

## MATICOVÉ OPERACE V SIMULINKU VERZE 4

*Daniel Honc, František Dušek*

Katedra řízení procesů a výpočetní techniky, FCHT, Univerzita Pardubice

### Abstrakt

Verze 4 SIMULINKu přinesla principiální změnu – možnost pracovat se signály typu matice. Od tohoto okamžiku lze na úrovni modelu používat maticové operace, což bylo dříve možné pouze na úrovni S-funkcí. Tímto se možnosti zápisu modelu v SIMULINKu přiblížily možnostem vlastního MATLABu. Zvláště užitečná je tato nová vlastnost např. pro oblast řízení, kde je nyní možné vytvořit kompletní diskrétní regulátor. Ukázka modelu SIMULINKu využívajícího nové možnosti pro realizaci adaptivního řízení je námětem tohoto příspěvku. Je ukázána průběžná identifikace modelu řízené soustavy, průběžný odhad stavu soustavy a LQ regulátor.

### Úvod

Při návrhu algoritmů řízení je velmi přínosná možnost experimentovat na fyzikálním modelu. Pouze numerická simulace chování uzavřené regulační smyčky je sice užitečná, ale případné problémy se projeví často až v reálu. Prvním krokem k realitě je řízení na fyzikálním modelu v reálném čase. Běžně se pro tyto účely používá MATLAB ve spojení s Real Time Toolboxem nebo Data Acquisition Toolboxem či jiným specializovaným toolboxem např. pro podporu signálových procesorů.

Zapsání příslušných algoritmů v MATLABu není až tak složitá záležitost. Pracnější je jejich ověření pomocí simulace uzavřené regulační smyčky, jejíž součástí je např. i nelineární model řízené soustavy. Právě pro zápis nelineárních dynamických modelů a simulace chování za různých podmínek je výhodné použít SIMULINK. Proč tedy úlohu neřešit rovnou v SIMULINKu? Problémem byla realizace složitějších diskrétních algoritmů obsahujících maticové operace, cykly a iterace rozložené v čase. Jediný rozumný způsob řešení bylo napsat vlastní S-funkci, což není triviální záležitost. Verze MATLABu 6.1 přinesla nové možnosti například v souvislosti s připojením externích zařízení (podpora sériové linky [Dušek 02]) a také rozšíření SIMULINKu verze 4 o podporu maticových signálů a operací.

V příspěvku je na konkrétním příkladu ukázáno adaptivní řízení elektrické soustavy. Spojení se soustavou je realizováno pomocí akviziční karty a Real Time Toolboxu. Parametry modelu soustavy ve tvaru diferenční rovnice jsou průběžně identifikovány rekurentní metodou nejmenších čtverců. Diferenční rovnice je převedena na stavový model, který je využíván jednak pro průběžnou identifikaci stavu Kalmánovým filtrem a jednak pro výpočet LQ regulátoru (aktualizovaného v každém kroku) s konečným horizontem řízení a penalizací akčního zásahu. Všechny použité bloky jsou napsány obecně tj. jsou parametrizovány pro volbu řádu soustavy, počáteční hodnoty pracovních matic a bez nutnosti volit explicitně interval vzorkování.

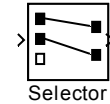
### Nové možnosti SIMULINKu verze 4

Signály mohou být jednorozměrové nebo dvourozměrové. Jednorozměrový 1-D signál je jednorozměrové pole vytvořené např. blokem *Mux*. Dvourozměrový 2-D signál je dvourozměrové pole - matice. Jednoprvkové pole je označováno jako skalár. Rozlišuje se řádkový vektor jako 2-D pole s jedním řádkem a sloupcový vektor je 2-D pole s jedním sloupcem.

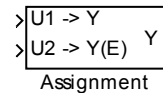
Bloky Simulinku se liší v rozměrech signálů, které mohou akceptovat nebo produkovat během simulace. Některé bloky podporují signály libovolných rozměrů, některé pouze skalární nebo 1-D signály. Detailní popis všech bloků je uveden v nápovědě Simulinku (v části Block Reference).

## Přehled použitých bloků a stručný popis jejich funkce

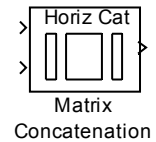
*Selector* - vybere prvky z vstupního vektorového nebo maticového signálu.



*Assignment* - přiřadí hodnoty určeným prvkům signálu.



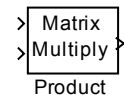
*Matrix Concatenation* - spojí vstupy horizontálně nebo vertikálně.



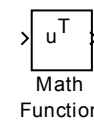
*Reshape* - změní rozměr signálu.



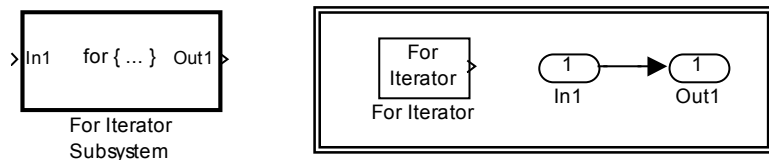
*Product* - provede násobení, dělení, maticové násobení nebo inverzi vstupů.



*Math Function* - provede matematickou funkci (např. transpozici).



*For Iterator Subsystem* - subsystem, který se cyklicky vykoná během jednoho simulačního kroku.



## Použité algoritmy

V této části jsou shrnuty použité algoritmy realizované jako obecné (parametrizované) knihovní bloky. Matematický maticový zápis je velmi stručný, popis v textu je zaměřen spíše na problémy či zajímavosti při převodu matematického zápisu do modelu SIMULINKu. Bližší popis algoritmů je ve většině běžných publikací zabývajících se klasickým diskretním řízením. Konkrétně použité verze algoritmů byly převzaty z publikace [Dušek 99], pro kterou byla jedním z hlavních zdrojů v této oblasti publikace [Havlena 96].

### Rekurentní metoda nejmenších čtverců s exponenciálním zapomínáním

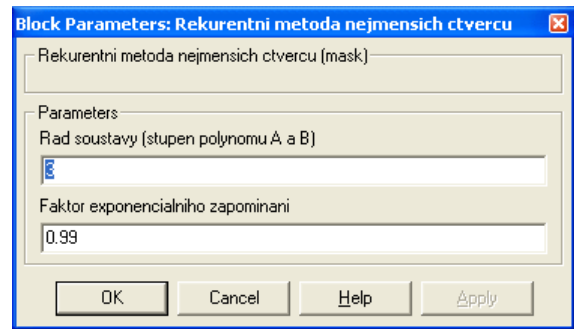
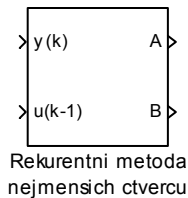
Neznámé parametry soustavy jsou odhadovány rekurentní metodou nejmenších čtverců s exponenciálním zapomínáním. Jsou identifikovány parametry diferenční rovnice  $\mathbf{A}\mathbf{y} = \mathbf{B}\mathbf{u}$ , kde  $\mathbf{A}$  a  $\mathbf{B}$  jsou vektory parametrů soustavy,  $\mathbf{y}$  a  $\mathbf{u}$  jsou vektory tvořené výstupy a vstupy soustavy v daných intervalech vzorkování.

$$\mathbf{A} = [1 \ a_1 \ a_2 \ \dots \ a_N], \quad \mathbf{B} = [0 \ b_1 \ b_2 \ \dots \ b_N], \quad \mathbf{y} = [y(k) \ y(k-1) \ \dots \ y(k-N)]', \quad \mathbf{u} = [u(k) \ u(k-1) \ \dots \ u(k-N)]'$$

V každém intervalu vzorkování je třeba provést následující posloupnost výpočtů:

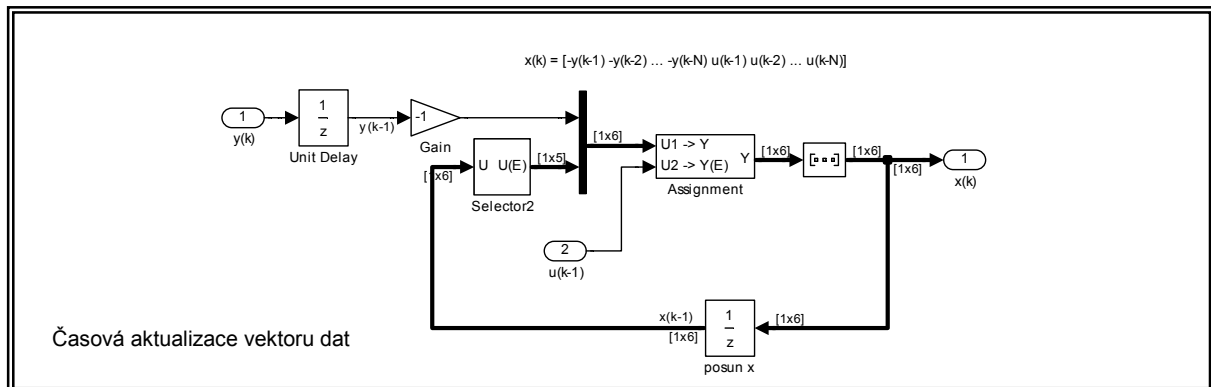
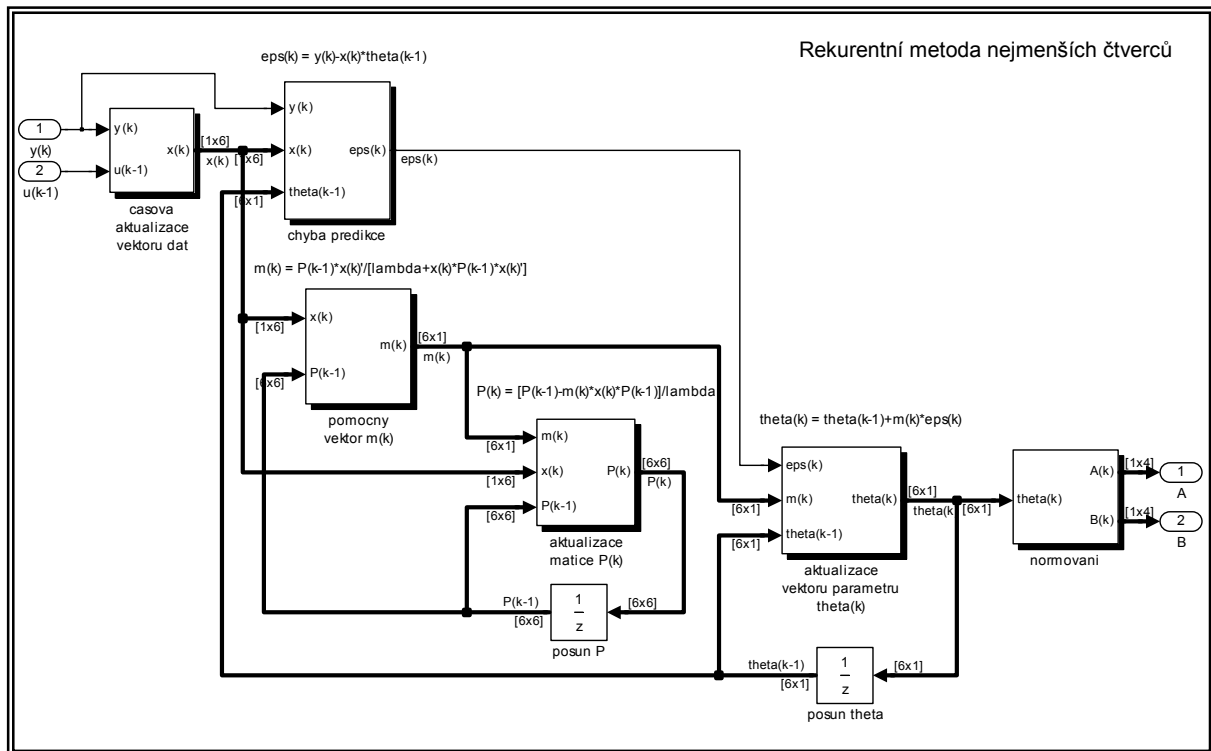
- |   |   |
|---|---|
| 1. časová aktualizace vektoru dat $\mathbf{x}$    | $\mathbf{x}(k) = [-y(k-1) \ -y(k-2) \ \dots \ -y(k-N) \ u(k-1) \ u(k-2) \ \dots \ u(k-N)]$  |
| 2. výpočet chyby predikce $\mathbf{eps}$          | $\mathbf{eps}(k) = y(k) - \mathbf{x}(k)\mathbf{theta}(k-1)$   |
| 3. výpočet pomocného vektoru $\mathbf{m}$         | $\mathbf{m}(k) = \mathbf{P}(k-1)\mathbf{x}(k) / [\lambda + \mathbf{x}(k)\mathbf{P}(k-1)\mathbf{x}(k)']$                                       |
| 4. aktualizace matice $\mathbf{P}$                | $\mathbf{P}(k) = [\mathbf{P}(k-1) - \mathbf{m}(k)\mathbf{x}(k)\mathbf{P}(k-1)] / \lambda$   |
| 5. aktualizace vektoru parametrů $\mathbf{theta}$ | $\mathbf{theta}(k) = \mathbf{theta}(k-1) + \mathbf{m}(k)\mathbf{eps}(k)$  |
| 6. normování vektorů $\mathbf{A}$ a $\mathbf{B}$  | $\mathbf{A} = [1 \ \mathbf{theta}(1) \ \dots \ \mathbf{theta}(N)], \quad \mathbf{B} = [0 \ \mathbf{theta}(N+1) \ \dots \ \mathbf{theta}(2N)]$ |

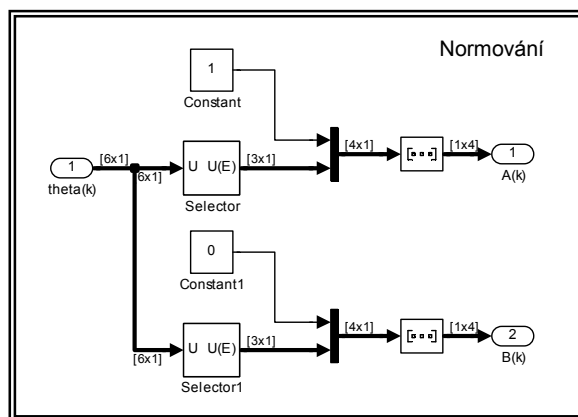
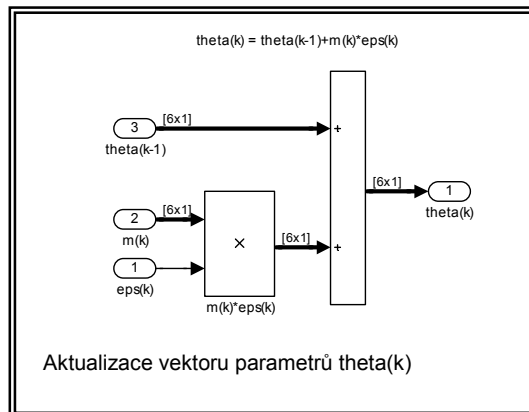
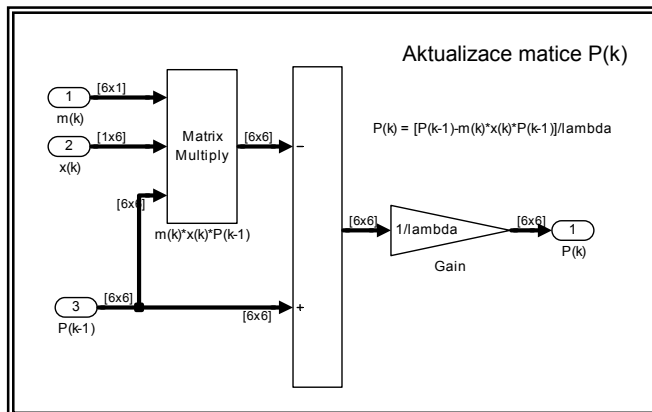
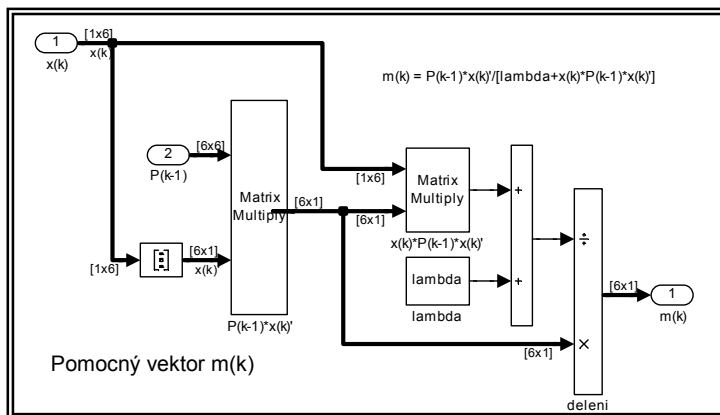
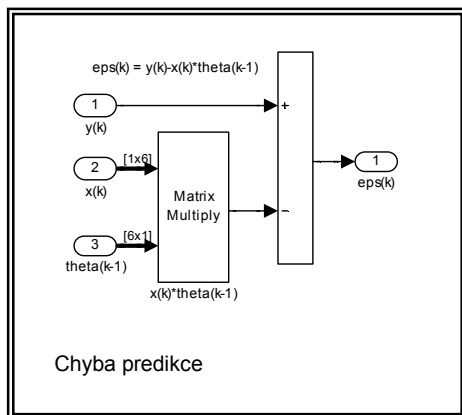
Vstupy zamaskovaného bloku Simulinku jsou regulovaná veličina  $y(k)$  a akční veličina  $u(k-1)$ . Výstupem jsou polynomy **A** a **B** s odhadovanými parametry. Parametry bloku jsou řád soustavy  $N$  (stupeň polynomu **A** a **B**) a faktor exponenciálního zapominání  $\lambda$ .



Pro dílčí operace 1-6 jsou vytvořeny samostatné subsystémy. Standardní bloky „posun P“ a „posun theta“ mají jako parametr počáteční hodnoty matice **P** a vektoru **theta**. Počáteční hodnoty vektoru dat  $x$  jsou zadávány v bloku „posun x“. Interval vzorkování celého subsystému je určen intervalem vzorkování vstupů  $u$  a  $y$  (intervalem vzorkování nadřazeného systému).

Časová aktualizace vektoru dat  $x$  je zajištěna následujícím způsobem. Ze zpožděného vektoru  $x(k-1)$  je v bloku *Selektor* odstraněn poslední prvek. V Bloku *Mux* je na první místo přidána hodnota  $-y(k-1)$ . Pomocí bloku *Assignment* je na pozici  $N+1$  přiřazena hodnota  $u(k-1)$ . Blok *Reshape* tvoří řádkový vektor  $x(k)$ . Ostatní bloky jsou přímočarým přepisem výpočetních vztahů.





## Pozorovatel stavu (stochastická soustava a kvadratické kritérium – Kalmánův filtr)

Stav soustavy je odhadován pomocí Kalmánova filtru. Model soustavy je ve stavovém popisu.  $\mathbf{x}$  je vektor stavů,  $\mathbf{u}$  je vektor vstupů a  $\mathbf{y}$  vektor výstupů soustavy.  $\mathbf{A}$ ,  $\mathbf{B}$  a  $\mathbf{C}$  jsou matice stavového popisu. U modelu je uvažován šum na vstupu a na výstupu soustavy.

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{B}_{in}\mathbf{e}_{in}(k)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{e}_{ou}(k)$$

$\mathbf{B}_{in}$  matice váhy šumu na složky stavu  
 $\mathbf{e}_{in}(k)$  vektor šumu na vstupu  
 $\mathbf{e}_{ou}(k)$  vektor šumu na výstupu

V každém intervalu vzorkování je třeba provést následující posloupnost výpočtů:

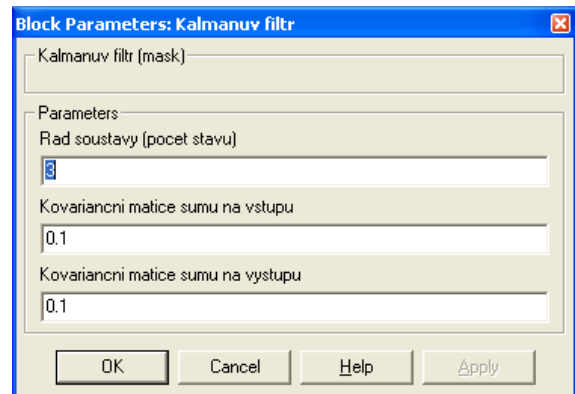
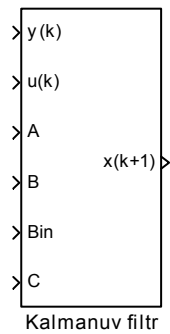
1. výpočet pomocného vektoru  $\mathbf{M}$   $\mathbf{M} = \mathbf{P}_K(k/k-1)\mathbf{C}'(\mathbf{R}_K + \mathbf{C}\mathbf{P}_K(k/k-1)\mathbf{C}')^{-1}$
2. datová aktualizace matice  $\mathbf{P}_K$   $\mathbf{P}_K(k/k) = (\mathbf{I} - \mathbf{M}\mathbf{C})\mathbf{P}_K(k/k-1)$
3. časová aktualizace matice  $\mathbf{P}_K$   $\mathbf{P}_K(k+1/k) = \mathbf{A}\mathbf{P}_K(k/k)\mathbf{A}' + \mathbf{B}_{in}\mathbf{Q}_K\mathbf{B}_{in}'$

4. datová aktualizace vektoru  $x$
5. časová aktualizace vektoru  $x$

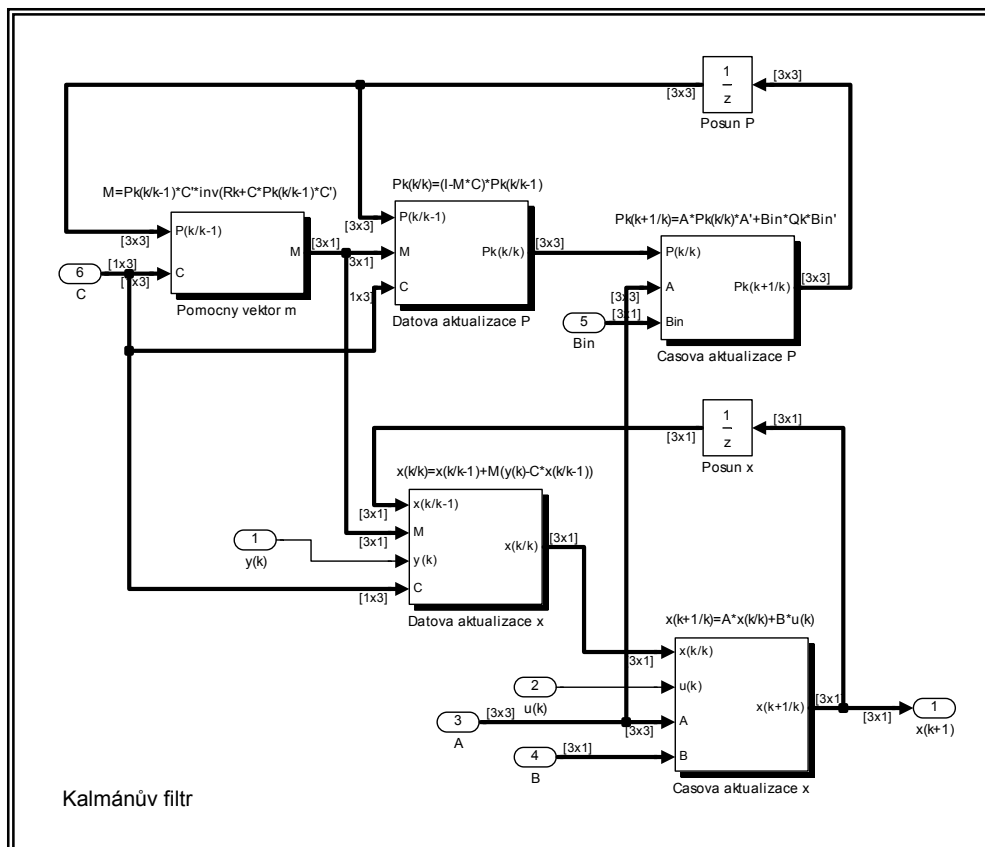
$$x(k/k) = x(k/k-1) + M(y(k) - Cx(k/k-1))$$

$$x(k+1/k) = Ax(k/k) + Bu(k)$$

Vstupy bloku Simulinku jsou regulovaná veličina  $y(k)$ , akční veličina  $u(k)$  a matice  $A$ ,  $B$ ,  $B_{in}$  a  $C$ . Výstupem je odhad stavu soustavy  $x(k+1)$ . Parametry bloku jsou řád soustavy  $N$  (počet stavů), kovarianční matice šumu na vstupu  $Q_k$  a kovarianční matice šumu na výstupu  $R_k$ .



Pro dílčí operace 1-5 jsou vytvořeny samostatné subsystemy. Bloky „posun P“ a „posun x“ mají jako parametr počáteční hodnoty matice  $P$  a vektoru  $x$ .



## LQ regulátor – programové řízení na konečném horizontu řízení s penalizací akční veličiny

Soustava je řízena pomocí LQ regulátoru na konečném horizontu řízení délky  $N$ . Model soustavy je ve stavovém popisu

$$x(k+1) = Ax(k) + Bu(k), \quad y(k) = Cx(k)$$

Kritérium je ve tvaru:  $J(N) = \frac{1}{2} x'(N)Q_N x(N) + \frac{1}{2} \sum_{i=k+1}^{k+N-1} [e'(i)Qe(i) + u'(i)Ru(i)]$ , kde

- $N$  je délka horizontu řízení (počet kroků)
- $x(N)$  je vektor stavu na konci horizontu řízení
- $Q_N$  je penalizační matice koncového stavu
- $e(i)$  je vektor regulačních odchylek v  $i$ -tém kroku
- $Q$  je penalizační matice regulačních odchylek

$$e(i) = w(i) - y(i), \quad w = [w(k+N-1) \ w(k+N) \ \dots \ w(k+1)]$$

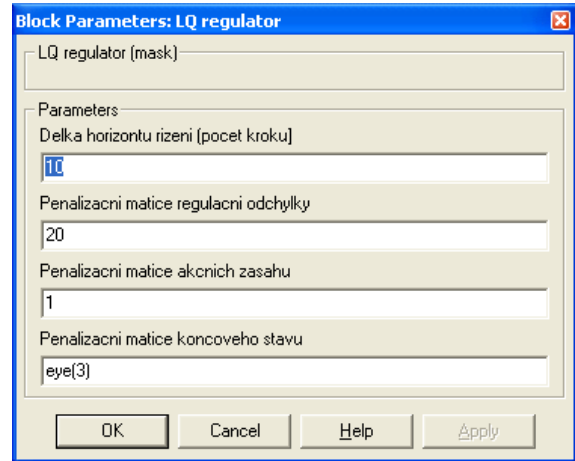
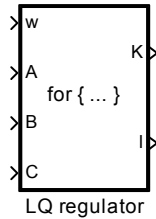
$u(i)$  je vektor akčních zásahů v  $i$ -tém kroku  
 $R$  je penalizační matice akčních zásahů

Regulační zákon:  $u(k) = K(1)x(k) + I(1)$

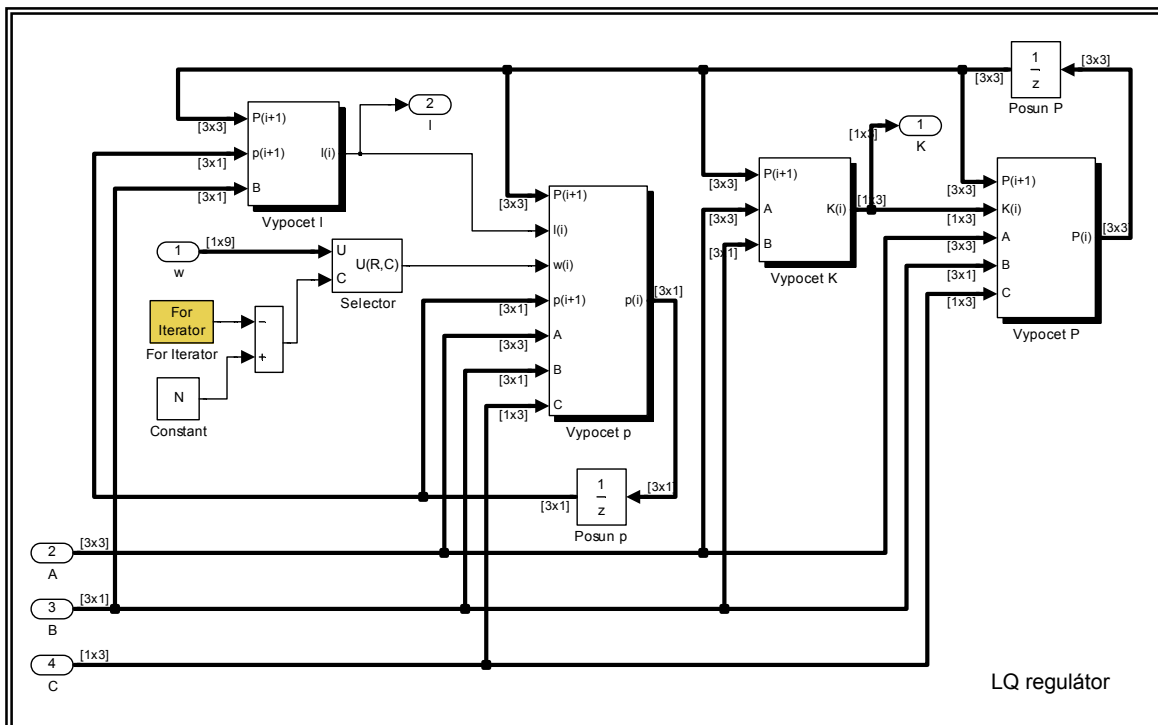
Pro  $i = N-1, \dots, 1$  je nutné rekurentně počítat parametry  $K(i)$  a  $I(i)$ , pomocnou matici  $P(i)$ , pomocný vektor  $p(i)$  podle vztahů:

- |                                  |   |
|----------------------------------|---|
| 1. výpočet parametru $K$         | $K(i) = -[R+B'P(i+1)B]^{-1}B'P(i+1)A$     |
| 2. výpočet parametru $I$         | $I(i) = -[R+B'P(i+1)B]^{-1}B'p(i+1)$      |
| 3. výpočet pomocné matice $P$    | $P(i) = C'QC+A'P(i+1)A+[B'P(i+1)A]K(i)$   |
| 4. výpočet pomocného vektoru $p$ | $p(i) = A'p(i+1)-C'Qw(i)+[B'P(i+1)A]I(i)$ |

Vstupy bloku Simulinku jsou budoucí průběh žádané hodnoty  $w$ , matice  $A$ ,  $B$  a  $C$ . Výstupem jsou matice  $K$  a vektor  $I$  regulačního zákona. Parametry bloku jsou délka horizontu řízení  $N$  a penalizační matice  $Q$ ,  $R$  a  $Q_N$ .

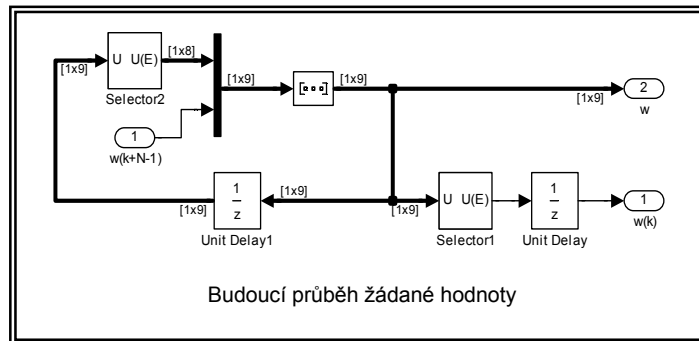
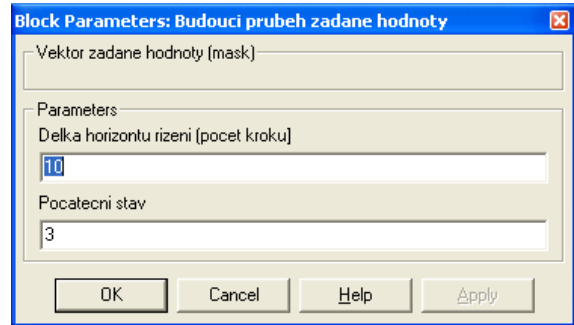
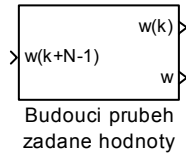


Iterace je provedena pomocí bloku *For Iterator Subsystem*. Počet iterací bloku „For Iterator“ je nastaven na  $N-1$ . Počáteční hodnoty pomocné matice  $P$  a pomocného vektoru  $p$  jsou zadány jako parametry bloků „Posun  $P$ “ a „Posun  $p$ “.



## Budoucí průběh žádané hodnoty

Pro LQ regulátor je nutné mít k dispozici budoucí průběh žádané hodnoty. Vstupem bloku Simulinku je žádaná hodnota v čase  $k+N-1$ . Výstupem je vektor  $w$  obsahující  $N-1$  budoucích hodnot a žádaná hodnota v čase  $k$ . Parametry bloku jsou délka horizontu řízení  $N$  a počáteční hodnota žádané.



## Převod soustavy do stavového popisu

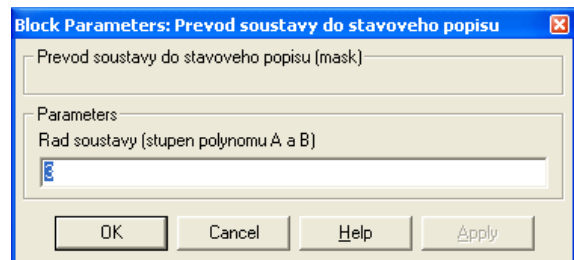
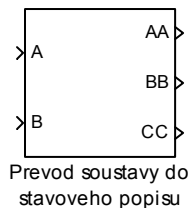
Rekurentní metodou nejmenších čtverců získáme diferenční rovnici. Pro Kalmánův filtr a LQ regulátor je nutné model soustavy převést ze vstupně-výstupního do stavového popisu.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_N(k+1) \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_{N-1} & -a_N \\ 1 & 0 & & 0 & 0 \\ & & & \vdots & \vdots \\ & & & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_N(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \cdot u(k), \quad y(k) = [b_1 \quad b_2 \quad \dots \quad b_N] \cdot \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_N(k) \end{bmatrix}$$

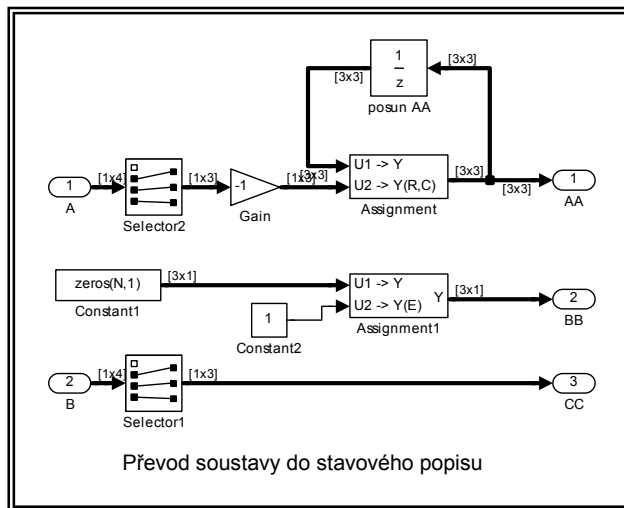
$$x(k+1) = AA \cdot x(k) + BB \cdot u(k)$$

$$y(k) = CC \cdot x(k)$$

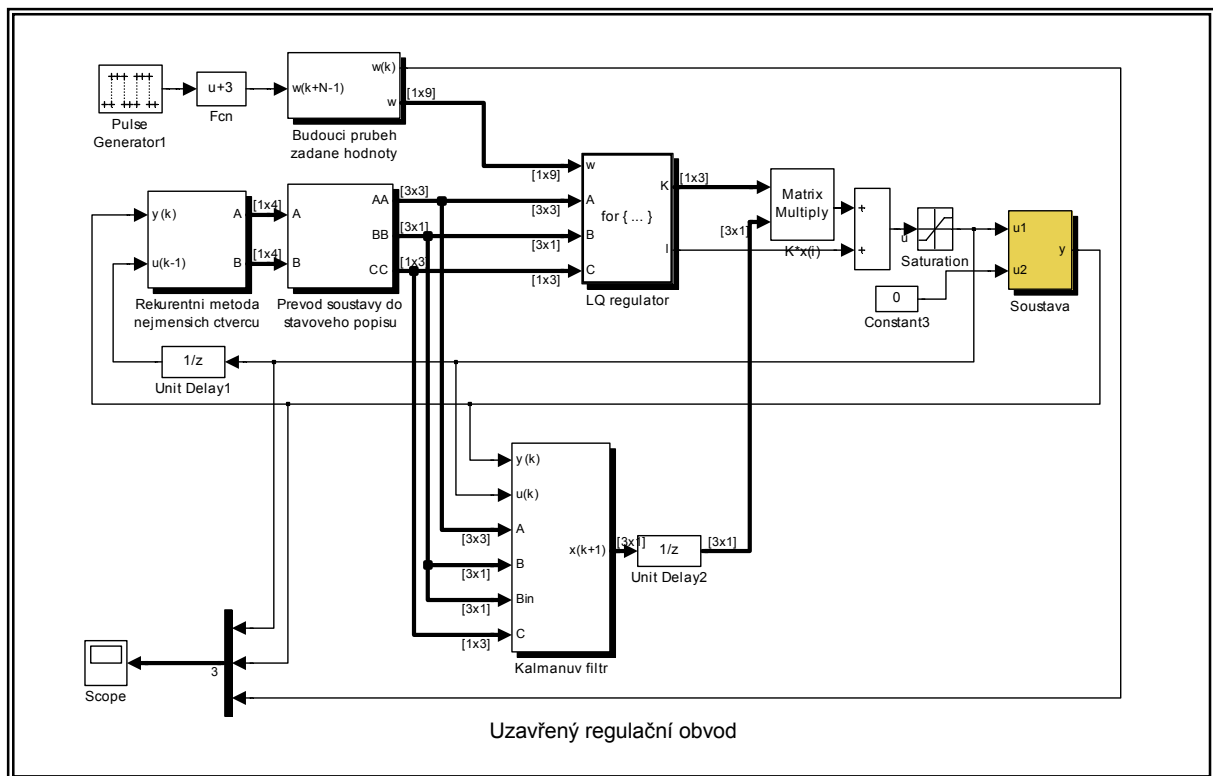
Vstupem bloku jsou polynomy  $A$  a  $B$ . Výstupem je matice  $AA$  a vektory  $BB$  a  $CC$  stavového popisu. Parametrem bloku je řád soustavy  $N$ .



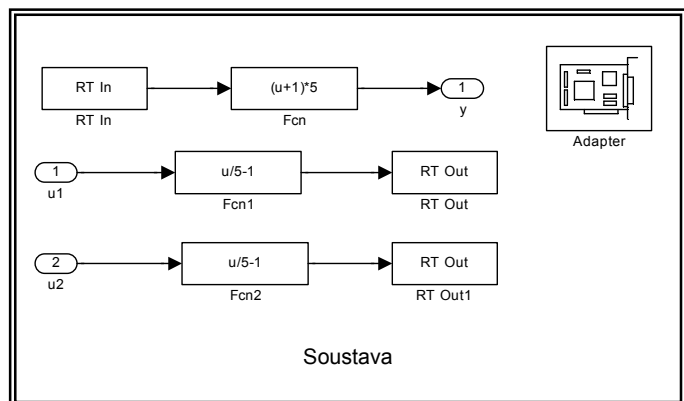
Počáteční naplnění matice  $AA$  je provedeno v bloku „posun  $AA$ “. V každém intervalu vzorkování je do prvního řádku matice  $AA$  přenesen vektor  $A$  se záporně vzatými koeficienty. Vektor  $BB$  je konstantní. Vektor  $CC$  je tvořený přímo koeficienty vektoru  $B$ .



## Uzavřený regulační obvod



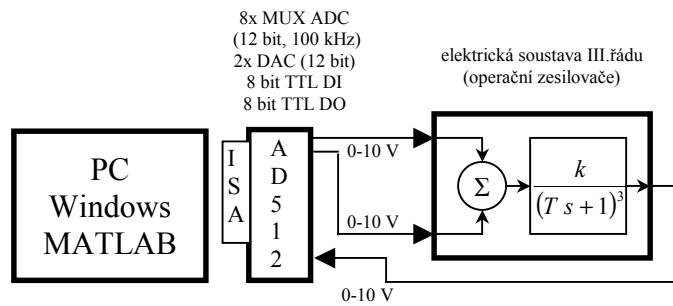
Propojením výše uvedených bloků je vytvořeno simulační schéma uzavřeného regulačního obvodu. Pro generování žádané hodnoty je použit blok „Pulse Generator“. Blok soustava je subsystem obsahující pouze bloky pro vstupní a výstupní signály měřicí karty „RT In“ a „RT Out“ a přepočítání na inženýrské jednotky. U Kalmanova filtru musí být blok „Unit Delay“ zpožďující predikovaný stav soustavy o jeden interval vzorkování. Vzorkování celého modelu je určováno vzorkováním bloku „Pulse Generator“ a bloků „RT In“ a „RT Out“.





## Řízená soustava a spojení s PC

Uspořádání zařízení je na obrázku. Řízenou soustavou je elektrická soustava (operační zesilovače) s dvěma vstupy a jedním výstupem. Tato soustava je třetího řádu s ustálením cca 10 sec a zesílením cca 1. Pro připojení soustavy je využita akviziční karta AD512 (Humusoft). Tato karta obsahuje 12 bitový analogočíslíkový převodník 100 kHz (ADC Analog to Digital Converter) s multiplexorem (8 kanálů) a dva 12 bitové číslicovoanalogové (DAC Digital to Analog Converter) převodníky, oba s nastaveným vstupním i výstupním rozsahem 0-10 V. Karta je vložena do ISA slotu počítač PC v konfiguraci Pentium 150 MHz, 128 MB EDO RAM a Windows 98 SE. Použití karty z MATLABU verze 6.1 a SIMULINKu zajišťuje Real Time Toolbox verze 3.1 (Humusoft).



## Realizace řízení soustavy v reálném čase

Je ukázán průběh regulačního pochodu při adaptivním řízení elektrické soustavy. Interval vzorkování je 0,5 s. Žádaná hodnota má obdélníkový průběh s periodou 50 s. Regulační pochod začíná s úplnou neznalostí modelu soustavy. K odhadu parametrů dochází v několika prvních krocích regulace. Dále jsou uvedeny parametry bloků Simulinku a počáteční hodnoty pomocných matic.

### Parametry bloků Simulinku

Rekurentní metoda nejmenších čtverců:  $N = 3$ ,  $\lambda = 0,99$

Kalmánův filtr:  $N = 3$ ,  $Q_K = 0,1$ ,  $R_K = 0,1$

LQ regulátor:  $N = 10$ ,  $Q = 20$ ,  $R = 1$ ,  $Q_N = \text{eye}(3)$

Budoucí průběh žádané hodnoty:  $N = 10$ ,  $w_0 = 3$  (obdélníkový průběh, perioda 50 s, úrovně 3 a 7)

Převod soustavy do stavového popisu:  $N = 3$

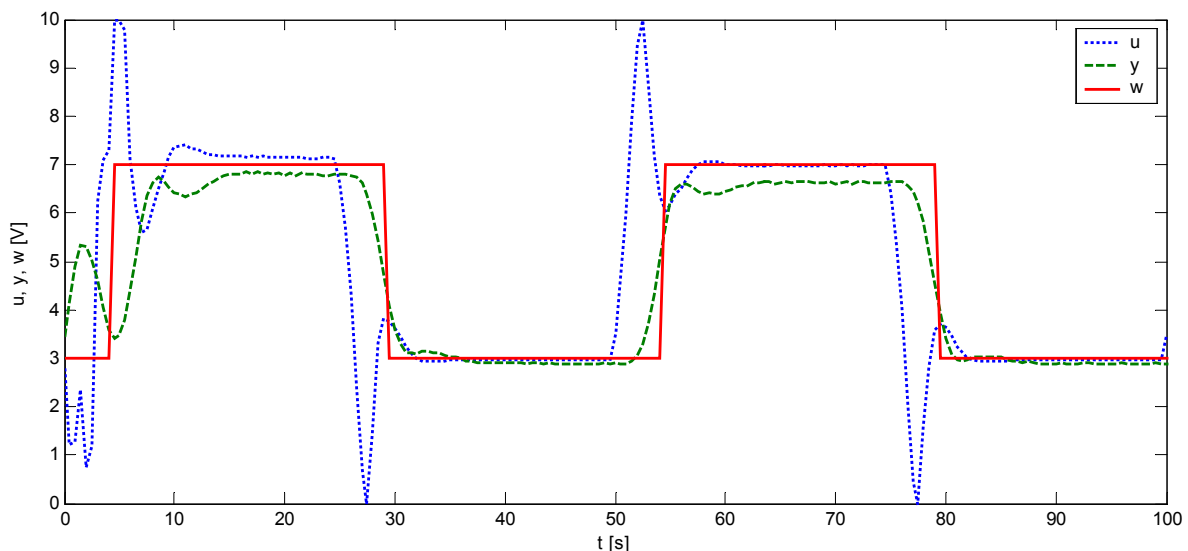
### Počáteční hodnoty pomocných matic

Rekurentní metoda nejmenších čtverců:  $x = \text{zeros}(1, 2*N)$ ,  $P = \text{eye}(2*N)*100$ ,  $\theta = \text{ones}(2*N, 1)$

Kalmánův filtr:  $P = \text{eye}(N)*0,01$ ,  $x = \text{zeros}(N, 1)$

LQ regulátor:  $p = 0$ ,  $P = Q_N$

Budoucí průběh žádané hodnoty:  $w = \text{ones}(1, N-1)*w_0$



Regulační pochod

## Závěr

Cílem článku není čtenáře seznámit s novým přístupem k řízení, ale na standardních příkladech demonstrovat možnosti Simulinku verze 4 v oblasti maticových výpočtů. Bloky lze většinou vytvořit přirozeným přepisem matematických vztahů. Práce s maticemi a jejich naplnění na začátku simulace je poněkud pracnější a proto je při vytváření blokových schémat velmi důležitá možnost průběžně kontrolovat rozměry signálů. Pro iterační výpočty disponuje Simulink subsystemy pro řízení výpočtů (rozhodování a cykly).

Bloky LQ regulátoru a Kalmánova filtru jsou sestaveny obecně a lze je použít pro mnohazměrovou soustavu (s větším počtem vstupů a výstupů). Bloky metody nejmenších čtverců a převodu do stavového popisu jsou uvažovány pro jednorozměrovou soustavu. U metody nejmenších čtverců jsou ukázána schémata všech použitých subsystemů. U ostatních metod jsou uvedeny pouze první úrovně. Interval vzorkování všech bloků je určován intervalem vzorkování nadřazeného systému a není jej nutné zadávat jako parametr.

Oproti skriptu Matlabu je řešení v Simulinku poněkud pracnější a méně přehledné. V případě, že jsou v modelu různé nelinearity, části simulované spojitě nebo systémy s různými intervaly vzorkování, má použití Simulinku své opodstatnění. Práce se vstupními a výstupními signály je v Simulinku pohodlná (načtení dat z prostředí Matlabu nebo ze souboru, interpolace, předání libovolně vyzorkovaných dat zpět do Matlabu). Simulaci lze spustit příkazem Matlabu, čímž jsou otevřené možnosti pro nejrůznější optimalizace.

## Literatura

**Dušek, F.:** *MATLAB a SIMULINK úvod do používání*. Druhé rozšířené vydání. [skriptum] Pardubice 2002, ISBN 80-7194-475-0

**Dušek, F.:** *Adaptivní řízení*. [učební text] Pardubice 1999

**Dušek, F.; Honc, D.:** *Využití sériové linky pod MATLABem verze 6*. [v tomto sborníku], 2002

**Havlena, V.; Štecha, J.:** *Moderní teorie řízení*. [skriptum] Vydavatelství ČVUT, 1996

**SIMULINK Help:** *Using SIMULINK/Block Reference*. 2001

Ing. Daniel Honc, Ph.D.  
doc. Ing. František Dušek, CSc.

e-mail: [daniel.honc@upce.cz](mailto:daniel.honc@upce.cz)  
e-mail: [frantisek.dusek@upce.cz](mailto:frantisek.dusek@upce.cz)

tel.: 040-603 7107  
tel.: 040-603 7125



Katedra řízení procesů a výpočetní techniky  
Fakulta chemicko-technologická  
Univerzita Pardubice  
nám. Čs. legií 565  
532 10 Pardubice

