

SIMULAČNÉ OVERENIE PREDIKTÍVNEHO RIADENIA MODELOV DYNAMICKÝCH SYSTÉMOV V NAVRHNUTOM GRAFICKOM POUŽÍVATEĽSKOM PROSTREDÍ

Štefan Jajčišin, Anna Jadlovská

Katedra kybernetiky a umelej inteligencie, FEI TU v Košiciach, Slovenská republika

Abstrakt

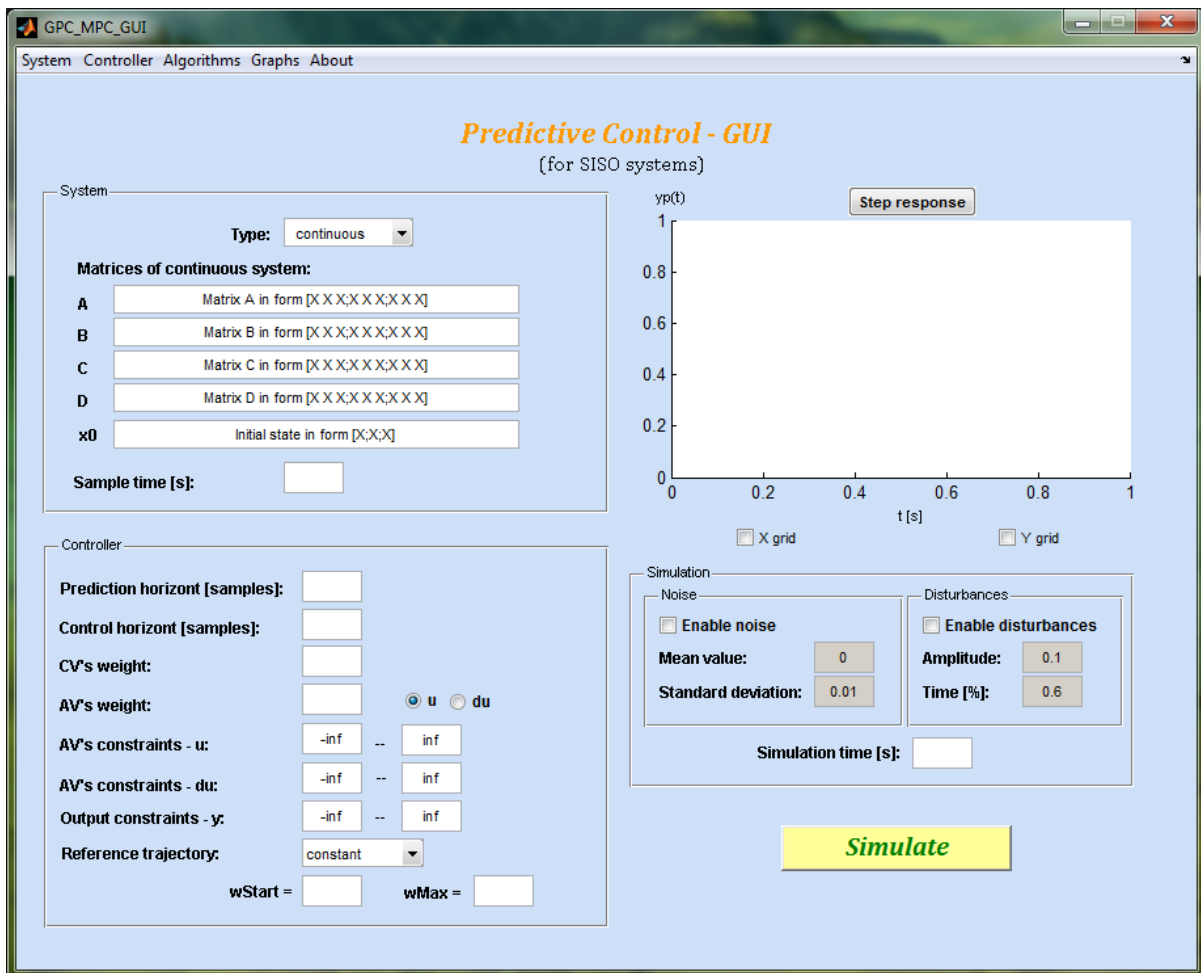
V článku je prezentované grafické používateľské prostredie, v ktorom sú implementované algoritmy prediktívneho riadenia založené na stavovom opise a na vstupno-výstupných regresných ARX a CARIMA modeloch dynamických systémov. Bližšie je rozoberaný spôsob algoritmickej realizácie konkrétnych metód prediktívneho riadenia a programová realizácia uzavretého regulačného obvodu pre potreby simulácie riadenia lineárnych modelov dynamických systémov. V rámci článku je predložený aj ukázkový príklad simulácie prediktívneho riadenia modelu fyzikálneho systému Guľôčka na ploche.

1 Úvod

V tomto článku sa venujeme opisu grafického používateľského prostredia (ďalej len GUI – *Graphical User Interface*), ktoré sme vytvorili v prostredí Matlab za účelom simulácie riadenia lineárnych modelov dynamických systémov pomocou algoritmov prediktívneho riadenia. Podstatnú časť GUI tvoria implementované algoritmy prediktívneho riadenia, naprogramované ako samostatné funkcie, ktoré vypočítavajú veľkosť akčného zásahu na základe konkrétnych hodnôt vstupných parametrov. GUI sme naprogramovali s využitím nástroja *guide*, ktorý je súčasťou programového vybavenia Matlabu [1]. Pomocou tohto nástroja sme v GUI vytvorili potrebné grafické objekty a definovali im príslušné udalosti, teda činnosti, ktoré sa majú vykonať po ich aktivácii.

Na obr. 1 je hlavné okno GUI, ktoré sa zobrazí po spustení navrhutej aplikácie. Môžeme ho rozdeliť na štyri časti. Ľavá horná časť slúži pre zadávanie lineárneho modelu dynamického systému v stavovom opise a v ľavej dolnej časti sa nachádzajú formuláre pre nastavenie parametrov algoritmov prediktívneho riadenia. V pravej hornej časti je možné pomocou tlačidla *Step response* zobrazíť prechodovú charakteristiku zadaného modelu dynamického systému. V grafe sa vykresľuje prechodová charakteristika s využitím funkcie *step*, ktorá je súčasťou *Control System Toolboxu*, ale aj pomocou nami naprogramovanej funkcie pre výpočet prechodovej charakteristiky *preChar*. Pomocou zaškrtnutých políčok pod grafom je možné zapnúť, resp. vypnúť mriežku v grafe pre os x a y . Pravá dolná časť hlavného okna slúži pre nastavenie parametrov simulácie, ako sú doba simulácie (*Simulation time*) v sekundách, šum (*Noise*), porucha (*Disturbances*). V tejto časti sa nachádza aj tlačidlo *Simulate*, ktorým sa spúšťa simulácia. Súčasťou okna je aj hlavné menu s položkami: *System*, *Controller*, *Algorithm*, *Graphs*, *About*.

V ďalšom bližšie popíšeme funkcionality GUI.



Obrázok 1: Hlavné okno grafického používateľského prostredia

2 Definovanie modelu dynamického systému v GUI

V rámci GUI je možné zadávať spojité, resp. diskrétny stavový opis lineárneho modelu SISO dynamického systému v tvare

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) & \mathbf{x}(k+1) &= \mathbf{A}_d\mathbf{x}(k) + \mathbf{b}_d u(k) \\ y(t) &= \mathbf{c}^T \mathbf{x}(t) + du(t) & y(k) &= \mathbf{c}^T \mathbf{x}(k) + du(k) \end{aligned} \quad \text{resp.} \quad (1)$$

kde \mathbf{A} , resp. \mathbf{A}_d je matica dynamiky s rozmermi $n_x \times n_x$, \mathbf{b} , resp. \mathbf{b}_d je stĺpcový vektor vstupu s dĺžkou n_x , \mathbf{c}^T je riadkový vektor výstupu s dĺžkou n_x , d je koeficient priamej väzby medzi vstupom a výstupom, $\mathbf{x}(t)$, resp. $\mathbf{x}(k)$ je stĺpcový vektor stavových veličín s dĺžkou n_x , $u(t)$, resp. $u(k)$ je hodnota vstupu, $y(t)$, resp. $y(k)$ je hodnota výstupu, pričom premenná n_x vyjadruje počet stavov dynamického systému.

Stavový opis (1) je možné získať linearizáciou nelineárnych diferenciálnych rovníc

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{f}(\mathbf{x}(t), u(t), t) \\ y(t) &= \mathbf{g}(\mathbf{x}(t), u(t), t) \end{aligned} \quad (2)$$

popisujúcich dynamiku konkrétneho fyzikálneho systému, kde t je čas, \mathbf{f} a \mathbf{g} sú vektorové, väčšinou nelineárne funkcie. Musíme podotknúť, že aktuálna verzia GUI je obmedzená na systémy s jedným vstupom a jedným výstupom. Rozšírenie na viacparametrové systémy je obsiahnuté v ďalšej verzii GUI, ktorá je momentálne vo vývoji.

Na základe voľby používateľ zadáva spojité alebo diskrétny stavový opis lineárneho modelu dynamického systému, ktorý bude použitý ako riadený systém. Zároveň sa zadaný model použije pri predikcii v rámci vybraných algoritmov prediktívneho riadenia. Je potrebné poznamenať, že v prípade zadávania diskrétného stavového opisu musí byť definovaná perióda vzorkovania v korešpondencii

s maticami stavového opisu. V prípade zadávania spojitého stavového opisu sa daná perióda vzorkovania použije pri diskretizácii modelu.

GUI umožňuje prostredníctvom položky *System* v hlavnom menu ukladať, prípadne načítavať skôr uložený lineárny model dynamického systému zo súboru.

3 Algoritmy prediktívneho riadenia implementované v GUI

Algoritmy prediktívneho riadenia, ktoré sú implementované v GUI patria k optimalizačným algoritmom a vo všeobecnosti minimalizujú funkcionál v tvare

$$J = \sum_{i=N_1}^{N_p} Q(i) [\hat{y}(k+i) - w(k+i)]^2 + \sum_{i=1}^{N_u} R(i) [u(k+i-1)]^2, \quad (3)$$

kde $u(k)$ je akčný zásah, $\hat{y}(k)$ predikovaná hodnota regulovanej veličiny a $w(k)$ referenčná trajektória. Hodnoty N_1 až N_p udávajú horizont predikcie (predikčný horizont), na ktorom sa vypočítava optimálna postupnosť hodnôt akčného zásahu $u(k)$. Kladná hodnota N_u predstavuje riadiaci horizont, pričom platí $N_u \leq N_p$. Ak sa pri metódach prediktívneho riadenia používa redukcia stupňov voľnosti akčného zásahu, platí $N_u < N_p$ [2].

Hodnoty $Q(i)$, resp. $R(i)$ reprezentujú váhové koeficienty regulačnej odchýlky na dĺžke horizontu predikcie, resp. akčného zásahu na dĺžke riadiaceho horizontu. Je potrebné poznamenať, že z hľadiska hodnôt váhových koeficientov je dôležitý predovšetkým koeficient pomeru $\lambda = R/Q$.

V niektorých prípadoch sa vo funkcionáli (3) namiesto akčného zásahu $u(k)$ používa zmena akčného zásahu $\Delta u(k)$, čím riadenie získava integračný charakter [2]. Dôležitým faktom v prípade metód prediktívneho riadenia je nutnosť poznať referenčnú trajektóriu $w(k)$ na celej dĺžke predikčného horizontu.

Algoritmy prediktívneho riadenia, ktoré sme implementovali do GUI môžeme rozdeliť do dvoch kategórií: algoritmy založené na stavovom opise – stavové MPC (Model Predictive Control), pre ktoré je riadiaca schéma zobrazená na obr. 2 a algoritmy vychádzajúce z prenosových funkcií – GPC (Generalized Predictive Control), ktorých použitie znázorňuje riadiaca schéma na obr. 3. V rámci každej kategórie sme naprogramovali dva typy algoritmov:

A. stavové MPC algoritmy

A1. stavový MPC algoritmus s prediktorom v tvare

$$\hat{y} = \mathbf{V}\mathbf{x}(k) + \mathbf{G}\mathbf{u} \quad (4)$$

(v GUI označený ako *MPC v.1*, detailne rozobraný v [2], prípadne [8]),

A2. stavový MPC algoritmus s prediktorom v tvare

$$\hat{y} = \mathbf{V}\mathbf{x}(k) + \mathbf{G}_1\mathbf{u}(k-1) + \mathbf{G}_2\Delta\mathbf{u} \quad (5)$$

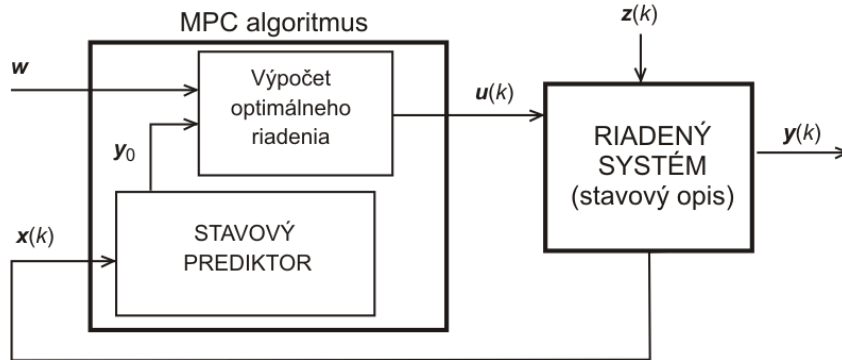
(v GUI označený ako *MPC v.2*, detailne rozobraný v [3], prípadne [8]),

kde $\mathbf{x}(k)$ je vektor aktuálneho stavu, \mathbf{u} , resp. $\Delta\mathbf{u}$ je postupnosť hodnôt, resp. zmien akčného zásahu. Za predpokladu, že v stavovom opise modelu dynamického systému (1) je koeficient priamej väzby $d=0$, má vektor predikovaných hodnôt výstupu \hat{y} , vektor postupnosti akčného zásahu \mathbf{u} , matica voľnej odozvy \mathbf{V} a matice vynútenej odozvy dynamického systému $\mathbf{G}, \mathbf{G}_1, \mathbf{G}_2$ tvar

$$\hat{y} = [\hat{y}(k+1) \quad \hat{y}(k+2) \quad \dots \quad \hat{y}(k+N_p)]^T, \quad \mathbf{u} = [u(k) \quad u(k+1) \quad \dots \quad u(k+N_p-1)]^T$$

$$\mathbf{V} = \begin{pmatrix} \mathbf{c}^T \mathbf{A}_d \\ \vdots \\ \mathbf{c}^T \mathbf{A}_d^{N_p} \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \mathbf{c}^T \mathbf{b}_d & 0 & \dots & \dots \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{c}^T \mathbf{A}_d^{N_p-1} \mathbf{b}_d & \dots & \mathbf{c}^T \mathbf{b}_d & \dots \end{pmatrix}, \quad \mathbf{G}_1 = \begin{pmatrix} \mathbf{c}^T \mathbf{b}_d \\ \mathbf{c}^T (\mathbf{A}_d + \mathbf{I}) \mathbf{b}_d \\ \vdots \\ \mathbf{c}^T (\mathbf{A}_d^{N_p-1} + \dots + \mathbf{A}_d + \mathbf{I}) \mathbf{b}_d \end{pmatrix},$$

$$\mathbf{G}_2 = \begin{pmatrix} \mathbf{c}^T \mathbf{b}_d & 0 & \cdots & 0 \\ \mathbf{c}^T (\mathbf{A}_d^{N_p} + \mathbf{I}) \mathbf{b}_d & \mathbf{c}^T \mathbf{b}_d & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \mathbf{c}^T (\mathbf{A}_d^{N_p-1} + \cdots + \mathbf{A}_d + \mathbf{I}) \mathbf{b}_d & \cdots & \mathbf{c}^T (\mathbf{A}_d + \mathbf{I}) \mathbf{b}_d & \mathbf{c}^T \mathbf{b}_d \end{pmatrix}.$$



Obrázok 2: Riadiaca štruktúra so stavovým MPC riadiacim algoritmom

B. GPC algoritmy

B1. GPC algoritmus vychádzajúci z ARX modelu dynamického systému

$$A_z(z^{-1})y(k) = B_z(z^{-1})u(k) + \xi(k) \quad (6)$$

pri vyjadrení prediktora v tvare

$$\hat{\mathbf{y}} = \mathbf{y}_0 + \mathbf{G}_3 \mathbf{u} \quad (7)$$

(v GUI označený ako *GPC ARX*, detailne rozobraný v [4], prípadne [8]),

B2. GPC algoritmus vychádzajúci z CARIMA modelu dynamického systému

$$A_z(z^{-1})y(k) = B_z(z^{-1})u(k-1) + \frac{C_z(z^{-1})}{\Delta} \xi(k) \quad (8)$$

pri vyjadrení prediktora v tvare

$$\hat{\mathbf{y}} = \mathbf{y}_0 + \mathbf{G}_4 \Delta \mathbf{u} \quad (9)$$

(v GUI označený ako *GPC CARIMA*, detailne rozobraný v [5], prípadne [8]),

kde - $B_z(z^{-1})$, resp. $A_z(z^{-1})$ sú polynómy čitateľa, resp. menovateľa rádu m , resp. n s koeficientmi b_i , resp. a_i , $C_z(z^{-1})$ je polynóm,

- $u(k)$ je vstup, $y(k)$ je výstup dynamického systému a $\xi(k)$ je chyba výstupu systému, resp. podľa [4] šum merania výstupu,

- $\Delta = 1 - z^{-1}$ predstavuje integrátor [5].

Vo vzťahoch (7) a (9) je \mathbf{y}_0 vektor predikcie hodnôt voľnej odozvy systému a výrazy $\mathbf{G}_3 \mathbf{u}$ a $\mathbf{G}_4 \Delta \mathbf{u}$ predstavujú vynútenú odozvu systému, pričom matice \mathbf{G}_3 a \mathbf{G}_4 je možné určiť na základe vzťahov uvedených v [8].

Po dosadení prediktorov (4), (5), (7) a (9) do funkcionálu (3) v maticovom tvare

$$J = (\hat{\mathbf{y}} - \mathbf{w})^T \mathbf{Q} (\hat{\mathbf{y}} - \mathbf{w}) + \mathbf{u}^T \mathbf{R} \mathbf{u}, \quad \text{resp.} \quad J = (\hat{\mathbf{y}} - \mathbf{w})^T \mathbf{Q} (\hat{\mathbf{y}} - \mathbf{w}) + \Delta \mathbf{u}^T \mathbf{R} \Delta \mathbf{u} \quad (10)$$

môžeme odvodiť kvadratickú formu (zvlášť pre príslušný algoritmus), ktorú je potrebné minimalizovať podľa \mathbf{u} , resp. $\Delta \mathbf{u}$:

$$\min_{\mathbf{u}} \frac{1}{2} \mathbf{u}^T \mathbf{H} \mathbf{u} + \mathbf{g}^T \mathbf{u}, \quad \text{resp.} \quad \min_{\Delta \mathbf{u}} \frac{1}{2} \Delta \mathbf{u}^T \mathbf{H} \Delta \mathbf{u} + \mathbf{g}^T \Delta \mathbf{u}. \quad (11)$$

Bližší postup a odvodenie je uvedené v [1], [3], [4], [5] a [8].

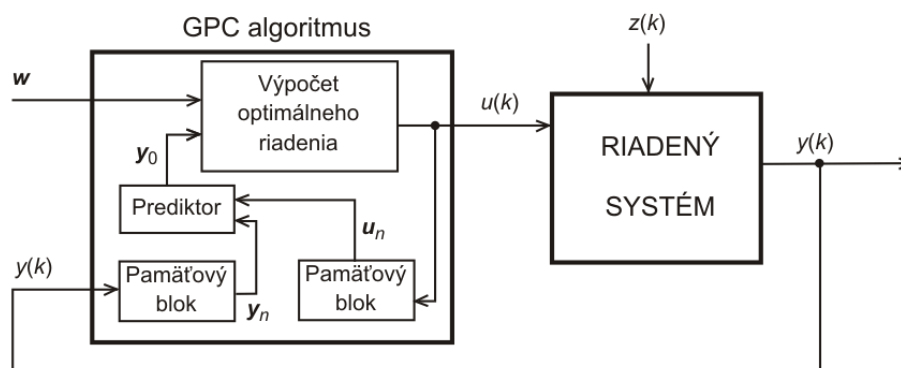
Akčný zásah je na základe podmienky o minimalizácii funkcionálu

$$\frac{\partial J}{\partial \mathbf{u}} = \mathbf{0}, \quad \text{resp.} \quad \frac{\partial J}{\partial \Delta \mathbf{u}} = \mathbf{0} \quad (12)$$

daný rovnicou

$$\mathbf{u} = -\mathbf{H}^{-1} \mathbf{g}, \quad \text{resp.} \quad \Delta \mathbf{u} = -\mathbf{H}^{-1} \mathbf{g}, \quad (13)$$

kde tvar matice \mathbf{H} a riadkového vektora \mathbf{g}^T závisí od použitého algoritmu riadenia, pričom ich konkrétny tvar pre jednotlivé algoritmy je uvedený v [8].



Obrázok 3: Riadiaca štruktúra s GPC riadiacim algoritmom

Navrhnuté GUI umožňuje definovať model dynamického systému iba vo forme stavového opisu. Z tohto dôvodu v prípade GPC riadiacich algoritmov dochádza pred akýmkoľvek ďalším výpočtom k transformácii matíc stavového opisu na prenosovú funkciu, ktorá predstavuje vstupno-výstupný opis modelov dynamických systémov.

Výhodou algoritmov prediktívneho riadenia je možnosť zakomponovať rôzne obmedzenia (akčného zásahu, zmeny akčného zásahu, výstupu) do výpočtu optimálneho riadenia. Tento výpočet sme realizovali pomocou *kvadratického programovania*, konkrétne pomocou funkcie *quadprog* v toolboxe Matlabu *Optimization Toolbox*, ktorá vypočítava vektor optimálnych hodnôt akčného zásahu minimalizáciou vzťahu (11) pri zohľadnení obmedzení $\mathbf{A}_{obn} \mathbf{u} \leq \mathbf{b}_{obn}$. Základná syntax použitia tejto funkcie na výpočet vektora optimálnych hodnôt \mathbf{u} , resp. $\Delta \mathbf{u}$ je

$$\mathbf{u} = \text{quadprog}(\mathbf{H}, \mathbf{g}, \mathbf{A}_{obn}, \mathbf{b}_{obn}), \quad \text{resp.} \quad \Delta \mathbf{u} = \text{quadprog}(\mathbf{H}, \mathbf{g}, \mathbf{A}_{obn}, \mathbf{b}_{obn}), \quad (14)$$

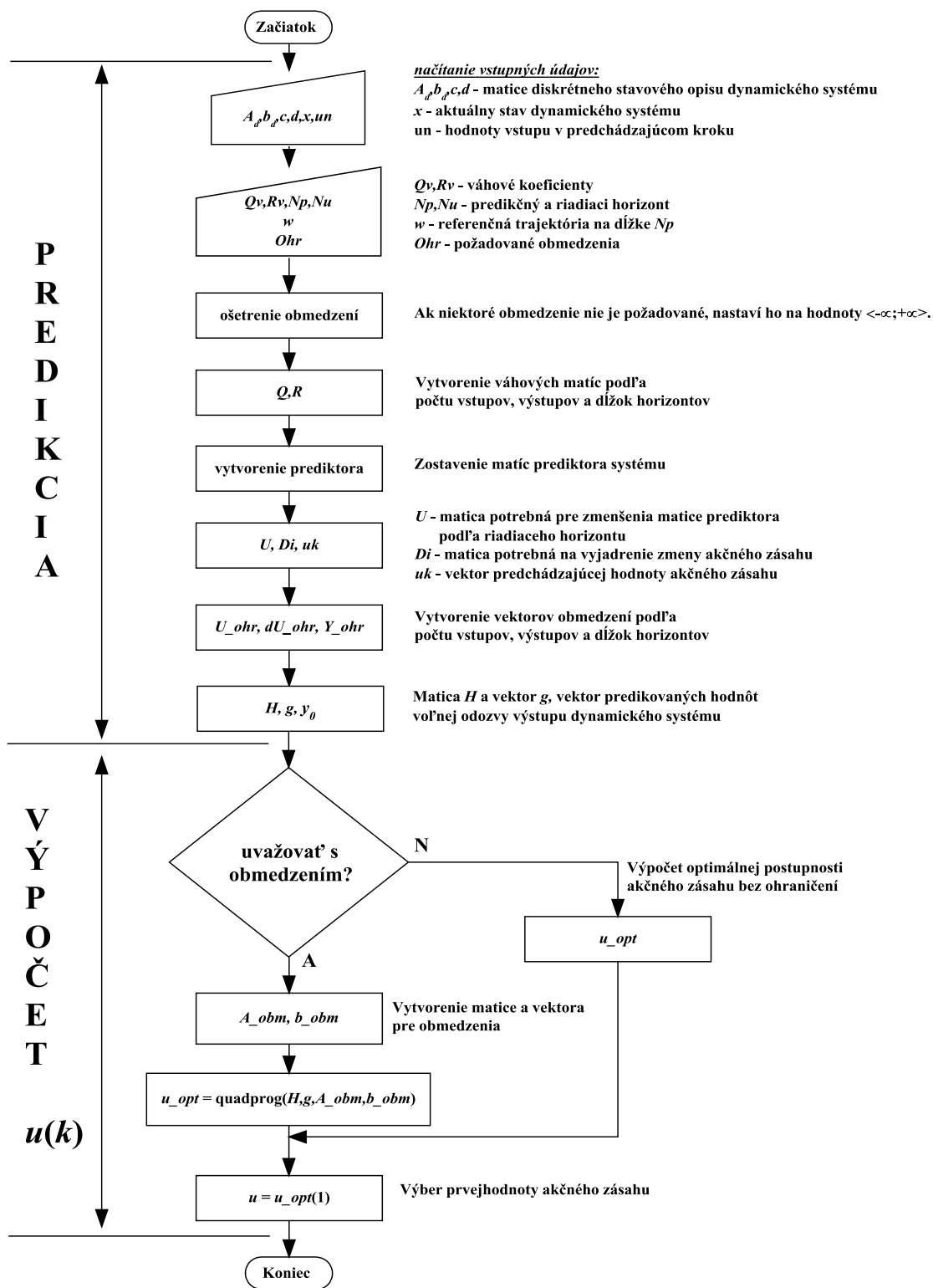
kde maticu \mathbf{A}_{obn} a vektor \mathbf{b}_{obn} je potrebné zostaviť podľa požadovaných obmedzení, pričom ich konkrétny tvar pre jednotlivé algoritmy je uvedený v [8].

Vývojový diagram, na základe ktorého sme realizovali implementáciu stavových MPC algoritmov ako funkcií jazyka Matlab je na obr. 4. Pre implementáciu GPC algoritmov je potrebné niektoré kroky špecificky upraviť, no celkový postup zostáva zachovaný.

Na určenie hodnoty akčného zásahu sme používali *výpočet s pohyblivým horizontom* [2], teda z optimálnej postupnosti akčného zásahu $\mathbf{u}_{opt} = [\mathbf{u}_{opt}(k) \quad \dots \quad \mathbf{u}_{opt}(k + N_u - 1)]$ - vypočítanej na základe vzťahu (13) alebo (14) - v každom kroku k predstavuje hodnotu výstupného parametra funkcie každého algoritmu iba prvý člen $\mathbf{u}_{opt}(k)$.

Algoritmus výpočtu optimálnej postupnosti akčného zásahu \mathbf{u}_{opt} s využitím prediktívneho riadenia s pohyblivým horizontom môžeme vyjadriť v týchto krokoch:

1. určenie referenčnej trajektórie na dĺžke horizontu predikcie \mathbf{w} ,
2. zistenie skutočného stavu $\mathbf{x}(k)$, resp. výstupu $\mathbf{y}(k)$ systému v danom kroku,
3. predikcia správania sa systému na horizonte predikcie na základe skutočných hodnôt akčného zásahu $\mathbf{u}_{opt}(k)$ a stavu $\mathbf{x}(k)$ v predošlých krokoch bez vplyvu akčného zásahu, tzv. voľná odozva systému,
4. výpočet postupnosti akčného zásahu \mathbf{u}_{opt} minimalizáciou funkcionálu J (3) pri známych parametroch $N_1, N_p, N_u, Q(i)$ a $R(i)$,
5. použitie $\mathbf{u}_{opt}(k)$ na vstupe systému a späť na krok č. 1.



Obrázok 4: Vývojový diagram algoritmickej stavových MPC algoritmov

Nastaviteľnými parametrami pre rozoberané riadiace algoritmy sú predikčný horizont N_p , riadiaci horizont N_u , váhové koeficienty $Q(i)$ a $R(i)$, prípadne požadované ohraničenia na akčný zásah, resp. výstup dynamického systému. Hodnotu parametra N_1 sme v našom prípade nastavili na 1.

Hodnoty uvedených parametrov môže používateľ nastavovať a meniť v ľavej dolnej časti GUI. V tejto časti je potrebné určiť aj typ, počiatočnú a maximálnu hodnotu referenčnej trajektórie, ktorá bude na základe týchto údajov vygenerovaná v rámci simulácie. Rovnako, ako v prípade modelu dynamického systému, aj v prípade nastavení parametrov riadiacich algoritmov, je možné tieto údaje ukladať a späť načítavať zo súboru prostredníctvom položky *Controller* v hlavnom menu.

4 Simulácia algoritmov prediktívneho riadenia a zobrazovanie výsledkov v GUI

V procese simulácie prediktívneho riadenia je možné použiť jeden stavový MPC a jeden GPC algoritmus naraz z dôvodu vzájomného porovnania ich účinkov v riadiacej štruktúre podľa obr. 2 a obr. 3. Výber algoritmov použitých v simulácii riadenia je potrebné vykonať prostredníctvom položky *Algorithms* v hlavnom menu.

Nastavenie doby simulácie (*Simulation time*) v sekundách, strednej hodnoty (*Mean value*) a smerodajnej odchýlky šumu (*Standard deviation*), ako aj amplitúdu (*Amplitude*) a čas pôsobenia poruchy (*Time*) v percentuálnom vyjadrení z doby simulácie je možné uskutočniť v pravej dolnej časti GUI. V prípade požiadavky na zašumenie riadeného procesu je prostredníctvom generátora šumu vygenerovaný šum s údajmi definovanými používateľom a pripočítaný k stavom, resp. výstupu systému. Porucha je realizovaná ako skokový prírastok a neskôr úbytok od akčného zásahu v čase určenom používateľom.

Po zadaní lineárneho modelu dynamického systému, výbere a nastavení parametrov riadiacich algoritmov, referenčnej trajektórie a prípadnom nastavení parametrov šumu a poruchy je simuláciu riadenia možné spustiť prostredníctvom tlačidla *Simulate*. Priebeh simulácie riadenia je indikovaný slovom *Simulating...* pod stlačeným tlačidlom, pričom po úspešne uskutočnenej simulácii sa na tomto mieste zobrazí informácia o dĺžke trvania danej simulácie.

Algoritmus, pomocou ktorého sme simulovali riadenie modelov dynamických systémov na základe riadiacich štruktúr uvedených na obr. 2 a obr. 3, má niekoľko krokov:

1. Načítanie vstupných údajov
 - matice diskrétného stavového opisu dynamického systému, v prípade GPC algoritmov transformácia na čitateľ a menovateľ diskrétnej prenosovej funkcie,
 - perióda vzorkovania T_{vz} , doba regulácie (simulácie) T_{sim} ,
 - vektor referenčnej trajektórie w ,
 - nastavenie šumu a poruchovej veličiny z .
2. Inicializácia údajov
 - nastavenie počiatočného výstupu $y(0)$, resp. stavu $x(0)$ systému,
 - nulovanie hodnôt, ktoré vo vstupno-výstupnom opise dynamického systému predstavujú hodnoty v zápornom čase,
 - určenie počtu vzoriek: $n_v = T_{sim} / T_{vz} + 1$, kde T_{sim} je doba simulácie a T_{vz} je perióda vzorkovania,
 - nastavenie počítadla vzoriek: $k \leftarrow n$, kde n je rád systému.
3. Výpočet regulačnej odchýlky $e(k) = w(k) - y(k)$.
4. Výpočet akčného zásahu $u(k)$ podľa konkrétneho vzťahu pre použitý typ riadenia + prípadná saturácia akčného zásahu.
5. Pripočítanie poruchovej veličiny k akčnému zásahu (ak je zadaná).
6. Výpočet nového výstupu $y(k)$ s využitím vypočítaného akčného zásahu $u(k)$ na základe:
 - a. diskrétnej prenosovej funkcie (pre GPC),
 - b. matíc stavového opisu (pre stavové MPC).
7. Pripočítanie šumu k stavom, resp. výstupu (ak chceme simulovať riadenie so šumom).
8. Nastavenie stavu, resp. výstupu lineárneho modelu dynamického systému podľa stavu, resp. výstupu riadeného systému.
9. $k \leftarrow k + 1$.
10. Ak $k \leq n_v + n$, potom skok na bod 3.
11. Odstránenie vzoriek v zápornom čase (ak je to potrebné).
12. Grafický výstup výsledkov simulácie.

Akonáhle je simulácia úspešne skončená, zobrazia sa jednotlivé časové priebehy akčného zásahu $u(k)$, referenčnej trajektórie $w(k)$ a výstupu dynamického systému $y(k)$ v príslušných grafoch podľa voľby používateľa v položke *Graphs* v hlavnom menu. V aktuálnej verzii GUI je možné výsledky zobrazit' nasledovne:

- **Extra** – výsledky simulácie sa zobrazia v osobitnom grafe pre vybraný MPC algoritmus a osobitnom grafe pre vybraný GPC algoritmus, pričom sú zobrazené výsledky, kde boli pri výpočte zohľadnené aj požadované ohraničenia na akčný zásah, prípadne výstup dynamického systému a taktiež výsledky bez týchto ohraničení,
- **Joined** – výsledky simulácie pre MPC a GPC algoritmus sa zobrazia spolu v jednom grafe, pričom zobrazené sú iba výsledky, kde boli vo výpočte zohľadnené požadované ohraničenia na akčný zásah, prípadne výstup dynamického systému.

GUI sme naprogramovali tak, aby výsledky simulácie pre používateľom zadaný typ vážená hodnota u alebo zmeny akčného zásahu Δu vo funkcionáli (10) boli zobrazené v osobitných grafoch. Je to tak navrhnuté z tohto dôvodu, aby bolo možné zobrazit' výsledky simulácie pri oboch typoch vážená naraz, avšak nie v jednom grafe.

Na základe vyššie uvedených spôsobov algoritmickej GUI je možné uskutočniť porovnanie výsledkov:

1. vybraného algoritmu, kde boli, resp. neboli zohľadnené ohraničenia v jednom grafe,
2. vybraného stavového MPC a GPC algoritmu v jednom grafe (iba s ohraničeniami),
3. vybraného algoritmu, kde bola vo funkcionáli vážená hodnota u a Δu (osobitne).

V položke *Graphs* v hlavnom menu je aj voľba *Export to..* \rightarrow *.mat file*, po ktorej zaškrtnutí je používateľovi po úspešnej simulácii ponúknutá možnosť uloženia časových priebehov vo forme *.mat* súboru.

Položka *Graphs/Close* v hlavnom menu uzatvára všetky otvorené okná s časovými priebehmi, čo zrýchľuje prácu s výsledkami simulácie.

5 Ukážkový príklad simulácie riadenia systému Gul'ôčka na ploche

V tejto časti uvádzame ukážkový príklad pre prácu s GUI. Ako riadený systém sme zvolili lineárny model fyzikálneho systému *Gul'ôčka na ploche*, ktorým sme sa bližšie zaoberali v [6] a [7]:

1. Načítanie lineárneho modelu zo súboru

Pomocou klávesovej skratky **CTRL+L** zobrazíme dialógové okno. Z podadresára *examples* vyberieme súbor **BP_system**. V hlavnom okne sa automaticky vyplní časť pre zadávanie modelu dynamického systému príslušnými maticami, ktoré vznikli linearizáciou nelineárneho modelu.

2. Nastavenie parametrov algoritmu prediktívneho riadenia

Parametre algoritmu prediktívneho riadenia nastavíme podľa týchto hodnôt:

Prediction horizon [samples] = 10;

Control horizon [samples] = 2;

CV's weight = 100;

AV's weight = 0,01;

AV's constraint – u = $\langle -1; 1 \rangle$; zaškrtneme **u**;

AV's constraint – Δu = $\langle -\text{inf}; \text{inf} \rangle$ (neobmedzene);

Output constraint – y = $\langle -\text{inf}; \text{inf} \rangle$ (neobmedzene);

Reference trajectory = ramps & stairs (rampy a schody),

wStart = -0,15;

wMax = 0,12.

3. Výber požadovaných algoritmov

V hlavnom menu vyberieme položku *Algorithms* \rightarrow *GPC* \rightarrow *CARIMA* a *Algorithms* \rightarrow *MPC* \rightarrow *v.I.* (Sú to zároveň východzie nastavenia).

4. Nastavenie zobrazovania grafov a export výsledkov do súboru

V menu vyberieme položku **Graphs** → **Joined** a **Graphs** → **Export to..** → **.mat file**.

5. Nastavenie šumu, poruchy a doby simulácie

Povolíme šum a poruchu a ich hodnoty nastavíme takto:

Mean value = 0;

Standard deviation = 0,001;

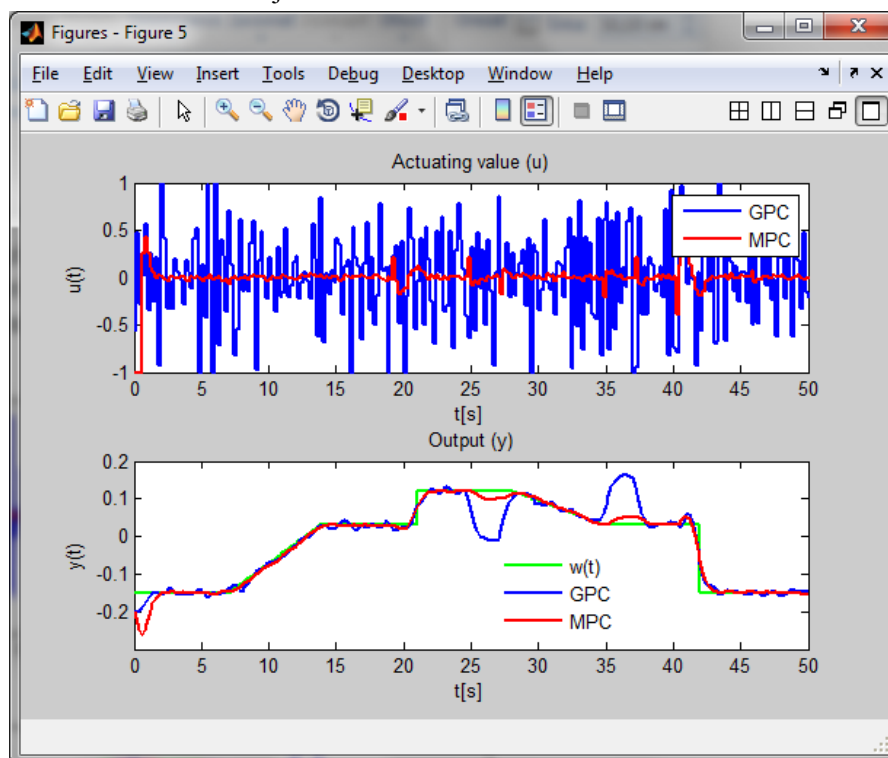
Amplitude = 0,2;

Time [%] = 0,5;

Simulation time [s] = 50;

6. Simulácia a zobrazenie výsledkov s ich uložením.

Po stlačení tlačidla *Simulate* sa zobrazí jedno okno rozdelené na dve časti podľa obr. 5, porovnali sme teda stavový MPC a GPC algoritmus pri zašumenom riadenom systéme. Získané časové priebehy sú zobrazené na nasledujúcom obrázku.



Obrázok 5: Výsledky simulácie prediktívneho riadenia lineárneho modelu systému Guľôčka na ploche pomocou GUI

Avšak, keďže sme vybrali možnosť exportovať výsledky do súboru, objavilo sa aj dialógové okno, v ktorom môžeme zadať názov súboru, do ktorého budú uložené výsledky simulácie riadenia.

6 Záver

V článku sme prezentovali prácu s vytvoreným grafickým používateľským prostredím pre simulačné riadenie modelov dynamických systémov pomocou algoritmov prediktívneho riadenia. Toto grafické používateľské prostredie môže byť plnohodnotne využité pri porovnávaní výsledkov riadenia medzi rôznymi navrhnutými algoritmi prediktívneho riadenia, ako aj na analýzu vplyvu zakomponovania požadovaných ohraničení do výpočtu optimálnej postupnosti akčného zásahu. Konkrétne využitie tohto prostredia bude vo výučbe predmetu *Riadenia a umelá inteligencia*, ktorý sa vyučuje na inžinierskom stupni štúdia na Katedre kybernetiky a umelej inteligencie FEI TU v Košiciach. V rámci tohto predmetu bude slúžiť ako pomôcka pre študentov a bude im poskytovať nástroj pre kontrolu správnosti výsledkov, ktoré získali na základe nimi naprogramovaných algoritmov prediktívneho riadenia.

Prezentované grafické používateľské prostredie je naďalej vo vývoji, postupne ho rozširujeme o nové funkcionality. Ďalšiu verziu doplníme o algoritmy prediktívneho riadenia s využitím nástrojov iných toolboxov, ktoré sa špecializujú na prediktívne riadenie a rozširujeme už existujúce riadiace algoritmy. Okrem toho rozširujeme množinu možných modelov dynamických systémov, konkrétne o mnohorozmerné modely a tiež nelineárne modely, ktoré sú matematicky vyjadrené vo forme nelineárnych diferenciálnych rovníc. Vytvárame prepracovanejší nástroj na generovanie referenčnej trajektórie a celkovo pridávame do GUI aj niekoľko menších vylepšení.

PodĎakovanie

Tento článok bol vytvorený realizáciou vedeckého projektu Vega č. 1/0617/08 Grantovej agentúry SR pod názvom Multiagentové sieťové riadiace systémy s automatickou rekonfiguráciou (50%) a projektu Centrum informačných a komunikačných technológií pre znalostné systémy (kód ITMS projektu:26220120020) na základe podpory operačného programu Výskum a vývoj financovaného z Európskeho fondu regionálneho rozvoja (50%).

Použitá literatúra

- [1] THE MATHWORKS: Creating Graphical User Interfaces (User's guide). Dostupné na internete: http://www.mathworks.com/help/pdf_doc/matlab/buildgui.pdf
- [2] ROUBAL, Jiří: Prediktívni regulátor (Příklady ze cvičení – Moderní teorie řízení). Dostupné na internete: http://support.dce.felk.cvut.cz/pub/roubalj/teaching/MTR/seminars/MTR_cv8_mpc.pdf.
- [3] BELDA¹, Květoslav – BÖHM¹, Josef – VALÁŠEK², Michael: Model-Based Control for Parallel Robot Kinematics. Praha: ¹Ústav teorie informace a automatizace AV ČR, ²ČVUT – FS, 2004.
- [4] BELDA, Květoslav – BÖHM, Josef: Adaptive Predictive Control for Simple Mechatronic Systems. Praha: Ústav teorie informace a automatizace AV ČR, 2006.
- [5] FIKAR, Miroslav: Predictive Control – An Introduction. Bratislava: Slovenská technická univerzita - FCHPT, 1999.
- [6] JADLOVSKÁ, Anna – JAJČIŠIN, Štefan – LONŠČÁK, Richard: Modelling and PID Control Design of Nonlinear Educational Model Ball & Plate. Košice: TU-FEI. Medzinárodná konferencia Process Control, 2009.
- [7] JAJČIŠIN, Štefan: Modelovanie a riadenie dynamických systémov v prostredí Matlab/Simulink. Bakalárska práca. Košice: TU-FEI, 2008.
- [8] JAJČIŠIN, Štefan: Aplikácia moderných metód v riadení nelineárnych výukových modelov. Diplomová práca. Košice: TU-FEI, 2010.