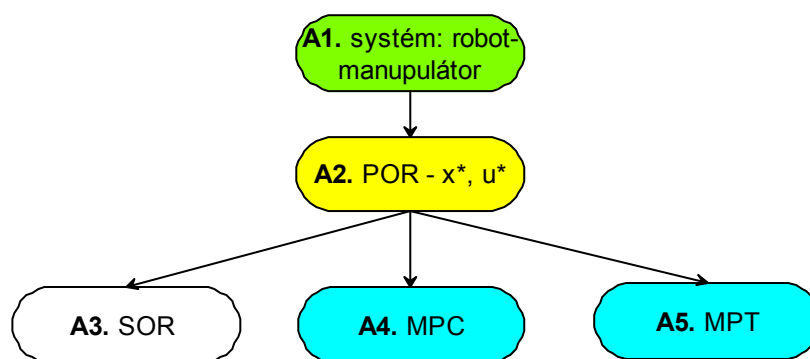


A. Optimálne riadenie NDS – Robot-manipulátor

Úlohy:

1. Implementujte matematický model NDS Robot-manipulátor do programového prostredia MATLAB /Simulink.
2. Navrhnite algoritmus pre riešenie úlohy programového optimálneho riadenia (POR) pre NDS Robot-manipulátor.
3. Navrhnite algoritmus spätно-väzobného optimálneho riadenia pre sledovanie referenčnej trajektórie získanej algoritmom POR.
4. Využitím funkcií toolboxu MPC vykonajte sledovanie optimálnej trajektórie nelineárneho dynamického systému Robot – manipulátor.
5. Využitím funkcií toolboxu MPT vykonajte sledovanie optimálnej trajektórie nelineárneho dynamického systému Robot - manipulátor.

Na nasledujúcom Obr. 1 sa nachádza rozdelenie úloh tutoriálu Optimálne riadenie NDS pre nelineárny dynamický systém Robot-manipulátor. Úlohy A3., A4., a A5. Je možné vykonať až po vykonaní úloh A1. a A2..



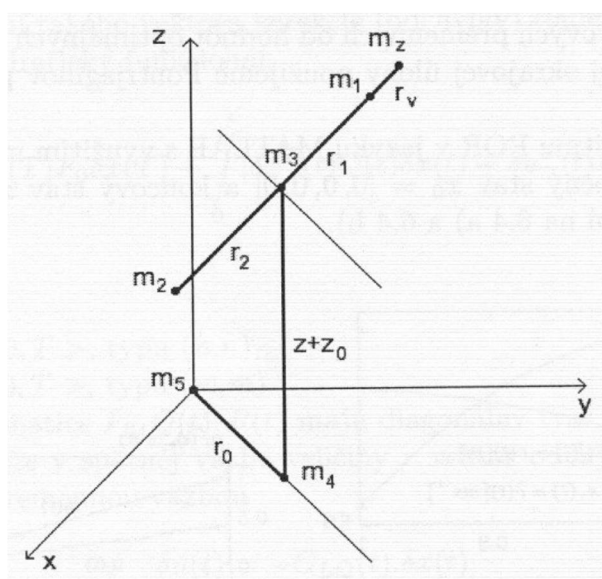
Obr. 1 Rozdelenie tutoriálu Optimálne riadenie NDS - Robot-manipulátor

ÚLOHA A1.

Implementácia matematického modelu NDS Robot-manipulátor do programového prostredia MATLAB/Simulink

Zostavenie matematického modelu Robot-manipulátor

Pri riešení tohto problému potrebujeme získať matematický popis modelu Robot-manipulátor. Robot manipulátor predstavuje robotické rameno s tromi stupňami voľnosti. Kinematická štruktúra robota manipulátora vyzerá nasledovne:



Parametre:

$r = r_1 + r_2$	dĺžka ramena robota
m_z	hmotnosť závažia
m_1	hmotnosť úchopovej hlavice
m_2	hmotnosť motorov v ramene
m_3	hmotnosť podpornej hlavice
m_4	hmotnosť rotujúcej časti stĺpa
m_5	hmotnosť nerotujúcej časti stĺpa
z	zdvih ramena
φ	rotácia ramena

Obr. 2 Kinematická štruktúra modelu Robot-manipulátor

Matematický model je odvodený na základe Lagrangeových rovníc, ktoré majú tvar:

$$\frac{d}{dt} \frac{\partial W_k}{\partial \dot{q}_i} - \frac{\partial W_k}{\partial q_i} + \frac{\partial W_p}{\partial q_i} = Q_i \quad (1)$$

pre $i = 1, 2, \dots, n$, kde n je počet stupňov voľnosti, W_k je kinetická a W_p je potenciálna energia, Q_i je zovšeobecnená sila.

Diferenciálne rovnice pre rotáciu a výsuv ramena:

$$(m_{rr} + m_z + m_1 + m_2)\ddot{r} - (m_z + m_1)r\dot{\varphi}^2 = K_1 u_1 \quad (2)$$

$$(I_{23} + (m_z + m_1)r^2)\ddot{\varphi} + 2(m_z + m_1)r\dot{r}\dot{\varphi} = K_2 u_2$$

a diferenciálna rovnica pre zdvih ramena:

$$(m_1 + m_2 + m_3 + m_z + m_{rr})(\ddot{z} - g) = K_3 u_3, \quad (3)$$

kde:

$I_{23} = I_{\varphi r} + (m_2 + m_3 + m_4)r_0^3$	je moment zotrvačnosti,
m_{rr}	je redukovaná hmotnosť elektromotora, prevodov, a hracej tyče,
$I_{\varphi r}$	je redukovaný moment zotrvačnosti elektromotora a prevodovky
K_1, K_2	sú konštanty akčných veličín.

Diferenciálna rovnica pre zdvih je nezávislá na rovniciach rotácie a výsuvu a v našom riešení bude zanedbávaná.

Pre vyjadrenie matematického modelu Robota-manipulátora v stavovom priestore volíme nasledujúce stavové premenné:

$$x_1 = r [m], x_2 = \dot{r}[m.s^{-1}], x_3 = \varphi [rad], x_4 = \dot{\varphi}[rad.s^{-1}]. \quad (4)$$

Matematický model robota v stavovom priestore má teda tvar:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{m_b}{m_r} x_1 x_4^2 + \frac{K_1}{m_r} u_1 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= -\frac{2m_b x_1 x_2 x_4}{I_{23} + m_b x_1^2} + \frac{K_2}{I_{23} + m_b x_1^2} u_2, \end{aligned} \quad (5)$$

kde $m_b = m_z + m_1$ a $m_r = m_{rr} + m_b + m_2$.

Pozn.: Viac informácií o tomto modeli je možné získať v knihe: Teória optimálnych a adaptívnych systémov [Sarnovský, Ján; Jadlovská, Anna; Kica, Peter] alebo na webovej stránke: http://matlab.fei.tuke.sk/applications/CyberVirtLabMWS/Robot-manipulator/popis_robota.html

ÚLOHA A2.

Navrhните algoritmus pre riešenie úlohy POR pre Robot-manipulátor

Úloha optimálneho riadenia predpokladá, že matematický model je tvorený sústavou diferenciálnych rovníc (5). Úlohou programového optimálneho riadenia je určiť taký vektor riadenia $\mathbf{u}(t)$, aby dynamický systém prešiel z daného počiatočného stavu $\mathbf{x}(t_0) = \mathbf{x}_0$ do určeného koncového stavu $\mathbf{x}(T) = \mathbf{x}_T$ a aby bolo minimalizované kritérium optimálneho riadenia v tvare:

$$J(\mathbf{u}(t)) = \int_{t_0}^T g(\mathbf{x}(t), \mathbf{u}(t), t) dt, \quad (6)$$

kde $\mathbf{u}^T(t)$ je transponovaný vektor k vektoru riadenia $\mathbf{u}(t)$. Získaný vektor riadenia $\mathbf{u}^*(t)$ a zodpovedajúcu trajektóriu $\mathbf{x}^*(t)$ nazývame optimálnymi.

Riešenie optimalizačnej úlohy

Riešte optimalizačnú úlohu, kde kritérium optimálneho riadenia (minimalizácia spotreby energie z bodu A do bodu B za daný čas) pre systém Robot-manipulátor je zadané nasledovné:

$$J(\mathbf{u}(t)) = \frac{1}{2} (u_1^2(t) + u_2^2(t)). \quad (7)$$

Je zadaný systém stavových rovníc NDS Robot-manipulátor:

$$\begin{aligned} \dot{x}_1 &= x_2 = f_1(x, u, t) \\ \dot{x}_2 &= \frac{m_b}{m_r} x_1 x_4^2 + \frac{K_1}{m_r} u_1 = f_2(x, u, t) \\ \dot{x}_3 &= x_4 = f_3(x, u, t) \\ \dot{x}_4 &= -\frac{2m_b x_1 x_2 x_4}{I_{23} + m_b x_1^2} + \frac{K_2}{I_{23} + m_b x_1^2} u_2 = f_4(x, u, t) \end{aligned} \quad (8)$$

A zadaný je aj počiatočný $t_0 = 0$ a koncový $T = 1s$ čas riešenia a počiatočný a koncový stav riadeného systému, ktorý pre NDS Robot-manipulátor je určený nasledovne:

$$\begin{aligned} x_1(0) = 0 \rightarrow x_1(T) = 0,2 & \quad x_3(0) = 0 \rightarrow x_3(T) = 0,2 \\ x_2(0) = 0 \rightarrow x_2(T) = 0 & \quad x_4(0) = 0 \rightarrow x_4(T) = 0, \end{aligned} \quad (9)$$

Riešte dvojbodovú okrajovú úlohu s pevným pravým koncom pre NDS Robot-manipulátor. Cieľom je nájsť také trajektórie, ktoré prevedú systém z počiatočného stavu $[0, 0, 0, 0]$ do koncového stavu $[0.2, 0, 0.2, 0]$ a zároveň bude minimalizovaná spotreba energie.

Pre riešenie tohto problému využijeme **programové optimálne riadenie** (POR):

1. Vytvoríme Hamiltonian:

$$H = -l + \sum_{i=1}^n g_i(x, p, t) * f_i(x, p, t), \quad (10)$$

kde l je jadro funkcionálu a n je rovné počtu stavových rovníc a g_i je kovektor stavu pre i -tu stavovú rovnicu. V prípade NDS Robot-manipulátor získame Hamiltonián v tvare:

$$H = -\frac{1}{2}(u_1^2(t) + u_2^2(t)) + p_1 * x_2 + p_2 * \left(\frac{m_b}{m_r} x_1 x_4^2 + \frac{K_1}{m_r} u_1\right) + p_3 * x_4 + \quad (11)$$

$$+ p_4 * \left(-\frac{2m_b x_1 x_2 x_4}{I_{23} + m_b x_1^2} + \frac{K_2}{I_{23} + m_b x_1^2} u_2\right)$$

2. Vytvoríme kovektory stavu p_i podľa vzorca:

$$\frac{\partial p_i}{\partial t} = -\frac{\partial H}{\partial x_i}, \quad (12)$$

čím získame štyri diferenciálne rovnice:

$$\dot{p}_1 = p_4 \cdot \left(\frac{2 \cdot m_b \cdot x_2 \cdot x_4}{I_{23} + m_b x_1^2} - \frac{4 \cdot m_b^2 \cdot x_1^2 \cdot x_2 \cdot x_4}{(I_{23} + m_b x_1^2)^2} + \frac{2 \cdot K_2 \cdot m_b \cdot u_2 \cdot x_1}{(I_{23} + m_b x_1^2)^2} \right) - \frac{m_b \cdot p_2 \cdot x_4^2}{m_r}$$

$$\dot{p}_2 = \frac{2 \cdot m_b \cdot p_4 \cdot x_1 \cdot x_4}{I_{23} + m_b x_1^2} - p_1$$

$$\dot{p}_3 = 0$$

$$\dot{p}_4 = \frac{2 \cdot m_b \cdot p_4 \cdot x_1 \cdot x_2}{I_{23} + m_b x_1^2} - p_3 - \frac{2 \cdot m_b \cdot p_2 \cdot x_1 \cdot x_4}{m_r}$$
(13)

kde poznáme iba koncové podmienky $p_i(T) = 0$ pre $i = 1, 2, 3, 4$.

3. Následne získame rovnice zákona riadenia:

$$\frac{\partial H}{\partial u} = 0, \quad (14)$$

ktoré dosadíme do systému diferenciálnych rovníc (8) a (13) za u_1 a u_2 . Takto sme získali súbor diferenciálnych rovníc:

$$\dot{x}(t) = f(x(t), p(t), t) \quad (15)$$

$$\dot{p}(t) = g(x(t), p(t), t)$$

4. Pre získanie počiatočných podmienok kovektoru stavu potrebujeme získať citlivostné rovnice. Tie vypočítame na základe vzorcov:

$$\begin{aligned} \dot{U}_{ij}(t) &= \sum_{k=1}^n \frac{\partial f_i(\bar{x}(t), p(t))}{\partial x_k(t)} * U_{kj}(t) + \sum_{k=1}^n \frac{\partial f_i(\bar{x}(t), p(t))}{\partial p_k(t)} * V_{kj}(t) \\ \dot{V}_{ij}(t) &= \sum_{k=1}^n \frac{\partial g_i(\bar{x}(t), p(t))}{\partial x_k(t)} * U_{kj}(t) + \sum_{k=1}^n \frac{\partial g_i(\bar{x}(t), p(t))}{\partial p_k(t)} * V_{kj}(t) \end{aligned} \quad (16)$$

Počiatkové podmienky pre citlivostné rovnice sú dané a to $U_{ij} = 0$ pre $i = 1, \dots, 4$ a $j = 1, \dots, 4$ a $V_{ij} = 1$ pre $i = j$ inak $V_{ij} = 0$.

5. Po získaní citlivostných rovníc, algoritmicky vyriešime sústavu 40-tich diferenciálnych rovníc, zloženú z rovníc systému (8), rovníc kovektoru stavu (13) a citlivostných rovníc. Následne vykonáme test na presnosť

$$|x_i^k(T) - x_{iT}| < \varepsilon,$$

kde $x_i^k(T)$ je vypočítaný stav v čase T , x_{iT} sú koncové podmienky stavových premenných, kde $i = 1, 2, 3, 4$ a ε je požadovaná presnosť.

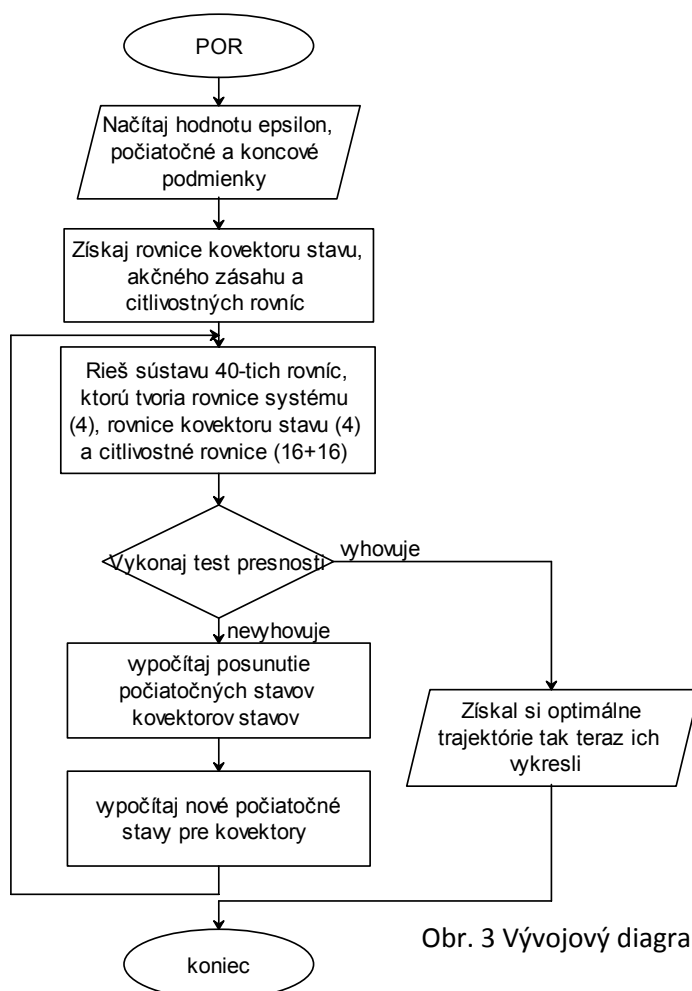
Ak podmienka presnosti nebola splnená, bolo potrebné vypočítať korekcie počiatkových podmienok kovektorov stavu $\Delta p_i(0)$ zo vzorca:

$$\Delta p_i(0) = - \sum_{l=1}^n U_{il}(T) \Delta p_l^{(k+1)} \quad (17)$$

6. kde novú počiatkovú podmienku kovektora p_i sme určili zo vzťahu:

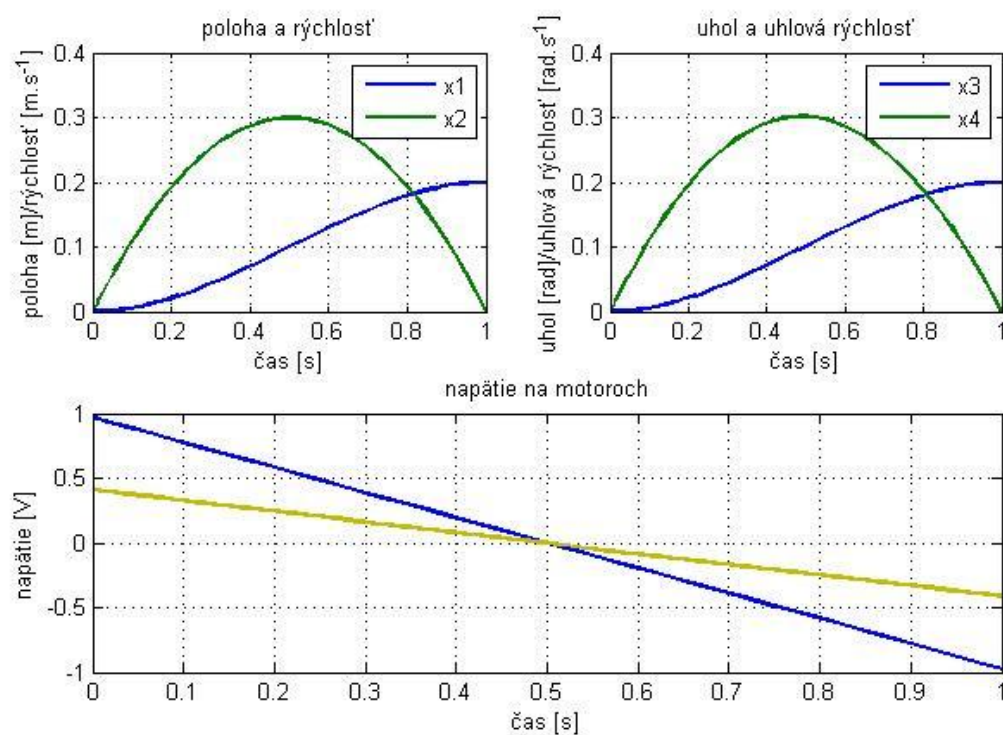
$$p_i^{(k+1)}(0) = p_i^k(0) + \Delta p_i^{(k+1)} \quad (18)$$

Po dosiahnutí požadovanej presnosti algoritmus riešenia končí a trajektórie poslednej iterácie sú optimálne.



Obr. 3 Vývojový diagram POR

Po implementácii do programového prostredia MATLAB a vykonaní experimentov sme získali nasledovné výsledky:



Obr. 4 priebehy jednotlivých stavových veličín

ÚLOHA A3.

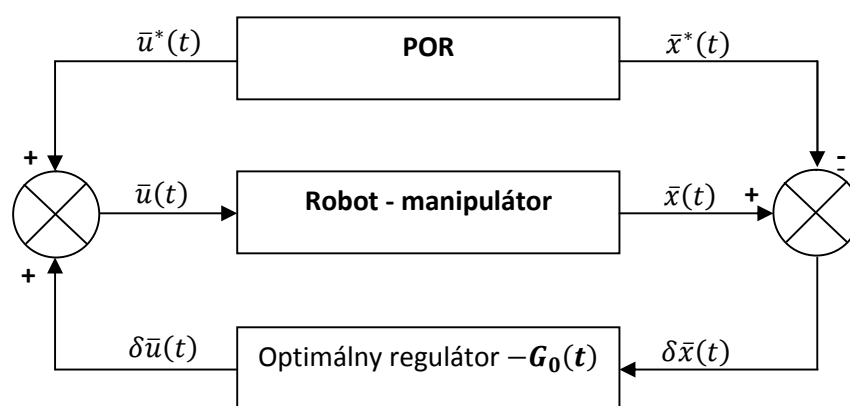
Algoritmicky vykonajte sledovanie optimálnej trajektórie pomocou spätno-väzobného optimálneho riadenia

Návrh deterministického regulátora podľa kvadratického kritéria vychádza z trajektórií získaných algoritmom POR:

- optimálnej trajektórie pohybu $\bar{x}^*(t)$,
- optimálneho časového priebehu riadiacej veličiny $\bar{u}^*(t)$,

ktoré boli vypočítané pre NDS Robot-manipulátor pomocou algoritmu POR.

Spätno-väzobné optimálne riadenie je zobrazené na Obr. 5 :



Obr. 5 SOR časovo variantný prípad

kde

- $\bar{x}(t)$ - skutočný stav,
- $\bar{x}^*(t)$ - požadovaný stav - získaný riešením POR,
- $\delta \bar{x}(t) = \bar{x}(t) - \bar{x}^*(t)$ - vektor odchýlky stavu,
- $\bar{u}(t)$ - skutočný vektor riadenia,
- $\bar{u}^*(t)$ - vektor optimálne riadenia – získaný riešením POR,
- $\delta \bar{u}(t)$ - vektor korekcie riadenia.

Pre výpočet deterministického regulátora je potrebné nájsť taký vektor odchýlky riadenia $\delta \bar{u}(t)$ na intervale $\langle 0, T \rangle$, ktorý minimalizuje kvadratický funkcionál:

$$J_{LQ} = \delta \bar{x}^T(t) F_0 \delta \bar{x}(t) + \int_0^T (\delta \bar{x}^T(t) Q_0 \delta \bar{x}(t) + \delta \bar{u}^T(t) R_0 \delta \bar{u}(t)) dt \quad (19)$$

Navrhovaný regulátor v spätnej väzbe viaže dve veličiny a to:

- vektor odchýlky stavu $\delta \bar{x}(t)$ a
- vektor korekcie riadenia $\delta \bar{u}(t)$

s lineárnou časovo-variantnou spätnou väzbou $\delta \bar{u}(t) = -G_0(t)\delta \bar{x}(t)$, kde $G_0(t)$ je časovo premenlivá matica riadenia, pre ktorú platí:

$$G_0(t) = R_0^{-1} B_0^T(t) K_0(t), \quad (20)$$

kde $K_0(t)$ je symetrická matica veľkosti $n \times n$, ktorá je riešením Riccatiho maticových diferenciálnych rovníc:

$$\dot{K}_0(t) = -K_0(t)A_0(t) - A_0^T(t)K_0(t) - Q_0(t) + K_0(t)B_0(t)R_0^{-1}(t)B_0^T(t)K_0(t) \quad (21)$$

s okrajovou podmienkou $K_0(T) = F_0$, kde $A_0(t)$, $B_0(t)$ sú jakobiany získane linearizáciou NDS Robot-manipulátor. $Q_0(t)$ a $R_0(t)$ sú váhové matice.

Existujú dva spôsoby riešenia SOR:

- časovo - variantný prípad (G_0 – časovo premenlivá) (Obr. 5)
- časovo – invariantný prípad ($\hat{G}_0$ – štruktúra spätnej väzby)

Algoritmus pre riešenie úlohy sledovania optimálnych trajektórií pohybu nelineárneho dynamického systému Robot - manipulátor s využitím princípu linear quadratic (SOR).

1. MODELOVANIE

- Vytvorenie matematického modelu Robot-manipulátor $\bar{x}'(t) = \bar{f}(\bar{x}(t), \bar{u}(t), t)$.
- Riešenie úlohy POR, ktorého výsledkom je získanie optimálnych trajektórií $\bar{x}^*(t)$ a $\bar{u}^*(t)$ na intervale $\langle 0, T \rangle$ s kritériom minima spotreby energie.
- Výber váhových rovníc Q_0, R_0 do kvadratického kritéria.

2. VÝPOČTY OFF-LINE

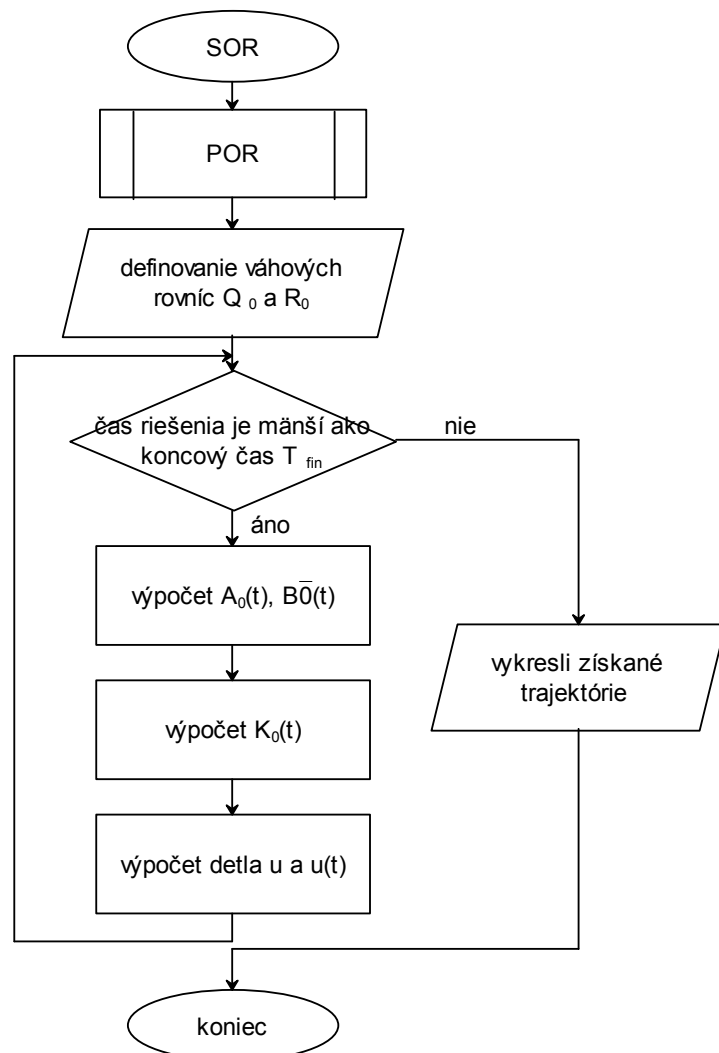
- Z stavových rovníc vytvorených v kroku a) získame jakobiany:
 - $A_0(t), B_0(t)$ - pre časovo – variantný prípad,
 - $\hat{A}_0, \hat{B}_0$ - pre časovo – invariantný prípad.
- Matice $A_0(t)$ a $B_0(t)$, Q_0 a R_0 predstavujú koeficienty Riccatiho diferenciálnej rovnice, ktorú riešime numericky na intervale $\langle 0, T \rangle$, aby sme získali maticu $K_0(t)$ a následne vypočítali $G_0(t)$

3. VÝPOČTY ON-LINE

- Simulácia riadiacej štruktúry (obr. 6) a výpočet regulačnej odchýlky

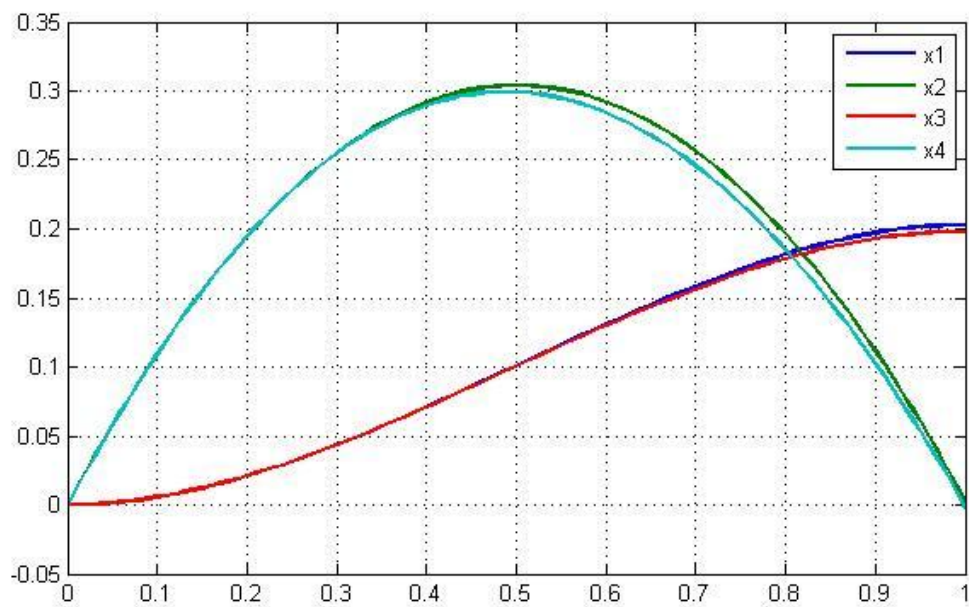
$$\delta \bar{x}(t) = \bar{x}(t) - \bar{x}^*(t).$$
- Výpočet korekcie riadenia :
 - $\delta \bar{u}(t) = -G_0(t)\delta \bar{x}(t)$ pre ON-LINE
 - $\delta \bar{u}(t) = -\hat{G}_0\delta \bar{x}(t)$ pre OFF-LINE
- Výpočet skutočného riadenia $\bar{u}(t) = \bar{u}^*(t) + \delta \bar{u}(t)$

Pre simuláciu boli zvolené matice R_0 a Q_0 ako diagonálne $q_{ii} = 1$ pre $i = 1, \dots, 4$ a $r_{ii} = 0,05$ pre $i = 1, 2$.



Obr. 6 Vývojový diagram návrhu spätno-väzobného regulátora

Riešením v prostredí MATLAB/ Simulink sme dostali nasledujúce priebehy:



Obr. 7 Časovo - variantné riešenie deterministického regulátora

ÚLOHA A4.

Využitím funkcií toolboxu MPC vykonajte sledovanie optimálnej trajektórie nelineárneho dynamického systému Robot – manipulátor

MPC je jeden z mnohých toolboxov programového prostredia MATLAB, ktorý poskytuje funkcie a bloky pre analýzu, dizajn a simuláciu prediktívneho regulátora MPC.

V tejto časti vytvoríme riadenie využitím funkcie *mpc*, ktorej syntax je:

$$\mathbf{MPC} = \mathbf{mpc}(\mathbf{sys}, T_s, \mathbf{P}, \mathbf{M}),$$

kde vstupné parametre sú:

- **systém *sys*** – pre riešenie úlohy využite zlinealizovaný model helikoptéry z úlohy B2.
- **periódu vzorkovania T_s** - perióda vzorkovania pre systém helikoptéry je 0.01 sekundy ($T_s = 0,01$).
- **predikčný horizont $\mathbf{P}$**
- **riadiaci horizont $\mathbf{M}$**

výstupom funkcie *mpc* je štruktúra, ktorá definuje regulátor.

MPC je prediktívny riadiaci algoritmus, ktorý je založený na minimalizácii funkcionálu:

$$J = \sum_{i=1}^N w_{x_i} (r_i - x_i)^2 + \sum_{i=1}^N w_{u_i} \Delta u_i^2, \quad (22)$$

kde

x_i je i – ta regulovaná veličina

r_i je i – ta referenčná trajektória

u_i je i – ta akčná veličina

w_{x_i} je i – ty váhový koeficient pre regulované veličiny x_i

w_{u_i} je i – ty váhový koeficient pre akčné veličiny u_i

Použitím funkcie *mpc* je možné definovať aj ohraničenia na riadenie a výstup.

Pre NDS Robot-manipulátor by sme mohli definovať nasledujúce obmedzenia:

- implementujte aj ohraničenie pre akčný zásah:

$$\Rightarrow u_1 \in \langle -1; 1 \rangle \vee a$$

$$\Rightarrow u_2 \in \langle -0.4; 0.4 \rangle \vee,$$

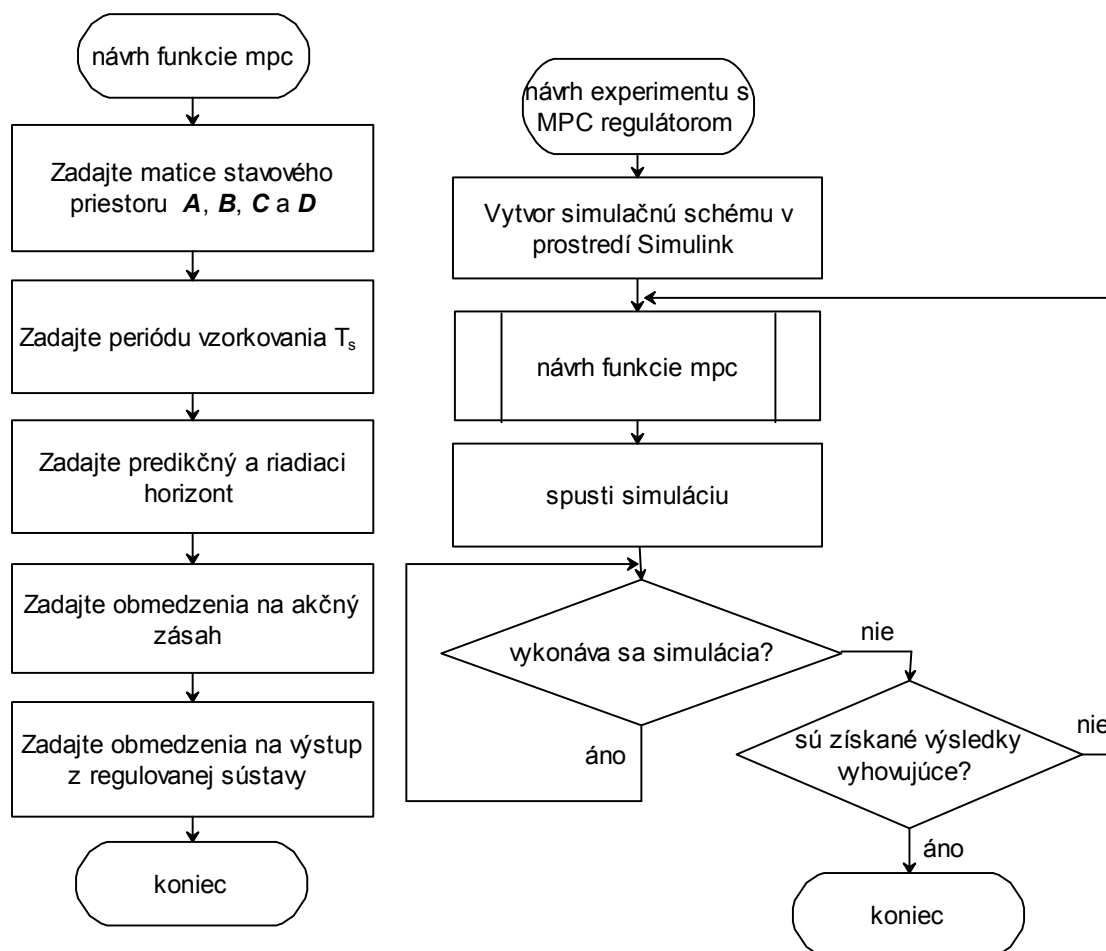
- a ohraničenia výstupu

$$\Rightarrow \text{výsuv } x_1 \in \langle -0.21; 21 \rangle \text{ m a}$$

$$\Rightarrow \text{natočenie } x_3 \in \langle -0.21; 0.21 \rangle \text{ rad}$$

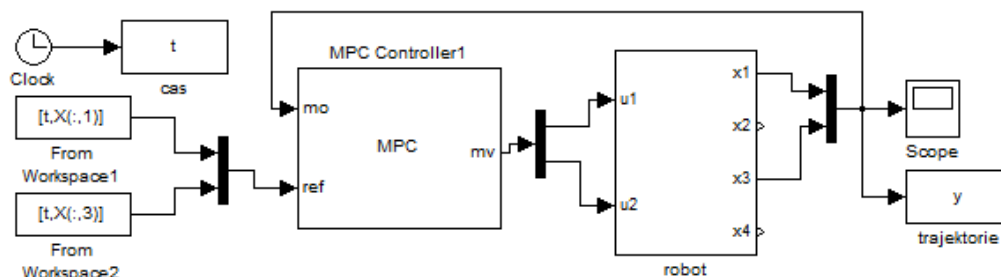
Nezabúdajte však, že riešenie je v radiánoch.

Následne vytvorte simulačnú schému zapojenia s využitím existujúceho bloku MPC regulátora. V nasledujúcom vývojovom diagrame je znázornený postup pre riešenie sledovania trajektórie s využitím toolboxu MPC.



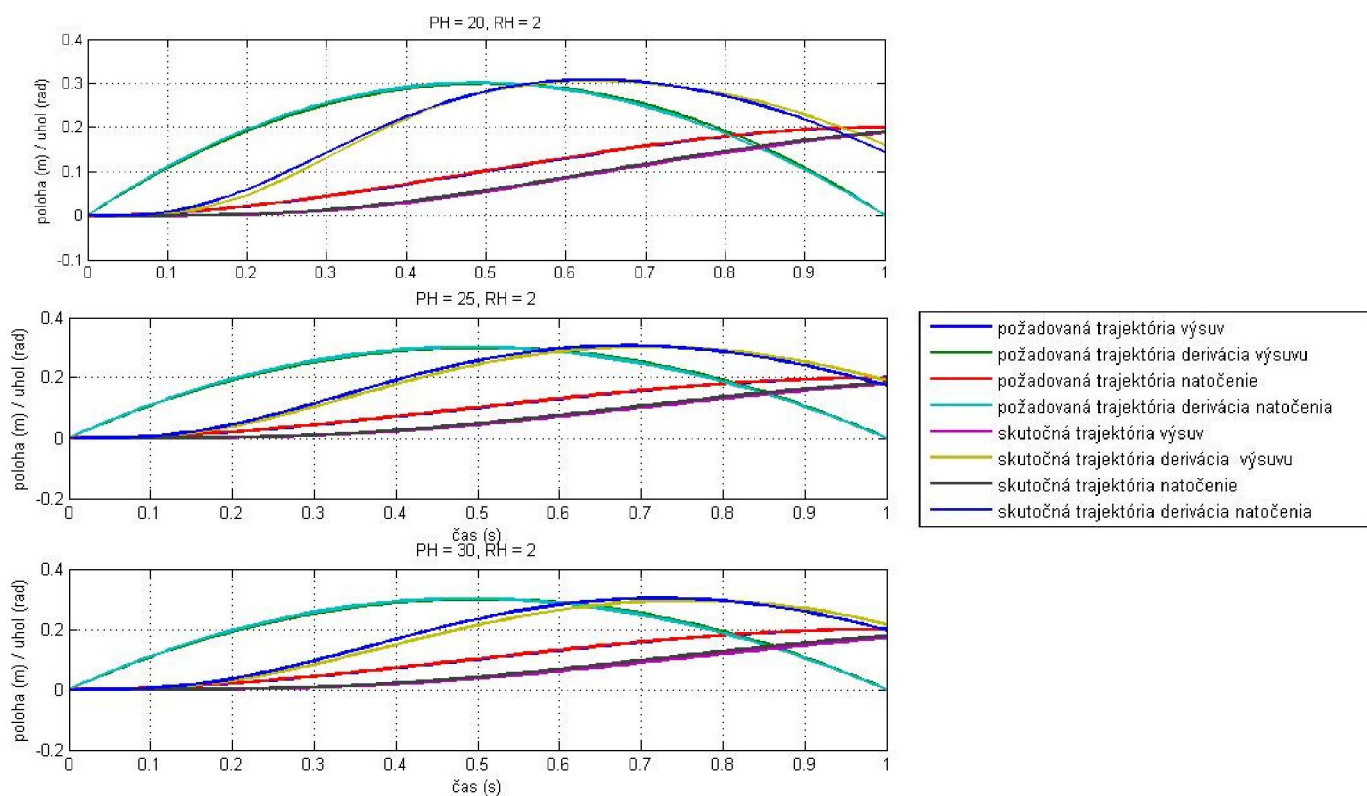
Obr. 8 Aplikácia funkcie *mpc* a simulácia riadiacej štruktúry

V simulačnom prostredí Simulink sme vytvorili simulačnú schému zapojenia:



Obr. 9 Schéma zapojenia MPC regulátora do uzavretého regulačného obvodu

Riešením v prostredí MATLAB/Simulink sme získali nasledujúce priebehy pre rôzny predikčné horizonty:



Obr. 10 Sledovanie optimálnej trajektórie získanej POR s regulátorom MPC

ÚLOHA A5.

Využitím toolboxu MPT vykonajte sledovanie optimálnej trajektórie nelineárneho dynamického systému Robot – manipulátor.

MPT je voľne dostupný toolbox, ktorý je možné prevziať zo stránky: <http://people.ee.ethz.ch/~mpt/3/>. Tento toolbox je potrebné rozbaľiť v adresári, kde sa nachádza programové prostredie MATLAB. Následne je potrebné pri spustení programového prostredia MATLAB prilinkovať odkaz na tento toolbox a aktivovať ho.

MPT toolbox prilinkujeme a aktivujeme nasledovne:

```
addpath(genpath('C:\cesta_k_adresaru_MATLAB\mpt\'))  
mpt init
```

Ak je už aktívny MPT toolbox je potrebné vytvoriť:

SysStruct

Štruktúru **SysStruct (System Structure)**, ktorá popisuje riadený systém.

MPT sa môže zaoberať dvoma typmi systémov:

1. Diskrétné lineárne časovo- invariantné (LTI) systémy
2. Diskrétné po častiach affine (PWA) systémy

Model Robot-manipulátor nie je hybridný systém a preto použijeme prvú možnosť zápisu.

V prípade systému zadaného v tvare LTI bude popis systému obsahovať:

```
sysStruct.A = A  
sysStruct.B = B  
sysStruct.C = C  
sysStruct.D = D
```

Po definovaní systému je potrebné zadať obmedzenia. MPT podporuje rôzne typy systémových obmedzení, ako napríklad:

1. Obmedzenia vstupu:
 $\text{sysStruct.umin} \leq u(k) \leq \text{sysStruct.umax}$
2. Obmedzenia výstupu (voliteľné pre väčšinu prípadov)
 $\text{sysStruct.ymin} \leq y(k) \leq \text{sysStruct.ymax}$
3. Obmedzenia stavu (voliteľné pre väčšinu prípadov)
 $\text{sysStruct.xmin} \leq x(k) \leq \text{sysStruct.xmax}$
4. Obmedzenia na rýchlosť vstupov
 $\text{sysStruct.dumin} \leq u(k) - u(k-1) \leq \text{sysStruct.dumax}$

Druhá možnosť zápisu je pre PWA systémy, ktorými popisujeme hybridné systémy. Dynamické správanie takýchto systémov je zachytené v nasledujúcich rovniciach:

$$x(k+1) = A_i x(k) + B_i u(k) + f_i,$$

$$y(k) = C_i x(k) + D_i u(k) + g_i,$$

kde f_i a g_i sú nelinearity systému. Každá dynamika „ i “ je definovaná v tzv. „*polyhedral partition*“ podľa ohraničenia takzvaného „*guardlines*“:

Popis SysStruct pre PWA systém:

```
sysStruct.A = {A1, ..., An}
sysStruct.B = {B1, ..., Bn}
sysStruct.C = {C1, ..., Cn}
sysStruct.D = {D1, ..., Dn}
sysStruct.f = {f1, ..., fn} [voliteľný]
sysStruct.g = {g1, ..., gn} [voliteľný]
```

n – počet všetkých dynamík

MPT je schopný vysporiadať sa aj s PWA systémami s obmedzeniami aditívnej poruchy:

$$x(k+1) = A_i x(k) + B_i u(k) + f_i + w(k),$$

kde sa predpokladá porucha $w(k)$ pre všetky časové inštancie polytopu W , to indikuje, ktorý systém je subjektom. To nastavíme pomocou `sysStruct.noise = W`, kde W je politopálny objekt príslušného rozmeru.

Pre problém sledovania (`probStruct.tracking=1`) je možné určiť hranice referenčných signálov nasledovne:

```
sysStruct.yrefmax, sysStruct.yrefmin
sysStruct.xrefmax, sysStruct.xrefmin
```

ProbStruct

štruktúru **ProbStruct (Problem Structure)**, ktorá rieši optimalizačný problém využitím MPT.

V tejto štruktúre nastavujeme hodnoty regulátora ako napríklad:

- U – vektor akčných veličín, nad ktorými sa vykonáva optimalizácia
- N – predikčný horizont
- p – lineárna norma, môže byť 1 alebo INF pre 1 až nekonečno normu
- Q – váhová matica stavov
- R – váhová matica akčných veličín
- P_N – váha terminálnych stavov
- $Tset$ – terminalne nastavenie

Návrh MPT regulátora v prostredí MATLAB/Simulink:

1. Vytvoríme štruktúru systému *SysStruct*, v ktorej definujeme systém na základe linearizovaného modelu popísaného v úlohe A1.
2. Zdefinujeme periódu vzorkovania.
3. Vytvoríme štruktúru *ProbStruct*, v ktorej budú definované obmedzenia na akčný zásah:

$$\Rightarrow u_1 \in \langle -1; 1 \rangle \vee \text{a}$$

$$\Rightarrow u_2 \in \langle -0.4; 0.4 \rangle \vee,$$

a obmedzenia na stavy:

$$\Rightarrow \text{Poloha } x_1 \in \langle -0.2; 0.2 \rangle$$

$$\Rightarrow \text{Rýchlosť } x_2 \in \langle 0; 0.3 \rangle$$

$$\Rightarrow \text{Natočenia } x_3 \in \langle -0.2; 0.2 \rangle$$

$$\Rightarrow \text{Uhlová rýchlosť } x_4 \in \langle 0; 0.3 \rangle$$

Ako sledovaný výstup sme definovali polohu x_1 .

4. V štruktúre *ProbStruct* definujeme penalizáciu výstupu, ktorú zadáme maticou:

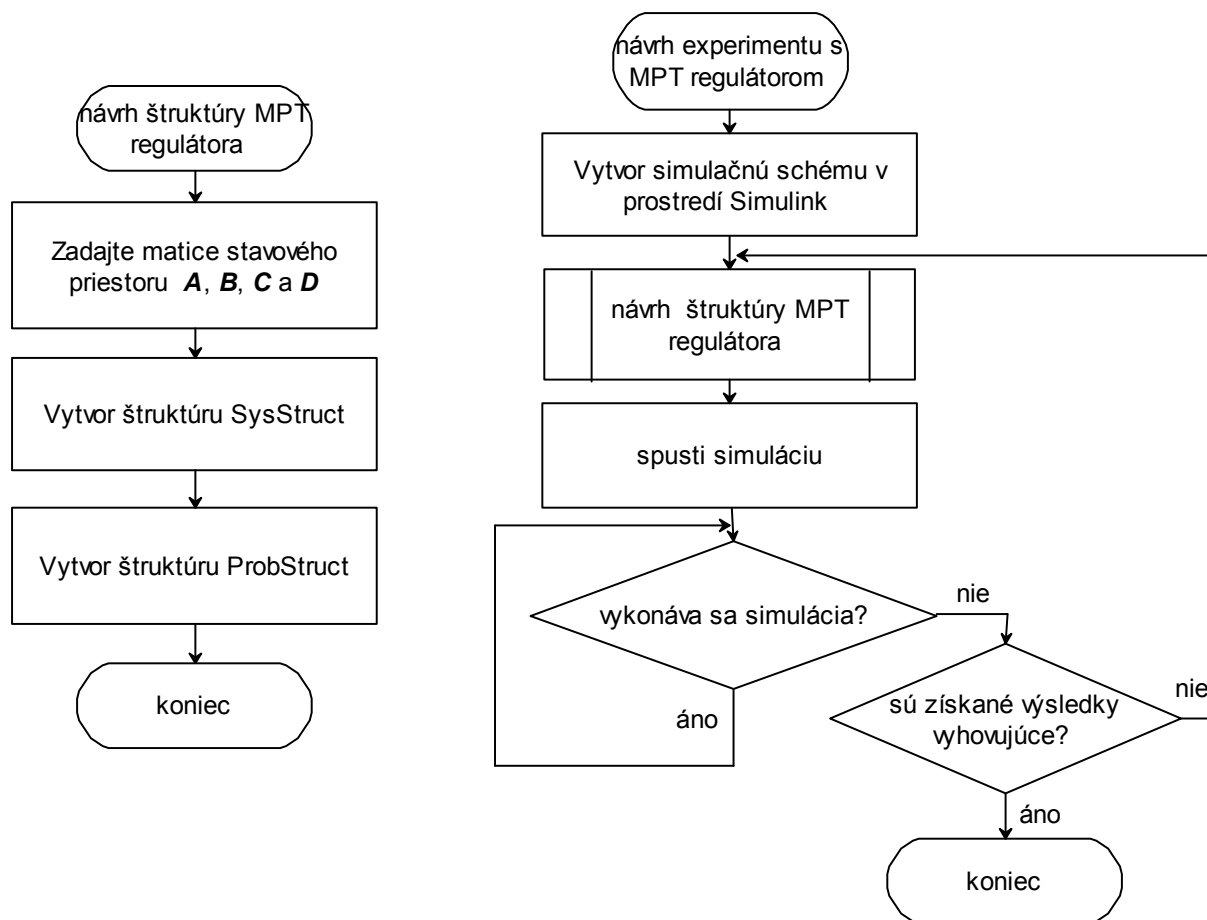
$$Q_y = 5000$$

a penalizáciu vstupov maticou:

$$R = \begin{bmatrix} 0,00001 & 0 \\ 0 & 0,00001 \end{bmatrix}.$$

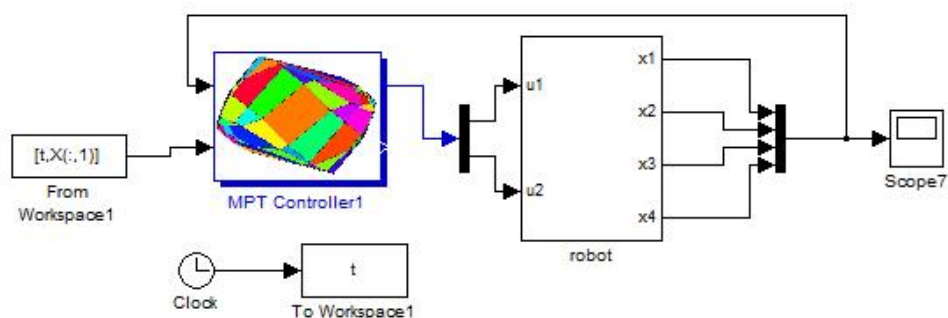
5. Pokúsime sa nájsť najlepšiu kombináciu predikčného a riadiaceho horizontu.

Na nasledujúcom vývojom diagrame je znázornený postup pre algoritmicke riešenie problému sledovania s využitím MPT toolboxu.



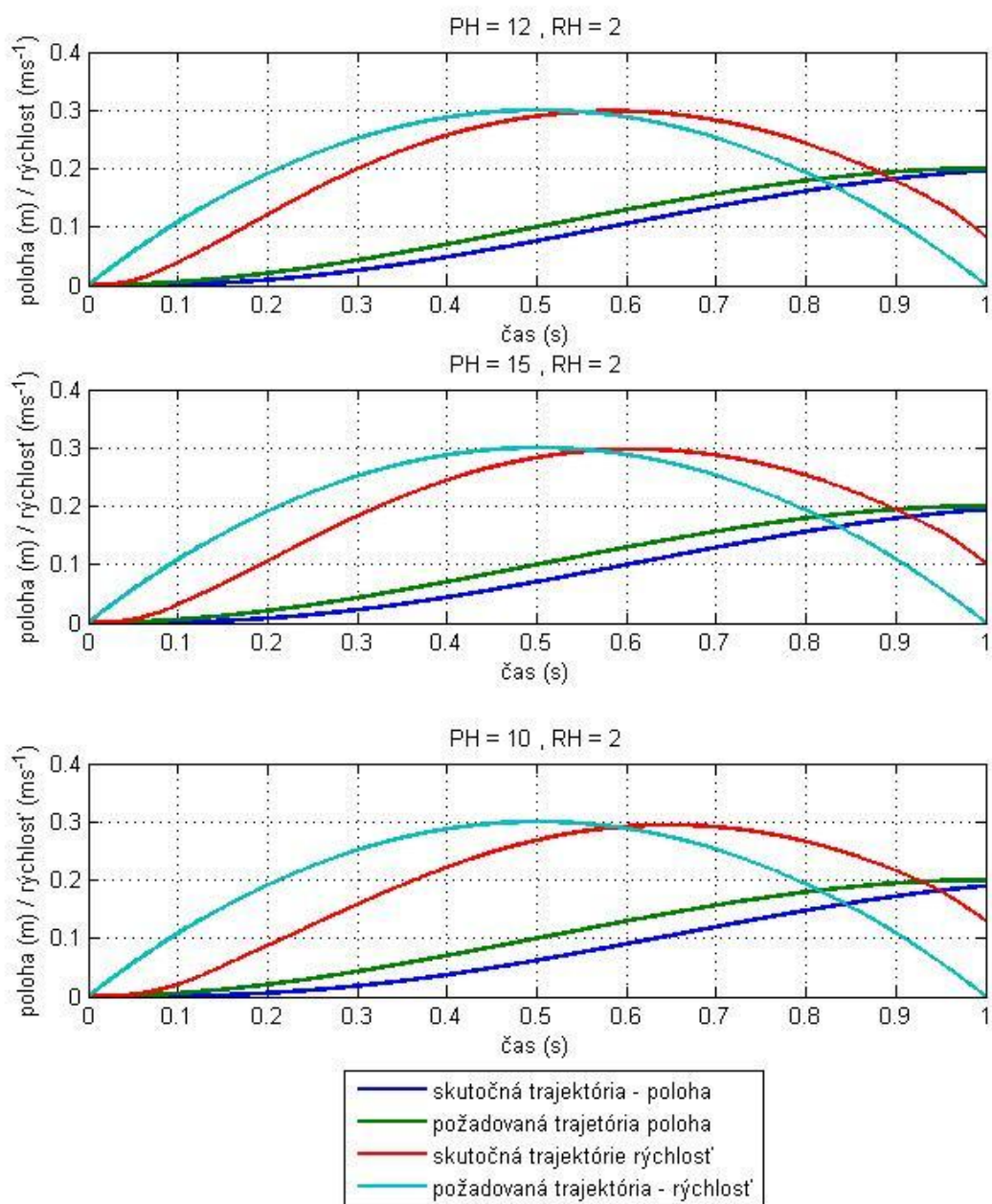
Obr. 11 Aplikácia MPT regulátora a simulácia riadiacej štruktúry

Následne sme vytvorili simulačnú schému v prostredí Simulink:



Obr. 12 Zapojenie MPT regulátora s nelineárnym dynamickým systémom Robot-manipulátor

Riešením v prostredí MATLAB/Simulink sme pri rôznych hodnotách predikčného horizontu dostali nasledujúce priebehy pre polohu:



Obr. 13 Sledovanie optimálnej trajektórie POR s MPT regulátorom