

# HIERARCHICKÝ OPTIMÁLNÍ REGULÁTOR

*Branislav Reháč*

ČVUT FEL, katedra řídicí techniky

## 1 Úvod

Teorie dynamických optimalizačních úloh je již delší dobu dobře rozpracována. Přesto není v praxi příliš často využívána. Jedním z důvodů je, že řešení obecných dynamických optimalizačních problémů vede na nutnost řešit obecné okrajové úlohy pro obyčejné diferenciální rovnice. Řešení těchto úloh je spojeno se značnými nároky, a to jak z hlediska algoritmizace, tak i z hlediska spotřeby času pro provedení výpočtu. Další nevýhodou je, že zákon řízení nelze v obecném případě formulovat ve tvaru zpětné vazby.

Pro řešení okrajových úloh pro obyčejné diferenciální rovnice lze využít metodu konečných prvků podobným způsobem, jakým se tato metoda používá pro řešení parciálních diferenciálních rovnic. Z toho vyplývá možnost využití nástrojů pro řešení parciálních diferenciálních rovnic, jako je např. Femlab, i pro řešení úloh dynamické optimalizace.

I při použití moderní výpočetní techniky nemusí být rychlost výpočtu řešení okrajových úloh vždy dostatečná na to, aby ji bylo možné použít pro sestavení řídicího algoritmu bez dalších úprav. Proto jsme navrhli regulátor hierarchický.

## 2 Hierarchické řízení

Úlohu optimálního řízení v reálném čase jsme rozdělili na dvě podúlohy. První z nich je výpočet predikce optimální trajektorie systému na zadaném horizontu, druhá je výpočet řízení, které zajistí optimální sledování této trajektorie. Tyto podúlohy se řeší v různých úrovních regulátoru, přičemž řešení první úlohy představuje vyšší, řešení druhé úlohy pak nižší úroveň.

Nyní stručně shrneme vlastnosti obou úrovní.

### Vyšší úroveň

- určuje optimální trajektorii systému,
- provádějí se v ní náročné výpočty,
- při výpočtech se bere v úvahu celá struktura systému, nelinearity apod.

### Nižší úroveň

- zajišťuje sledování trajektorie vypočtené ve vyšší úrovni,
- kompenzují se zde vlivy poruchových veličin,
- zákon řízení by měl být co nejjednodušší, nejlépe ve tvaru zpětné vazby,
- pro výpočet řízení systém dekomponovat, využívat jen nejpodstatnějších vazeb mezi subsystémy, nelinearity nahradit linearizací atd.

Při návrhu optimálního řízení je nutné vzít v úvahu i zpoždění vzniklé při provádění výpočtů ve vyšší úrovni. Po tuto dobu nižší úroveň používá trajektorii vypočítanou v předchozím kroku. Dobu, po kterou se používá trajektorie vypočítaná v průběhu jednoho výpočtu vyšší úrovně, označíme  $T_v$ . Tato doba musí být delší než délka nejdelsího výpočtu ve vyšší úrovni.

## 3 Návrh řídicího algoritmu

Je dán systém  $\dot{x} = f(x(t), u(t))$  s počáteční podmínkou  $x(0) = x_0$ .

Nejprve formulujme úlohu, kterou řeší vyšší úroveň: je-li  $T_H > 0$  horizont predikce a  $\tau \geq 0$ , máme nalézt  $x^*$ ,  $u^*$  tak, že

$$\int_{\tau}^{\tau+T_H} J(x^*(t), u^*(t)) dt$$

je minimální, za podmínky

$$\dot{x}^* = f(x^*(t), u^*(t)), \quad x^*(\tau) = x_0.$$

Tento problém označíme jako problém (P) na intervalu  $(\tau, \tau + T_H)$  s počáteční podmínkou  $x_0$ .

Je-li čas  $t = kT_v$ ,  $k$  celé, pak se provedou následující akce:

- vyšší úroveň zjistí hodnoty stavů a začne provádět výpočet,
- do bufferu se nahraje hodnota optimální přímé vazby pro interval  $(kT_v, (k+1)T_v)$ .

Výpočet ve vyšší úrovni se skládá z těchto fází (pro čas  $t = kT_v$ ):

1. spočítá se predikce stavu v čase  $(k+1)T_v$  pomocí hodnot přímé vazby zjištěné v předchozím kroku. Tuto predikci stavu označíme  $\hat{x}((k+1)T_v)$ .
2. Vyřeší se optimalizační problém (P) na intervalu  $((k+1)T_v, (k+2)T_v)$  s počáteční podmínkou  $\hat{x}((k+1)T_v)$ .
3. Z hodnot predikce stavu v intervalu  $((k+1)T_v, (k+2)T_v)$  se spočítá hodnota přímé vazby v tomto intervalu.

V čase  $(k+1)T_v$  se hodnota přímé vazby nahraje do bufferu.

Stručně představíme úlohu, která se ve vyšší úrovni řeší. Dynamická optimalizace se převede na problém minimalizace Lagrangeovy funkce

$$L = \int_{(k+1)T_v}^{(k+2)T_v} J(x(t), u(t)) + \lambda(t)(\dot{x} - f(x(t), u(t)))dt,$$

kde funkce  $\lambda$  je Lagrangeův multiplikátor neboli kostav. Pro optimální řízení platí, že splňuje následující okrajovou úlohu:

$$\dot{x} = f(x(t), u(t)), \tag{1}$$

$$\dot{\lambda} = D_x J(x(t), u(t)) + D_x f(x(t), u(t))\lambda(t) \tag{2}$$

$$0 = D_u J(x(t), u(t)) + D_u f(x(t), u(t))\lambda(t). \tag{3}$$

Okrajové podmínky pro stavy jsou dány jejich hodnotami v bodě  $(k+1)T_v$ , kostavy splňují podmínku  $\lambda((k+2)T_v) = 0$ . Více viz [1].

Regulátory v nižší úrovni by měly být co nejjednodušší, např. navržené pomocí LQ teorie. Výhodou těchto regulátorů je konstantní zesílení ve zpětné vazbě, vliv vyšší úrovně se projevuje jen prostřednictvím přímé vazby.

Předpokládáme, že stavová rovnice je lineární a má tvar

$$\dot{x}(t) = f(x(t), u(t)) = Ax(t) + Bu(t).$$

a dále, že kritérium lze psát ve tvaru

$$J(x, u) = x^T Qx + u^T Ru + \Pi(x, u),$$

kde funkce  $\Pi$  obsahuje členy řádu vyššího než 2.

Předpokládejme, že systém je složen z  $N$  subsystémů, které spolu "slabě" interagují. Pak existují matice  $A_1, \dots, A_N, A_{vazba}, B_1, \dots, B_N, B_{vazba}, Q_1, \dots, Q_N$  a  $R_1, \dots, R_N$  takové, že lze matici  $A$  zapsat takto:

$$A = \begin{pmatrix} A_1 & 0 & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & A_N \end{pmatrix} + A_{vazba},$$

předpokládáme, že matice  $B, Q, R$  lze zapsat analogicky. Přitom vliv vazeb mezi systémy je malý a při definici nižší úrovně ho nebudeme uvažovat, stejně jako vliv členů vyššího řádu v kritériu. Pak se úloha nalezení regulátorů nižší úrovně redukuje na problém nalézt  $N$  regulátorů pro systém

$$\dot{x}_i(t) = A_i x_i(t) + B_i u_i(t)$$

takových, že je minimalizováno kritérium

$$\int_0^{+\infty} x_i^T(t)Q_i x_i(t) + u_i^T(t)R_i u_i(t) dt.$$

To je standardní úloha lineárně-kvadratického řízení (LQ řízení). Předpokládáme, že matice  $A_i, B_i, Q_i, R_i$  splňují podmínky zaručující existenci řešení tohoto problému. Více viz [2]. O otázkách decentralizovaného řízení systémů pojednává např. [3].

Z této teorie plyne, že optimální řízení  $i$ -tého subsystému je dáno vztahem

$$u_i = u_i^{zv}(t) - R_i^{-1} B_i^T p_i(t),$$

kde:

- $u_i^{zv}(t) = -R_i^{-1} B_i^T P_i x_i(t)$  je zpětná vazba (ta se nemění),
- $-R_i^{-1} B_i^T p_i(t)$  je přímá vazba. Funkce  $p$  je definována vztahem

$$\dot{p}_i(t) = -(A_i - P_i B_i^T R_i^{-1} B_i) p_i(t) + Q_i r_i(t).$$

Matice  $P_i$  je řešení Riccatiho rovnice:

$$A_i^T P_i + P_i A_i - P_i B_i R_i^{-1} B_i^T P_i - Q_i = 0$$

a  $r_i$  je  $i$ -tá reference. Ta je spočítána ve vyšší úrovni, jako příslušná část predikce optimální trajektorie.

Pokud by původní systém byl nelineární, pak je pro výpočet regulátorů nižší úrovně nutné provést linearizaci. Pokud by byla provedena jen jednou, je možné provést předchozí postup beze změn, s to výjimkou, že k výslednému stavu, resp. řízení je nutno přičíst stav, resp. řízení v bodě, kde se počítá linearizace. Jestliže by se bod, okolo něhož se systém linearizuje, měnil (na základě informací o predikci ve vyšší úrovni), pak je nutné kromě toho přepočítávat i zesílení matice zpětné vazby.

## 4 Implementace vyšší úrovně

Pro implementaci vyšší úrovně regulátoru byl adaptován kód vytvořený GUI systému Femlab 2.1. Pro řešení úlohy byla zvolena obecná forma pro soustavu šesti jednodimenzionálních rovnic na intervalu  $(0, T_H)$ , použit byl nelineární řešič.

Zde uvádíme systém rovnic, který byl řešen ve Femlabu:

$$\dot{x}_1 = x_1 + 0.01x_2 + u_1 \quad (4)$$

$$\dot{x}_2 = 0,01x_1 + 2x_2 + u_2 \quad (5)$$

$$\dot{\lambda}_1 = q_1(x_1 - \bar{x}_1) + \lambda_1 + 0.01\lambda_2 \quad (6)$$

$$\dot{\lambda}_2 = q_2(x_2 - \bar{x}_2) + 0.01\lambda_1 + 2\lambda_2 \quad (7)$$

$$0 = r_1 u_1 + \lambda_1 \quad (8)$$

$$0 = r_2 u_2 + \lambda_2 \quad (9)$$

pro neznámé  $x_1, x_2, \lambda_1, \lambda_2, u_1, u_2$ . Do seznamu proměnných byly zařazeny  $q_1, q_2, r_1, r_2, xx10, xx20$ . Dirichletovy okrajové podmínky byly

$$x_1(0) = xx10, x_2(0) = xx20, \lambda_1(T_H) = 0, \lambda_2(T_H) = 0.$$

Dirichletovy podmínky pro stavy v čase  $T_H$  a kostavy v čase 0 byly formulovány tak, aby byly vždy splněné, tj. např.

$$x_1(T_H) - x_1(0) = 0.$$

Neumannova podmínka byla také zadána tak, aby byla vždy splněna a neměla tedy vliv na řešení. Tím jsme dosáhli toho, že úloha byla korektně zadaná. Obě tyto podmínky je nutné zadat, neboť to vyžaduje použitá verze systému Femlab (viz [4]).

Funkce  $\bar{x}_1, \bar{x}_2$  provádějí výpočet žádané hodnoty. Musí být implementovány v souladu s požadavky na syntaxi funkcí počítajících hodnoty koeficientů rovnic, jak ji vyžaduje Femlab.

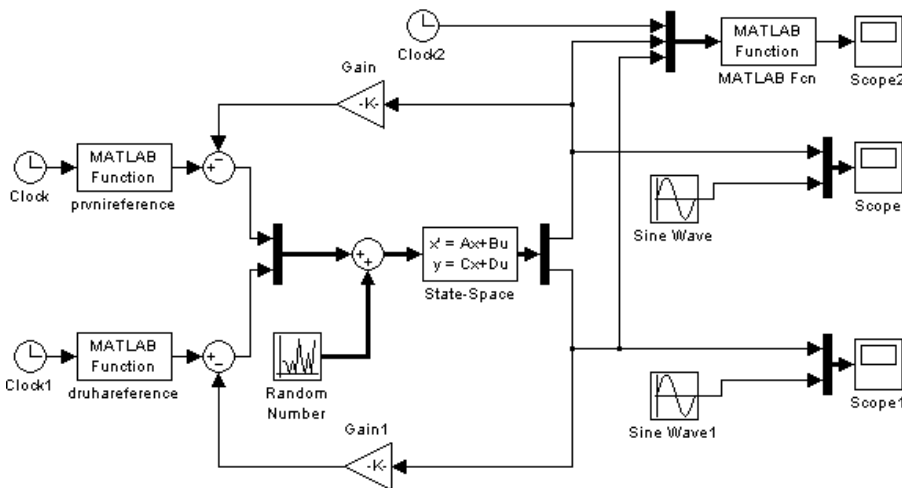
Vytvořený m-file byl dále upraven, aby obsahoval všechny fáze popsané ve třetím odstavci:

- byla z něj vytvořena funkce `predfun2` s argumenty (`cas`, `qq1`, `qq2`, `rr1`, `rr2`, `x1cas`, `x2cas`). Argumenty `qq1`, `qq2`, `rr1`, `rr2` odpovídají proměnným  $q_1, q_2, r_1, r_2$ . Argument `cas` předává funkci hodnotu času, kdy dochází k volání funkce `predfun2`, v argumentech `x1cas`, `x2cas` se předávají hodnoty stavů v tomto čase.
- interval, na kterém se řeší okrajový problém, byl změněn na (`cas`, `cas+TH`)
- před výpočtem okrajového problému se vyřeší stavové rovnice na intervalu (`cas`, `cas+1`). Tím získáme predikci stavů v čase `cas+1`. Tu použijeme jako okrajovou podmínku pro stavu.
- následuje řešení okrajové úlohy, tj. část, kterou vygenerovalo GUI Femlabu
- struktura `fem` se uloží do souboru. Dále se vykoná spočítá hodnota přímé vazby na intervalu (`cas`, `cas+TH`) a opět se uloží do souboru.

## 5 Simulační schéma

Pro modelování systému byl použit Simulink. Pro správné odsimulování by bylo potřebné provádět výpočty v obou úrovních v odlišných procesech, což jsme neprováděli. Výpočet se proto při každém volání vyšší úrovně zastaví.

Schéma je na obr. (1). Zesilovače `Gain` a `Gain1`, sumátory a systém tvoří nižší úroveň. Funkce



Obrázek 1: Simulační schéma

`prvnireference` a `druhareference` vybírají správnou hodnotu přímé vazby z hodnot spočítaných ve vyšší úrovni. Vazba z nižší úrovně do vyšší je realizována funkcí `MATLAB Fcn`, která vždy po uplynutí doby  $T_v = 1$  aktivuje výpočet ve vyšší úrovni, tj. funkci `predfun1` a předá jí potřebné parametry.

## 6 Výsledky simulací

Pro simulace byl použit systém (4) s kvadratickým kritériem. Váhové matice v kritériu byly zvoleny takto:

$$Q = \begin{pmatrix} q_1 & 0 \\ 0 & q_2 \end{pmatrix}, \quad R = \begin{pmatrix} r_1 & 0 \\ 0 & r_2 \end{pmatrix},$$

kde  $q_1 = q_2 = 10$ ,  $r_1 = r_2 = 0.2$ . Regulátory v nižší úrovni mají zesílení, které je optimální pro řízení systému  $\dot{x}_1 = x_1 + u_1$  s kvadratickým kritériem s váhovými maticemi  $q_1$ ,  $r_1$ , resp.  $\dot{x}_2 = x_2 + u_2$ , v kritériu jsou váhové matice  $q_2$ ,  $r_2$ . To znamená, že vazby mezi systémy jsou zanedbány.

Stav  $x_1$  by měl sledovat referenci  $\bar{x}_1 = \sin 4t$  a stav  $x_2$  sleduje referenci  $\bar{x}_2 = \cos 5t$ .

Dále byly provedeny simulace stejného systému s tím, že kritérium bylo doplněno o člen

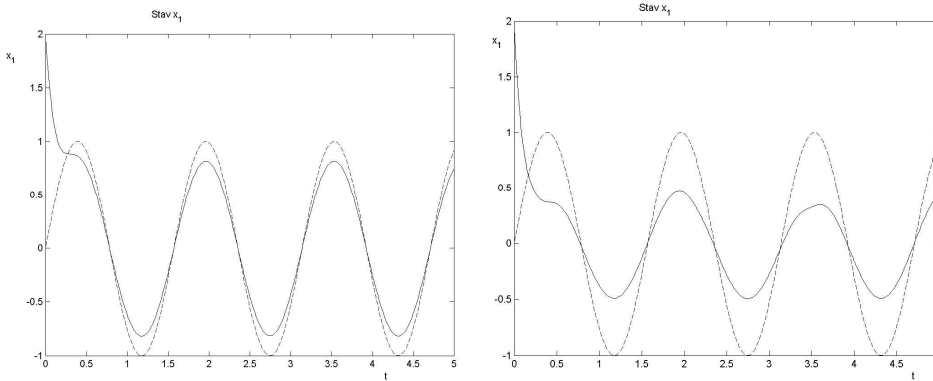
$$b(x_1, x_2) = \begin{cases} 0.005 \left( \frac{1}{((x_1-0.6)^2 + x_2^2 + 0.05)} - 5 \right)^4 & \text{je-li } \frac{1}{((x_1-0.6)^2 + x_2^2 + 0.05)} > 5 \\ 0 & \text{jinak.} \end{cases}$$

Tato bariérová funkce penalizuje trajektorie, které se příliš blíží bodu  $x_p = (0.6, 0)$  ve stavovém prostoru. Rovnice pro adjungované stavy se po přidání bariérové funkce změny takto:

$$\dot{\lambda}_1 = q_1(x_1 - \bar{x}_1) + \lambda_1 + 0.01\lambda_2 + \frac{\partial b(x_1, x_2)}{\partial x_1} \quad (10)$$

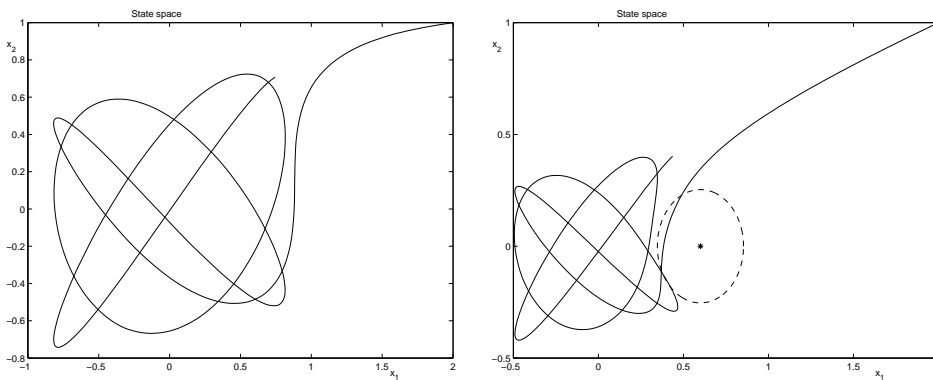
$$\dot{\lambda}_2 = q_2(x_2 - \bar{x}_2) + 0.01\lambda_1 + 2\lambda_2 + \frac{\partial b(x_1, x_2)}{\partial x_2} \quad (11)$$

Vývoj stavů systému v čase je zobrazen na obr 2 bez bariérové funkce (vlevo) a s přidanou bariérovou funkcí (vpravo). Vývoj stavů ve stavovém prostoru je znázorněn na obr. 3, z něhož je zřejmý účinek



Obrázek 2: Stav  $x_1$ , plná čára - skutečný stav, čárkovaná čára - reference

penalizace stavů v okolí bodu  $x_p$ . Čárkovanou kružnicí jsou označeny body, na nichž bariérová funkce nabývá hodnoty 1.



Obrázek 3: Stavový prostor

## 7 Poznámky k výpočtům

Je-li zvolena příliš velká váha na odchylku stavů od reference ( $q_i = 1000$ ), začnou se objevovat problémy se splněním okrajových podmínek při řešení okrajové úlohy - tyto podmínky nejsou splněny ani po velkém počtu iterací. Použití takto nepřesné aproximace výrazně zhorší kvalitu regulace.

Další problémy se vyskytnou, je-li bariérová funkce příliš strmá. Pak výpočet obvykle končí chybovým hlášením.

Je nutné upozornit i na značnou časovou náročnost výpočtu, a to zejména při výpočtu optimální přímé vazby pro nižší úroveň. Zde se řeší diferenciální rovnice, v jejíž pravé straně se vyskytuje optimální trajektorie  $x$  vypočtená ve vyšší úrovni. Z toho vyplývá nutnost volání funkce `postinterp` při každém vypočítání pravé strany.

## 8 Závěr

Odsimulovali jsme systém s víceúrovňovým regulátorem, v němž se řeší okrajová úloha pomocí systému Femlab.

## Reference

- [1] Bryson, A.E., Ho, Y., *Applied Optimal Control*, J. Wiley, New York, 1975
- [2] Havlena, V., Štecha, J, *Moderní teorie řízení*, Vyd. ČVUT, Praha, 1996
- [3] Trave, L., Titli, A., Tarras, A., *Large Scale Systems: Decentralization, Structure Constraints and Fixed Modes*, Springer, Berlin 1989
- [4] *Femlab Reference Manual v. 2.0*, COMSOL AB, 2000

## Kontakt

Branislav Reháček  
ČVUT FEL, katedra řídicí techniky  
Karlovo nám. 13  
121 35 Praha 2  
  
tel. 02 / 2435 7336  
rehakb@control.felk.cvut.cz