

SPRAWOZDANIE 11

Modelowanie Systemów Dynamicznych

Temat ćwiczenia: Układy z opóźnieniem

Michał Midor, gr. 5, 7.01.2026 r. - Środa 13.15

Cel ćwiczenia:

Celem ćwiczenia jest ręczne skonstruowanie układów z opóźnieniem, by poznać jak działają one "pod maską". Wykorzystana zostanie również wiedza z poprzednich zajęć dotycząca używania funkcji `fminsearch` do optymalizacji parametrów obiektu tak, by błąd był minimalny.

Rozwiązanie zadań:

Ćwiczenie rozpoczyna się od aproksymacji układu z opóźnieniem θ za pomocą metody Pade'go 1-go oraz 2-go rzędu.

Punktem odniesienia jest 'G_base' (transmitancja idealna), która za pomocą właściwości 'InputDelay' ustawione ma idealne opóźnienie ' θ '.

Aproksymowane obiekty stworzone są dzięki mnożeniu transmitancji bazowej (bez opóźnienia) przez aproksymację Pade'go, najpierw pierwszego, a następnie drugiego rzędu (w kodzie to obiekty 'P1' oraz 'P2').

```
clear;

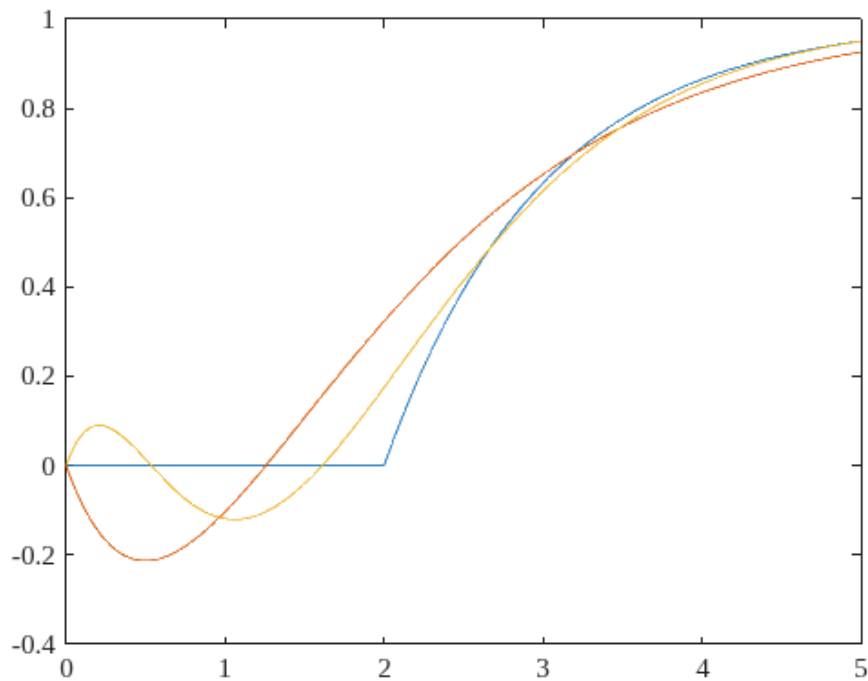
K = 1;
T = 1;
theta = 2;
t_sim = 0:0.01:5; % czas symulacji 5s

G = tf([K], [T, 1]);
G_base = G;
G_base.InputDelay = theta;
[y_ideal, t1] = step(G_base, t_sim);

P1 = tf([-theta/2, 1], [theta/2, 1]);
G_delay1 = G * P1;
[y_p1, t2] = step(G_delay1, t_sim);

P2 = tf([theta.^2/12, -theta/2, 1], [theta.^2/12, theta/2, 1]);
G_delay2 = G * P2;
[y_p2, t3] = step(G_delay2, t_sim);

plot(t1, y_ideal, t2, y_p1, t3, y_p2);
```



Na wykresie widać, że odpowiedź skokowa aproksymowanego pierwszym rzędem obiektu (kolor pomarańczowy) ma tylko jedną zmianę kierunku i jedno przecięcie się z zerem. Dodatkowo, w obszarze aproksymowanego opóźnienia, jego jej amplituda jest dwukrotnie większa od amplitudy odpowiedzi skokowej obiektu aproksymowanego równaniem drugiego rzędu.

Drugim etapem zadania jest aproksymacja układu wysokiego rzędu z transmitancją:

$$G(s) = \frac{5^{10}}{(s + 5)^{10}},$$

za pomocą układu niskiego rzędu z opóźnieniem z transmitancją:

$$G(s) = \frac{e^{-\theta s}}{as^2 + bs + 1}.$$

Na początek przyjęte zostały parametry ręcznie dobrane parametry, dla których odchyłka od układu wysokiego rzędu była jak najmniejsza.

Używając funkcji 'pade' oraz podając rząd przybliżenia, otrzymuje się licznik i mianownik transmitancji układu niskiego rzędu, którego używamy do aproksymacji.

```
n = 5;
a = 0.25;
b = 0.7;

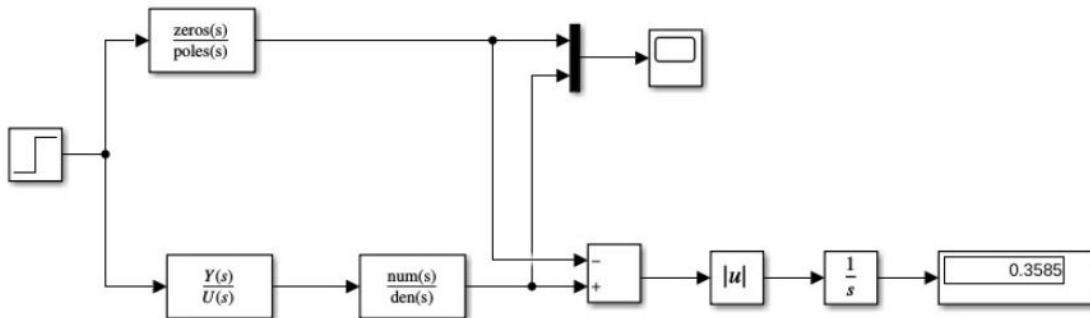
[licz, mian] = pade(theta, n)
```

```

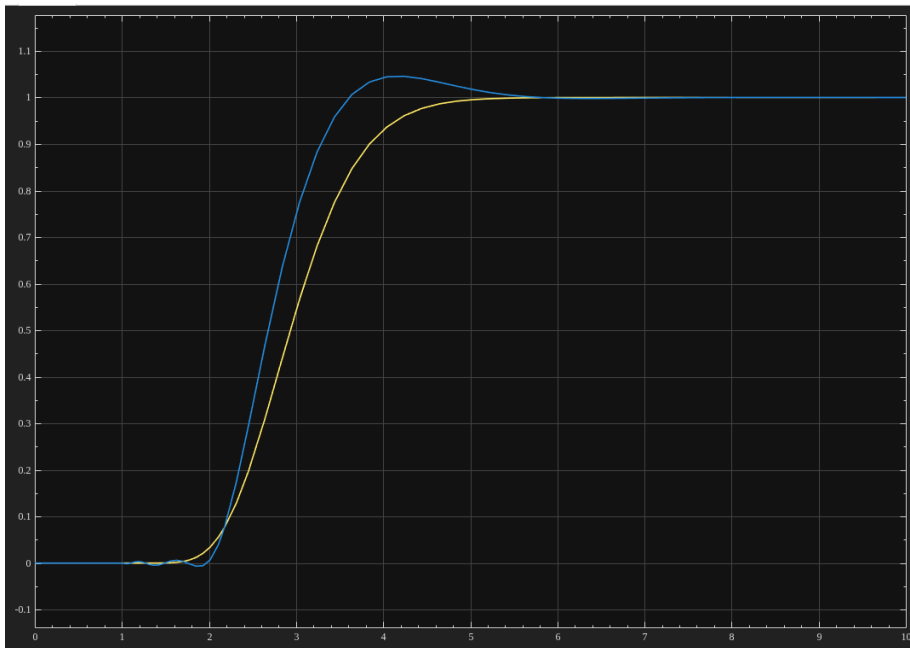
licz = 1×6
-1.0000    15.0000   -105.0000    420.0000   -945.0000    945.0000
mian = 1×6
 1.0000    15.0000    105.0000    420.0000    945.0000    945.0000

```

Mając te dane, można stworzyć model w simulinku, który graficznie pokaże proces uzyskiwania obydwu odpowiedzi skokowych, a także wylicza błąd.



Jak widać, błąd ten jest dosyć duży, natomiast wykresy odpowiedzi skokowych przedstawione są poniżej. Kolor żółty - obiekt wysokiego rzędu, kolor niebieski - obiekt, którego parametrów szukamy.



Następnie, w celu numerycznego wyznaczenia najbardziej optymalnych parametrów, dla których uzyskany błąd jest faktycznie znikomy, użyta jest funkcja 'fminsearch' poznana na poprzednich zajęciach.

Optymalizowana będzie funkcja błędu, której kod wygląda następująco:

```

function result_error = blad_wynik(X)
    a = X(1);
    b = X(2);
    theta = X(3);

    t_sim = 0:0.01:5;

    obiekt_10 = zpk([], -5 * ones(1,10), 5^10);
    [y_reference, ~] = step(obiekt_10, t_sim);

    n = 5;
    [licz, mian] = pade(theta, n);
    delay_tf = tf(licz, mian);
    obiekt_optymalizowany = tf([1], [a b 1]);
    obiekt_wynikowy = delay_tf * obiekt_optymalizowany;
    [y_optymalny, ~] = step(obiekt_wynikowy, t_sim);

    e = y_reference - y_optymalny;
    result_error = sum(e.^2)/length(e);
end

```

Na wejście zatem jest podany wektor parametrów a, b, theta.

```

X = [a, b, theta];

[parametry, blad] = fminsearch('blad_wynik', X)

```

```

parametry = 1×3
    0.3826    1.0318    0.9641
blad =
1.2689e-04

```

```

a = parametry(1);
b = parametry(2);
theta = parametry(3);

```

Błąd otrzymany tą metodą jest zdecydowanie mniejszy i wynosi 1.2689e-04. Z wyznaczonymi najbardziej optymalnymi parametrami można ostatecznie stworzyć obiekt wynikowy niskiego rzędu, a jego odpowiedź skokowa w porównaniu do odpowiedzi skokowej obiektu wysokiego rzędu jest przedstawiona na poniższym wykresie (kolor pomarańczowy).

```

[licz, mian] = pade(theta, n);
delay_tf = tf(licz, mian);
obiekt_optymalizowany = tf([1], [a b 1]);
obiekt_wynikowy = delay_tf * obiekt_optymalizowany;
[y_wynik, ~] = step(obiekt_wynikowy, t_sim);

obiekt_10 = zpk([], -5 * ones(1,10), 5^10);
[y_reference, ~] = step(obiekt_10, t_sim);

plot(t_sim, y_reference, t_sim, y_wynik);

```

