



## Linearyzacja układów nieliniowych

### 1. Linearyzacja

Układy liniowe to układy, których opis ma postać zależności liniowych. W szczególności, musi być spełniona *zasada superpozycji*, która stanowi, że reakcja układu liniowego na wymuszenie o postaci:

$$u = c_1 u_1 + c_2 u_2$$

ma postać:

$$y = c_1 y_1 + c_2 y_2$$

przy czym  $y_1$  i  $y_2$  stanowią wynik oddziaływania oddzielnych wymuszeń  $u_1$  i  $u_2$  ( $c_1$  i  $c_2$  są dowolnymi stałymi). Zatem w opisie równania liniowego nie mogą występować żadne operacje nieliniowe na zmiennych układu (np. iloczyny lub potęgi zmiennych), a parametry układu (współczynniki równań) nie mogą zależeć od zmiennych.

Z formalnego punktu widzenia, w rzeczywistości układy liniowe nie istnieją. Postulat liniowości wiąże się bowiem z niezwykle ostrymi warunkami, w szczególności z wymaganiem, aby żadna ze zmiennych układu nie podlegała żadnym ograniczeniom. Stoi to w oczywistej sprzeczności z doświadczeniem, które uczy, że w otaczającym nas świecie nie istnieją zmienne fizyczne, które mogłyby przyjmować wartości dowolnie duże. Można więc mówić jedynie o modelu liniowym układu – aby taki model można było zastosować do danego układu fizycznego, należy znaleźć przynajmniej pewien obszar wartości zmiennych, w którym postulat liniowości jest spełniony z zadowalającą dokładnością [2].

Z drugiej strony, klasyczna teoria sterowania została wyprowadzona dla układów liniowych i dlatego może być stosowana tylko dla nich lub liniowych aproksymacji układów nieliniowych. Wygoda oraz dostępność efektywnych metod analizy układów liniowych oraz ścisłych rozwiązań analitycznych dotyczących większości przypadków praktycznych sprawia, że modele liniowe stosuje się powszechnie.

Aby uzyskać model liniowy układu nieliniowego należy przeprowadzić tzw. *linearyzację*. Jest to uproszczenie układu nieliniowego w taki sposób, że charakterystykę nieliniową przybliża się lokalnie (tzn. w pewnym obszarze) odpowiednio dobraną zależnością liniową.

Założmy, że jest dany model dynamiki układu o postaci:

$$\frac{dy}{dt} = f(y, u)$$

gdzie  $y$  jest wyjściem, a  $u$  wejściem układu. Liniowa aproksymacja takiego układu może być przeprowadzona za pomocą rozwinięcia równania w szereg Taylora i odrzucenia członów wyższego rzędu, czyli:

$$f(y, u) \cong f(\bar{y}, \bar{u}) + \left. \frac{\partial f}{\partial y} \right|_{\bar{y}, \bar{u}} (y - \bar{y}) + \left. \frac{\partial f}{\partial u} \right|_{\bar{y}, \bar{u}} (u - \bar{u})$$

gdzie  $(\bar{y}, \bar{u})$  oznacza tzw. *punkt pracy*, wokół którego dokonujemy linearyzacji. Taka linearyzacja jest możliwa tylko lokalnie, w otoczeniu wybranego punktu pracy i ma sens tylko dla małych odchyłeń od tego punktu. To znaczy że odchylenie  $y - \bar{y}$  punktu bieżącego  $y$  od punktu  $\bar{y}$ , wokół którego dokonujemy rozwinięcia, musi być dostatecznie małe (to samo tyczy się oczywiście odchylenia  $u - \bar{u}$ ).

Z definicji, w punkcie pracy występuje tzw. stan ustalony, wobec tego pierwszy człon z szeregu Taylora się zeruje:

$$f(\bar{y}, \bar{u}) = 0$$

Wprowadzając jeszcze tzw. zmienne odchyłkowe:  $y' = y - \bar{y}$  oraz  $u' = u - \bar{u}$  (czyli odchyłki od punktu pracy), równanie Taylora można uprościć do postaci:

$$f(y, u) \cong \left. \frac{\partial f}{\partial y} \right|_{\bar{y}, \bar{u}} y' + \left. \frac{\partial f}{\partial u} \right|_{\bar{y}, \bar{u}} u'$$

W ten sposób otrzymujemy zależność liniową, ale sensowną tylko w pobliżu punktu pracy.

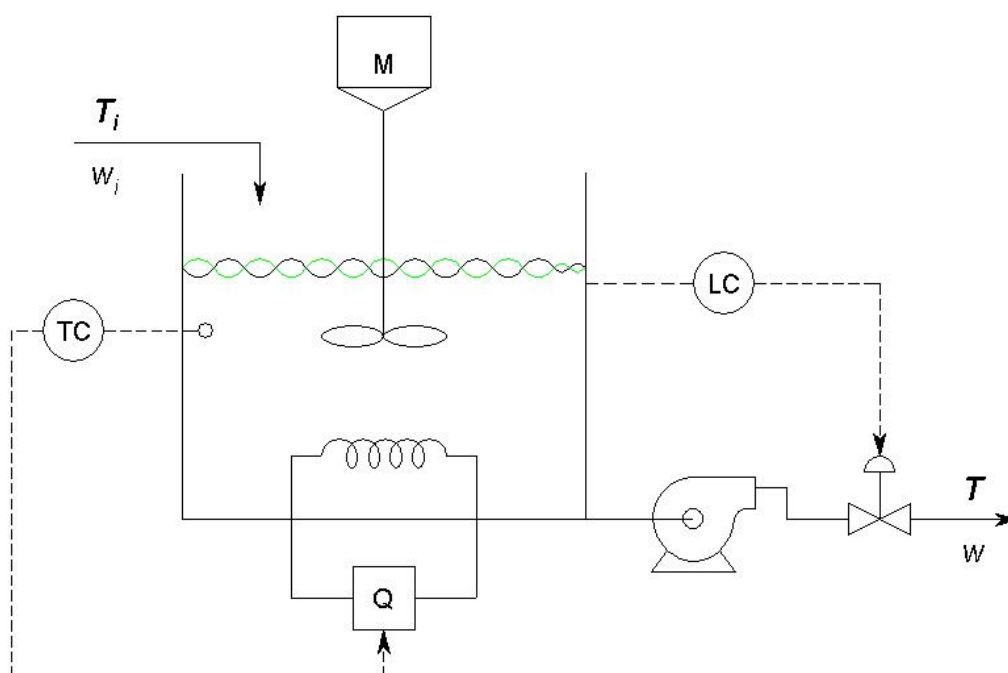
## 2. Obiekt nieliniowy

Obiektem, który będziemy linearyzować jest model zbiornika ze stałym dopływem i wypływem oraz grzaniem, z założeniem idealnego mieszania. Model taki można opisać równaniami różniczkowymi wyprowadzonymi na podstawie zasady zachowania masy [3]:

$$\rho \frac{dV}{dt} = w_i - w \quad (1)$$

i zasady zachowania energii:

$$V\rho \frac{dT}{dt} = w_i(T_i - T) + \frac{Q}{C} \quad (2)$$



Rys. 1. Zbiornik ze stałym wypływem i grzaniem.

### Oznaczenia:

$w_i$ [kg/s]	– strumień wejściowy masowy,
$w$ [kg/s]	– strumień wyjściowy masowy,
$\rho$ [kg/m <sup>3</sup> ]	– gęstość cieczy,
$V$ [m <sup>3</sup> ]	– objętość cieczy w zbiorniku,
$T_i$ [K]	– temperatura strumienia wejściowego,
$T$ [K]	– temperatura strumienia wyjściowego,
$Q$ [W]	– moc grzałki,
$C$ [J/(kg*K)]	– ciepło właściwe cieczy.

### 3. Symulacja modelu nieliniowego

Aby dokonać symulacji układu w Matlabie, czyli rozwiązać układ równań różniczkowych, należy je odpowiednio zapisać w postaci funkcji :

```
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 = 1820;      % 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];
```

Powyższa funkcja zwraca stan układu dla bieżącej chwili czasowej. Funkcję tę należy następnie użyć jako argument wejściowy odpowiedniego solvera, np. **ode45**:

```
[t,x] = ode45(@zbiornik_stan, [0 600], [V0,T0], [], wi, w, Ti, Q);
```

gdzie:

[0 600] – czas symulacji (od zera do 600 sek.)  
[V0,T0] – wartości początkowe  
[] – pusty nawias, oznaczający domyślne opcje solvera  
wi,w,Ti,Q – wartości wejściowe

przykładowe wartości:

wejścia	stan początkowy
$Q = 8000$ [W]	$T_0 = 293$ [K]
$w = 0.4$ [kg/s]	$V_0 = 0.04$ [m <sup>3</sup> ]
$w_i = 0.4$ [kg/s]	
$T_i = 293$ [K]	

Funkcja zwraca  $[t,x]$ , gdzie  $t$  jest wektorem czasu, a  $x$  macierzą rozwiązań, w której pierwsza kolumna przechowuje wartości  $x(1)$  – objętość cieczy w zbiorniku, a druga  $x(2)$  – temperaturę strumienia wyjściowego.



Wykonaj symulację i narysuj odpowiedzi układu na osobnych wykresach w jednym oknie graficznym (wykorzystując funkcję `subplot`). Zbadaj odpowiedzi dla różnych wartości wejść (np. dla dopływu i wypływu o różnych wartościach).

#### 4. Stan ustalony

Proces dynamiczny znajduje się w **stanie równowagi**, jeśli zmienne stanu pozostają stałe w pewnym skończonym przedziale czasu. Wartości zmiennych stanu określają wówczas punkt równowagi układu. Stanowi równowagi mogą towarzyszyć zmiany wejść kompensujące się wzajemnie. Jeśli w stanie równowagi wszystkie wejścia pozostają stałe przez okres dłuższy niż największe opóźnienie czasowe występujące w układzie, to proces znajduje się w **stanie ustalonym**. Stan ustalony procesu ma ważną własność *samopodtrzymania*, tzn. utrzymuje się aż do wystąpienia zmian na wejściu procesu. Większość procesów przemysłowych, szczególnie procesów dużej skali, jest projektowana z myślą o ich prowadzeniu w stanie ustalonym. W związku z tym poszukiwanie optymalnych oraz bezpiecznych stanów ustalonych odgrywa istotną rolę w sterowaniu [1].

Układy nieliniowe linearyzuje się zazwyczaj wokół stanu ustalonego, co prowadzi do modelu o ograniczonej adekwatności, ale stacjonarnego (czyli takiego, którego parametry nie zmieniają się w czasie). W związku z powyższym, zanim dokonamy linearyzacji naszego układu, musimy znaleźć jego stan ustalony.

W dziedzinie czasu, wartości wyjść z stanie ustalonym można wyliczyć analitycznie z równań stanu, podstawiając zera w miejsce pochodnych stanu (lewe strony równań (1,2)). W dziedzinie Laplace'a (dla modeli liniowych), można je wyliczyć za pomocą twierdzenie o wartości granicznej:

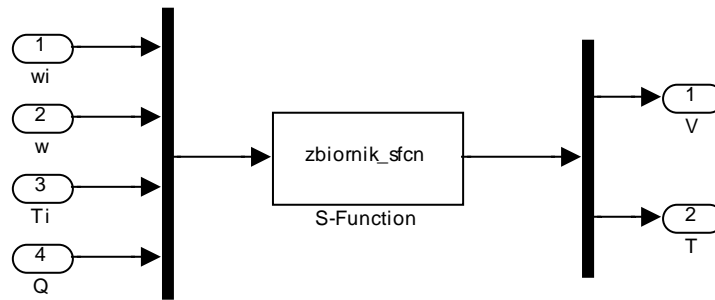
$$y_u = \lim_{s \rightarrow 0} sY(s)$$

Stan ustalony można także policzyć numerycznie za pomocą funkcji Matlab'a `trim`. Aby to było możliwe trzeba zapisać model naszego zbiornika za pomocą tzw. s-funkcji, czyli funkcji którą można wywołać w Simulinku:

```
function [sys,x0,str,ts]=zbiornik_sfcn(t,x,u,flag,V0,T0)

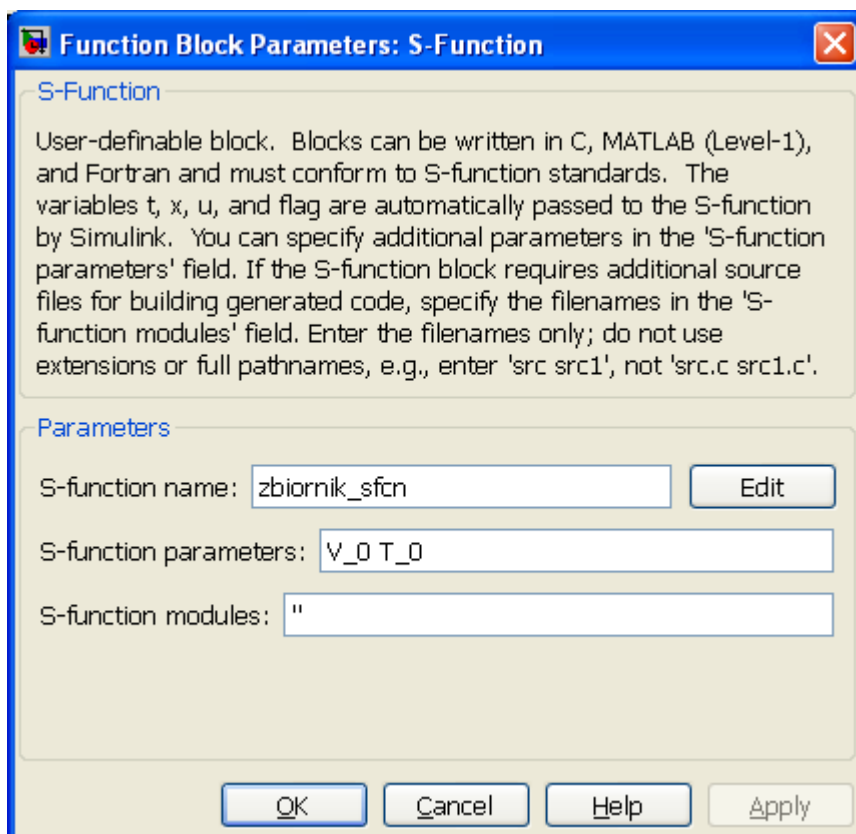
switch flag
case 0 % inicjalizacja
    str = [];
    ts = [0 0];
    s = simsizes;
    s.NumContStates = 2; % liczba stanów ciągłych
    s.NumDiscStates = 0; % liczba stanów dyskretnych
    s.NumOutputs = 2; % liczba wyjść
    s.NumInputs = 4; % liczba wejść
    s.DirFeedthrough = 0; % wejście nie przenosi 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
```

Na podstawie powyższej funkcji – `zbiornik_sfcn.m` – można już utworzyć model w Simulinku:



Wejścia i wyjścia w powyższym modelu muszą być utworzone za pomocą bloków *Inport* i *Outport*.

Parametry s-funkcji (po kliknięciu w blok S-Function) są następujące:



W polu *S-Function name* wpisujemy nazwę stworzonej s-funkcji, a w polu *S-Function parameters* wpisujemy wartości początkowe  $V_0$  i  $T_0$ . Zapisujemy tak utworzony model Simulinka do pliku `zbiornik_sys.mdl`.

Teraz możemy użyć funkcji Matlaba `trim` do znalezienia stanu ustalonego:

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

Znaczenie argumentów wejściowych wraz z przykładowymi wartościami:

`zbiornik_sys` – nazwa pliku s-funkcji z modelem

`X0=[]` – wektor stanu

`U0=[0.1;0.1;293;8000]` – wektor wejść [ $w_i, w, T_i, Q$ ]

`Y0=[0.04;303]` – wektor wyjść (chcemy aby objętość cieczy w zbiorniku była stała i wynosiła  $0.04 \text{ m}^3$  oraz aby temperatura wyjściowa wynosiła 303 K)

$x_0, u_0, y_0$  stanowią punkt wokół którego funkcja `trim` stara się znaleźć stan ustalony. Można również zablokować wybrane wartości wejść, wyjść i stanu aby pozostały stałe (tzn. aby funkcja `trim` ich nie zmieniała). Służą do tego wektory  $IX, IU, IY$ , np.:

$IX=[]$  – wartości stanu nie są blokowane

$IU=[3]$  – trzecia zmienna wejściowa (czyli  $T_1$ ) jest zablokowana (na wartości 293)

$IY=[1;2]$  – pierwsza ( $V=0.04$ ) i druga ( $T=303$ ) zmienna wyjściowa jest zablokowana

Funkcja `trim` zwraca wyniki w postaci wektorów  $x, u, y$  (wartości stanu, wejść i wyjść w stanie ustalonym). Jeśli znalezienie punktu równowagi nie będzie możliwe, funkcja zwróci najbliższy punkt, który minimalizuje następujące kryterium:

$$\text{abs}([x(ix)-x_0(ix); u(iu)-u_0(iu); y(iy)-y_0(iy)])$$



Wykonaj obliczenia stanu ustalonego i narysuj odpowiedzi układu w stanie ustalonym (na osobnych wykresach w jednym oknie graficznym). Zbadaj działanie funkcji `trim` dla różnych wartości parametrów wejściowych.

## 5. Model liniowy

Po obliczeniu stanu ustalonego możemy przystąpić do linearyzacji układu nieliniowego w punkcie pracy. Służą do tego funkcja Matlaba `linmod`:

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

gdzie  $x$  jest wektorem stanu, a  $u$  wektorem wejść w stanie ustalonym (czyli argumenty wyjściowe funkcji `trim`). W wyniku dostaniemy model liniowy w przestrzeni stanu, czyli macierze  $A, B, C, D$ . Model ten można teraz opisać za pomocą układu równań:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

gdzie  $u$  jest wejściem,  $y$  – wyjściem,  $x$  – stanem układu.

Model można także zapisać w postaci transmitancji, za pomocą funkcji `ss2tf`:

```
[licz,mian] = ss2tf(A,B,C,D)
```

Aby policzyć odpowiedzi układu na zadane wejście, można użyć funkcji `lsim`:

```
[y,t] = lsim(A,B,C,D,u,t,x0)
```

Gdzie macierz  $u$  oznacza wektor wejść – musi mieć tyle rzędów ile wynosi czas symulacji (czyli `length(t)`) i tyle kolumn ile wejść posiada układ. Wektor  $t$  oznacza czas symulacji, a wektor  $x_0$  wartości początkowe symulacji (stan początkowy). Funkcja zwraca wektor odpowiedzi  $y$  (ma tyle kolumn ile wynosi liczba wyjść układu) i czas symulacji  $t$ .



Narysuj przebiegi odpowiedzi układu zlinearyzowanego w stanie ustalonym. Narysuj także odpowiedzi w stanach przejściowych – poeksperymentuj z różnymi wartościami wejść.

## 6. Literatura

1. Duda, J.T., *Modele matematyczne, struktury i algorytmy nadrzędnego sterowania komputerowego*, Uczelniane Wydawnictwa Naukowo-Dydaktyczne AGH, Kraków, 2003.
2. Markowski A., Kostro J., Lewandowski A., *Automatyka w pytaniach i odpowiedziach*, Wydaw. Naukowo-Techniczne, Warszawa, 1985.
3. Seborg, D.E., *Process dynamics and control*, John Wiley & Sons, New York , 1989