

## Laboratorium

### **Sterowanie napędami elektrycznymi – zagadnienia wybrane**

#### Ćwiczenie

Badanie struktury sterowania predykcyjnego prędkością i położeniem napędu z silnikiem prądu stałego.

Opracował: mgr inż. Piotr J. Serkies

## 1. Cel ćwiczenia

Celem ćwiczenia jest:

- Zapoznanie się z możliwościami wykorzystania regulatorów predykcyjnych w napędzie prądu stałego.
- Ocena wpływu długości horyzontu predykcji wyjść na właściwości dynamiczne napędu.
- Porównanie działania regulatora predykcyjnego z kaskadową strukturą regulacji położenia napędu prądu stałego.

## 2. Zakres ćwiczenia

- Budowa modelu dla predykcyjnego regulatora położenia.
- Budowa funkcji celu przy sterowaniu prędkością i położeniem.
- Badania symulacyjne.
- Badania eksperymentalne.

## 3. Literatura

- [1] materiały z wykładu Podstawy Automatyki
- [2] materiały z wykładu Sterowanie napędami elektrycznymi – zagadnienia wybrane
- [3] Tunia H., Kaźmierkowski M.P., Automatyka napędu przekształtnikowego, WNT, 1987
- [4] MACIEJOWSKI J.M, Predictive Control with Constraints, Prentice Hall, UK, 2002.
- [5] EL-SHARKAWI M. A., Fundamentals of electric al drivers, Brooks/Cole, 2000
- [6] TATJEWSKI P., Sterowanie zaawansowane obiektów przemysłowych, struktury i algorytmy, Akademicka Oficyna Wydawnicza EXIT, 2002.
- [7] KVASNICA M., GRIEDER P., BAOTIC M., MORARI M., Multi-Parametric Toolbox (MPT), HSCC (Hybrid Systems: Computation and Control), Lecture Notes in Computer Science, vol. 2993, 2004, 448-46.
- [8] <http://control.ee.ethz.ch/~mpt/>

## 1. ZASADA REGULACJI PREDYKCYJNEJ

Zasada działania dyskretnych algorytmów predykcyjnych [4],[6] polega na minimalizacji różnicy pomiędzy wartościami wielkości regulowanych  $y(k+p/k)$  predyktowanych (przewidywanymi) w chwili  $k$  na przyszłą chwilę  $k+p$  (przyszłe wartości wyjść  $k+p$  są wyznaczane z modelu, na podstawie wartości z chwili  $k$ ), a wartościami zadanymi dla tych chwil  $y^{zad}(k+p/k)$  na horyzoncie predykcji  $N$  ( $p=1,2,\dots,N$ ). Opisywana minimalizacja różnicy jest rozumiana w sensie minimalizacji określonej funkcji kryterialnej. W kolejnym okresie próbkowania (chwila  $k+1$ ) następuje ponowny pomiar sygnałów wyjściowych obiektu i cały proces jest powtarzany z takim samym horyzontem predykcji  $N$ . Przedstawiona powyżej zasada sterowania predykcyjnego jest więc sterowaniem repetycyjnym. W opisywanym algorytmie zakłada się również, iż po upływie horyzontu sterowania  $N_u$  ( $N_u < N$ ) przyrost sygnału sterującego wynosi zero, co oznacza, że regulator predykcyjny posiada właściwości całkujące. W przypadku zadania sterowania z ograniczeniami regulacja predykcyjna prowadzi do rozwiązania problemu optymalizacji na bieżąco.

Funkcja kryterialna ma najczęściej postać [6]:

$$J(k) = \sum_{p=N_1}^N (y^{zad}(k+p|k) - y(k+p|k))^T \mathbf{Q}(p) (y^{zad}(k+p|k) - y(k+p|k)) + \sum_{p=0}^{N_u-1} \Delta u(k+p|k) \mathbf{R}(p) \Delta u(k+p|k) \quad (1)$$

gdzie:  $y^{zad}(k+p/k)$  jest wektorem wartości zadanych o wymiarze  $n_y = \dim(y)$ ,  $y(k+p/k)$  jest wektorem wartości regulowanych o wymiarze  $n_y = \dim(y)$ ,  $\Delta u(k+p/k)$  jest wektorem przyrostów sterowań o wymiarze  $n_u = \dim(u)$ ,  $\mathbf{Q}(p) > 0$  jest macierzą wagową składowych wektora uchybu  $y^{zad}(k+p/k) - y(k+p/k)$  przewidywanych na chwilę  $k+p$ . Jest to najczęściej macierz diagonalna,  $\mathbf{R}(p) \geq 0$  jest macierzą wagową wektora przyrostów sterowań przewidywanych na chwilę  $k+p$ . Macierz ta określa wartości kary za zmienność sterowań w stosunku do redukcji uchybów regulacji. W przypadku, gdy elementy macierzy  $\mathbf{R}(p)$  są zerowe, regulator staje się mało odporny na szumy pomiarowe. Przy założeniu występowania ograniczeń można zapisać pełne zadanie optymalizacji [4],[5]:

$$\min_{\Delta u(k|k), \dots, \Delta u(k+N_u-1|k)} \left\{ \sum_{p=N_1}^N \|y^{zad}(k+p|k) - y(k+p|k)\|_{\mathbf{Q}(p)}^2 + \sum_{p=0}^{N_u-1} \|\Delta u(k+p|k)\|_{\mathbf{R}(p)}^2 \right\} \quad (2)$$

$$u_{\min} \leq u(k+p|k) \leq u_{\max} \quad p = 0, 1, \dots, N_u$$

$$\Delta u_{\min} \leq \Delta u(k+p|k) \leq \Delta u_{\max} \quad p = 0, 1, \dots, N_u$$

$$y_{\min} \leq y(k+p|k) \leq y_{\max} \quad p = N_1, N_1 + 1, \dots, N$$

Algorytm sterowania predykcyjnego może być zaimplementowany na dwa sposoby. Sposób tradycyjny polega na rozwiązaniu problemu optymalizacji on-line dla danego stanu układu  $x(k)$  i przyjętej w algorytmie długości horyzontu predykcji. Należy podkreślić, że zwiększenie długości okresu przewidywania umożliwia uzyskanie lepszych właściwości dynamicznych obiektu, jednakże w znacznym stopniu wpływa na komplikację algorytmu sterowania (zwiększa wymagany nakład obliczeniowy). W praktyce bardzo często przyjmuje się stałą wartość sygnału sterującego dla całego horyzontu predykcji. W znacznym stopniu redukuje to złożoność obliczeniową całego algorytmu. Po wyznaczeniu optymalnej wartości sygnału sterującego w chwili  $k$  jest ona podawana na sterowany obiekt. Kolejnej chwili obliczeniowej ( $k+1$ ) cała procedura jest powtarzana od nowa uwzględniając aktualne wartości wektora stanu obiektu  $x(k+1)$  (mierzone bądź estymowane). Opisana strategia sterowania wymaga zastosowania szybkich układów mikroprocesorowych i dlatego jest stosowane w przypadku obiektów o relatywnie dużych stałych czasowych.

W drugim przypadku problem optymalizacji dla danej postaci funkcji celu (2) jest rozwiązany off-line dla wszystkich kombinacji wektora stanu  $x \in X_f$  przy użyciu programowania wieloparametrycznego (ang. multiparametric programming) [7],[8]. Korzystając z niego można wykazać, że przestrzeń  $X_f$  może być podzielona na regiony w których optymalizator jest wyrażony jako funkcja jawna dla danych wartości wektora  $x$ . Prawo sterowania może być traktowane jako odcinkami ciągłe i wyrażone następująco:

$$U(x) = \mathbf{K}_r x + g_r, \quad \forall x \in P_r \quad (3)$$

gdzie  $P_r$  są wielościenneymi zbiorami zdefiniowanymi jako:

$$P_r = \{x \in \mathfrak{R}^n \mid H_r x \leq d_r\}, \quad r = 1, \dots, N_r \quad (4)$$

Algorytmy projektowania wielościennej zbiorów i wyliczenia prawa sterowania są opisane szczegółowo w [7],[8]. W najprostszym przypadku prawo sterowania może być wyliczane przez przeszukiwanie po kolei wszystkich regionów, aż do znalezienia regionu aktywnego i kolejno zaimplementowania prawa sterowania zgodnie z (4). Należy zaznaczyć, że w literaturze można znaleźć bardziej efektywne algorytmy generacji bieżącej wartości prawa sterowania. Zastąpienie optymalizacji wykonywanej on-line wersją off-line umożliwi praktyczną realizację sterowania predykcyjnego do układów o małych stałych czasowych takich jak napędy elektryczne.

## 2. MODEL NAPĘDU

W przypadku przeważającej części napędów przyjmuje się liniowy model matematyczny części mechanicznej. Jednak w przypadku precyzyjnych zastosowań należy uwzględnić zjawiska związane z tarciem [3], [5]. Model uwzględniający tarcie można opisać poniższymi równaniami:

$$\begin{cases} T_e \frac{d}{dt} i_t = k_t u_t - i_t - k_t \psi_w \omega \\ m_e = \psi_w i_t \\ T_m \frac{d}{dt} \omega = m_e - m_L \\ \frac{d}{dt} \alpha = \frac{1}{T_\alpha} \omega \end{cases} \quad (5)$$

gdzie:  $T_e$  – stała elektromagnetyczna,  $k_t$  – współczynnik wzmocnienia twornika,  $\psi_w$  -strumień wzbudzenia,  $T_m$  – stała mechaniczna,  $i_t$  – prąd twornika,  $\omega$  – prędkość napędu,  $\alpha$  – położenie wału,  $m_L$  – moment obciążenia,  $T_m$  – mechaniczna stała czasowa ( $T_m=0.406s$ ),  $T_\alpha$  – stała pozycjonowania.

Model (5) można zapisać w formie równań stanu:

$$\frac{d}{dt} \begin{bmatrix} i_t \\ \omega \\ \alpha \end{bmatrix} = \underbrace{\begin{bmatrix} -1 & -k_t \psi_w & 0 \\ T_e & T_e & 0 \\ \psi_w & 0 & 0 \\ T_m & 0 & 0 \\ 0 & 1 & 0 \\ & T_\alpha & 0 \end{bmatrix}}_{A_m} \begin{bmatrix} i_t \\ \omega \\ \alpha \end{bmatrix} + \underbrace{\begin{bmatrix} k_t \\ T_e \\ 0 \\ 0 \end{bmatrix}}_{B_m}^T \begin{bmatrix} i_t \\ \omega \\ \alpha \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ -1 \\ T_m \\ 0 \end{bmatrix}}_{B_d}^T \begin{bmatrix} i_t \\ \omega \\ \alpha \end{bmatrix} \quad (6)$$

W kolejnym kroku model należy rozszerzyć o dwie dodatkowe zmienne stanu: moment obciążenia i zadane położenie. Dynamika tych zmiennych stanu jest nieznaną:

$$\frac{d}{dt} m_L = 0 \quad (7)$$

$$\frac{d}{dt} \alpha^{ref} = 0 \quad (8)$$

Po wprowadzeniu zmian model przyjmie postać:

$$\frac{d}{dt} \begin{bmatrix} i_t \\ \omega \\ \alpha \\ m_L \\ \alpha^{ref} \end{bmatrix} = \underbrace{\begin{bmatrix} -1 & \frac{-k_t \psi_w}{T_e} & 0 & 0 & 0 \\ \frac{\psi_w}{T_m} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{T_\alpha} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{A_m} \begin{bmatrix} i_t \\ \omega \\ \alpha \\ m_L \\ \alpha^{ref} \end{bmatrix} + \underbrace{\begin{bmatrix} k_t \\ T_e \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{B_m}^T \begin{bmatrix} i_t \\ \omega \\ \alpha \\ m_L \\ \alpha^{ref} \end{bmatrix} \quad (9)$$

Zapiszmy ogólny liniowy model stanowy:

$$\begin{cases} x(k+1) = \mathbf{A}x(k) + \mathbf{B}u(k) \\ y(k) = \mathbf{C}x(k) \end{cases} \quad (10)$$

gdzie:  $x$  jest wektorem stanu o wymiarze  $n_x$  natomiast  $y$  jest wektorem wyjść regulowanych dostępnych pomiarowo o wymiarze  $n_y$ . Na podstawie wzoru (4.4), zakładając pomiar pełnego wektora stanu oraz brak zakłóceń można wyprowadzić zależność na przewidywaną wartość wyjść [4],[10]. Oznaczmy jak poprzednio przez  $x(k+p|k)$  stan przewidywany w chwili  $k$  na chwilę  $k+p$  horyzontu predykcji i postępując rekurencyjnie dostajemy:

$$\begin{aligned} x(k+1|k) &= \mathbf{A}x(k) + \mathbf{B}u(k|k) = \mathbf{A}x(k) + \mathbf{B}(\Delta u(k|k) + u(k-1)) \\ x(k+2|k) &= \mathbf{A}x(k+1|k) + \mathbf{B}(\Delta u(k+1|k) + \Delta u(k|k) + u(k-1)) = \\ &= \mathbf{A}^2 x(k+1|k) + \mathbf{A}\mathbf{B}(\Delta u(k|k) + u(k-1)) + \mathbf{B}(\Delta u(k+1|k) + \Delta u(k|k) + u(k-1)) = \\ &= \mathbf{A}^2 x(k) + (\mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k|k) + \mathbf{B}\Delta u(k+1|k) + (\mathbf{A} + \mathbf{I})\mathbf{B}u(k-1) \\ &\quad \vdots \\ x(k+N_u|k) &= \mathbf{A}x(k+N_u-1|k) + \mathbf{B}(\Delta u(k+N_u-1|k) + \dots + \Delta u(k|k) + u(k-1)) = \\ &= \mathbf{A}^{N_u} x(k) + (\mathbf{A}^{N_u-1} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k|k) + (\mathbf{A}^{N_u-2} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k+1|k) + \\ &\quad + \dots + \mathbf{B}\Delta u(k+N_u-1|k) + (\mathbf{A}^{N_u-1} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}u(k-1) \\ &\quad \vdots \\ x(k+N|k) &= \mathbf{A}^N x(k) + (\mathbf{A}^{N-1} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k|k) + (\mathbf{A}^{N-2} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k+1|k) + \\ &\quad + \dots + (\mathbf{A}^{N-N_u} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}\Delta u(k+N_u-1|k) + (\mathbf{A}^{N-1} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B}u(k-1) \end{aligned}$$

Zdefiniujmy wektory:  $\mathbf{X}$  i  $\mathbf{U}$  następująco:

$$\mathbf{X}(k) = \begin{bmatrix} x(k+N_1|k) \\ \vdots \\ x(k+N|k) \end{bmatrix}, \quad \mathbf{U}(k) = \begin{bmatrix} \Delta u(k|k) \\ \vdots \\ \Delta u(k+N_u-1|k) \end{bmatrix}$$

Uzyskane wcześniej zależności można przedstawić w formie macierzowej, dla  $p=N_1, \dots, N$

$$\begin{aligned}
X(k) = & \begin{bmatrix} \mathbf{A}^{N_1} \\ \vdots \\ \mathbf{A}^{N_u} \\ \mathbf{A}^{N_u+1} \\ \vdots \\ \mathbf{A}^N \end{bmatrix} x(k) + \begin{bmatrix} (\mathbf{A}^{N_1-1} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B} \\ \vdots \\ (\mathbf{A}^{N_u-1} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B} \\ (\mathbf{A}^{N_u} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B} \\ \vdots \\ (\mathbf{A}^{N-1} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B} \end{bmatrix} u(k-1) + \\
& + \begin{bmatrix} (\mathbf{A}^{N_1-1} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B} & \dots & 0 \\ \vdots & \ddots & \vdots \\ (\mathbf{A}^{N_u-1} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B} & \dots & \mathbf{B} \\ (\mathbf{A}^{N_u} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B} & \dots & (\mathbf{A} + \mathbf{I})\mathbf{B} \\ \vdots & \ddots & \vdots \\ (\mathbf{A}^{N-1} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B} & \dots & (\mathbf{A}^{N-N_u} + \dots + \mathbf{A} + \mathbf{I})\mathbf{B} \end{bmatrix} \Delta U(k)
\end{aligned} \tag{11}$$

Stosując zapis równania (10) do równania (11) otrzymujemy:

$$X(k) = \tilde{\mathbf{A}}x(k) + \tilde{\mathbf{B}}u(k-1) + \mathbf{P}\Delta u(k) \tag{12}$$

Aby uzyskać możliwość wyznaczenia zależności opisującej trajektorię przewidywanych wyjść potrzebna jest definicja macierzy wyjść  $\tilde{\mathbf{C}}$  o wymiarze:  $(N - N_1 + 1)n_y \times (N - N_1 + 1)n_x$ :

$$\tilde{\mathbf{C}} = \begin{bmatrix} \mathbf{C} & 0 & \dots & 0 \\ 0 & \mathbf{C} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{C} \end{bmatrix}$$

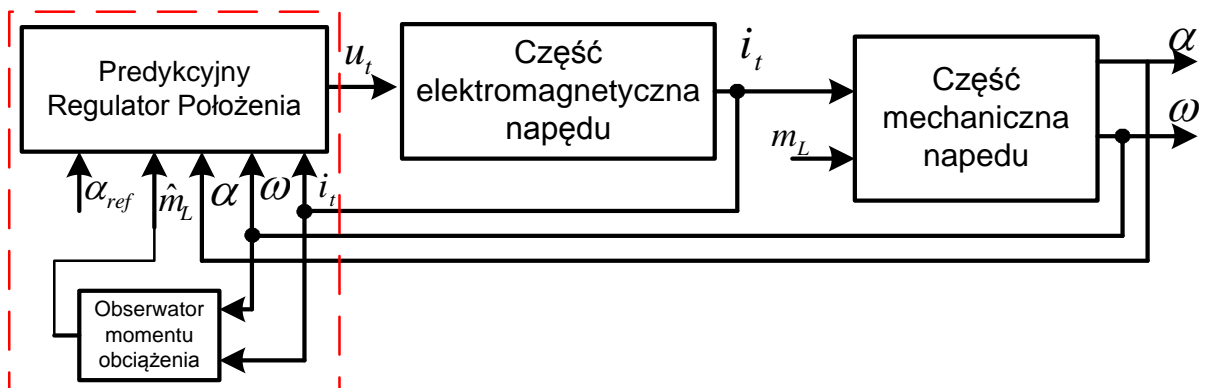
Na tej podstawie można sformułować zależność na  $Y^{pred}(k)$ :

$$Y^{pred}(k) = \tilde{\mathbf{C}}X(k) = [\tilde{\mathbf{C}}\tilde{\mathbf{A}}x(k) + \tilde{\mathbf{C}}\tilde{\mathbf{B}}u(k-1)] + \tilde{\mathbf{C}}\mathbf{P}\Delta U(k) = Y^0(k) + \Delta Y(k) \tag{13}$$

gdzie  $\Delta Y$  jest składową wymuszoną odpowiedzi, natomiast  $Y^0$  jest składową swobodną zależną od poprzednich sterowań.

### 3. Struktura sterowania

Na rys. 1. przedstawiono strukturę sterowania rozpatrywanego napędu. Regulator predykcyjny jest nadrzędnym regulatorem wystawiającym sygnał zadanego napięcia twornika. Do tego celu wykorzystuje informacje o zmiennych stanu napędu.



Rys.1. Struktura sterowania.

Regulator predykcyjny minimalizuje wyjście postaci:

$$y = \alpha^{ref} - \alpha \quad (14)$$

Przy zachowaniu ograniczeń:

$$-i_t^{max} \leq i_t \leq i_t^{max} \quad (15a)$$

$$-\omega^{max} \leq \omega \leq \omega^{max} \quad (16b)$$

Jako podstawowe parametry ograniczeń można przyjąć  $i_t^{max}=3$ ,  $\omega^{max}=1$ .

Jako prosty estymator momentu obciążenia można wykorzystać przekształcone równanie ruchu:

$$T_m \frac{d}{dt} \omega = m_e - m_L \Rightarrow m_L = m_e - T_m \frac{d}{dt} \omega \quad (17)$$

Ze względu na zaszumienie sygnału prędkości należy wprowadzić filtrację sygnałów, co zmniejszy ich oddziaływanie, wprowadzając opóźnienie. Symulator przyjmie postać:

$$m_L = \frac{m_e}{T_q s + 1} - \frac{T_m \omega s}{T_q s + 1} \quad (18)$$

gdzie:  $T_q$  – stała filtru dobrana jako kompromis pomiędzy filtracją szumu a opóźnieniem w estymacji momentu obciążenia.

#### 4. Budowa regulatora z wykorzystaniem toolbox'u MPT

Toolbox można pobrać ze strony: <http://control.ee.ethz.ch/~mpt/>

```
%% parametry modelu
Tm = 0.203; % stała mechaniczna
Ta = 1; % stała pozycjonowania
Te=0.00994; % stała elektromagnetyczna
Kt=8.6759; % wzmacnienie twornika
fi=1.0; % znamionowy strumień wzbudzenia
Ts = 0.5e-3; % okres próbkowania regulatora (czas dyskretyzacji modelu)
Tss=1e-5; % czas dyskretyzacji modelu symulacyjnego

%% Budowa modelu dla regulatora predykcyjnego
A = [-1/Te      -Kt*fi/Te      0      0      0;
      fi/Tm      0      0      -1/Tm      0;
      0      1/Ta      0      0      0;
      0      0      0      0      0;
      0      0      0      0      0];
B = [1*Kt/Te      0      0      0      0]';
D= [0]';
C = [0      0      -1      0      1];

nym = size(C,1);
model_cont = ss(A,B,C,D);
[nx,nu] = size(model_cont.B);
```

```

%% -- EXPLICIT MPC DESIGN & COMPUTATION -----
% System structure
sysStruct      = mpt_sys(model_cont,Ts); % dyskretyzacja modelu
sysStruct.umax = 1.3; % ograniczenie sygnału sterującego od góry
sysStruct.umin = -1.3; % ograniczenie sygnału sterującego od dołu
w_max = 1; % ograniczenie prędkości
ia_max=3; % ograniczenie prądu twornika
sysStruct.xmax  = [ia_max;w_max;Inf;Inf;Inf]; % budowa wektorów ograniczeń od góry
sysStruct.xmin  = -[ia_max;w_max;Inf;Inf;Inf]; % budowa wektorów ograniczeń dołu

db = [ia_max;w_max;1.2;1.1;1.1];% definicja multipowierzchni ograniczającej
                                przestrzeń rozważań
sysStruct.Pbnd  = polytope([eye(5);-eye(5)], [db;db])% definicja
                                multipowierzchni
                                ograniczającej przestrzeń
                                rozważań

SM=110e-4; % podstawowa definicja poziomu niepewności modelu
db = [SM 0.5*SM 0.0001*SM 0.0001*SM 0.0001*SM]' % i dla poszczególnych
                                zmiennych stanu
sysStruct.noise = polytope([eye(5);-eye(5)], [db;db]) % wprowadzenie niepewności
                                do multipowierzchni

% Problem structure
probStruct.Q    = eye(nx); % macierz różnicująca oddziaływanie zmiennych stanu
probStruct.Qy  = [1.2] % macierz różnicująca oddziaływanie minimalizowanych
                                wyjść
probStruct.R    = 0.6e-6; % macierz różnicująca zmienność sterowania
probStruct.N    = 5; % horyzont predykcji wyjść
probStruct.Nc   = 2; % horyzont predykcji sterowań
probStruct.norm = 2; % kwadratowa funkcja celu

explctrl = mpt_control(sysStruct,probStruct); % obliczenie regulatora

```

Model w simulinku:

