Linearyzacja

Sprawozdanie

Akademia Górniczo-Hutnicza im. Stanisława Staszica
Modelowanie Systemów Dynamicznych 2023
WEAIiIB, Automatyka i Robotyka

Data wykonania ćwiczenia:
21.12.2022 r.

Data oddania sprawozdania:
03.01.2023 r.

Jakub Górski

Grupa dziekańska nr 3

Spis Treści

1. Cel ćwiczeń

Celem laboratorium jest zapoznanie się z tematyką linearyzacji modelów dynamicznych nieliniowych.

2. Wstęp teoretyczny

Układ liniowy – układ opisany liniową zależnością (w szczególności model musi spełniać zasadę superpozycji).

Zasada superpozycji:

$$c_1 u_1 + c_2 u_2 = c_1 y_1 + c_2 y_2$$

- c_i – wartości stałe

- y_i – wyniki odziaływań wymuszeń u_i

- u_i – wymuszenia

Układ nieliniowy jest przedstawiony w postaci modelu zawierającego nieliniowe operacje na zmiennych układu (np. mnożenie) lub/i jego parametry są zależne od zmiennych.

W rzeczywistości nie istnieją układy liniowe, gdyż postulat liniowości powiązany jest z ostrymi warunkami (np. brakiem ograniczeń zmiennych układu, co nie jest fizyczne możliwe). Można jednak korzystać z modelu liniowego układu (jeśli istnieje taki obszar parametrów, w których postulat liniowości jest spełniony z określoną dokładnością).

Klasyczna teoria sterowania została wyprowadzona dla modeli liniowych.

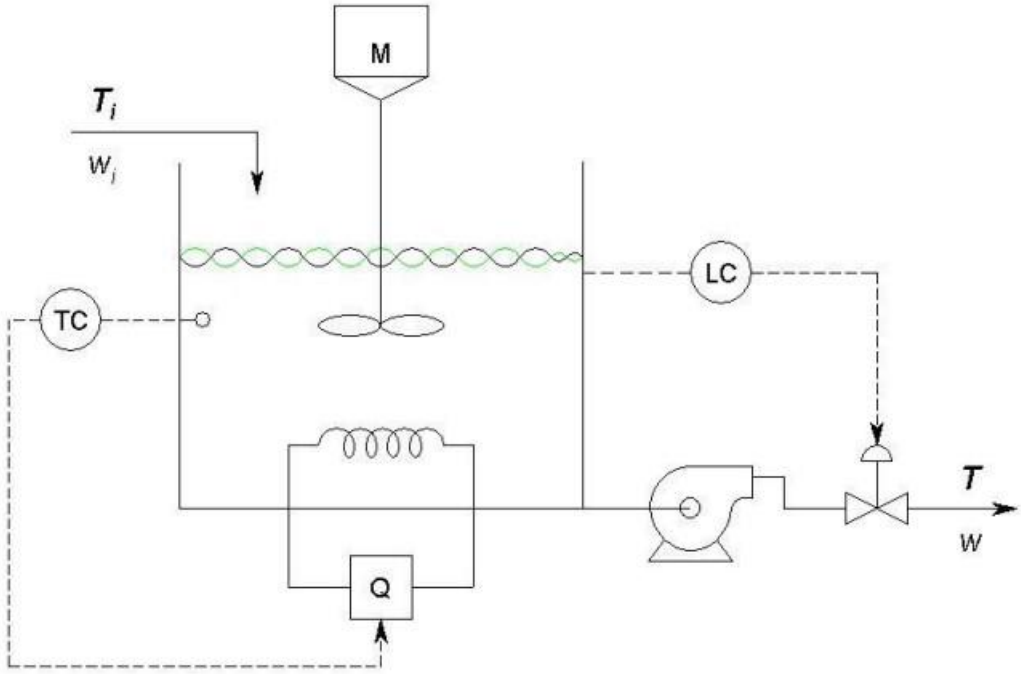
Linearyzacja – proces uproszczenia modelu nieliniowego, aby zależność nieliniowa była przybliżana lokalnie (w pewnym obszarze) przez odpowiednio wybraną zależność liniową.

Stan równowagi – zmienne stanu pozostają stałe w pewnym skończonym przedziale czasowym (przy czym wartości wejść mogą się zmieniać, lecz się kompensują). Wartości tych zmiennych określaną punkt równowagi.

Stan ustalony – zmienne wejściowe są niezmienne przez dłuższy czas niż największe opóźnienie czasowe zawarte w układzie. Stan ten posiada ważną własność samopodtrzymywania, polegającą na braku występowania zmian do momentu zaistnienia zmian na wejściu.

Linearyzację układu nieliniowego dokonuje się wokół stanu ustalonego.

W dziedzinie czasu, wartość wyjść w stanie ustalonym można wyznaczyć z równań stanu, podstawiając zera w miejsce pochodnych stanu.



Schemat modelu potrzebnego do wykonania laboratorium

Równania różniczkowe opisujące zadany model:

Z zasady zachowania masy:

$$\rho \frac{dV}{dt} = w_i - w$$

Z zasady zachowania energii:

$$V\rho \frac{dT}{dt} = w_i(T_i - T) + \frac{Q}{C}$$

Oznaczenia:

T_i – temperatura strumienia wejściowego [K]

T – temperatura strumienia wyjściowego [K]

w_i – masowy strumień wejściowy [kg/s]

w – masowy strumień wyjściowy [kg/s]

V – objętość cieczy w zbiorniku [m^3]

Q – moc grzałki [W]

ρ – gęstość cieczy [kg/m^3]

C – ciepło właściwe cieczy [J/(kg·K)]

3. Wykonanie zadań

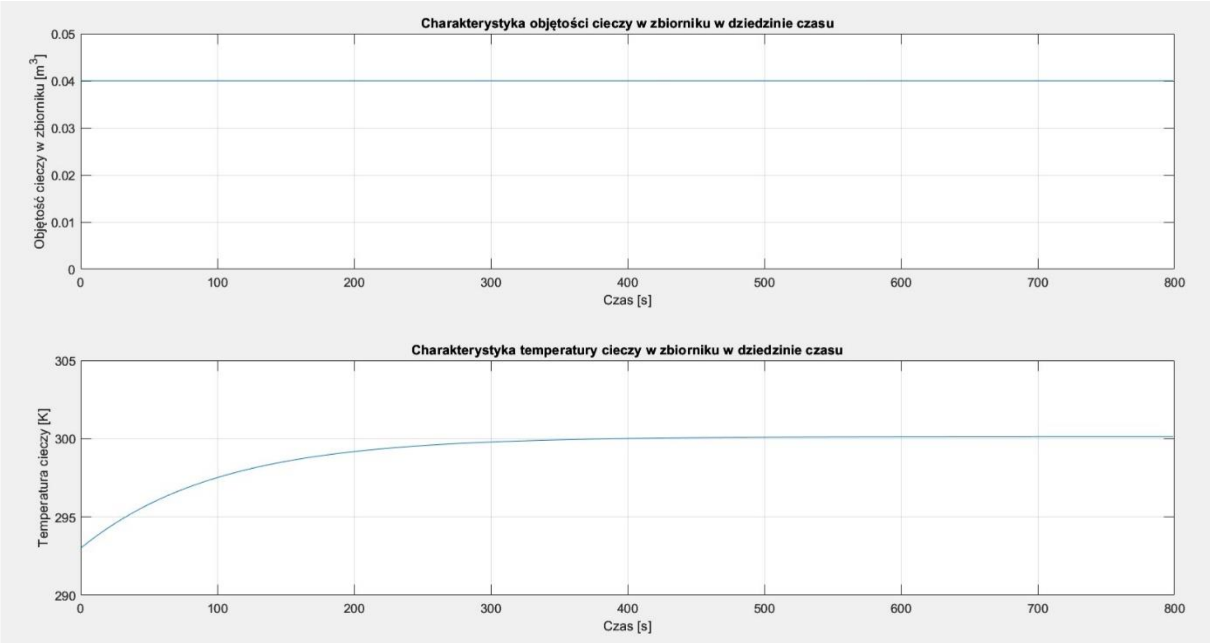
3.1. Model nieliniowy

```
function dx = zbiornik_stan(t, x, wi, w, Ti, Q)
    % Argumenty wejściowe:
    % t - czas
    % x - wektor stanu układu
    % wi - dopływ
    % w - wypływ
    % Ti - temperatura cieczy dopływającej
    % Q - moc dostarczana (grzanie)
    %---------------------------- zmienne stanu ---

    x1 = x(1); % objętość
    x2 = x(2); % temperatura

    %------------------------------ parametry -----

    C = 4200; % ciepło właściwe [J/(Kg*K)]
    ro = 1000; % gęstość [kg/m3]

    %%---------------------------- równania stanu --

    dx1 = 1/ro*(wi - w);
    dx2 = wi*(Ti - x2)/(ro*x1) + Q/(ro*x1*C);

    dx = [dx1; dx2]; % pochodne stanu
end
```

Funkcja reprezentująca układ równań różniczkowych opisujących zadany model

```
%% Model nieliniowy (I)
Q = 12000;
w = 0.4;
wi = 0.4;
Ti = 293;
T0 = 293;
V0 = 0.04;

[t, x] = ode45(@zbiornik_stan, [0:799], [V0, T0], [], wi, w, Ti, Q);

figure;
subplot(2, 1, 1);
plot(t, x(:, 1));
ylim([0 0.05]);
title("Charakterystyka objętości cieczy w zbiorniku w dziedzinie czasu");
xlabel("Czas [s]"); ylabel("Objętość cieczy w zbiorniku [m^3]");
grid on;
subplot(2, 1, 2);
plot(t, x(:, 2));
ylim([290 305]);
title("Charakterystyka temperatury cieczy w zbiorniku w dziedzinie czasu");
xlabel("Czas [s]"); ylabel("Temperatura cieczy [K]");
grid on;
```

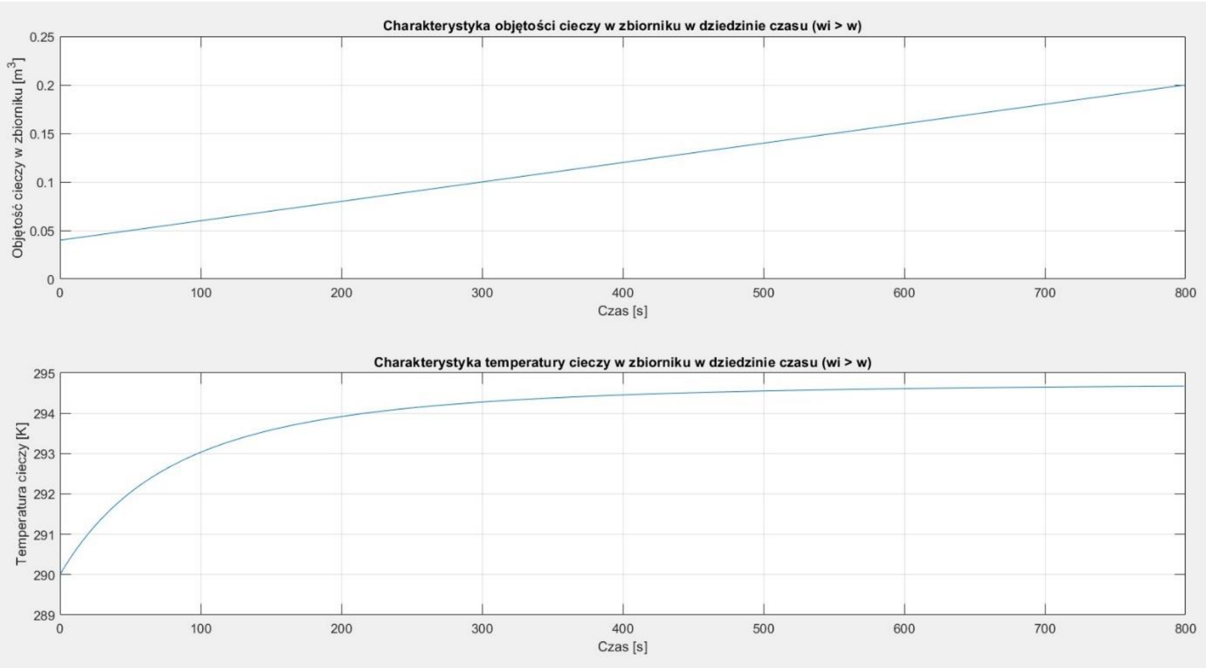
Kod programu – rozwiązanie układu równań różniczkowych i wizualizacja wyników (w = wi)



Wykresy objętości i temperatury cieczy w zbiorniku w czasie (w = wi)

```
%% Model nieliniowy (II)
Q = 10000;
w = 0.3;
wi = 0.5;
Ti = 290;
T0 = 290;
V0 = 0.04;

[t, x] = ode45(@zbiornik_stan, [0:799], [V0, T0], [], wi, w, Ti, Q);

figure;
subplot(2, 1, 1);
plot(t, x(:, 1));
ylim([0 0.25]);
title("Charakterystyka objętości cieczy w zbiorniku w dziedzinie czasu (wi > w)");
xlabel("Czas [s]"); ylabel("Objętość cieczy w zbiorniku [m^3]");
grid on;
subplot(2, 1, 2);
plot(t, x(:, 2));
ylim([289 295]);
title("Charakterystyka temperatury cieczy w zbiorniku w dziedzinie czasu (wi > w)");
xlabel("Czas [s]"); ylabel("Temperatura cieczy [K]");
grid on;
```

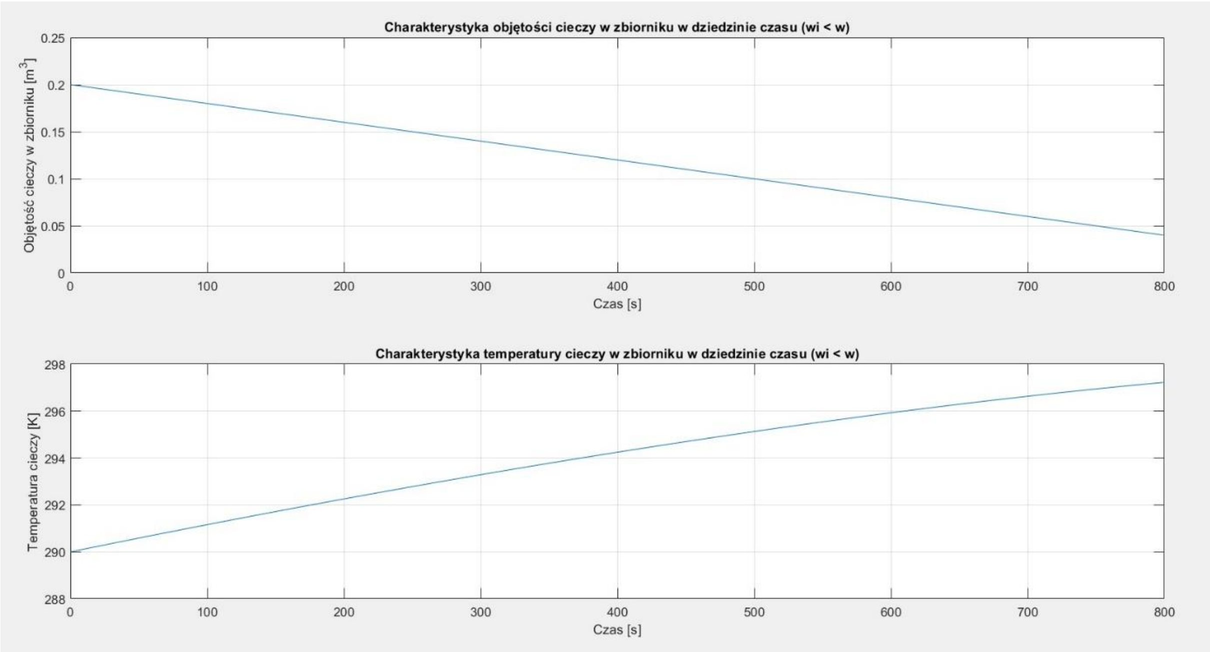
Kod programu – rozwiązanie układu równań różniczkowych i wizualizacja wyników (w < wi)



Wykresy objętości i temperatury cieczy w zbiorniku w czasie (w < wi)

```
%% Model nieliniowy (III)
Q = 10000;
w = 0.5;
wi = 0.3;
Ti = 290;
T0 = 290;
V0 = 0.2;

[t, x] = ode45(@zbiornik_stan, [0:799], [V0, T0], [], wi, w, Ti, Q);

figure;
subplot(2, 1, 1);
plot(t, x(:, 1));
ylim([0 0.25]);
title("Charakterystyka objętości cieczy w zbiorniku w dziedzinie czasu (wi < w)");
xlabel("Czas [s]"); ylabel("Objętość cieczy w zbiorniku [m^3]");
grid on;
subplot(2, 1, 2);
plot(t, x(:, 2));
ylim([288 298]);
title("Charakterystyka temperatury cieczy w zbiorniku w dziedzinie czasu (wi < w)");
xlabel("Czas [s]"); ylabel("Temperatura cieczy [K]");
grid on;
```

Kod programu – rozwiązanie układu równań różniczkowych i wizualizacja wyników (w > wi)



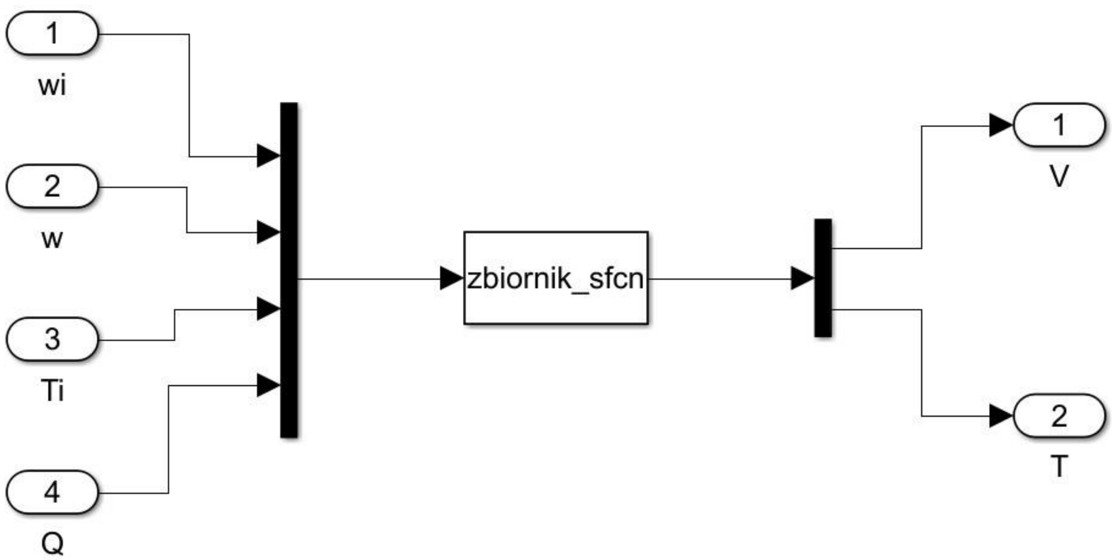
Wykresy objętości i temperatury cieczy w zbiorniku w czasie (w > wi)

3.2. Stan ustalony

W *Matlabie* stan ustalony można obliczyć metodami numerycznymi, korzystając z funkcji *trim*.

```
function [sys, x0, str, ts] = zbiornik_sfcn(t, x, u, flag, V0, T0)
    switch flag
        case 0 % inicjalizacja
            str = [];
            ts = [0 0];
            s = simsizes;
                s.NumContStates = 2; % liczba stanów ciągłych
                s.NumDiscStates = 0; % liczba stanów dyskretnych
                s.NumOutputs = 2; % liczba wyjść
                s.NumInputs = 4; % liczba wejść
                s.DirFeedthrough = 0; % wejście nie przenosi się bezpośrednio na wyjście
                s.NumSampleTimes = 1; % czas próbkowania
            sys = simsizes(s);
            x0 = [V0, T0];
        case 1 % pochodne
            wi = u(1);
            w = u(2);
            Ti = u(3);
            Q = u(4);
            sys = zbiornik_stan(t, x, wi, w, Ti, Q);
        case 3 % wyjście
            sys = x;
        case {2 4 9}
            sys = [];
        otherwise
            error(['unhandled flag =', num2str(flag)]);
    end
end
```

Model zbiornika zapisany za pomocą *s-funkcji*



Model w *Simulinku* potrzebny do rozwiązania zadania

```
%% Stan ustalony (I)
V0 = 0.04;
T0 = 293;

X0 = [0.04; 303];
U0 = [0.4; 0.4; 293; 12000];
Y0 = [0.04; 303];

IX = [];
IU = [1; 2; 3];
IY = [1; 2];

[x, u, y, dx] = trim('zbiornik_sys', X0, U0, Y0, IX, IU, IY)

[t_, x_] = ode45(@zbiornik_stan, [0:799], [V0, T0], [], u(1), u(2), u(3), u(4));

figure;
subplot(2, 1, 1);
plot(t_, x_(:, 1));
title("Charakterystyka objętości cieczy w zbiorniku w dziedzinie czasu (w stanie ustalonym)");
xlabel("Czas [s]"); ylabel("Objętość cieczy w zbiorniku [m^3]");
grid on;
subplot(2, 1, 2);
plot(t_, x_(:, 2));
ylim([290 305]);
title("Charakterystyka temperatury cieczy w zbiorniku w dziedzinie czasu (w stanie ustalonym)");
xlabel("Czas [s]"); ylabel("Temperatura cieczy [K]");
grid on;
```

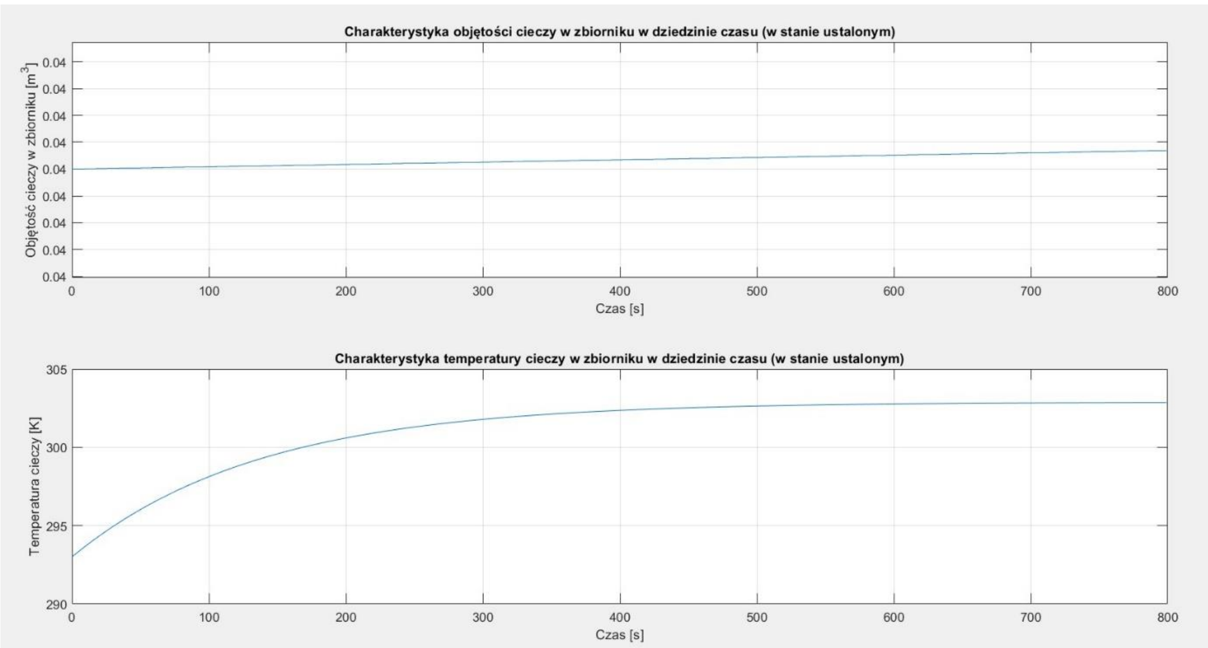
Kod programu – znalezienie stanu ustalonego i wizualizacja wyników

Przy zadanych danych nie można znaleźć punktu równowagi układu. W tym momencie funkcja *trim* poszukuje najbliższego punktu, który minimalizuje kryterium:

$$\mathrm{abs}([\mathrm{x}(\mathrm{ix}) - \mathrm{x0}(\mathrm{ix});\ \mathrm{u}(\mathrm{iu}) - \mathrm{u0}(\mathrm{iu});\ \mathrm{y}(\mathrm{iy}) - \mathrm{y0}(\mathrm{iy})]).$$

```
x =

    0.1435
  302.8920


u =

   1.0e+04 *

    0.0000
    0.0000
    0.0293
    1.2000


y =

    0.1435
  302.8920
```

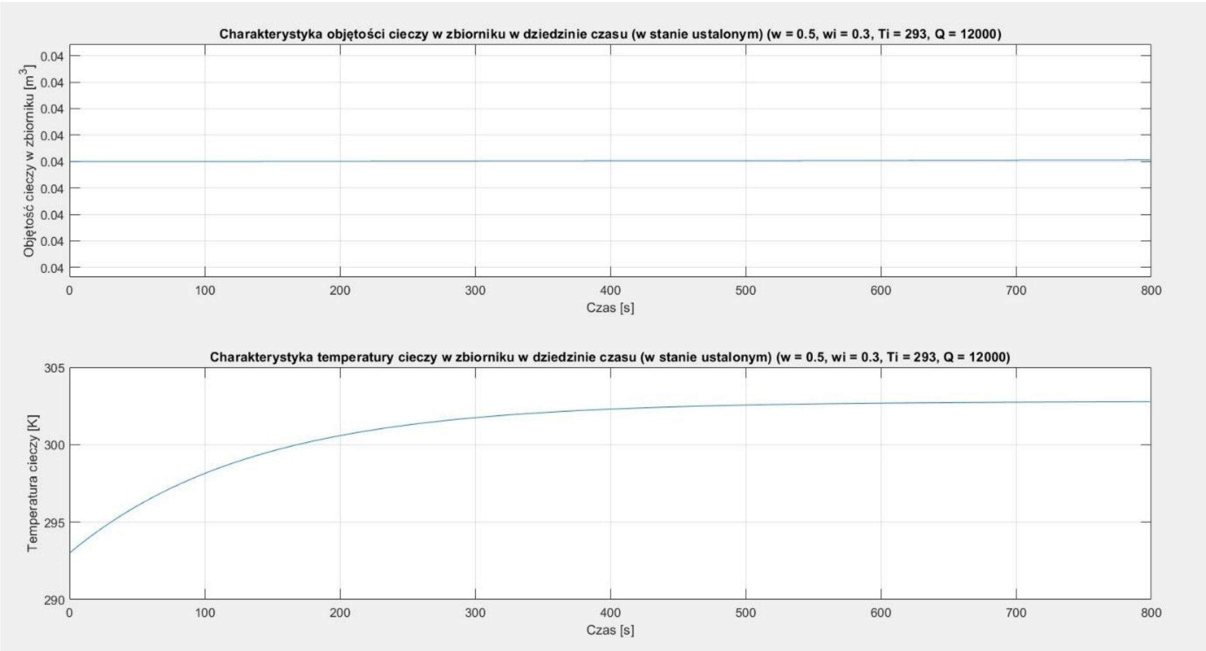
Wartości zwracane przez funkcję *trim*



Wykresy objętości i temperatury cieczy w zbiorniku w dziedzinie czasu (model w stanie ustalonym)

```
%% Stan ustalony (II)
V0 = 0.04;
T0 = 293;

X0 = [0.04; 303];
U0 = [0.3; 0.5; 293; 12000];
Y0 = [0.04; 303];

IX = [];
IU = [1; 2; 3];
IY = [1; 2];

[x, u, y, dx] = trim('zbiornik_sys', X0, U0, Y0, IX, IU, IY)

[t_, x_] = ode45(@zbiornik_stan, [0:799], [V0, T0], [], u(1), u(2), u(3), u(4));

figure;
subplot(2, 1, 1);
plot(t_, x_(:, 1));
title("Charakterystyka objętości cieczy w zbiorniku w dziedzinie czasu (w stanie ustalonym) (w = 0.5, wi = 0.3, Ti = 293, Q = 12000)");
xlabel("Czas [s]"); ylabel("Objętość cieczy w zbiorniku [m^3]");
grid on;
subplot(2, 1, 2);
plot(t_, x_(:, 2));
ylim([290 305]);
title("Charakterystyka temperatury cieczy w zbiorniku w dziedzinie czasu (w stanie ustalonym) (w = 0.5, wi = 0.3, Ti = 293, Q = 12000)");
xlabel("Czas [s]"); ylabel("Temperatura cieczy [K]");
grid on;
```

Kod programu – znalezienie stanu ustalonego i wizualizacja wyników (w = 0.5, wi = 0.3, Ti = 293, Q = 12000)



Wykresy objętości i temperatury cieczy w zbiorniku w dziedzinie czasu (model znajduje się w stanie ustalonym) (w = 0.5, wi = 0.3, Ti = 293, Q = 12000)

3.3. Model liniowy

Do linearyzacji (na zmiennych odchyłkowych, odchyłkach od punktu pracy) układu nieliniowego w *Matlabie* służy funkcja *linmod*. Po zastosowaniu wspomnianego elementu otrzyma się model liniowy w przestrzeni stanu (w postaci macierzy *A*, *B*, *C*, *D*).

Aby znaleźć odpowiedź układu na zadane wymuszenie wykorzystano funkcję *lsim*.

```
%% Model liniowy
V0 = 0.04;
T0 = 293;

X0 = [0.04; 303];
U0 = [0.4; 0.4; 293; 12000];
Y0 = [0.04; 303];

IX = [];
IU = [1; 2; 3];
IY = [1; 2];

[x, u, y, dx] = trim('zbiornik_sys', X0, U0, Y0, IX, IU, IY);

[A, B, C, D] = linmod('zbiornik_sys', x, u);

for iu = 1:4
    [licz, mian] = ss2tf(A, B, C, D, iu);

    printsys(licz, mian);
end

t = 0:799;
U = zeros(length(t), length(u));
X_pocz = [0.04 293];
X_ust = [0.04 303];
x0 = X_pocz - X_ust;

y_ = lsim(A, B, C, D, U, t, x0);

figure;
subplot(2,1,1);
plot(t, y_(:,1) + X_ust(1));
ylim([0 0.06]);
title('Model liniowy');
xlabel('Objętość');
grid on;
subplot(2,1,2);
plot(t, y_(:,2) + X_ust(2));
ylim([292 302]);
xlabel('Temperatura');
grid on;
```

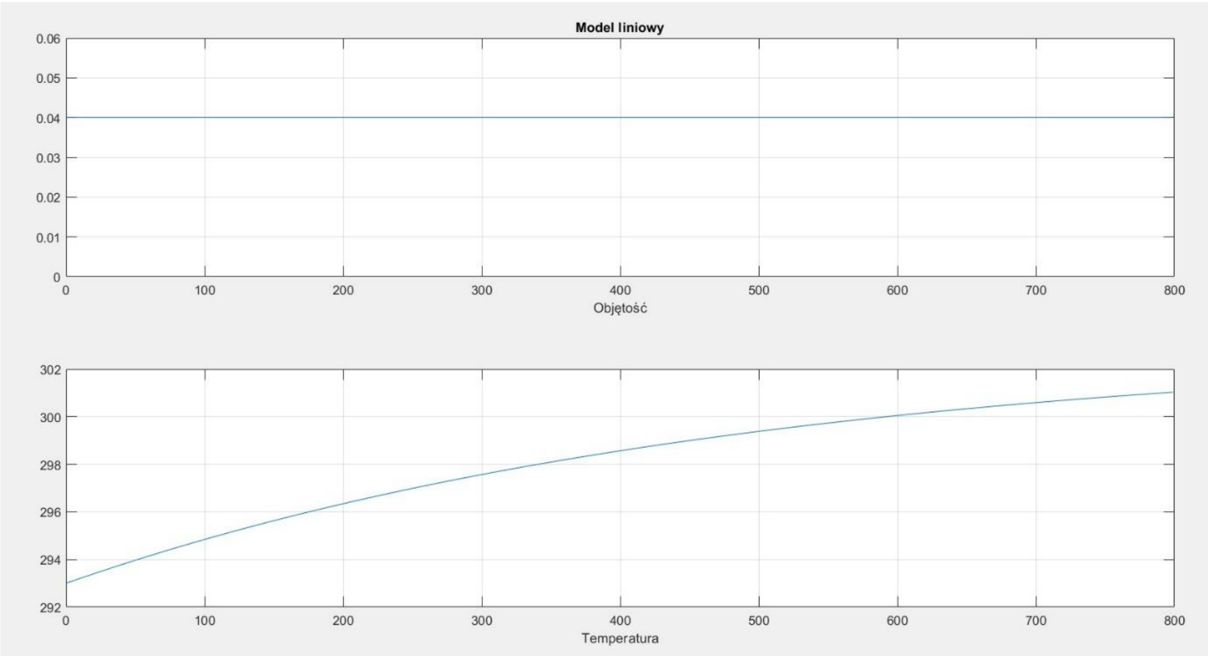
Kod programu – linearyzacja modelu nieliniowego

```
num(1)/den =

   0.001 s + 2.0351e-06
   --------------------
     s^2 + 0.0020351 s

num(2)/den =

   -0.068185 s + 1.5823e-13
   ------------------------
       s^2 + 0.0020351 s

num(1)/den =

   -0.001 s - 2.0351e-06
   ---------------------
     s^2 + 0.0020351 s

num(2)/den =

      -1.5823e-13
   -----------------
   s^2 + 0.0020351 s
```

Transmitancję opisujące model (I)

```
num(1)/den =

           0
   -----------------
   s^2 + 0.0020351 s

num(2)/den =

     0.0020351 s
   -----------------
   s^2 + 0.0020351 s

num(1)/den =

           0
   -----------------
   s^2 + 0.0020351 s

num(2)/den =

     1.6593e-06 s
   -----------------
   s^2 + 0.0020351 s
```

Transmitancję opisujące model (II)



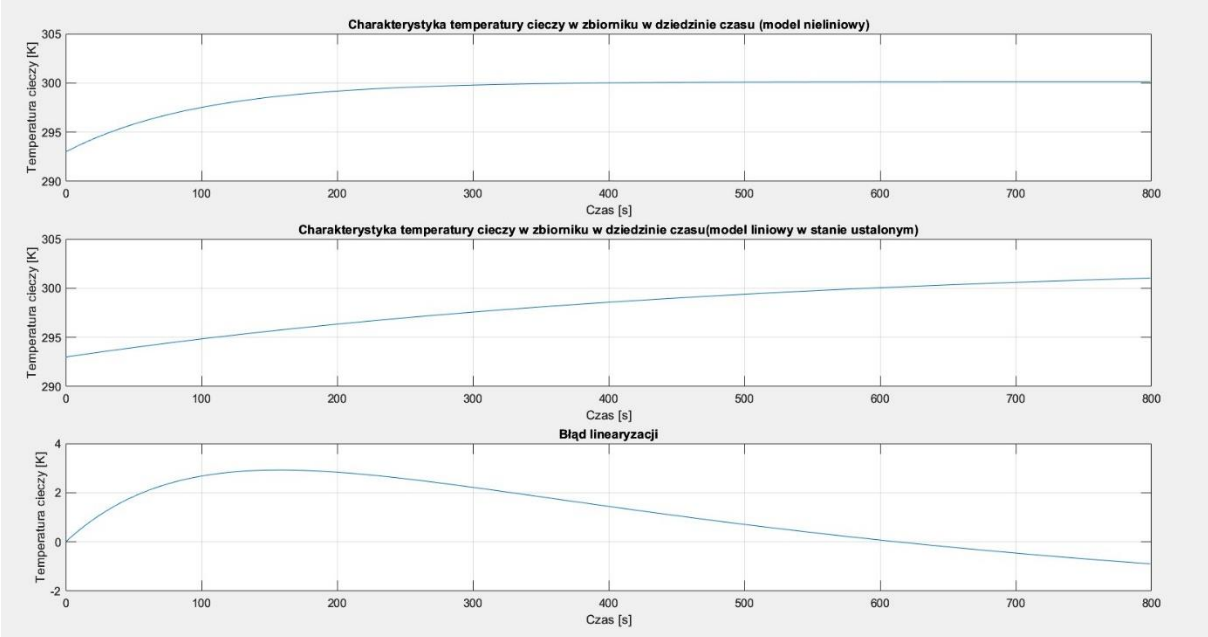
Wykresy – linearyzacja modelu nieliniowego

```
%% Porównanie układu nieliniowego i układu liniowego w stanie ustalonym
Q = 12000;
w = 0.4;
wi = 0.4;
Ti = 293;

T0 = 293;
V0 = 0.04;

% układ nieliniowy
[t, x] = ode45(@zbiornik_stan, [0:799], [V0, T0], [], wi, w, Ti, Q);

X0 = [0.04; 303];
U0 = [0.4; 0.4; 293; 12000];
Y0 = [0.04; 303];

IX = [];
IU = [1; 2; 3];
IY = [1; 2];

[x_, u_, y_, dx_] = trim('zbiornik_sys', X0, U0, Y0, IX, IU, IY);

[A, B, C, D] = linmod('zbiornik_sys', x_, u_);

U = zeros(length(t), length(u_));
X_pocz = [0.04 293];
X_ust = [0.04 303];
x0 = X_pocz - X_ust;

% układ liniowy w stanie ustalonym
y__ = lsim(A, B, C, D, U, t, x0);

% błąd linearyzacji
Y = x(:, 2) - (y__(:, 2) + X_ust(2));
```

Kod programu – porównanie układu nieliniowego i liniowego w stanie ustalonym (I)

```
figure;
subplot(3, 1, 1);
plot(t, x(:, 2));
ylim([290 305]);
title("Charakterystyka temperatury cieczy w zbiorniku w dziedzinie czasu (model nieliniowy)");
xlabel("Czas [s]"); ylabel("Temperatura cieczy [K]");
grid on;
subplot(3, 1, 2);
plot(t, y__(:, 2) + X_ust(2));
ylim([290 305]);
title("Charakterystyka temperatury cieczy w zbiorniku w dziedzinie czasu" + ...
    "(model liniowy w stanie ustalonym)");
xlabel("Czas [s]"); ylabel("Temperatura cieczy [K]");
grid on;
subplot(3, 1, 3);
plot(t, Y);
ylim([-2 4]);
title("Błąd linearyzacji");
xlabel("Czas [s]"); ylabel("Temperatura cieczy [K]");
grid on;
```

Kod programu – porównanie układu nieliniowego i liniowego w stanie ustalonym (II)



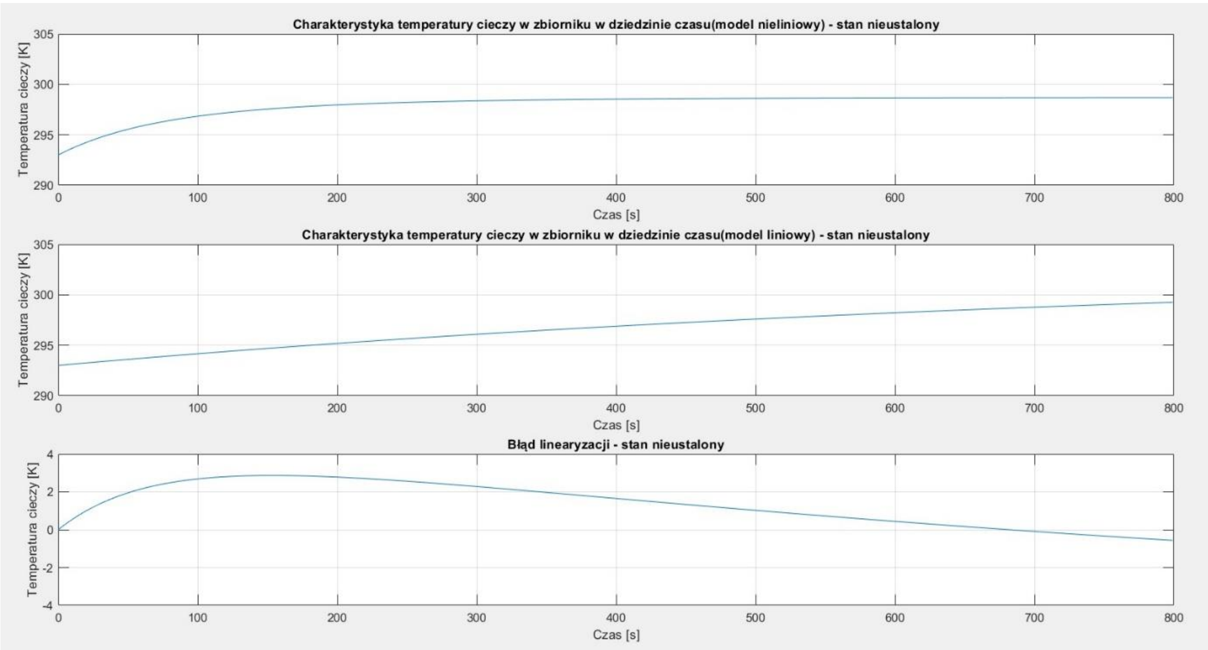
Wykresy – porównanie układu nieliniowego i liniowego w stanie ustalonym

```
%% Stan nieustalony
Q = 12000;
w = 0.4;
wi = 0.5;
Ti = 293;

T0 = 293;
V0 = 0.04;

% układ nieliniowy
[t, x] = ode45(@zbiornik_stan, [0:799], [V0, T0], [], wi, w, Ti, Q);

X0 = [0.04; 303];
U0 = [0.5; 0.4; 293; 12000];
Y0 = [0.04; 303];

IX = [];
IU = [1; 2; 3];
IY = [1; 2];

[x_, u_, y_, dx_] = trim('zbiornik_sys', X0, U0, Y0, IX, IU, IY);

[A, B, C, D] = linmod('zbiornik_sys', x_, u_);

U = zeros(length(t), length(u_));
X_pocz = [0.04 293];
X_ust = [0.04 303];
x0 = X_pocz - X_ust;

% układ liniowy
y__ = lsim(A, B, C, D, U, t, x0);

% błąd linearyzacji
Y = x(:, 2) - (y__(:, 2) + X_ust(2));
```

Kod programu – porównanie układu nieliniowego i liniowego w stanie nieustalonym (I)

```
figure;
subplot(3, 1, 1);
plot(t, x(:, 2));
ylim([290 305]);
title("Charakterystyka temperatury cieczy w zbiorniku w dziedzinie czasu" + ...
    "(model nieliniowy) - stan nieustalony");
xlabel("Czas [s]"); ylabel("Temperatura cieczy [K]");
grid on;
subplot(3, 1, 2);
plot(t, y__(:, 2) + X_ust(2));
ylim([290 305]);
title("Charakterystyka temperatury cieczy w zbiorniku w dziedzinie czasu" + ...
    "(model liniowy) - stan nieustalony");
xlabel("Czas [s]"); ylabel("Temperatura cieczy [K]");
grid on;
subplot(3, 1, 3);
plot(t, Y);
ylim([-4 4]);
title("Błąd linearyzacji - stan nieustalony");
xlabel("Czas [s]"); ylabel("Temperatura cieczy [K]");
grid on;
```

Kod programu – porównanie układu nieliniowego i liniowego w stanie nieustalonym (II)



Wykresy – porównanie układu nieliniowego i liniowego w stanie nieustalonym

4. Wnioski

- Linearyzacja układu nieliniowego polega na znalezieniu liniowego przybliżenia nieliniowej zależności w pewnym określonym obszarze.
- Linearyzację zazwyczaj przeprowadza się wokół stanu ustalonego.
- Aby dokonać linearyzacji modelu nieliniowego w *Matlabie* należy zastosować funkcje *trim oraz linmod.*

5. Bibliografia

- Konspekt do zajęć „Linearyzacja”