulation`:

1. Z formularza pobierane są $J_1, J_2, b_1, b_2, n_1, n_2$ i obliczane parametry zastępcze:

$$J_{\mathrm{eq}} = J_1 + J_2 \left(\frac{n_1}{n_2}\right)^2, \qquad b_{\mathrm{eq}} = b_1 + b_2 \left(\frac{n_1}{n_2}\right)^2.$$

2. Na tej podstawie tworzone są macierze stanu:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -b_{\mathrm{eq}}/J_{\mathrm{eq}} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 1/J_{\mathrm{eq}} \end{bmatrix}, \quad C = \begin{bmatrix} \frac{n_1}{n_2} & 0 \\ 0 & \frac{n_1}{n_2} \end{bmatrix}, \quad D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

3. Generowana jest siatka czasu $t\_k \in [t_0, t_f)$ z krokiem h (funkcja `np.arange`).
4. Warunek początkowy: $x_0 = \left[\theta_1(0),\ \omega_1(0)\right]^\top$, gdzie $\theta_1(0)$ konwertowane jest ze stopni do radianów.

Sygnał wejściowy $u(t)$

W zależności od wyboru użytkownika ustawiana jest funkcja:

$$\text{sinus: } u(t) = A\sin(2\pi f t + \varphi),$$
$$\text{prostokąt: } u(t) = \begin{cases} +A, & \text{jeśli } \operatorname{fract}\left(\frac{t+\varphi}{T}\right) < d, \\ -A, & \text{w przeciwnym razie,} \end{cases}$$
$$\text{piła: } u(t) = A\left(2\operatorname{fract}\left(\frac{t+\varphi}{T}\right) - 1\right),$$

$$\text{trójkąt (symetryczny):} \qquad u(t) = \begin{cases} A\Big(4\,\mathrm{fract}(\frac{t+\varphi}{T}) - 1\Big), & \text{jeśli } \mathrm{fract}\Big(\frac{t+\varphi}{T}\Big) < \frac{1}{2}, \\ A\Big(3 - 4\,\mathrm{fract}(\frac{t+\varphi}{T})\Big), & \text{w przeciwnym razie.} \end{cases}$$

gdzie A – amplituda, f – częstotliwość, T – okres, φ – faza, $d \in (0,1)$ – wypełnienie.

Pętla symulacji

Dla każdego kroku czasu wykonywane są:

1. Obliczenie $u_k = u(t_k)$.
2. **Krok Eulera:** $x_{k+1}^{\mathrm{E}} = x_k^{\mathrm{E}} + h\, f(t_k, x_k^{\mathrm{E}}, u_k)$.
3. **Krok RK4:** zgodnie ze wzorem:
$$x_{k+1}^{\mathrm{RK4}} = x_k^{\mathrm{RK4}} + \frac{h}{6}\,(k_1 + 2k_2 + 2k_3 + k_4),$$
gdzie k_i liczone są dla $f(t,x,u) = Ax + Bu$.
4. Wyjście: $y_k^{\mathrm{E}} = C\,x_k^{\mathrm{E}} + D\,u_k, \quad y_k^{\mathrm{RK4}} = C\,x_k^{\mathrm{RK4}} + D\,u_k$.

Kąt θ_2 przedstawiany jest na wykresie w stopniach [°] (konwersja z radianów), prędkość ω_2 w rad/s.

Wizualizacja wyników

Tworzone są trzy wykresy:

1. Sygnał wejściowy $u(t) = \tau_m(t)$ [N m].
2. $\theta_2(t)$ dla Eulera i RK4.
3. $\omega_2(t)$ dla Eulera i RK4 .

Przed rysowaniem osie są czyszczone (`ax.clear()`), a następnie aktualizowane (`canvas.draw()`). Dodano pasek narzędzi Matplotlib do interaktywnej obsługi wykresów (przybliżanie, zapisywanie).

Uwagi i wnioski

- Zwiększanie skoku powoduje utratę dokładności wyników obliczonych przy pomocy metody Eulera. Można uzyskać bardziej dokładny wynik poprzez ustawienie małej wartości skoku, ale wiąże się to ze zwiększeniem liczby obliczeń
- Układ o większym tłumieniu szybciej osiąga stan ustalony, ale ma mniejszą amplituda ustalonej odpowiedzi dla danego wymuszenia
- Układ o mniejszym tłumieniu ma większe przeregulowania i drgania, co dobrze pokazuje wpływ parametrów fizycznych.
- Przy małych krokach różnice pomiędzy metodą Eulera a RK4 są niezauważalne.
- Na podstawie odpowiedzi układu na pobudzenia o różnych częstotliwościach, można wywnioskować, że układ pełni funkcje filtru dolnoprzepustowego - tłumi wyższe częstotliwości i przepuszcza niskie