

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE



**FAKULTA CHEMICKÉJ A POTRAVINÁRSKEJ TECHNOLOGIE
ÚSTAV INFORMATIZÁCIE, AUTOMATIZÁCIE A MATEMATIKY
ODDELENIE INFORMATIZÁCIE A RIADENIA PROCESOV**



Konvexná optimalizácia a prediktívne riadenie s modelom

BAKALÁRSKA PRÁCA

FCHPT-5415-50907

Bratislava 2010

Ján Drgoňa

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE



FAKULTA CHEMICKÉJ A POTRAVINÁRSKEJ TECHNOLOGIE
ÚSTAV INFORMATIZÁCIE, AUTOMATIZÁCIE A MATEMATIKY
ODDELENIE INFORMATIZÁCIE A RIADENIA PROCESOV



Konvexná optimalizácia a prediktívne riadenie s modelom

BAKALÁRSKA PRÁCA

FCHPT-5415-50907

Študijný program: Automatizácia, informatizácia a manažment v chémii a potravinárstve

Číslo a názov študijného odboru: 5.2.14 automatizácia, 5.2.52 priemyselné inžinierstvo

Školiace pracovisko: Fakulta chemickej a potravinárskej technológie

Vedúci záverečnej práce/školiteľ: Ing. Michal Kvasnica, PhD.

Bratislava 2010

Ján Drgoňa



ZADANIE BAKALÁRSKEJ PRÁCE

Študent: **Ján Drgoňa**
ID študenta: **50907**
Študijný program: **automatizácia, informatizácia a manažment v chémii a potravinárstve**
Kombinácia študijných odborov: **5.2.14 automatizácia, 5.2.52 priemyselné inžinierstvo**
Vedúci práce: **Ing. Michal Kvasnica, PhD.**

Názov práce: **Konvexná optimalizácia a prediktívne riadenie s modelom**

Špecifikácia zadania:

Prediktívne riadenie je spôsob riadenia, ktorý využíva model reálneho zariadenia na vytvorenie predikcií (čiže predpovedí) budúceho správania sa systému. Následne sa na základe predikcií pomocou optimalizácie určí taký spôsob riadenia procesu, aby bol napr. maximalizovaný zisk, minimalizované náklady a pod. Z tohto pohľadu je prediktívne riadenie najpokročilejšou a najviac využívanou technológiou pri riadení procesov všade tam, kde je úloha zisku na prvom mieste. Úlohou bakalárskeho projektu je zoznámenie sa s problematikou prediktívneho riadenia a formulovanie týchto úloh ako problémov konvexnej optimalizácie.

Rozsah práce: **30**

Riešenie zadania práce od: **15. 02. 2010**

Dátum odovzdania práce: **22. 05. 2010**

L. S.

Ján Drgoňa
Študent

prof. Ing. Miroslav Fikar, DrSc.
Vedúci pracoviska

prof. Ing. Miroslav Fikar, DrSc.
Garant študijného programu

Čestné prehlásenie:

Čestne prehlasujem, že som bakalársku prácu písal sám s použitím danej literatúry, ktorú uvádzam v mojej práci.

.....
Ján Drgoňa

Pod'akovanie

Ďakujem vedúcemu bakalárskej práce Ing. Michalovi Kvasnicovi, PhD. za vedenie, rady, pripomienky, poskytnuté vedomosti a za ochotu mi vždy pomôcť pri akomkoľvek probléme ktorý sa vyskytol pri vypracovaní bakalárskej práce.

Abstrakt

Cieľom tejto práce je oboznámiť sa problematikou konvexnej optimalizácie, jej základné vlastnosti a využitie v prediktívnom riadení. Ďalej sa práca zaoberá práve prediktívnym riadením s modelom, ktoré patrí v súčasnosti medzi veľmi populárne pokročilé metódy riadenia. Práca je zameraná hlavne na matematickú formuláciu prediktívneho riadenia ako konvexného optimalizačného problému. Nakoniec sú zadefinované niektoré problémy prediktívneho riadenia, ktoré sú riešené teoreticky, ale aj prakticky na konkrétnom príklade riadenia dvoch zásobníkov kvapaliny.

Kľúčové slová: konvexná optimalizácia, polytopy, ohraničenia, normy, matematický model, prediktívne riadenie s modelom, regulácia stavov, regulácia výstupov

Abstract

Aim of this thesis is present and analyze the topic of convex optimization, its basic properties and utilization in predictive control. I am also dealing with model predictive control, which is in present days a very popular part of advanced process control methods. The main focus is on the mathematical formulation of predictive control as a convex optimization problem. Finally we define some problems of model predictive control , which are solved theoretically and also practically on concrete example of two liquid tanks.

Key words: convex optimization, polytopes, constraints, norms, mathematical model, model predictive control, state regulation, output regulation

Obsah

1	ÚVOD	9
1.1	VÝVOJ PREDIKTÍVNEHO RIADENIA	9
1.2	ZÁKLADNÉ MYŠLIENKY	9
2	OPTIMALIZÁCIA	11
3	KONVEXNÁ OPTIMALIZÁCIA [1]	11
3.1	KONVEXNÁ MNOŽINA A SPOJITÝ PRIESTOR	12
3.2	NADROVINA A POLPRIESTOR	14
3.3	POLYTOPY	16
3.4	KONVEXNÉ FUNKCIE	17
3.4.1	<i>Normy</i>	<i>17</i>
3.5	KONVEXNÝ OPTIMALIZAČNÝ PROBLÉM	19
3.5.1	<i>Všeobecný konvexný optimalizačný problém</i>	<i>19</i>
3.5.2	<i>Lineárny konvexný optimalizačný problém</i>	<i>19</i>
3.5.3	<i>Kvadratický konvexný optimalizačný problém</i>	<i>21</i>
4	MATEMATICKÝ MODEL	22
4.1	SPOJITÝ MATEMATICKÝ MODEL	22
4.2	LINEARIZÁCIA	23
4.3	DISKRETIZÁCIA MATEMATICKÉHO MODELU	25
5	MATEMATICKÁ FORMULÁCIA PREDIKTÍVNEHO RIADENIA S MODELOM [4]	26
5.1	STAVOVÁ REGULÁCIA	26
5.1.1	<i>Rovnica predikcie [5]</i>	<i>28</i>
5.2	REGULÁCIA VÝSTUPOV	29
5.3	NENULOVÁ REFERENCIA	29
5.4	ODSTRÁNENIE TRVALEJ REGULAČNEJ ODCHÝLKY	31
5.4.1	<i>Riešenie pomocou výpočtu ustálených stavov</i>	<i>31</i>
5.4.2	<i>Riešenie pomocou penalizácie zmeny vstupov</i>	<i>36</i>
5.5	VPLYV VÁHOVÝCH MATÍC	39
5.6	SPOMALOVACIE OHRANIČENIA	40
5.7	ČASOVO MENIACA SA REFERENCIA	41
6	ZÁVER	46
7	LITERATÚRA	47

1 Úvod

1.1 Vývoj prediktívneho riadenia

Prvé koncepty prediktívneho riadenia vznikli už v sedemdesiatych rokoch dvadsiateho storočia. V tej dobe však išlo len o matematické teórie na papieri a praktické využitie bola zatiaľ len hudba budúcnosti. Z počiatku týmto pokročilým myšlienkam bránila v rozvoji hlavne slabá výkonnosť výpočtovej techniky, čo sa malo začiatkom deväťdesiatych rokov postupne zmeniť a do sveta prenikli prvé algoritmy využívané aj v praxi. Tieto prvé systémy riadené pomocou prediktívneho riadenia však museli byť pomalé a to vzhľadom na vysokú výpočtovú náročnosť prediktívneho riadenia. Avšak prudký rozvoj výpočtovej techniky v posledných desaťročiach takmer úplne zmazal tieto nedostatky a ponúkol priestor pre nové metódy prediktívneho riadenia, ktoré sa v súčasnosti využívajú aj pri procesoch s rýchlou dynamikou a rýchlo sa rozšírili tak, že prediktívne riadenie patrí v súčasnosti medzi najpokročilejšie a najviac využívané metódy riadenia.

Najvýznamnejšie úspechy boli dosiahnuté pri využití prediktívneho riadenia v ropných rafinériách a petrochemickom priemysle, veľmi dobré výsledky sa dosiahli taktiež v chemickom priemysle, metalurgii, energetike a robotike. [5]

Súčasný trend rozvoja prediktívneho riadenia, len potvrdzuje jeho dominantnú úlohu na trhu, vďaka jeho schopnostiam optimálneho riadenia pri rešpektovaní daných ohraničení, čo zaručuje veľkú ekonomickú rentabilitu riadených procesov. Práve preto sa dnes prediktívne riadenie využíva všade tam, kde je otázka peňazí na prvom mieste a úlohou je minimalizovať náklady, respektíve maximalizovať zisk.

1.2 Základné myšlienky

Základným stavebným kameňom prediktívneho riadenia je presný matematický model riadeného systému, na základe ktorého sme schopný pri známych vstupoch predpovedať budúce stavy systému. Čiže ak dobre poznám vlastnosti a správanie sa systému, viem v akom počítačnom stave sa nachádza a aké akčné zásahy naňho pôsobia, viem predpovedať ako na ne systém zareaguje. Tieto budúce stavy sú predpovedané pre istý časový interval ktorý sa nazýva horizont predikcie, ide o dopredu určený počet krokov, pre ktoré sa v každom kroku vypočíta stav systému, berúc do úvahy už predošlé predikcie stavov a vstupov. Keďže, horizont predikcie je postupnosť určitých krokov do budúcnosti, tak musíme vedieť

určiť veľkosť týchto krokov v jednotkách času a tým je perióda vzorkovania. Horizont predikcie nám určuje koľko krokov do budúcnosti spravíme a perióda vzorkovania nám hovorí aké veľké tie kroky budú, napríklad sekunda, dve sekundy, minúta... Takže pomocou horizontu predikcie a periódy vzorkovania sme schopní povedať, ako ďaleko do budúcnosti môžeme vidieť cez dané predpovede.

Pre efektívne a bezpečné riadenie sa v praxi používa takzvaná stratégia pohyblivého horizontu, keď sa na riadenie používa len prvý člen postupnosti a následne sa určí nová postupnosť, z ktorej sa v riadení použije taktiež prvý člen. Pri samotnej predikcii sa už jedná o optimalizáciu daných riadiacich vstupov, respektíve výstupov na základe známych ohraničení a požadovaných hodnotách výstupov, respektíve vstupov, čo môžeme inak nazvať aj trajektóriou riadenia. Práve vďaka tomu potom pri prediktívnom riadení hovoríme o optimálnej postupnosti krokov riadenia, ktoré sa dokážu prispôbiť zmenám ohraničení, alebo zmene referenčnej trajektórie riadenia. Týmto sa dostávame k ďalšiemu základnému pilieru prediktívneho riadenia, a to k optimalizácii.

Optimalizačný algoritmus pomocou účelovej funkcie, a známych ohraničeníach vyberie z množiny možných riešení to najlepšie možné riešenie a tým sa zabezpečí optimálny akčný zásah. Ďalšou hlavnou výhodou prediktívneho riadenia je jeho schopnosť aktívne pracovať s ohraničeniami vstupných, výstupných ako aj stavových veličín a to práve vďaka optimalizácii. Preto ide problematika optimalizácie ruka v ruke s problematikou prediktívneho riadenia.

2 Optimalizácia

Odpoveď na otázku, prečo je úloha optimalizácie v ľudskej spoločnosti, taká dôležitá je zrejmá už na prvý pohľad. Každá ľudská činnosť je spojená s optimalizáciou, či už vedomou, alebo podvedomou. Minimalizácia vstupných nákladov, efektívnejšie využívanie prírodných zdrojov, alebo maximalizácia zisku sú hlavnými kritériami pre všetky priemyselné, ekonomické a technologické procesy v súčasnosti. Avšak optimalizáciu môžeme chápať aj inak, predstavme si že sa chceme dostať z bodu A, kde práve stojíme do vzdialenejšieho bodu B. Ak nám v ceste nestojí žiadna prekážka, tak túto vzdialenosť prejdeme po najkratšej možnej ceste, po priamke AB, nebudeme robiť kľučky, ani pred tým nenavštívime bod C ktorý neleží na spojnici bodov A a B. Tým sme vykonali optimalizáciu našej činnosti vzhľadom na vynaloženú vstupnú energiu a čas za ktorý prejdeme danú vzdialenosť, pretože sme vybrali najlepšiu možnosť z nekonečného množstva trajektórií, ktoré sme si mohli zvoliť.

Optimalizácia je teda vo všeobecnosti, nájdenie najlepšieho možného (optimálneho) riešenia vzhľadom na dané podmienky a požiadavky. Slová ako optimálnejší alebo najoptimálnejší nedávajú zmysel, pretože lepší stav ako optimálny už pri daných podmienkach nemôže existovať.

V matematike sa pod pojmom optimalizácie rozumie hľadanie extrému cieľovej, alebo účelovej funkcie. Množina, na ktorej sa stanovuje extrém je množina prístupných riešení a je zvyčajne opísaná sústavou rovníc, respektíve nerovníc ktoré sú obmedzeniami pre danú množinu. Aby mala optimalizácia zmysel, musí mať účelová (objektívna, kritériálna) funkcia viac ako jedno riešenie.

3 Konvexná optimalizácia [1]

Využitie konvexnej optimalizácie v prediktívnom riadení, nie je náhodné, ale vyplýva zo základných vlastností konvexnej optimalizácie, ktoré sú pre potreby prediktívneho riadenia, veľmi výhodné. A to hlavne preto, že v praxi často ohraničenia riadenia, vytvárajú konvexné množiny, alebo množiny, ktoré je možné upraviť tak, aby boli konvexné. Čo znamená, že akýkoľvek problém prediktívneho riadenia sme schopný definovať ako konvexný optimalizačný problém. Jednou z najväčších výhod konvexnej optimalizácie je to, že ak nájdeme lokálny extrém konvexnej funkcie, tak je zároveň aj globálnym extrémom. Konvexné optimalizačné problémy sú preto efektívne riešiteľné pomocou známych

numerických metód a taktiež prirodzene vyplývajú z inžinierskych problémov, akými sú minimalizácia vstupnej energie a maximalizácia zisku.

3.1 Konvexná množina a Spojitý priestor

Konvexná množina tvorí jednu zo základných definícií konvexnej optimalizácie a preto je veľmi dôležité aby sme ju spomenuli. Vo všeobecnosti môžeme chápať konvexnú množinu, ako množinu bodov, pre ktoré musí platiť, že pre ľubovoľné dva body v tejto množine môžeme zostrojiť takú spojnicu týchto bodov, že všetky body ležiace na tejto spojnici budú patriť do konvexnej množiny. Čiže daná množina musí tvoriť spojitý priestor.

Uvedme si príklad, predpokladajme, že máme dva body $x_1, x_2 \in R^n$ a platí že $x_1 \neq x_2$. Pre definíciu spojitého priestoru musí platiť nasledujúca rovnica, ktorá opisuje všetky body y ležiace na priamke prechádzajúcej bodmi x_1 a x_2 :

$$y = \theta x_1 + (1 - \theta)x_2 \quad (1)$$

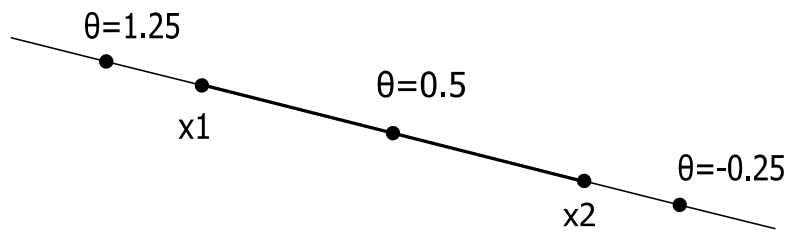
Parameter $\theta \in R$ pre konvexnú množinu nadobúda hodnoty od 0 po 1, čo zodpovedá uzatvorenému sektoru priamky od bodu x_1 po bod x_2 . Ak $\theta=1$, tak $y=x_1$ a ak $\theta=0$, tak $y=x_2$.

Vyjadrením rovnice vo forme:

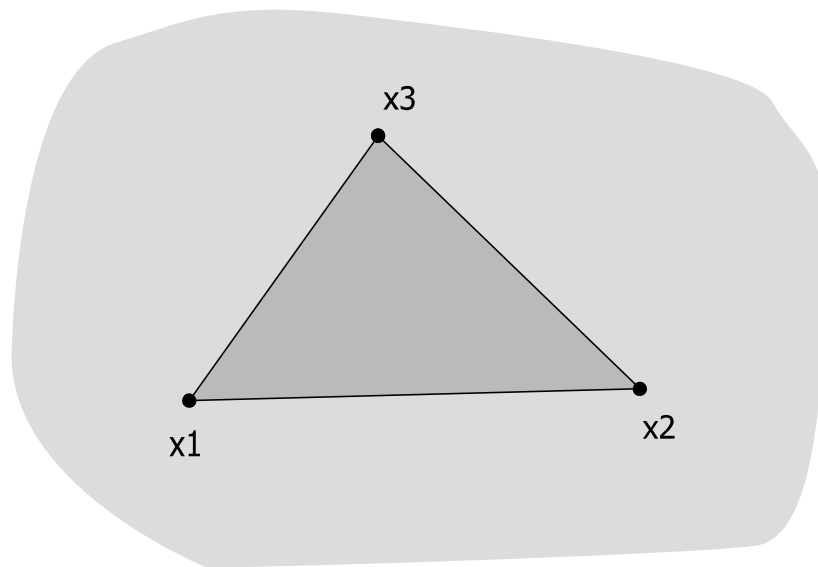
$$y = x_2 + \theta(x_1 - x_2) \quad (2)$$

Získame novú interpretáciu a to, že bod y je súčtom základného bodu x_2 a vzdialenosti medzi bodmi x_2 a x_1 vynásobenú parametrom θ . Z toho vyplýva že parameter θ udáva podiel vzdialenosti od x_1 po x_2 kde leží bod y .

Ako meníme hodnoty parametra θ od 0 po 1, tak sa mení aj poloha bodu y od x_2 po x_1 , keď prekročíme hodnotu parametra $\theta > 1$ tak bod y leží za hraničným bodom x_1 , a naopak ak $\theta < 0$ tak bod y leží za hraničným bodom x_2 . Ak teda nemáme ohraničenia pre parameter θ , ten môže nadobúdať ľubovoľné hodnoty z množiny reálnych čísel, vtedy hovoríme o afinnej množine. Afinnú množinu si môžeme geometricky predstaviť v jednom rozmere, ako priamku, ktorá prechádza danými dvoma bodmi, konvexnú množinu si môžeme predstaviť ako úsečku ohraničenú týmito bodmi. Dvojrozmerné by to bola pre afinnú množinu neohraničená plocha, určená tromi nezávislými bodmi a pre konvexnú množinu plocha ohraničená práve týmito tromi bodmi.



Obr.1: Interpretácia spojitej množiny v jednorozmernom priestore. O priamke určenej bodmi x_1 a x_2 môžeme hovoriť ako o afinnej množine, o úsečke x_1, x_2 hovoríme ako o konvexnej množine.



Obr.2: Interpretácia spojitej množiny v dvojrozmernom priestore, trojuholník vytvorený bodmi x_1, x_2 a x_3 predstavuje konvexnú množinu a neohraničená plocha určená týmito bodmi predstavuje afinnú množinu.

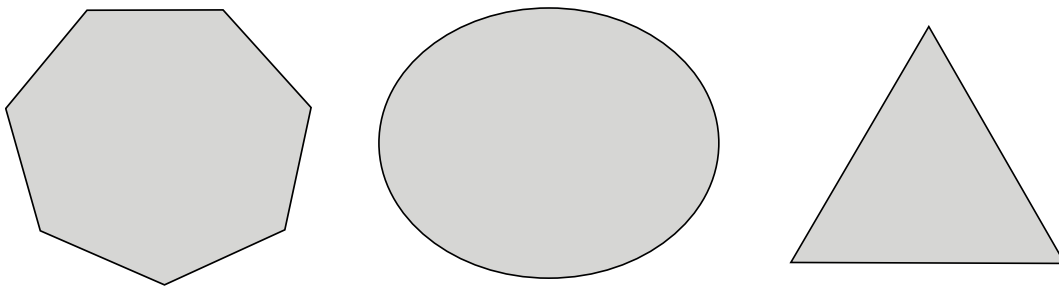
Na základe predošlej definície dvoch spojitých bodov, môžeme definovať spojitý priestor ako množinu všetkých bodov $C \subseteq \mathbb{R}^n$ pre ktorú platí, že úsečka prechádzajúca určená dvomi rozdielnymi bodmi patriacimi do množiny C , musí patriť do množiny C .

Inými slovami, spojitá množina C obsahuje lineárnu kombináciu hocijakých dvoch bodov patriacich množine C , za predpokladu, že súčet koeficientov v danej lineárnej kombinácii je rovný jednej a všetky koeficienty sú nezáporné reálne čísla, čo môžeme matematicky vyjadriť nasledovne:

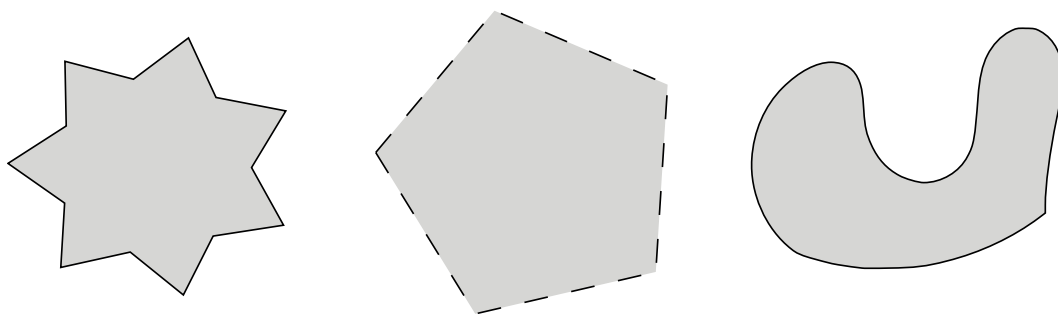
Predpokladajme že máme body $x_1, \dots, x_k \in C$, a súčet koeficientov $\theta_1 + \dots + \theta_k = 1$, potom bod $y = \theta_1 x_1 + \dots + \theta_k x_k$ taktiež patrí do množiny C . Súbor všetkých takýchto lineárnych kombinácií sa potom nazýva afinná množina.

Matematické vyjadrenie konvexnej množiny, priamo vyplýva z matematickej formulácie afinnej množiny a líši sa len ohraňčením hodnoty parametra $\theta_i \geq 0, i = 1, \dots, k$. Všeobecný zápis konvexnej množiny C má potom tvar:

$$\text{conv}C = \{\theta_1 x_1 + \dots + \theta_k x_k \mid x_i \in C, \theta_i \geq 0, i = 1, \dots, k, \theta_1 + \dots + \theta_k = 1\} \quad (3)$$



Obr.3: Príklady dvojrozmerných konvexných útvarov



Obr.4: Príklady dvojrozmerných nekonvexných útvarov

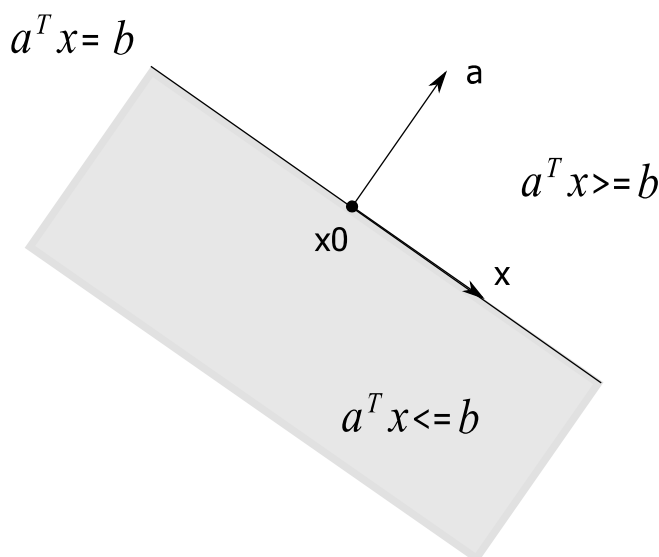
3.2 Nadrovina a polpriestor

Geometrické predstavy Nadrovín a polpriestorov nám značne pomáhajú pri formulácii konvexných optimalizačných problémov. Nadroviny aj polpriestory sú konvexné útvary. Môžeme ich chápať aj ako dva základné typy ohraňčenia, a to ohraňčenia v tvare rovnosti, ktoré si môžeme predstaviť v priestore ako nadroviny a ohraňčenia v tvare nerovnosti, ktoré môžeme chápať ako akési polpriestory.

Predstavme si teda že máme n -rozmerný priestor a jedno ohraňčenie v tvare rovnosti, potom v tomto priestore nám toto ohraňčenie vytvorí akúsi nadrovinu, na ktorej sa budú nachádzať

všetky možné riešenia daného problému. Teraz si naše ohraničenie v tvare rovnosti rozšírime na ohraničenie v tvare nerovnosti, potom náš n -rozmerný priestor bude rozdelený na dve časti, na jeden polpriestor kde sa riešenia môžu nachádzať a druhý kde sa riešenia nachádzať nemôžu.

Napríklad v prípade dvojrozmerného priestoru by bola nadrovina priamkou pretínajúcou daný dvojrozmerný priestor a polpriestor by bola polrovina určená danou priamkou.



Obr.5: Na obrázku môžeme vidieť v dvojrozmernom priestore zobrazenú nadrovinu, ktorá je definovaná prostredníctvom príslušného ohraničenia v tvare rovnosti, táto nadrovina pretína priestor na dva polpriestory, pre ktoré zodpovedajú príslušné ohraničenia v tvare nerovnosti.

Matematicky je nadrovina množina všetkých bodov x , pre ktoré platí: $a^T x = b$, pričom $a \in R^n$, $a \neq 0$, a $b \in R$.

Podobne pre polpriestor platí, že je to množina všetkých bodov x , pre ktoré platí: $a^T x \leq b$, pričom $a \in R^n$, $a \neq 0$, a $b \in R$.

Vektor a je normálovým vektorom, vektor b určuje posunutie od počiatočného bodu x_0 , vektor x je vektorom premenných.

Pre lepšie geometrické chápanie môžeme upraviť predošlé rovnice a nadrovinu vyjadriť aj ako množinu $\{x \mid a^T (x - x_0) = 0\}$ a polpriestor ako $\{x \mid a^T (x - x_0) \leq 0\}$, kde bod x_0 je začiatkový bod sústavy a uspokojuje podmienku $a^T x_0 = b$.

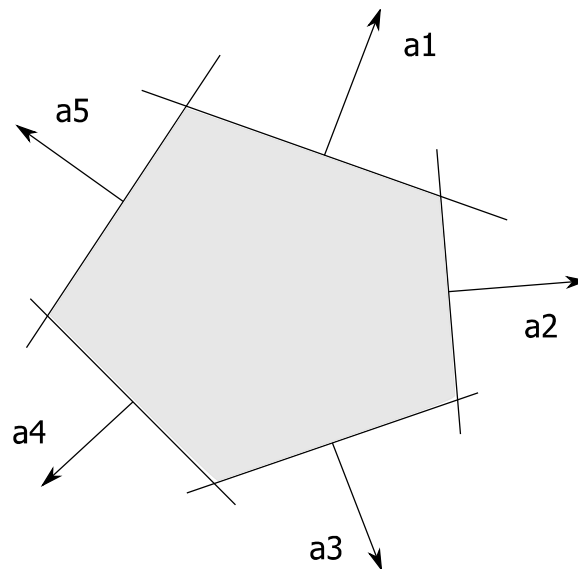
3.3 Polytopy

Polytop alebo aj polyhedrón je vo všeobecnosti definovaný ako prienik konečného počtu lineárnych ohraňení v tvare rovnosti a nerovnosti. A keďže polroviny aj polpriestory sú útvary konvexné, tak aj výsledný polytopický útvar vzniknutý z ich prienikov musí byť konvexný, preto v konvexnej optimalizácii tvorí základnú množinu možných riešení daného problému práve polytop.

V polytope vždy platí, že nájdenie lokálneho extrému znamená automaticky nájdenie aj globálneho extrému, vďaka konvexnosti daného útvaru, preto je jeho použitie v optimalizácii veľmi výhodné vzhľadom na zložitosť riešenia a výpočtového času optimalizácie. Ďalšou výhodou polytopu je že sa skladá z lineárnych ohraňení, ktoré sú dostupnými metódami ľahko a efektívne riešiteľné, na rozdiel od ohraňení nelineárnych.

Matematicky môžeme polytop zapísať pomocou množiny $P = \{x \mid Ax \leq B, Cx = D\}$, kde:

$$A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix}, \quad B = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}, \quad C = \begin{bmatrix} c_1^T \\ \vdots \\ c_n^T \end{bmatrix} \quad \text{a} \quad D = \begin{bmatrix} d_1 \\ \vdots \\ d_n \end{bmatrix} \quad (4)$$



Obr.6: Na tomto obrázku môžeme vidieť polytopický útvar, vytvorený pomocou piatich lineárnych ohraňení, kde a_1 až a_5 sú smerové vektory nadrovin prislúchajúcich k daným ohraňeniam.

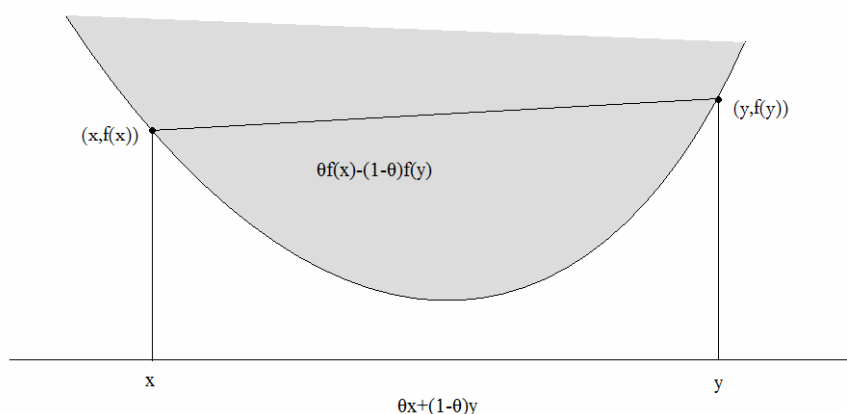
3.4 Konvexné funkcie

Konvexný tvar účelovej funkcie je v konvexnej optimalizácii veľmi dôležitý, práve preto, že nájdenie lokálneho extrému takejto funkcie, znamená, že sme našli aj extrém globálny. To znamená že zakaždým sme schopní nájsť, najlepšie možné riešenie a nemôže sa nám stať, že uviazneme v lokálnom extréme, čo by v praxi mohlo znamenať nesprávny výpočet akčných zásahov a tým aj nefunkčnosť celého riadiaceho systému.

Matematicky môžeme konvexnú funkciu f definovať tak, že pre všetky x, y patriace do definičného oboru funkcie f a pre parameter θ , $0 \leq \theta \leq 1$, platí rovnica:

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y) \quad (5)$$

Geometricky, to znamená že úsečka, medzi bodmi $(x, f(x))$ a $(y, f(y))$, leží nad grafom funkcie f .



Obr.7: konvexná funkcia

3.4.1 Normy

Normy sú konvexné funkcie, ktoré môžeme vo všeobecnosti chápať ako vyjadrenia vzdialenosti, respektíve dĺžky vektorov a spomíname ich práve preto, že ich využívame pri formulovaní konvexných optimalizačných problémov. Normy udávajú vzdialenosť bodu x od iného referenčného bodu y , ak sú body x a y totožné, jedná sa o takzvaný nulový vektor a jeho veľkosť, čiže norma je rovná nule.

Určime si referenčný vektor y rovný nule, parameter p je väčší alebo rovný ako jedna a patrí do oboru reálnych čísel, potom matematicky môžeme normu pre vektor $x = [x_1, x_2, \dots, x_n]^T$ všeobecne definovať podľa nasledujúcej rovnice:

$$\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p} \quad (6)$$

Keď položíme parameter $p=2$, získame takzvanú Euklidovskú normu, alebo tiež dva normu. Táto norma udáva najkratšiu možnú vzdialenosť bodu x od referenčného stavu a priamo vyplýva z Pytagorovej vety, preto sa s ňou môžeme často stretnúť pri formulovaní optimalizačných problémov.

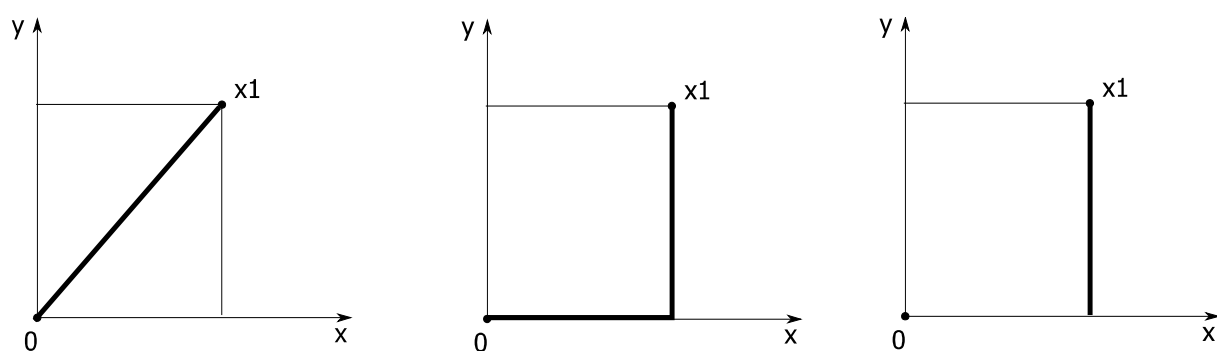
$$\|x\|_2 = \left(\sum_{i=1}^n |x_i|^2 \right)^{1/2} \quad (7)$$

Ďalšou normu, ktorú si predstavíme je jedna norma, kde $p=1$ a ide o súčet veľkostí jednotlivých súradníc vektora x .

$$\|x\|_1 = \sum_{i=1}^n |x_i| \quad (8)$$

A nakoniec tu máme nekonečno normu, ktorú môžeme chápať ako dĺžku najväčšej súradnice z celého vektora x .

$$\|x\|_\infty = \max |x_i| \quad (9)$$



Obr.8: Na obrázku máme znázornenú vzdialenosť bodu x_1 od nuly v troch rôznych normách. Vľavo ide o dva normu, v strede máme jedna normu a napravo nekonečno normu.

3.5 Konvexný optimalizačný problém

3.5.1 Všeobecný konvexný optimalizačný problém

Štandardne môžeme všeobecný konvexný optimalizačný problém zapísať nasledovne:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \dots, m \\ & g_j(x) = 0, \quad j = 1, \dots, n \end{array} \quad (10)$$

Kde x je optimalizovaná premenná, f_0 je účelová funkcia, $f_i(x) \leq 0$, sú ohraničenia v tvare nerovnosti a $g_j(x) = 0$ sú ohraničenia v tvare rovnosti.

Tento zápis môžeme potom chápať, ako problém hľadania takého x , pre ktoré je hodnota funkcie $f_0(x)$ minimálna a zároveň spĺňa dané ohraničenia. Zápis pomocou anglických výrazov minimize (minimalizovať) a subject to (vzhľadom na) je štandardným zápisom používajúcim sa pri definovaní optimalizačných problémov.

3.5.2 Lineárny konvexný optimalizačný problém

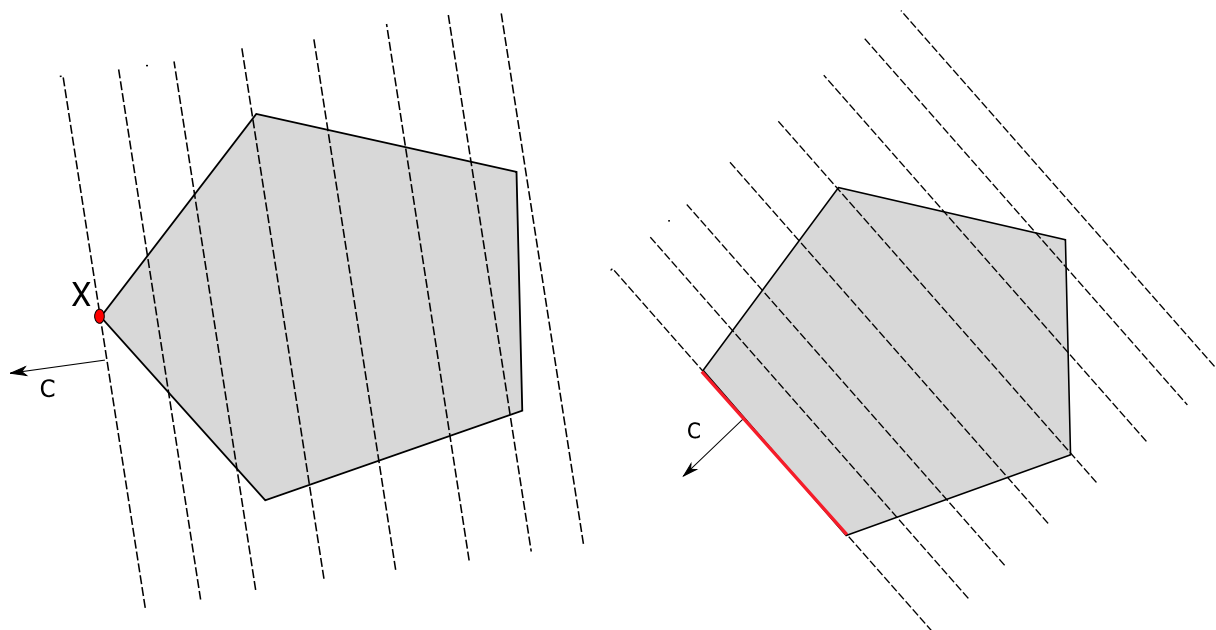
O lineárnom konvexnom optimalizačnom probléme hovoríme vtedy, ak účelová funkcia, aj funkcie ohraničení sú lineárne funkcie.

Všeobecný zápis lineárneho konvexného optimalizačného problému je:

$$\begin{array}{ll} \text{minimize} & c^T x + d \\ \text{subject to} & Gx \leq h \\ & Ax = b \end{array} \quad (11)$$

Vektor x je vektorom hodnôt optimalizovanej premennej, vektor c je vektorom konštánt účelovej funkcie a konštanta d vyjadruje posunutie účelovej funkcie o jej hodnotu, často sa môžeme stretnúť aj so zápisom, kde sa vynecháva konštanta d , pokiaľ nemá vplyv na riešenie optimalizačného problému. Matica G je matica hodnôt obmedzení v tvare nerovnosti a matica A je matica hodnôt obmedzení v tvare rovnosti. A nakoniec vektory h a b tvoria hranice ohraničení v tvare nerovnosti a rovnosti.

Nakoľko máme v tomto prípade lineárne ohraničenia, výslednou množinou možných riešení bude polytop. Geometricky si potom môžeme účelovú funkciu v dvojrozmernom priestore predstaviť ako priamku a hľadanie minima, ako posúvanie tejto priamky vo vnútri polytopu v smere vektora C dovtedy, pokiaľ nenarazíme na okraj polytopu. Z čoho vyplýva že riešenie optimalizačného problému sa vždy nachádza na okrajových ohraničení polytopu a nikdy nie vo vnútri polytopu. Hlavným nedostatkom lineárneho programovania je fakt, že môže nastať situácia keď sa účelová funkcia pri optimalizácii prekryje s jedným z lineárnych ohraničení polytopu, čo znamená že potom nemôžeme jednoznačne určiť optimum.



Obr.9: Geometrická ilustrácia lineárneho programovania, polytop predstavujúci množinu možných riešení je na obrázku šedý útvar, prerušované čiary predstavujú účelovú funkciu, vektor c je smerovým vektorom účelovej funkcie. Na obrázku vľavo červený bod x znázorňuje nájdenie optimálneho riešenia vo vrchole polytopu. Obrázok napravo ilustruje situáciu, keď nemôžeme jednoznačne určiť optimálne riešenie, pretože účelová funkcia sa prekryva s jedným z ohraničení.

3.5.3 Kvadratický konvexný optimalizačný problém

Ak má účelová funkcia tvar kvadratickej funkcie, a ohraničenia majú lineárny tvar, hovoríme o konvexnom kvadratickom optimalizačnom probléme, alebo tiež o kvadratickom programovaní.

Matematicky môžeme zapísať problém kvadratického programovania nasledovne:

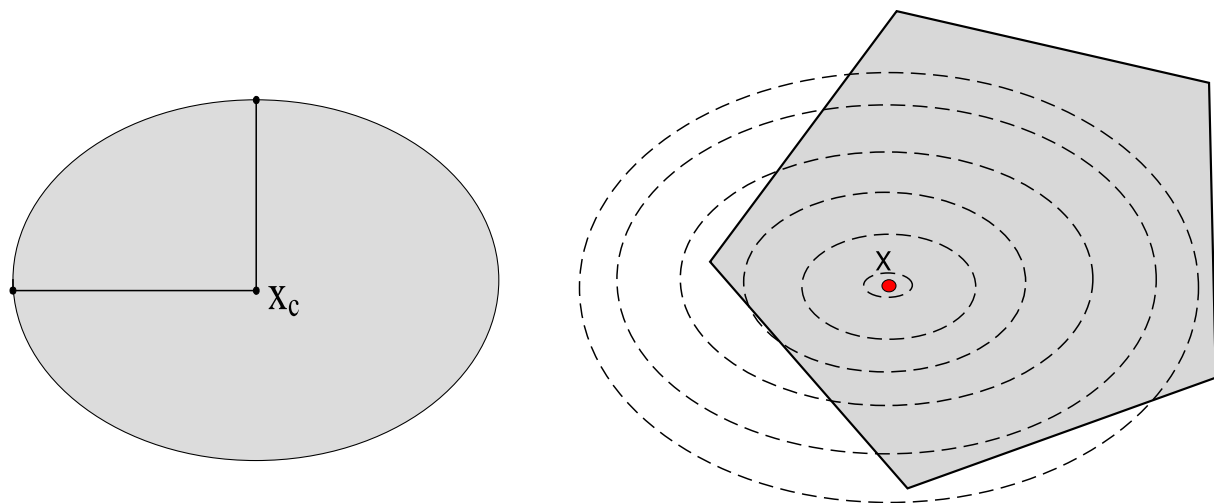
$$\begin{array}{ll} \text{minimize} & x^T P x + 2Q^T x + R \\ \text{subject to} & Gx \leq h \\ & Ax = b \end{array} \quad (12)$$

Geometricky môžeme kvadratickú funkciu chápať ako elipsu, definovanú ako množinu bodov, pre ktoré platí:

$$\mathcal{E} = \left\{ x \mid (x - x_c)^T P^{-1} (x - x_c) \leq 1 \right\}, \quad (13)$$

kde vektor x_c vyjadruje stred elipsy, matica P vyjadruje sploštenie elipsy, respektíve rozdiel v dĺžke dvoch hlavných polosí.

Matematickou úpravou definície elipsy, potom získavame taký tvar kvadratickej funkcie, aký máme definovaný v účelovej funkcii, kde matica P vyjadruje sploštenie a matice Q a R reprezentujú posunutie elipsy v priestore. Vo všeobecnosti potom minimalizujeme účelovú funkciu, v tvare elipsy cez lineárne ohraničenia tvoriace polytopický útvar. Hľadaný extrém funkcie, sa potom môže nachádzať v celom polytope, nielen na jeho okrajoch ako tomu bolo pri lineárnom programovaní. Hlavnou výhodou použitia kvadratickej účelovej funkcie oproti lineárnej, je fakt, že nájdený extrém je unikátny a nemôže sa nám stať že nájdeme extrémov viacero, ako pri lineárnom programovaní.



Obr.10: Naľavo je znázornená elipsa o strede x_c a jej dve hlavné polosi.

Napravo máme geometrickú ilustráciu kvadratického programovania, kde šedý útvar znamená množinu možných riešení, prerušované čiary znázorňujú účelovú funkciu a červený bod x znamená optimálne riešenie problému.

4 Matematický model

4.1 Spojitý matematický model

Existencia dobrého matematického modelu je v prediktívnom riadení základným stavebným kameňom celého riadenia. Pretože práve prostredníctvom matematického vyjadrenia správania sa systému dokážeme na základe známych vstupov predpovedať budúce výstupy, čo je hlavnou výhodou prediktívneho riadenia od ostatných metód riadenia.

Matematické modely môžu mať viacero podôb, ako sú lineárne či nelineárne modely statické alebo dynamické modely a tak ďalej. V našom prípade sa budeme zaoberať dynamickým lineárnym stavovým modelom.

Stavovými modelmi, môžeme nazvať také matematické modely fyzikálneho systému, ktoré sú definované pomocou súboru vstupných výstupných a stavových premenných. Dynamickými preto, že uvažujeme o systémoch meniacich sa v čase, takéto modely sú zvyčajne zapisované pomocou diferenciálnych rovníc. A lineárne modely použijeme preto, že sú jednoduchšie ako nelineárne a tým nám poskytujú, pri menšej strate presnosti veľkú úsporu výpočtového času potrebného na výpočet optimalizácie, z toho dôvodu sú najčastejšie využívané v praxi.

Takýto časovo spojitý model môžeme matematicky vo všeobecnosti zapísať nasledovne:

$$\begin{aligned} \dot{x}(t) &= A(t)x(t) + B(t)u(t) \\ y(t) &= C(t)x(t) + D(t)u(t) \end{aligned} \quad (14)$$

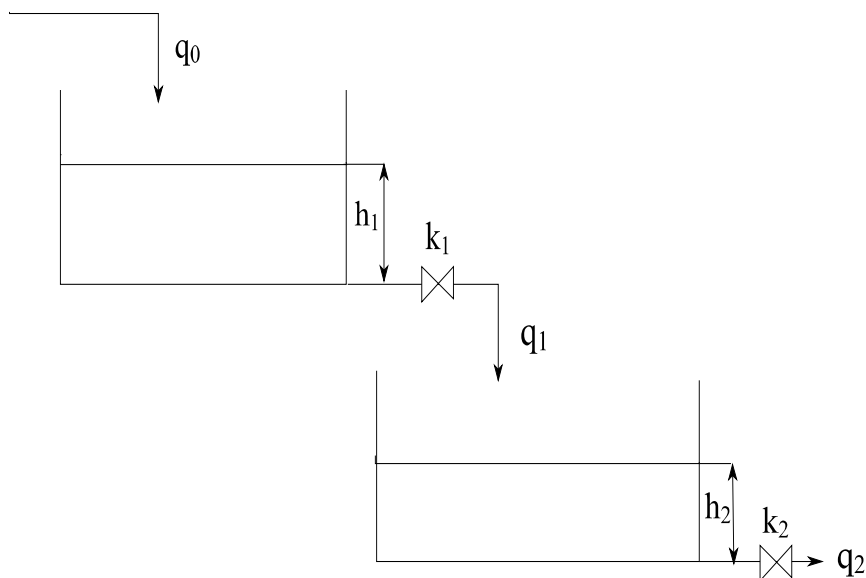
Kde premenná $x(t)$ je stavovým vektorom, premenná $y(t)$ je vektorom výstupov, $u(t)$ je vektorom vstupov a premenná $\dot{x}(t)$ vyjadruje vektor budúcich stavov systému. Matica A je stavová matica, B je matica vstupov, C je matica výstupov a D je matica vstupov pôsobiaca na výstupy.

Ak sa takýto matematický model nemení v čase tak ho nazývame časovo nemenným spojitým systémom a jeho stavové, vstupové a výstupové matice sú konštantné.

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (15)$$

4.2 Linearizácia

Lineárne modely získavame pomocou linearizácie nelineárnych modelov. Uvedieme si príklad nelineárneho dynamického matematického modelu dvoch zásobníkov kvapaliny a následnej linearizácie pomocou Taylorovho rozvoja.



Obr.11: Schéma dvoch zásobníkov kvapaliny bez interakcie

Nelineárny dynamický matematický model:

$$\begin{aligned} F \frac{dh_1}{dt} &= q - k\sqrt{h_1} \\ F \frac{dh_2}{dt} &= k\sqrt{h_1} - k\sqrt{h_2} \end{aligned} \quad (16)$$

Kde F je prierezová plocha zásobníkov kvapaliny, q je vstupný prietok, h_1 a h_2 sú výšky hladín v prvom a druhom zásobníku kvapaliny a k je prietoková konštanta zásobníkov.

Model ustáleného stavu:

$$\begin{aligned} k\sqrt{h_1^s} &= q \\ k\sqrt{h_2^s} &= k\sqrt{h_1^s} \end{aligned} \quad (17)$$

Linearizáciou Taylorovým rozvojom získame nasledovný tvar lineárneho matematického modelu:

$$\begin{aligned} F \frac{dx_1}{dt} &= u - \frac{k}{2\sqrt{h_1^s}} x_1 & x_1 &= h_1 - h_1^s \\ & & x_2 &= h_2 - h_2^s \\ F \frac{dx_2}{dt} &= \frac{k}{2\sqrt{h_1^s}} x_1 - \frac{k}{2\sqrt{h_2^s}} x_2 & u &= q - q^s \end{aligned} \quad (18)$$

Zjednodušené môžeme tento stavový opis zapísať nasledovne:

$$\begin{aligned} x' &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad x' = \begin{bmatrix} x_1' \\ x_2' \end{bmatrix}, \quad y = x_2$$

$$A = \begin{pmatrix} \frac{-k}{2F\sqrt{h_1^s}} & 0 \\ \frac{k}{2F\sqrt{h_1^s}} & \frac{-k}{2F\sqrt{h_2^s}} \end{pmatrix} \quad B = \begin{pmatrix} 1/F \\ 0 \end{pmatrix}, \quad C = (0 \quad 1), \quad D = 0 \quad (19)$$

4.3 Diskretizácia matematického modelu

Nakoľko riadiace systémy sú implementované v praxi pomocou modernej výpočtovej techniky, ktorá dokáže pracovať len s diskrétnymi dátami, všetky spojité stavové systémy použité v týchto riadiacich systémoch musíme najprv diskretizovať. Diskretizácia znamená prechod zo spojitých stavov k diskrétnym, v matematickom poňatí ide o nahradenie derivácie, zmenou Δ .

$$\frac{dx}{dt} \approx \frac{x(t + \Delta t) - x(t)}{\Delta t} \quad (20)$$

V diskrétnom systéme na rozdiel od spojitého systému sa všetky premenné menia skokovo v časových intervaloch nazývaných aj perióda vzorkovania, na každom takomto intervale je hodnota danej premennej konštantná, až dovtedy, pokiaľ sa zas skokovo nezmení na inú hodnotu.

Diskrétny matematický model v základnom tvare, kde Δt je perióda vzorkovania.

$$\begin{aligned} x(t + \Delta t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (21)$$

Zjednodušený zápis:

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t \\ y_t &= Cx_t + Du_t \end{aligned} \quad (22)$$

Nakoľko od správneho výberu periódy vzorkovania závisí aj presnosť nového diskretizovaného modelu, tak tento výber nie je náhodný. Z pravidla sa určuje z nasledovného intervalu:

$$\Delta t \in \left(\frac{T_{90}}{5}, \frac{T_{90}}{15} \right) \quad (23)$$

Označenie T_{90} predstavuje čas v ktorom hodnoty výstupov zodpovedajú 90% z maximálnej hodnoty výstupov. Inak povedané, perióda vzorkovania sa určuje tak, aby spravila aspoň 5 vzoriek za časovú konštantu T . [4]

5 Matematická formulácia prediktívneho riadenia s modelom [4]

5.1 Stavová regulácia

Úlohu prediktívneho riadenia vo všeobecnosti zapisujeme ako konvexný optimalizačný problém. Princípu prediktívneho riadenia najlepšie porozumieme pri vysvetlení nasledujúceho matematického zápisu.

$$\begin{aligned} \min \quad & \sum_{k=0}^{N-1} \left(\|Q_x x_{t+k}\|_p + \|Q_u u_{t+k}\|_p \right) \\ \text{s.t.} \quad & x_{t+1} = Ax_t + Bu_t \\ & x_t = x(t) \\ & x_t \in \bar{X} \\ & u_t \in \bar{U} \end{aligned} \tag{24}$$

Hlavným cieľom optimalizácie je minimalizovať účelovú funkciu, ktorá je vyjadrená pomocou noriem, predstavujúcich vzdialenosť predpovedaných stavov x_{t+k}

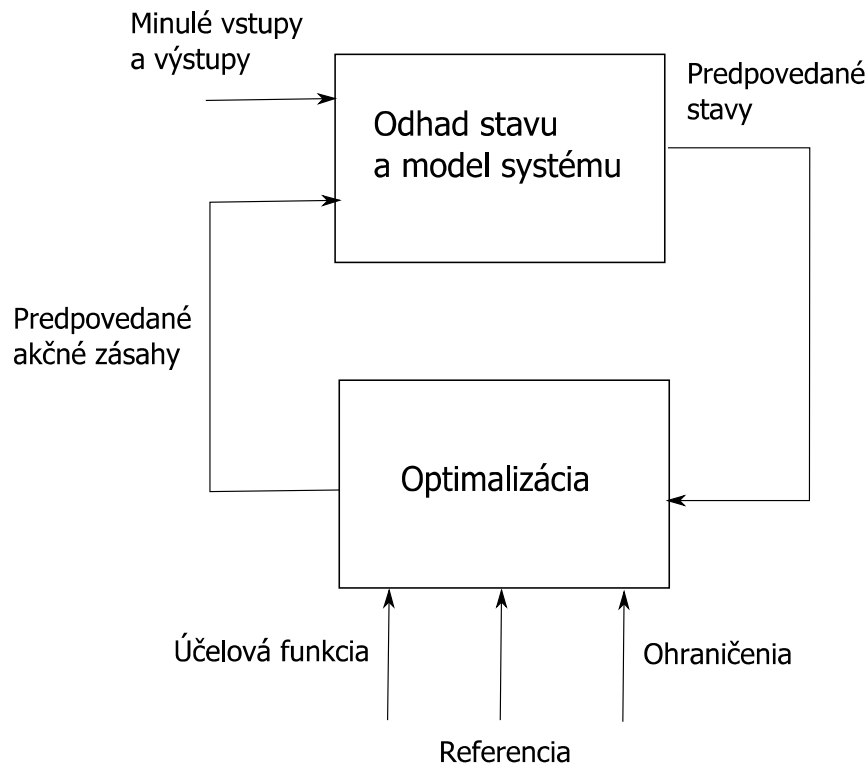
a predpovedaných vstupov u_{t+k} od nulového referenčného bodu. V tomto prípade ide teda o stavovú reguláciu, pretože jej účelom je priviesť stavy systému do referenčného bodu.

Konštanta N nám určuje horizont predikcie, čiže počet predikčných krokov, parameter p nám definuje stupeň normy a Q_x , Q_u sú váhovými maticami stavov a vstupov. Váhové matice nám určujú cenu daných stavov a vstupov pre optimalizáciu, významu váhových matic a ich dopadu na riadenie sa v tejto práci neskôr ešte budeme venovať.

V ohraničeniach nám rovnica $x_{t+1} = Ax_t + Bu_t$ reprezentuje stavový model systému,

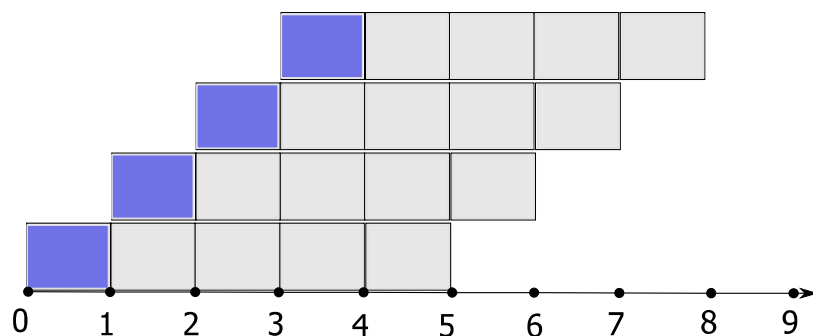
$x_t = x(t)$ predstavuje počiatočný stav v ktorom sa daný systém nachádza, $x_t \in \bar{X}$ nám predstavuje ohraničenia stavov a $u_t \in \bar{U}$ ohraničenia vstupov.

Výsledkom je snaha dostať budúce stavy na nulovú referenciu pri použití minimálnej energie, pri akceptovaní daných ohraničení.



Obr.12: Základná štruktúra prediktívneho riadenia s modelom [2]

Celý algoritmus prediktívneho riadenia beží v uzavretej slučke, ako reálny akčný zásah použijeme len prvý člen z celého horizontu predikcie, po aplikácii akčného zásahu v riadenom systéme, získame nový počiatočný stav systému, potom sa pomocou stavového modelu spravia predpovede pre budúce stavy a optimalizácia vypočíta ďalšiu postupnosť optimálnych akčných zásahov, z ktorých sa opäť pre akčný zásah použije len prvý člen.



Obr.13: Zobrazenie metódy takzvaného pohyblivého horizontu predikcie na časovej osi. Veľkosť horizontu predikcie je 5 krokov, využije sa len prvý vypočítaný akčný zásah, z celého horizontu.

5.1.1 Rovnica predikcie [5]

Na predpovede budúcich stavov sa používajú takzvané predikčné rovnice, ktoré sú odvodené rekurzívnym riešením rovnice stavového modelu posúvaním sa dopredu v čase. [5]

$$\begin{aligned}
 x_{t+1} &= Ax_t + Bu_t \\
 x_{t+2} &= Ax_{t+1} + Bu_{t+1} = A^2x_t + ABu_t + Bu_{t+1} \\
 x_{t+3} &= Ax_{t+2} + Bu_{t+2} = A^3x_t + A^2Bu_t + ABu_{t+1} + Bu_{t+2} \\
 &\vdots = \vdots \\
 x_{t+k} &= A^k x_t + \sum_{i=0}^{k-1} A^{k-i-1} Bu_{t+i} \quad k = 1, \dots, N
 \end{aligned} \tag{25}$$

Z predchádzajúcich rovníc vyplýva, že pre úspešné predikcie budúcich stavov potrebujeme poznať počiatočný stav systému x_t .

Všeobecne môžeme predchádzajúce rovnice pre horizont predikcie $k = 1, \dots, N$ zjednotiť pomocou maticového zápisu:

$$X = \hat{A}x_t + \hat{B}U, \tag{26}$$

kde matice X a U predstavujú predpovedané stavy a vstupy:

$$X = \begin{bmatrix} x_{t+1} \\ x_{t+2} \\ \vdots \\ x_{t+N} \end{bmatrix}, \quad U = \begin{bmatrix} u_t \\ u_{t+1} \\ \vdots \\ u_{t+N-1} \end{bmatrix}, \tag{27}$$

matice vzniknuté rozvojom stavového modelu v čase potom sú:

$$\hat{A} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix} \tag{28}$$

5.2 Regulácia výstupov

Tento koncept využívame vtedy, keď riadime výstupy systému, čiže v účelovej funkcii máme za úlohu minimalizovať vzdialenosť výstupných veličín od danej referencie. Užitočné je to najmä vtedy, keď nás zaujíma len výstup systému a nie stavy v ktorých sa systém počas riadenia nachádza.

Rozdiel v matematickom zápise stavovej regulácie a regulácie výstupov spočíva v tom, že v účelovej funkcii nahradíme stavové veličiny výstupnými veličinami a váhovú maticu stavov vymeníme za váhovú maticu výstupov. Do ohraničení pridáme rovnicu výstupov

$y_t = Cx_t + Du_t$ a ohraničenia pre výstupy $y_t \in Y$, všetky ostatné ohraničenia sú nezmenené:

$$\begin{aligned} \min \quad & \sum_{k=0}^{N-1} \left(\|Q_y y_{t+k}\|_p + \|Q_u u_{t+k}\|_p \right) \\ \text{s.t.} \quad & x_{t+1} = Ax_t + Bu_t \\ & y_t = Cx_t + Du_t \\ & x_t = x(t) \\ & x_t \in \bar{X}, y_t \in \bar{Y}, u_t \in \bar{U} \end{aligned} \tag{29}$$

Pretože riadiaci systém je schopný regulovať len tie signály, ktoré sú definované v účelovej funkcii, pri regulácii výstupov môže nastať situácia, že niektoré stavy systému pri optimalizácii uletia a nadobudnú fyzikálne nerealizovateľné hodnoty. To by v praxi mohlo viesť k havarijnému stavu, preto sú pri definovaní optimalizačného problému stále potrebné aj stavové ohraničenia $x_t \in \bar{X}$. Taktiež ako si môžeme všimnúť, stále potrebujeme aj definíciu počiatočného stavu systému, kvôli následným predikciám stavov, vstupov a výstupov.

5.3 Nenulová referencia

V predchádzajúcich častiach tejto práce sme hovorili o riadení stavov, respektíve výstupov na nulovú referenciu. Avšak pri implementácii prediktívneho riadenia do praxe je táto teória nepostačujúca, nakoľko vo väčšine praktických problémov sa stretávame s potrebou riadenia na nenulové referencie.

Matematicky ide len o pridanie vyjadrenia referenčného stavu do účelovej funkcie optimalizácie, čo pre stavový reguláciu vyzerá nasledovne:

$$\begin{aligned}
\min \quad & \sum_{k=0}^{N-1} \left(\|Q_x (x_{t+k} - x_{ref})\|_p + \|Q_u u_{t+k}\|_p \right) \\
s.t. \quad & x_{t+1} = Ax_t + Bu_t \\
& x_t = x(t) \\
& x_t \in \bar{X} \\
& u_t \in \bar{U}
\end{aligned} \tag{30}$$

Takže v novej účelovej funkcii už minimalizujeme vzdialenosť aktuálnych stavov od konkrétnej referenčnej hladiny, nie od nuly. Ako si môžeme všimnúť ohraničenia zostávajú nezmenené.

Pre reguláciu výstupov, to platí obdobne:

$$\begin{aligned}
\min \quad & \sum_{k=0}^{N-1} \left(\|Q_y (y_{t+k} - y_{ref})\|_p + \|Q_u u_{t+k}\|_p \right) \\
s.t. \quad & x_{t+1} = Ax_t + Bu_t \\
& y_t = Cx_t + Du_t \\
& x_t = x(t) \\
& x_t \in \bar{X}, y_t \in \bar{Y}, u_t \in \bar{U}
\end{aligned} \tag{31}$$

V účelovej funkcii minimalizujeme súčasne so vzdialenosťou daných stavov alebo výstupov od referencie aj veľkosť akčného zásahu, čiže sa snažíme o to aby sa dodávaná energia do systému približovala nulovej hodnote. To sa prejavuje tak, že niektoré zo stavov alebo výstupov vykazujú regulačnú odchýlku od žiadanej hodnoty. Nakoľko dosiahnutie daných referenčných hladín pre stavy, respektíve výstupy pri nulovom akčnom zásahu väčšinou nie je možné. Optimalizácia preto vyberie také riešenie, pre ktoré má účelová funkcia minimálnu hodnotu, bez ohľadu nato, či sa stavy alebo výstupy dostali do požadovaných referencií.

Zmenou hodnôt váhových matíc Q_u a Q_x , respektíve Q_y môžeme meniť výsledky optimalizácie, napríklad zväčšením matice Q_u nám narastie odchýlka stavov alebo výstupov a akčný zásah sa bude blížiť nulovej hodnote, naopak zväčšením matíc Q_x a Q_y zas odchýlku môžeme zmenšiť na malú hodnotu, avšak odstrániť sa nám týmto spôsobom nepodarí.

5.4 Odstránenie trvalej regulačnej odchýlky

V tejto práci si povieme o dvoch spôsoboch ako odstrániť regulačnú odchýlku, spôsobenú minimalizáciou akčného zásahu na nulovú hodnotu. V oboch prípadoch budeme meniť matematické vyjadrenie akčných zásahov v účelovej funkcii.

5.4.1 Riešenie pomocou výpočtu ustálených stavov

V tomto riešení minimalizujeme v účelovej funkcii rozdiel medzi predikovanými akčnými zásahmi a hodnotou akčného zásahu u_{ss} , ktorá zodpovedá požadovaným hodnotám príslušných ustálených stavov. Lepšie pochopenie nám umožňuje nasledovná formulácia účelovej funkcie optimalizácie pre reguláciu výstupov:

$$\min \sum_{k=0}^{N-1} \left(\|Q_y (y_{t+k} - y_{ref})\|_p + \|Q_u (u_{t+k} - u_{ss})\|_p \right) \quad (32)$$

Výpočet ustálených stavov odvodený zo stavového modelu sa realizuje nasledovne:

$$\begin{bmatrix} (I-A) & -B \\ C & D \end{bmatrix} \begin{bmatrix} x_{ss} \\ u_{ss} \end{bmatrix} = \begin{bmatrix} 0 \\ y_{ref} \end{bmatrix} \quad (33)$$

Toto riešenie môže byť použité rovnako aj pre stavovú reguláciu, namiesto referenčnej hodnoty stavov x_{ref} použijeme v účelovej funkcii vypočítanú hodnotu ustáleného stavu x_{ss} ktorý zodpovedá danej referencii výstupov y_{ref} , účelová funkcia má potom tvar:

$$\min \sum_{k=0}^{N-1} \left(\|Q_x (x_{t+k} - x_{ss})\|_p + \|Q_u (u_{t+k} - u_{ss})\|_p \right) \quad (34)$$

Ohraničenia v oboch prípadoch, v stavovej regulácii aj v regulácii výstupov zostávajú nezmenené, preto ich nespomíname.

Príklad 1 [3]:

Doterajšie teoretické poznatky, môžeme najlepšie demonštrovať praktickým príkladom, realizovaným v programe MATLAB, pomocou programového balíka Yalmip, ktorý nám umožňuje zjednodušený zápis ohraničení a účelovej funkcie a následné riešenie optimalizácie, pomocou zodpovedajúcich metód.

Budeme uvažovať o systéme dvoch zásobníkov kvapaliny bez interakcie, ako matematický model použijeme linearizovaný model zásobníkov, tak ako sme si ho definovali v kapitole

MATEMATICKÝ MODEL. Ako ustálený stav pri ktorom spravíme linearizáciu, si určíme stav zodpovedajúci vstupnému prietoku $q = 10 \text{ cm}^3 \text{ s}^{-1}$. Zadefinujeme si konštanty, ako sú plocha prierezu zásobníkov $F = 12.566 \text{ cm}^2$, a prietoková konštanta zásobníkov $k = 3.667 \text{ cm}^{2.5} \text{ s}^{-1}$. Potom hladiny oboch zásobníkov v tomto ustálenom stave majú výšku $h_1^s = h_2^s = 7.4367 \text{ cm}$. Ako referenčnú hladinu si určíme nový ustálený stav, napríklad o 30% väčší ako prvý ustálený stav: $h_1^{s2} = h_2^{s2} = 9.6677 \text{ cm}$. Spojitý stavový model diskretizujeme, použijeme periódu vzorkovania 5 sekúnd. Zadefinujeme si horizont predikcie N a následne všetky stavové, výstupné a vstupné premenné, pomocou funkcie `sdpvar` (Yalmip). Počet predikovaných premenných je priamo úmerný horizontu predikcie. Fyzikálne ohraničenia pre výšky hladín sú nasledovné $0 \leq h_{1,2} \leq 25 \text{ cm}$, pre vstupný prietok zas platí $0 \leq q \leq 20 \text{ cm}^3 \text{ s}^{-1}$. Tieto ohraničenia si zadefinujeme v odchyľkovom tvare, ako rozdiel od prvého ustáleného stavu. Táto skutočnosť nám umožní si počiatočný stav pre optimalizáciu určiť rovný nule, nakoľko skutočný počiatočný stav sa rovná práve prvému ustálenému stavu. Preto aj premenné x , y , u budú v odchyľkovom tvare vyjadrovať zmenu od prvého ustáleného stavu. Zadefinujeme si účelovú funkciu a ohraničenia, a následná optimalizácia už prebieha pomocou funkcie `solvesdp` z programového balíka Yalmip v uzavretej slučke pri dopredu určenom počte krokov simulácie. Jeden krok simulácie reálne zodpovedá časovému úseku periódy vzorkovania. Takže ak si určíme počet krokov simulácie napríklad na 30, tak pri perióde vzorkovania 5 sekúnd, by nám riadenie reálneho systému trvalo 150 sekúnd.

Príklad 1: Výstupová regulácia s výpočtom ustálených stavov

Zápis v MATLABE:

```
clear all
clc
% vstupny prietok v 1 us
qs1=10;
% konstanty
k=3.667;
F=12.566;
% 1 ustaleny stav
h1s1=(qs1/k)^2;
h2s1=h1s1;
% 2 ust stav o 30% vacsi ako 1. us
h1s2=1.3*h1s1;
h2s2=h1s2;

% matice A,B odvodene zo spojiteho stavoveho modelu
Ac=[ -k/(F*(2*sqrt(h1s1))), 0;
```

```

    k/(F*(2*sqrt(h1s1))), -k/(F*(2*sqrt(h2s1))) ]];
Bc=[1/F;0];
Cc=[0 1];
Dc = 0;
% spojity stavovy model
sc = ss(Ac, Bc, Cc, Dc);

% diskretizacia, perioda vzorkovania 5 sec
Ts = 5;
sd = c2d(sc, Ts);
% matice diskretizovaného stavového modelu
A = sd.A; B = sd.B; C = sd.C; D = sd.D;

% definovanie rozmerov matic
nx=size(A,2);
nu=size(B,2);
ny=size(C,1);

% urcenie horizontu predikcie N
N = 3;
% definicia premennych x,u,y
x = {}; u = {}; y = {};
for k = 1:N+1, x{k} = sdpvar(nx, 1); end
for k = 1:N, u{k} = sdpvar(nu, 1); end
for k = 1:N, y{k} = sdpvar(ny, 1); end

% vahove matice
Qx=eye(nx);
Qu=eye(nu);
Qy=eye(ny);

% ohranicenia v odch. tvare
hmin=[0;0];hmax=[25;25];qmin=[0];qmax=[20];
xmin=hmin-[h1s1;h2s1];xmax=hmax-[h1s1;h2s1];
umin=qmin-qs1;umax=qmax-qs1;
ymin=xmin(2,:);ymax=xmax(2,:);

% ref stav odch. model
yref=h2s2-h2s1;
% vypocet us
M = [(eye(nx)-A) -B; C D];
ss = pinv(M)*[zeros(nx, 1); yref];
u_ss = ss(nx+1:end);

% definovanie ucelovej funkcie a ohraniceni
objective = 0; constraints = [];
for k = 1:N
objective=objective+(y{k}-yref)'*Qy*(y{k}-yref) + ...
(u{k}-u_ss)'*Qu*(u{k}-u_ss);
constraints=constraints+[x{k+1}==A*x{k}+B*u{k}; ...
y{k}==C*x{k}+D*u{k};xmin<=x{k}<=xmax;umin<=u{k}<=umax; ...

```

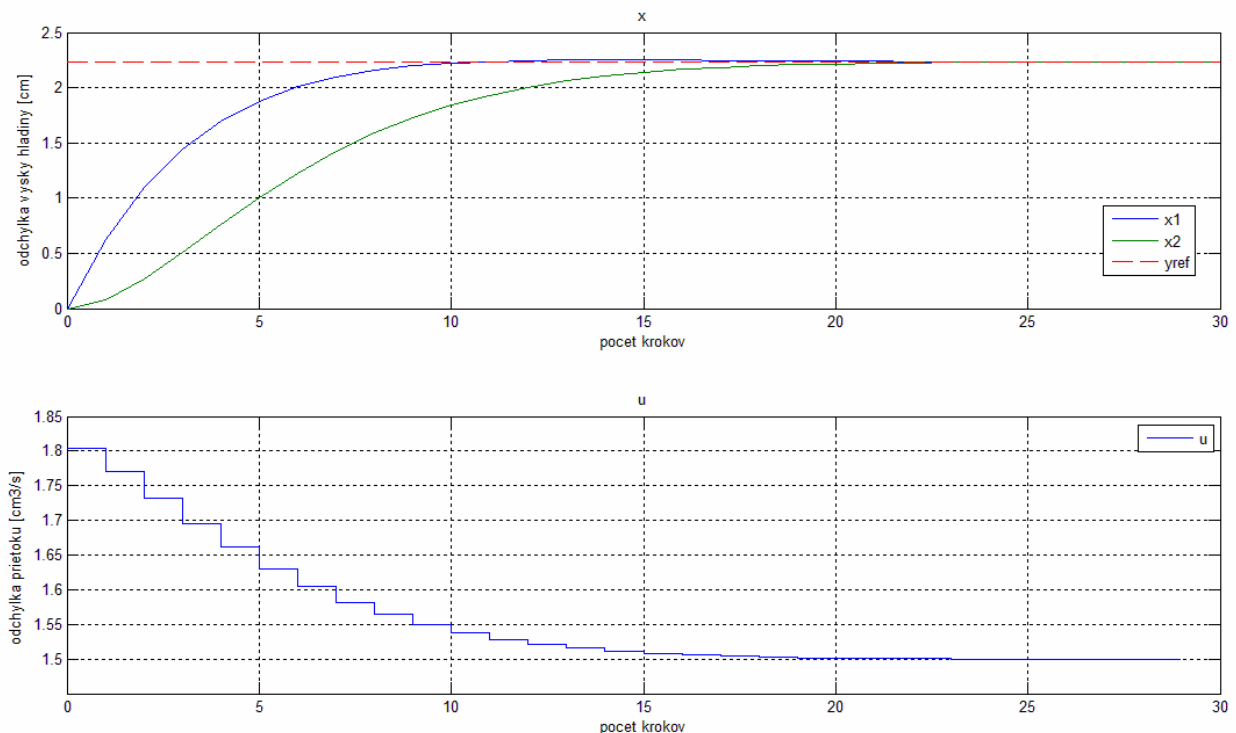
```

ymin<=y{k}<= ymax];
end

% pocet krokov simulacie
N_simsteps = 30;
% pociatocny stav nulovy
xt = [0;0];
X = [xt']; U = [];
% simulacia
for t = 1:N_simsteps
    % riesenie optimalizacie pomocou yalmipu
    info = solvesdp(constraints+[x{1}==xt], objective);
    if info.problem~= 0, error('Problem infeasible'),end
    % optimalny akcny zasah
    u0opt = double(u{1});
    % novy stav
    xt = A*xt + B*u0opt;
    X = [X; xt'];
    U = [U; u0opt'];
end

```

Grafický výsledok simulácie 1:



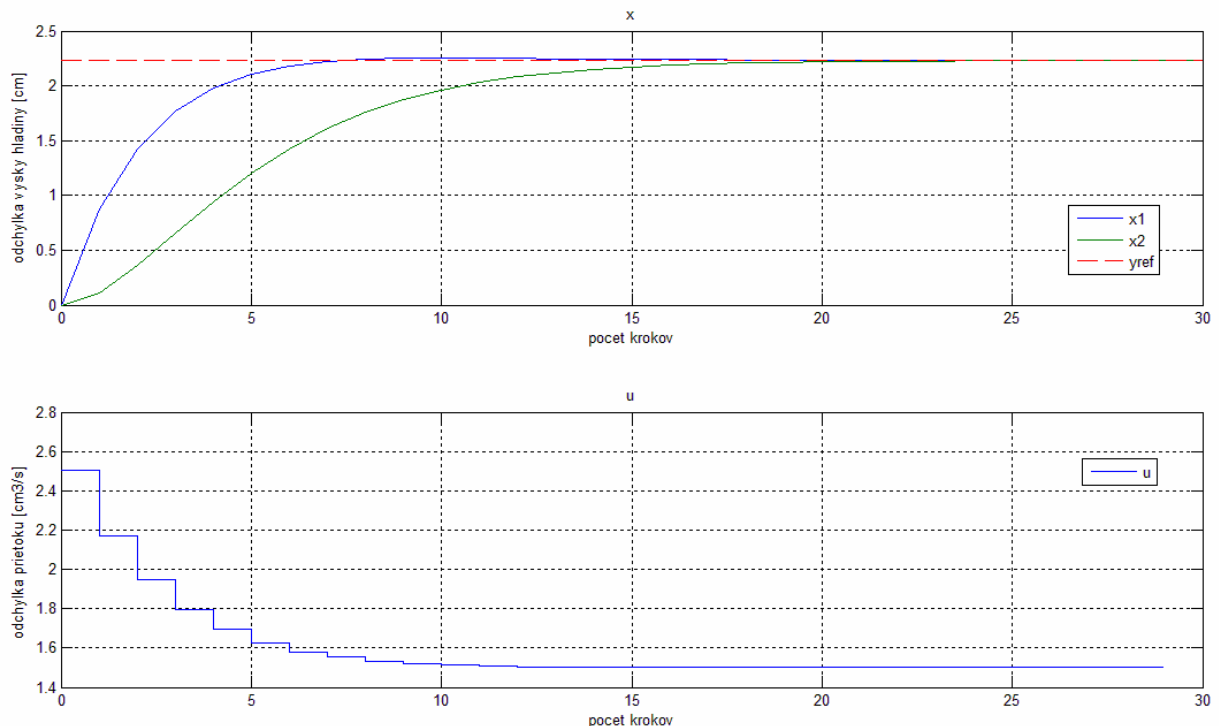
Príklad 1: Stavová regulácia s výpočtom ustálených stavov

Zmena oproti regulácii výstupov spočíva, len v zadefinovaní účelovej funkcie a výpočte ustálených stavov, priamo v ohraničeniach, ako sme si to vysvetlili v teoretickej časti. Preto už nebudeme písať celý kód v Matlabe, ale len tie časti, ktoré sa líšia od predošlého zápisu:

```
% ref stav
yref=h2s2-h2s1;
% matica M potrebna na vypocet us
M = [(eye(nx)-A) -B; C D];
% nove premenne potrebne na vypocet us
x_ss = sdpvar(nx, 1); u_ss = sdpvar(nu, 1);

% def ucelovej funkcie a obmedzeni
objective = 0; constraints = [];
for k = 1:N
    objective=objective+(x{k}-x_ss)'*Qx*(x{k}-x_ss) + ...
        (u{k}-u_ss)'*Qu*(u{k}-u_ss);
    constraints=constraints+[x{k+1}==A*x{k}+B*u{k}; ...
        xmin<=x{k}<=xmax;umin<=u{k}<=umax];
end
% vypocet us v obmedzeniach
constraints = constraints + ...
    [ M*[x_ss;u_ss] == [zeros(nx,1);yref] ];
```

Grafický výsledok simulácie 2:



Ako môžeme sledovať na grafoch z predchádzajúcich príkladov, tak riadenie zásobníkov prebehlo úspešne, obe hladiny sa v oboch prípadoch dostali pomocou optimalizovaných akčných zásahov na danú referenčnú hladinu. Avšak pri tomto spôsobe definovania účelovej funkcie si môžeme všimnúť isté nedostatky, nakoľko nemáme penalizovaný prírastok budúceho akčného zásahu, náš riadiaci systém v prvých krokoch simulácie vykazuje veľké akčné zásahy, ktoré postupne znižuje, až na hodnotu korešpondujúcu danému referenčnému stavu. V praxi by mohlo mať takéto riadenie problémy, nakoľko všetky zariadenia majú svoju dynamiku a pri menších hodnotách periódy vzorkovania, nemusíme byť schopný za taký krátky čas zabezpečiť adekvátne veľké akčné zásahy. Konkrétne v našom prípade nemusíme stihnúť v prvom kroku zvýšiť výkon čerpadla, nakoľko aby nám zabezpečil potrebný vstupný prietok kvapaliny.

5.4.2 Riešenie pomocou penalizácie zmeny vstupov

Problémy akčných zásahov, spomenuté v predchádzajúcom príklade, sa dajú riešiť pomocou penalizácie zmeny vstupov v účelovej funkcii. Čiže minimalizujeme rozdiel medzi súčasným akčným zásahom a budúcim akčným zásahom, čo môžeme vyjadriť pomocou rovnice:

$$\Delta u_{t+k} = u_{t+k} - u_{t+k-1} \quad (35)$$

Nakoľko máme v účelovej funkcii novú premennú Δu_{t+k} , potrebujeme si zadať aj novú váhovú maticu $Q_{\Delta u}$ a pretransformovať ohraničenia, optimalizačný problém potom môžeme zapísať nasledovne:

$$\begin{aligned} \min \quad & \sum_{k=0}^{N-1} \left(\|Q_y (y_{t+k} - y_{ref})\|_p + \|Q_{\Delta u} \Delta u_{t+k}\|_p \right) \\ \text{s.t.} \quad & \tilde{x}_{t+1} = \tilde{A} \tilde{x}_t + \tilde{B} \Delta u_t \\ & y_t = \tilde{C} \tilde{x}_t + \tilde{D} \Delta u_t \\ & \tilde{x}_t = \begin{bmatrix} x(t) \\ u(t-1) \end{bmatrix} \\ & \tilde{H}_{xu} \tilde{x}_t \leq \tilde{K}_{xu} \\ & H_y y_t \leq K_y \end{aligned} \quad (36)$$

Kde nové matice $\tilde{A}$, $\tilde{B}$, $\tilde{C}$ a $\tilde{D}$ sú:

$$\tilde{A} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \quad \tilde{B} = \begin{bmatrix} B \\ I \end{bmatrix} \quad (37)$$

$$\tilde{C} = [C \quad D] \quad \tilde{D} = D \quad (38)$$

A nové matice ohraničení stavov sú:

$$\tilde{H}_{xu} = \begin{bmatrix} H_x & 0 \\ 0 & H_u \end{bmatrix} \quad \tilde{K}_{xu} = \begin{bmatrix} K_x \\ K_u \end{bmatrix} \quad (39)$$

Príklad 1: Výstupná regulácia s penalizáciou zmeny vstupov

Tento nový zápis účelovej funkcie a ohraničení, implementujeme do príkladu dvoch zásobníkov kvapaliny ktorý sme už riešili pomocou výpočtu ustálených stavov.

Zápis v MATLABE:

```
clear all
clc
%vstupny prietok v 1 us
qs1=10;
%konstanty
k=3.667;
F=12.566;
% 1 ustalene stavy vyp z dif rov.
h1s1=(qs1/k)^2;
h2s1=h1s1;
% 2 ust stav o 30% vacsi ako 1. us
h1s2=1.3*h1s1;
h2s2=h1s2;
% matice A,B odvodene zo stavoveho opisu
Ac=[-k/(F*(2*sqrt(h1s1))), 0;
     k/(F*(2*sqrt(h1s1))), -k/(F*(2*sqrt(h2s1))) ];
Bc=[1/F;0];
Cc=[0 1];
Dc = 0;
% stavovy model spojity
sc = ss(Ac, Bc, Cc, Dc);

% diskretizacia perioda vzorkovania 5 sec
Ts = 5;
sd = c2d(sc, Ts);
A = sd.A; B = sd.B; C = sd.C; D = sd.D;
nx=size(A,2);
nu=size(B,2);
```

```

ny=size(C,1);
% urcenie matic Atilda Btilda Ctilda Dtilda
At = [A B; zeros(nu, nx) eye(nu)];
Bt = [B; eye(nu)]; Ct = [C D]; Dt=D;

% urcenie horizontu predikcie
N = 3;
% xtilda deltau y
xt = {}; du = {}; y = {};
for k = 1:N+1, xt{k} = sdpvar(nx+nu, 1); end
for k = 1:N, du{k} = sdpvar(nu, 1); end
for k = 1:N, y{k} = sdpvar(ny, 1); end
% vahove matice
Qx=eye(nx);
Qdu=eye(nu);
Qy=eye(ny);

% ohranicenia odch. tvar
hmin=[0;0];hmax=[25;25];qmin=[0];qmax=[20];
xmin=hmin-[h1s1;h2s1];xmax=hmax-[h1s1;h2s1];
umin=qmin-qs1;umax=qmax-qs1;
ymin=xmin(2,:);ymax=xmax(2,:);
% ref stav odch. tvar
yr=h2s2-h2s1;
yref = sdpvar(ny,1);

% definovanie ucelovej funkcie a ohraniceni
objective = 0; constraints = [];
for k = 1:N
objective=objective+(y{k}-yref)'*Qy*(y{k}-yref) + ...
du{k}'*Qdu*du{k};
constraints=constraints+[xt{k+1}==At*xt{k}+Bt*du{k}; ...
y{k}==Ct*xt{k}+Dt*du{k}; ...
[xmin;umin]<=xt{k}<=[xmax;umax];ymin<=y{k}<=ymax];
end

% pociatocny stav nulovy, transofmacia do tvaru xtilda
uprev = zeros(nu, 1);
x = [0;0];
x0 = [x; uprev];
% vektor stavov, zasahov a zmeny zasahov
X = [x']; U = [0]; DU= [0];

% simulacia pre dany pocet krokov
N_simsteps = 100;
for t = 1:N_simsteps
    % riesenie opt. problemu yalmip
    info = solvesdp(constraints+[xt{1}==x0;yref==yr],
objective);
    if info.problem~= 0, error('Problem infeasible'),end
    % vysledna optimalna zmena akcneho zasahu

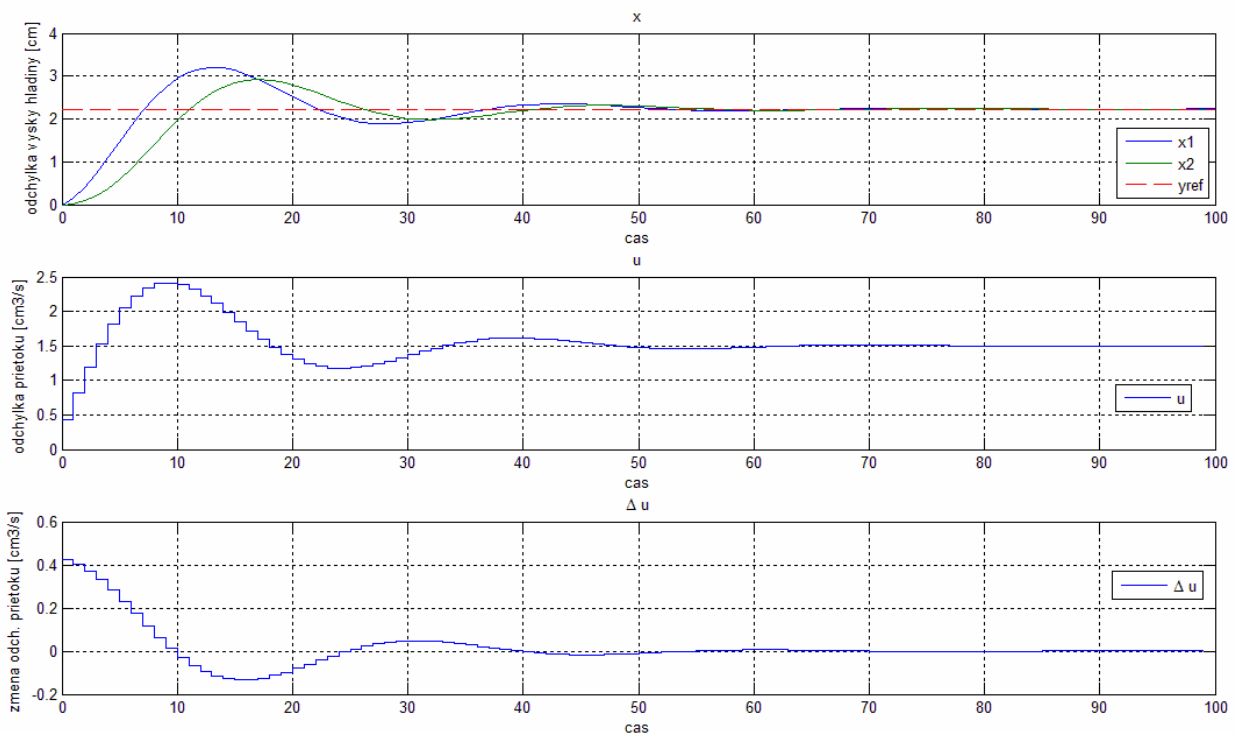
```

```

duopt = double(du{1});
% optimalny akcny zasah
uprev = duopt + uprev;
% novy stav
x = A*x + B*uprev;
x0 = [x; uprev];
X = [X; x'];
U = [U; uprev];
DU = [DU; duopt];
end

```

Grafický výsledok simulácie 3:

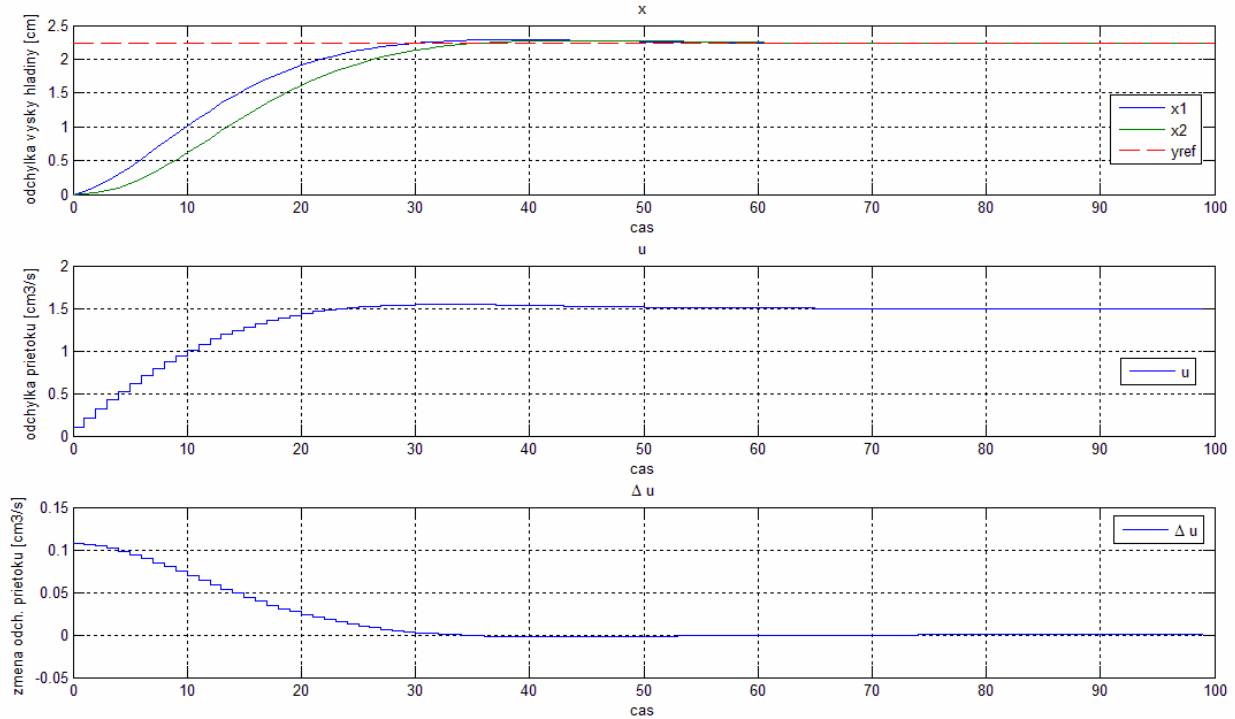


5.5 Vplyv váhových matíc

Z predošlých výsledkov riadenia sa môžeme presvedčiť, že platí to čo sme predpovedali v teoretickej časti a to, že pri riadení nám už naše akčné zásahy nerobia také prudké zmeny, ako v predošlých simuláciách, čo je spôsobené práve minimalizáciou ich zmien. Avšak nič nie je zadarmo, a v tomto prípade sme na to doplatili zväčšením času regulácie takmer trojnásobne, to by nám ani tak nevadilo ako fakt, že priebeh riadenia má zrazu kmitavý priebeh. Tieto kmity sú spôsobené tým, že zmena akčných zásahov nie je dostatočne penalizovaná v účelovej funkcii, to sa dá napraviť celkom jednoducho zmenou váhových matíc. Skusmo sme zistili že najideálnejší priebeh riadenia bude ak nastavíme váhovú maticu

$Q_{\Delta u}$, štyri krát väčšiu ako zvyšné váhové matice, potom získame nekmitavý priebeh riadenia, čo dokazuje aj nasledovný graf.

Grafický výsledok simulácie 4:



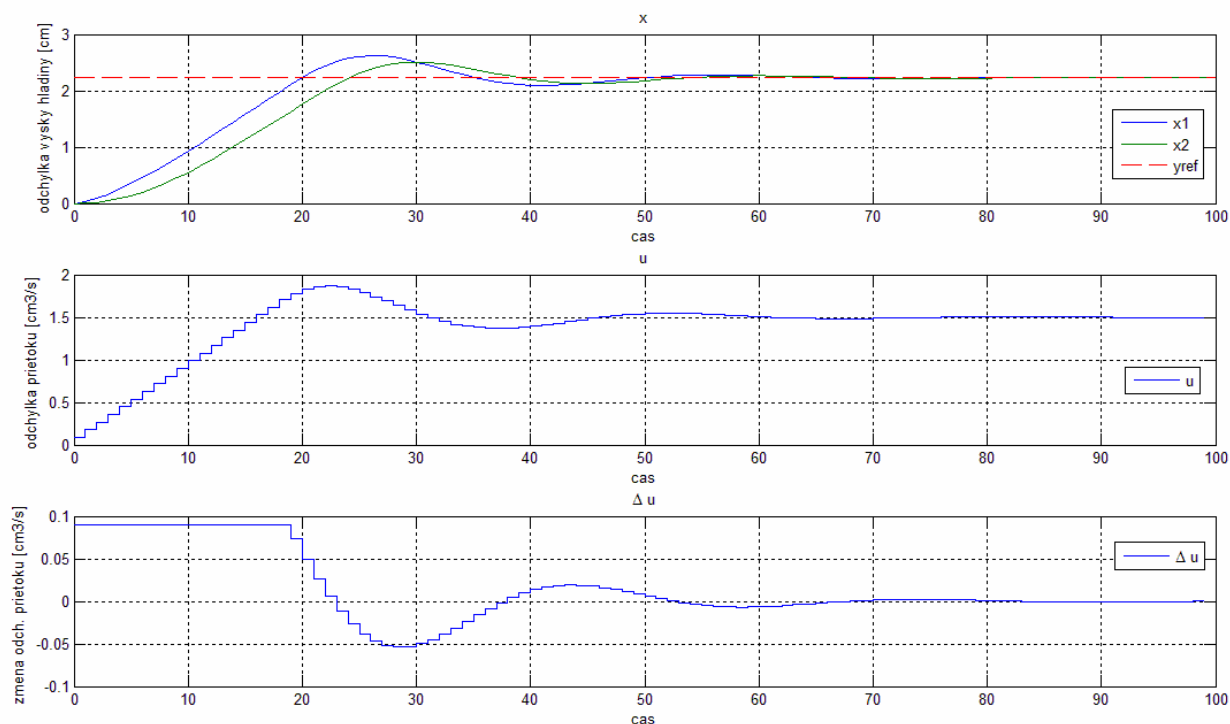
5.6 Spomalovacie ohraničenia

Druhým spôsobom ako obmedziť zmeny akčného zásahu, sú spomalovacie ohraničenia, ktoré môžeme matematicky definovať nasledovne: $\Delta u_{\min} \leq \Delta u_t \leq \Delta u_{\max}$, celý optimalizačný problém prediktívneho riadenia má potom nasledovný tvar:

$$\begin{aligned}
 \min \quad & \sum_{k=0}^{N-1} \left(\|Q_y (y_{t+k} - y_{ref})\|_p + \|Q_{\Delta u} \Delta u_{t+k}\|_p \right) \\
 \text{s.t.} \quad & \tilde{x}_{t+1} = \tilde{A} \tilde{x}_t + \tilde{B} \Delta u_t \\
 & y_t = \tilde{C} \tilde{x}_t + \tilde{D} \Delta u_t \\
 & \tilde{x}_t = \begin{bmatrix} x(t) \\ u(t-1) \end{bmatrix} \\
 & \tilde{H}_{xu} \tilde{x}_t \leq \tilde{K}_{xu} \\
 & H_y y_t \leq K_y \\
 & \Delta u_{\min} \leq \Delta u_t \leq \Delta u_{\max}
 \end{aligned} \tag{40}$$

Nastavením spomalovacích ohraňení v našom príklade na hodnoty $\Delta u_{\min} = -0.06$ a $\Delta u_{\max} = 0.09$ sa nám evidentne podarí zmenšiť maximálne preregulovanie, aj počet kmitov pri riadení, čo môžeme pozorovať na nasledujúcom grafe priebehu riadenia.

Grafický výsledok simulácie 5:



5.7 Časovo meniac sa referencia

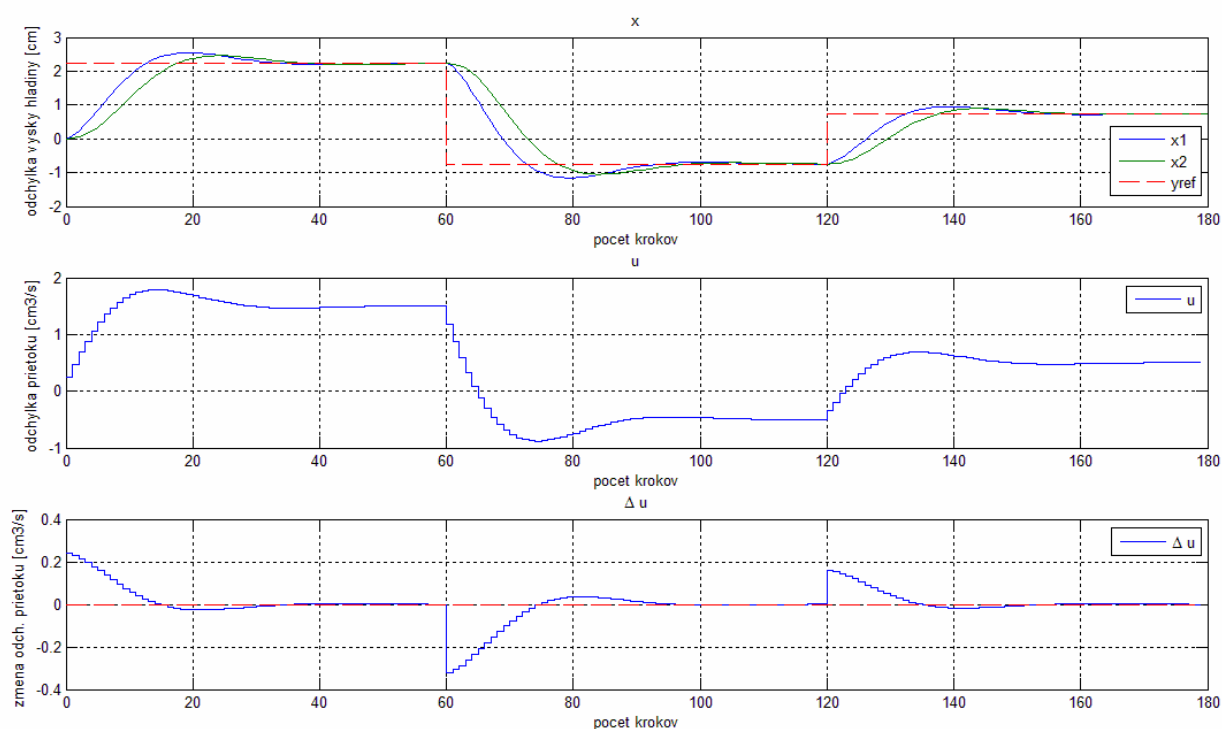
Potrebami praxe bola motivovaná aj myšlienka časovo meniacej sa referencie pre výstupné veličiny, keďže pre riadenie väčšiny reálnych zariadení potrebujeme flexibilný riadiaci systém, ktorý je schopný reagovať aj na meniace sa požiadavky žiadaných výstupných veličín v čase. Väčšina klasických regulátorov reaguje zmenou akčných zásahov až bezprostredne po zmene referenčnej hladiny, práve tu sa prejavuje silná stránka prediktívneho riadenia, ktoré sa nielenže s ľahkosťou prispôsobuje zmene referencie, ale taktiež sa na ňu dokáže dopredu pripraviť a reagovať s predstihom, v závislosti od veľkosti horizontu predikcie. Analogicky by sme to mohli prirovnať k jazde autom pozerajúc sa na cestu cez predné sklo (prediktívne riadenie), v porovnaní s jazdou na aute a pozerajúc sa na cestu cez bočné sklo (klasické regulátory). [4]

Príklad 1: časovo meniaci sa referencia

V našom príklade dvoch zásobníkov kvapaliny, budeme uvažovať o počte krokov riadenia 180, počas ktorých nastanú dve zmeny referenčnej hladiny. Prvých 60 krokov simulácie budeme riadiť systém do ustáleného stavu, ktorý je o 30% väčší ako náš počiatočný stav, následne sa nám referencia zmení pre ďalších 60 krokov na ustálený stav, ktorý je o 10% menší ako počiatočný stav a nakoniec posledných 60 krokov simulácie budeme riadiť systém na referenčnú hladinu zodpovedajúcu ustálenému stavu, ktorý je o 10% väčší ako počiatočný stav.

Pre porovnanie si ukážeme oba spôsoby sledovania referencie, ako prvý si ilustrujeme príklad bez predikcie budúcej referencie. V tomto prípade je matematická formulácia problému rovnaká ako v predošlej kapitole, čiže v účelovej funkcii použijeme konštantné hodnoty referencie y_{ref} . Simuláciou v MATLABE, získame nasledovný priebeh riadenia.

Grafický výsledok simulácie 6:



Ako je zrejmé z obrázku, tak v tomto prípade, nevyužívame možnosť predikcie budúcej referencie, takže nie sme na jej zmenu schopný zmenou akčného zásahu reagovať v predstihu. Avšak pre potreby praxe je takáto predikcia referenčnej hladiny kľúčová, nakoľko pri jej implementácii, sme schopný daný systém riadiť efektívnejšie ale aj bezpečnejšie, nakoľko

máme prehľad, kam do budúcnosti smeruje naše riadenie a prediktívny regulátor je potom schopný sa ľahšie vyrovnáť s prípadnými poruchami.

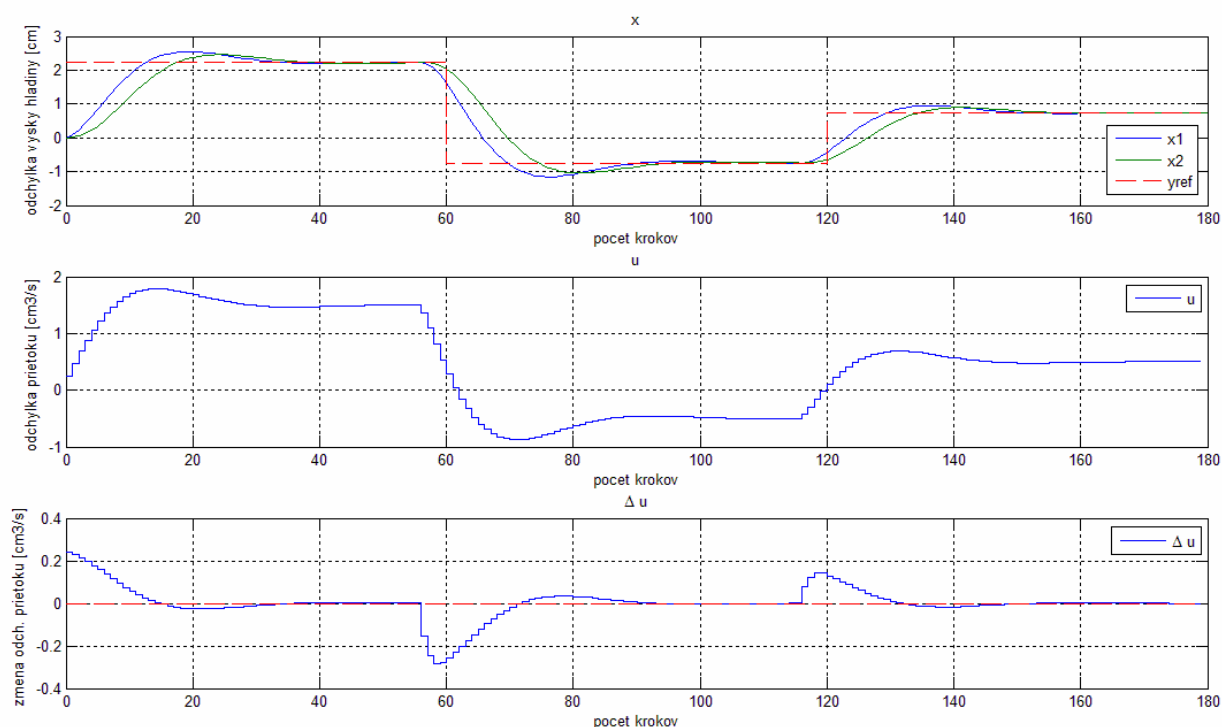
Túto flexibilitu, sme schopný zabezpečiť, tak že referencia výstupov v účelovej funkcii už naďalej nebude konštanta y_{ref} , ako sme to robili doteraz, ale nahradíme ju za premennú $y_{ref,t}$.

Keďže ohraničenia zostávajú nezmenené, účelová funkcia optimalizácie potom bude mať tvar:

$$\min \sum_{k=0}^{N-1} \left(\|Q_y (y_{t+k} - y_{ref,t})\|_p + \|Q_{\Delta u} \Delta u_{t+k}\|_p \right) \quad (41)$$

Simulácia v MATLABE, pre horizont predikcie N rovný 5, bude mať potom nasledovný priebeh.

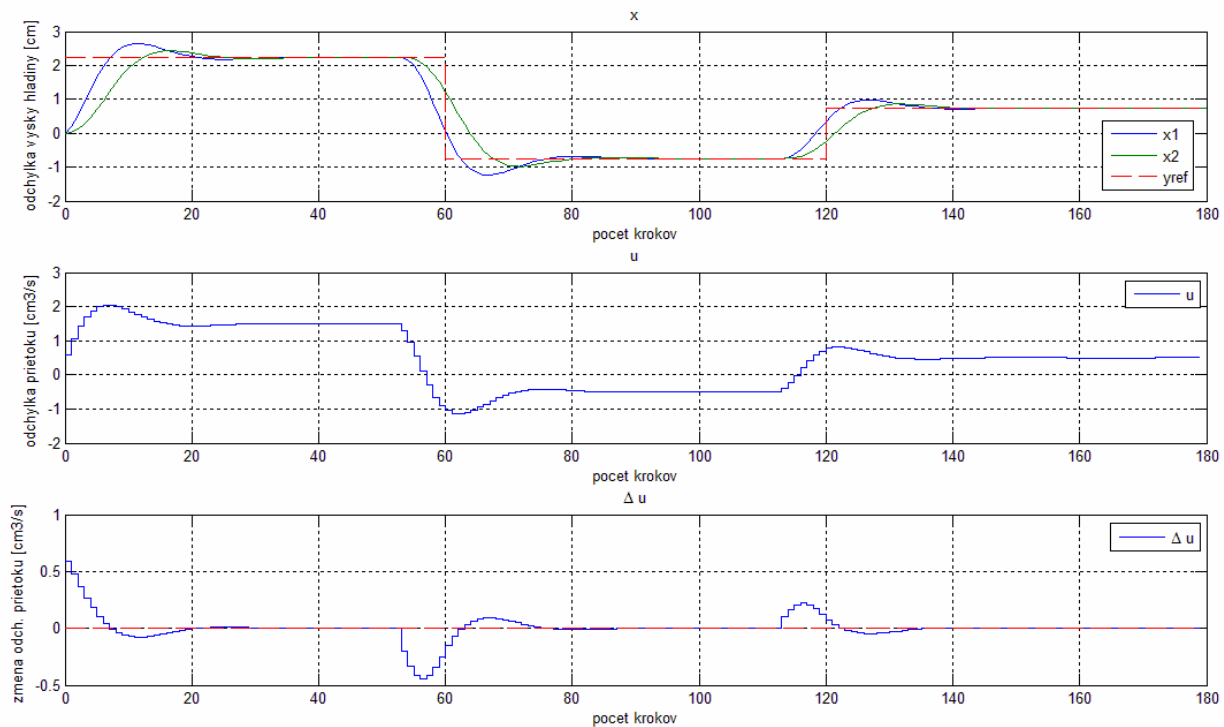
Grafický výsledok simulácie 7:



Z priebehu riadenia je evidentná zmena, oproti simulácii bez predikcie reference. Nakoľko náš prediktívny regulátor je schopný v tomto konkrétnom prípade pozerať sa 5 krokov dopredu a sledovať aj budúcu trajektóriu reference, tak na očakávanú zmenu reaguje vopred, čo zabezpečí plynulejší priebeh riadenia a rýchlejší prechod z jednej referenčnej hladiny do druhej.

Pri zvýšení horizontu predikcie na hodnotu $N=8$ môžeme dostať ešte efektívnejší priebeh riadenia, čo je ilustrované na nasledovnom grafickom zobrazení.

Grafický výsledok simulácie 8:



Ako si môžeme všimnúť pri predikcii trajektórie referencie, tak akčné zásahy sa prispôsobujú zmene referencie skôr o $N-1$ krokov v porovnaní s formuláciou bez predikcie trajektórie. Čo nám umožnilo evidentne rýchlejšie uradiť zásobníky kvapaliny na požadované hodnoty referenčných hladín.

Všeobecný zápis v MATLABE:

```
% vektor meniacej sa referencie
for t = 1:(N_simsteps+N)
    if t <= 60
        yr=h2s2-h2s1;
    elseif (t<=120 & t>60)
        yr=h2s3-h2s1;
    else
        yr=h2s4-h2s1;
    end
    refy=[refy;yr];
end
```

```

% pouzitie sledovania trajektorie
use_preview = true; % true alebo false

for t = 1:N_simsteps
    % podmienka pouzitia predikcie referencie
    if use_preview
        % predikovaná referencia
        refy_prediction = refy(t:t+N-1);
    else
        % statická referencia
        refy_prediction = repmat(refy(t), N, 1);
    end

    % riesenie optimalizacneho problemu
    info = solvesdp(constraints + [ xt{1} == x0; ...
        cat(1, yref{:}) == refy_prediction ], objective);
    if info.problem~= 0, error('Problem infeasible'),end

    % vysledna optimalna zmena akcneho zasahu
    duopt = double(du{1});
    % optimalny akcny zasah
    uprev = duopt + uprev;
    % novy stav
    x = A*x + B*uprev;
    x0 = [x; uprev];
end

```

6 Záver

Úlohou tejto práce v jej prvej časti bolo stručne predstaviť problematiku konvexnej optimalizácie. Teoreticky sme si vysvetlili pojmy ako konvexná množina, či spojitý priestor. Pomocou lineárnych ohraničení sme definovali polytopy, ako základné množiny možných riešení optimalizačných úloh riešených v tejto práci. Ďalej sme sa teoreticky zaoberali dvomi základnými typmi konvexných optimalizačných problémov, a to problémom lineárneho a kvadratického programovania, pričom sme sa dotkli aj problematiky konvexných funkcií a noriem. V druhej časti práce sme sa venovali prediktívnemu riadeniu s modelom a využitiu práve konvexnej optimalizácie v tomto progresívne sa vyvíjajúcom odvetví riadenia procesov. V práci sme okrajovo spomenuli aj problematiku matematického modelovania a diskretizácie daných modelov, nakoľko dobrý matematický model je základným predpokladom úspešného riadenia pomocou prediktívneho riadenia s modelom. Avšak hlavným cieľom tejto práce bolo priblíženie princípu fungovania prediktívneho riadenia, jeho matematická definícia a teoretické riešenie rôznych problémov riadenia ktoré sa môžu vyskytnúť v praxi pomocou metód prediktívneho riadenia. Všetky teoretické príklady, sme následne aj prakticky realizovali pomocou programu MATLAB, pričom sme získali aj grafické výsledky pre konkrétne príklady riadenia dvoch zásobníkov kvapaliny bez interakcie. Naším cieľom bolo zefektívniť riadenie týchto zásobníkov, čo sa nám implementáciou rôznych metód prediktívneho riadenia aj podarilo. V práci sme sa snažili poukázať hlavne na silné stránky prediktívneho riadenia a tým vysvetliť fakt, že prediktívne riadenie dnes patrí medzi najúspešnejšie a najvyužívanéjšie metódy riadenia procesov.

7 Literatúra

- [1] **Stephen Boyd - Lieven Vandenberghe:** Convex Optimization, Cambridge University Press 2004
- [2] **E. F. Camacho - C. Bordons:** Model predictive control, Springer-Verlag, London, 2004
- [3] **I. Rauová - M. Kvasnica - Ľ. Čirka - M. Fikar:** Real-time model predictive control of a laboratory liquid tanks system, STU Bratislava
- [4] **M. Kvasnica:** Prednášky k predmetu prediktívne riadenie, STU Bratislava
- [5] **Adrián Karas – Boris Rohál'-Ilkiv – Cyril Belavý:** Praktické aspekty prediktívneho riadenia, STU Bratislava, 2007