Podstawy automatyki i robotyki

Laboratorium 5

Michał Łaskawski

Identyfikacja z użyciem algorytmu optymalizacji

Do celów identyfikacji parametrów modelu matematycznego opisującego dynamikę obiektu sterowania jak również parametrów użytego regulatora można użyć jednej ze znanych metod optymalizacji funkcji wielu zmiennych.

Metody te identyfikują położenia punktów krytycznych (a dokładnie minimów) zdefiniowanej *funkcji celu*. Taką funkcją może być funkcja definiująca *kryterium najmniejszych kwadratów*.

$$S = \sum_{i=1}^{n} (v_i - y_i)^2$$

gdzie: v_i jest obserwowaną w i -tej chwili wartością odpowiedzi obiektu dynamicznego, y_i jest modelowaną w i -tej chwili wartością odpowiedzi modelu matematycznego obiektu dynamicznego, n jest ilością przeprowadzonych obserwacji.

Należy zauważyć, że wartości y_i będą wynikami obliczanymi na podstawie odpowiedzi na zdefiniowane wymuszenie funkcji modelu obiektu.

Przykład 1

Dokonać identyfikacji parametrycznej modelu obiektu sterowania na podstawie uzyskanych danych pomiarowych (wartości: v_i). Przyjąć iż modelem obiektu jest model inercyjny 3 rzędu o nieznanych parametrach: K oraz T.

$$G_o = \frac{K}{(1 + s \cdot T_1)\,(1 + s \cdot T_2)\,(1 + s \cdot T_3)}$$

Rozwiązanie:

Poniższy skrypt generuje wektor odpowiedzi na wymuszenie skokiem jednostkowym. Dane uzyskane z tego skryptu: `t_data` oraz `y_data`, symulują dane pomiarowe i posłużą do identyfikacji modelu obiektu.

```
% Nazwa pliku: pomiary.m
clc; clear;

% Parametry obiektu
gain = 1;
tau1 = 5;
tau2 = 6;
tau3 = 7;

% Parametry symulacji
simulationTime = 50;    % Całkowity czas symulacji [s]
numPoints = 1000;       % Liczba punktów w wektorze czasu
noiseAmplitude = 0.02;  % Amplituda dodawanego szumu

% Realizacja symulacji
[timeVector, noisyResponse] = noisyStepResponse(gain, tau1, tau2, tau3, ...
                                                simulationTime, numPoints, noiseAmplitude);

% Wykres wyników symulacji
figure;
plot(timeVector, noisyResponse, 'r', 'LineWidth', 1.5);
xlabel('Time [s]');
ylabel('Response');
title('Step Response of Third-Order Inertial System with Noise');
grid on;

t_data = timeVector;
y_data = noisyResponse';
```

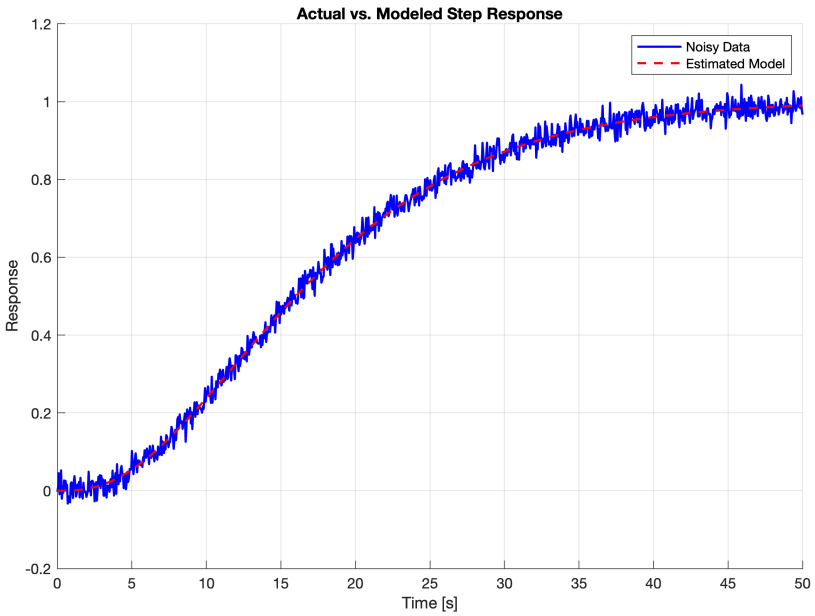
```
function [timeVector, noisyResponse] = noisyStepResponse(gain, tau1, tau2, tau3, ...
                                                         simulationTime, numPoints, noiseAmplitude)
    % Definicja transformaty operatorowej modelu inercyjnego trzeciego rzedu

    % Licznik transformaty
    numerator = gain;

    % Mianownik transformaty
    denominator = [tau1*tau2*tau3, tau1*tau2 + tau1*tau3 + tau2*tau3, tau1 + tau2 + tau3, 1];

    % Transformata modelu
    system = tf(numerator, denominator);

    % Utworzenie wektora czasu
    timeVector = linspace(0, simulationTime, numPoints);

    % Wyznaczenie odpowiedzi skokowej
    [response, ~] = step(system, timeVector);

    % Dodanie szumu do odpowiedzi
    % Wytworzenie wektora szumu
    noise = noiseAmplitude * randn(size(response));
    % Dodanie szumu do odpowiedzi
    noisyResponse = response + noise;
end
```

Poniższy skrypt implementuje metodę identyfikacji parametrów modelu inercyjnego na podstawie wygenerowanych wektorów: czasu (`t_data`) i odpowiedzi (`y_data`).

```
% Define the objective function for parameter estimation
% Defninicja funkcji celu, która implementuje kryterium najmniejszych kwadratów
objectiveFunction = @(params) computeSSE(params, t_data, y_data);

% Wartości początkowe identyfikowanych parametrów [K, T1, T2, T3]
initialGuesses = [1, 3, 3, 3];

% Uruchomienia algorytmu optymalizacji estymującego wartości identyfikowanych parametrów
estimatedParams = fminsearch(objectiveFunction, initialGuesses);

% Wydrukowanie na konsolę zidentyfikowanych parametrów
K_est = estimatedParams(1);
T1_est = estimatedParams(2);
T2_est = estimatedParams(3);
T3_est = estimatedParams(4);

fprintf('Estimated Parameters:\n');
fprintf('K = %.4f\n', K_est);
fprintf('T1 = %.4f\n', T1_est);
fprintf('T2 = %.4f\n', T2_est);
fprintf('T3 = %.4f\n', T3_est);

% Wykreślenie odpowiedzi na wymuszenie skokowe zidentyfikowanego modelu
figure;
plot(t_data, y_data, 'b', 'LineWidth', 1.5); hold on;
[~, y_model] = modelStepResponse(estimatedParams, t_data);
plot(t_data, y_model, 'r--', 'LineWidth', 1.5);
xlabel('Time [s]');
ylabel('Response');
title('Actual vs. Modeled Step Response');
legend('Noisy Data', 'Estimated Model');
grid on;

% Implementacja funkcji wyznaczającej wartość kryterium najmnieszych kwadratów
function sse = computeSSE(params, t, y_actual)
```

```
    K = params(1);
    T1 = params(2);
    T2 = params(3);
    T3 = params(4);

    % Defninicja transformaty modelu
    numerator = K;
    denominator = [T1*T2*T3, T1*T2 + T1*T3 + T2*T3, T1 + T2 + T3, 1];
    system = tf(numerator, denominator);

    % Wyznaczenie odpowiedzi na wymuszenie skokiem jednostkowym identyfikowanego modelu
    [y_model, ~] = step(system, t);

    % Interpolacja odpowiedzi modelu, tak aby punkty odpowiedzi zgadzały się z punktami danych
    y_model_interp = interp1(t, y_model, t, 'linear', 'extrap');

    % Wyznaczenie sumy kwadratów błędów
    errors = y_actual - y_model_interp;
    sse = sum(errors.^2);
end

function [t_model, y_model] = modelStepResponse(params, t)
    K = params(1);
    T1 = params(2);
    T2 = params(3);
    T3 = params(4);

    % Definicja transmitancji modelu
    numerator = K;
    denominator = [T1*T2*T3, T1*T2 + T1*T3 + T2*T3, T1 + T2 + T3, 1];
    system = tf(numerator, denominator);

    % Wyznaczenie odpowiedzi na wymuszenie skokiem jednostkowym modelu matematycznego
    [y_model, t_model] = step(system, t);
end
```

Uzyskany wynik:



Rysunek 1: Odpowiedzi na wymuszenie skokowe obiektu i zidentyfikowanego modelu

Optymalny dobór nastaw regulatora

Do optymalnego dobory nastaw regulatora (np. PI czy też PID), można wykorzystać techniki optymalizacji numerycznej (funkcja `fminsearch`) oraz *całkowe kryteria jakości*.

Całkowe kryteria jakości

Całkowe kryteria jakości są metodą oceny jakości układów dynamicznych. Wyznaczane są one poprzez całkowanie funkcji (sygnału) błędu $e(t)$ w czasie i opcjonalnie samego czasu t. Poniżej przedstawiono najważniejsze kryteria wraz z ich zaletami i wadami.

1. Kryterium ISE (Integral of Squared Error)

- Wyrażenie:
$$ISE = \int_0^\infty e^2(t)\,dt$$
- Penalizuje większe błędy bardziej niż mniejsze, ponieważ stosuje kwadrat błędu.
- Zalety:
  - Skupia się na minimalizacji dużych błędów.
  - Odpowiednie dla systemów, gdzie istotne są duże odchylenia.
- Wady:
  - Nie uwzględnia czasu występowania błędu.
  - Może prowadzić do wydłużenia czasu regulacji.

2. Kryterium ITSE (Integral of Time-weighted Squared Error)

- Wyrażenie:
$$ITSE = \int_0^\infty t \cdot e^2(t)\,dt$$
- Penalizuje błędy występujące w późniejszych etapach czasu.
- Zalety:
  - Sprzyja szybszemu tłumieniu oscylacji.
  - Lepsze dla systemów z priorytetem na stabilność.
- Wady:
  - Bardziej złożone obliczeniowo.
  - Wrażliwe na błędy w późniejszych etapach.

3. Kryterium IAE (Integral of Absolute Error)

- Wyrażenie:
$$IAE = \int_0^\infty |e(t)|\,dt$$
- Traktuje wszystkie błędy równomiernie, niezależnie od ich wielkości.
- Zalety:
  - Mniej wrażliwe na duże błędy niż ISE.
  - Łatwe w implementacji.
- Wady:
  - Może prowadzić do wydłużonego czasu regulacji.
  - Nie uwzględnia czasu występowania błędu.

4. Kryterium ITAE (Integral of Time-weighted Absolute Error)

- Wyrażenie:
$$ITAE = \int_0^\infty t \cdot |e(t)|\,dt$$

- Ważone czasem, co sprzyja minimalizacji błędów w późniejszych etapach regulacji.
- Zalety:
  - Dobre dla systemów wymagających szybkiego tłumienia oscylacji.
  - Zapewnia krótszy czas regulacji.
- Wady:
  - Bardziej złożone w implementacji.
  - Wrażliwe na zakłócenia w późniejszych fazach regulacji.

5. Kryterium RMS (Root Mean Square Error)

Chociaż nie jest klasycznym kryterium całkowym, RMS pozwala ocenić średnią wielkość błędu:

- Wyrażenie:

$$RMS = \sqrt{\frac{1}{T}\int_0^T e^2(t)\,dt}$$

- Mierzy średnią wartość błędu w zadanym okresie czasu.
- Zalety:
  - Intuicyjne i łatwe do interpretacji.
  - Proste w implementacji.
- Wady:
  - Wynik zależy od długości okresu obserwacji.
  - Nie różnicuje błędów względem czasu ich występowania.

Podsumowanie kryteriów:

Kryterium	Penalizuje duże błędy	Uwzględnia czas błędu	Szybkie tłumienie oscylacji
ISE	Tak	Nie	Średnie
ITSE	Tak	Tak	Dobre
IAE	Nie	Nie	Średnie
ITAE	Nie	Tak	Bardzo dobre
RMS	Tak	Nie	Średnie

Wybór kryterium:

Wybór odpowiedniego kryterium zależy od wymagań projektowych, charakterystyki systemu i jego zastosowania:

- ISE: Minimalizacja dużych błędów.
- ITSE: Stabilność i szybkie tłumienie oscylacji.
- IAE: Zrównoważone traktowanie błędów.
- ITAE: Szybkie tłumienie błędów w późniejszych etapach.
- RMS: Intuicyjne dla analizy średniego błędu w czasie.

Przykład 2

Problem, Dobierz optymalne nastawy regulatora PI, pracującego w zamkniętym układzie regulacji, który połączony jest szeregowo z obiektem sterowania, opisanym modelem inercyjnym 3 rzędu:

$$G_o = \frac{K}{(1 + s \cdot T_1)\,(1 + s \cdot T_2)\,(1 + s \cdot T_3)}$$

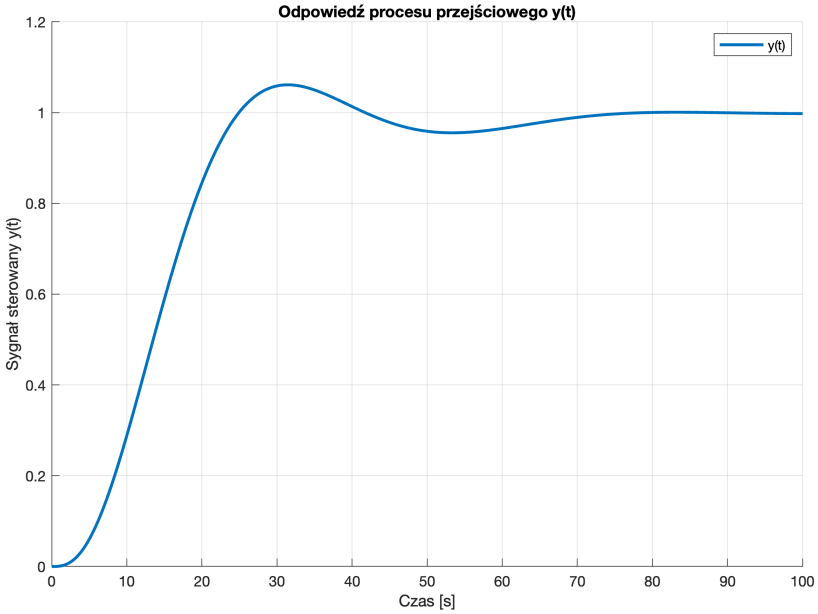
Przyjmij: $K = 1$, $T_1 = 5$, $T_2 = 6$, $T_3 = 7$. Jako kryterium jakości przyjmij kryterium ITAE.

Rozwiązanie:

```
clc; clear;

% Definicja parametru czasu symulacji
timeSpan = [0 100];
```

```
% Definicja operatora Laplace'a
s = tf('s');

% Definicja stałych czasowych obiektu
T1 = 5; % 1 Stała czasowa
T2 = 6; % 2 Stała czasowa
T3 = 7; % 3 Stała czasowa

% Funkcja transferu obiektu inercyjnego 3 rzędu
G = 1 / ((T1*s + 1)*(T2*s + 1)*(T3*s + 1));

% Funkcja celu dla optymalizacji ITAE
objectiveFunction = @(params) calculateITAE(params, G, timeSpan);

% Poczatkowe przybliżenie dla parametrów regulatora PI
initialParams = [1, 1]; % [Kp, Ki]

% Optymalizacja parametrów regulatora PI z użyciem fminsearch
optimizedParams = fminsearch(objectiveFunction, initialParams);
Kp_opt = optimizedParams(1);
Ki_opt = optimizedParams(2);

% Wyświetlenie wyników optymalizacji
fprintf('Optymalne parametry regulatora PI:\n');
fprintf('Kp = %.4f\n', Kp_opt);
fprintf('Ki = %.4f\n', Ki_opt);

% Zbudowanie regulatora PI z optymalnymi parametrami
C = Kp_opt + Ki_opt / s;

% Zamknięty układ regulacji (sprzężenie zwrotne)
T = feedback(C*G, 1);

% Symulacja odpowiedzi układu zamkniętego
[y, t] = step(T, timeSpan);

% Wyświetlenie informacji o odpowiedzi układu
info = stepinfo(T);
fprintf('Informacje o odpowiedzi układu:\n');
fprintf('Czas ustalania: %.4f s\n', info.SettlingTime);
fprintf('Przeregulowanie: %.2f%%\n', info.Overshoot);
fprintf('Czas narastania: %.4f s\n', info.RiseTime);

% Wizualizacja odpowiedzi układu
figure;
plot(t, y, 'LineWidth', 2);
grid on;
title('Odpowiedź procesu przejściowego y(t)');
xlabel('Czas [s]');
ylabel('Sygnał sterowany y(t)');
legend('y(t)');

% Funkcja obliczająca ITAE
function itae = calculateITAE(params, G, timeSpan)
    s = tf('s');
    Kp = params(1);
    Ki = params(2);
    C = Kp + Ki / s; % Regulator PI
    T = feedback(C*G, 1); % Zamknięty układ
    [y, t] = step(T, timeSpan); % Odpowiedź skokowa
    e = 1 - y; % Błąd regulacji
    itae = trapz(t, t .* abs(e)); % Całkowanie ITAE
end
```

Uzyskany wynik:

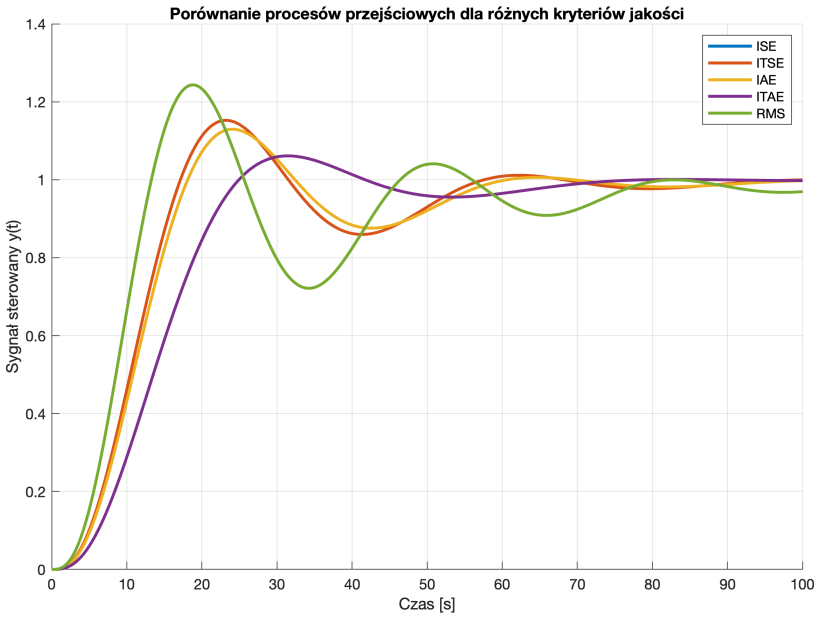


Rysunek 2: Proces przejściowy optymalnego układu regulacji

Zadanie 1

Zmodyfikuj przykład 2, w taki sposób aby dokonać wizualizacji procesów przejściowych oraz wyznaczyć czasowe kryteria jakości dla układów regulacji, w których parametry regulatora PI dobierane będą względem pozostałych przedstawionych kryteriów, to jest ISE, ITSE, IAE, ITAE oraz RMS.

Spodziewany wynik:



Rysunek 3: Procesy przejściowe optymalnych układów regulacji z regulatorem PI

Wyniki optymalizacji parametrów regulatora PI dla różnych kryteriów jakości

Kryterium	Kp	Ki	Czas ustalania [s]	Przeregulowanie [%]	Czas narastania [s]
ISE	2.8128	0.0835	105.0370	24.34	8.0056
ITSE	1.7890	0.0873	83.8081	15.24	10.4821
IAE	1.6565	0.0838	56.6768	12.96	11.0864
ITAE	1.0308	0.0719	65.8197	6.12	15.2358
RMS	2.8128	0.0835	105.0370	24.34	8.0056

Zadanie 2

Dobierz optymalne nastawy regulatora PID opisanego transmitancją:

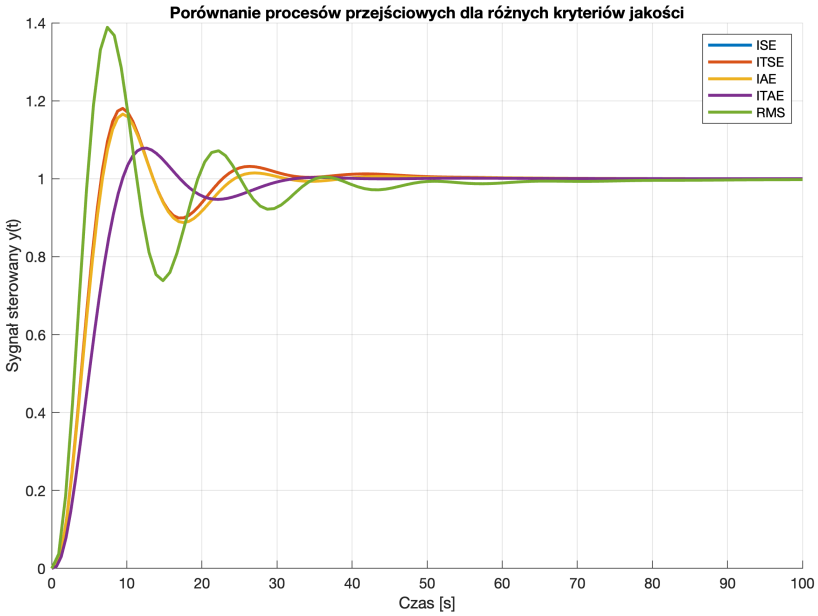
$$C_{PID}(s) = K_P \left(1 + \frac{1}{T_r s} + \frac{T_d s}{\tau_D s + 1}\right)$$

gdzie $\tau_D = 0.1 T_d$.

Uwagi:

- Jako kryteria optymalności wykorzystaj całkowe kryteria jakości.
- Dokonaj wizualizacji procesów przejściowych projektowanych układów oraz zestawienia czasowych wskaźników jakości (patrz zadanie 1).

Spodziewany wynik:



Rysunek 4: Procesy przejściowe optymalnych układów regulacji z regulatorem PID

Wyniki optymalizacji parametrów regulatora PID dla różnych kryteriów jakości

Kryterium	Kp	Ki	Kd	Czas ustalania [s]	Przeregulowanie [%]	Czas narastania [s]
ISE	9.2999	0.2951	37.0170	46.1936	38.79	2.9758
ITSE	4.2766	0.3050	27.3740	29.3624	18.05	4.1809
IAE	4.3470	0.2621	26.0436	22.9998	16.59	4.2901
ITAE	3.2654	0.1975	16.2572	28.0224	7.89	6.0003
RMS	9.2999	0.2951	37.0170	46.1936	38.79	2.9758