

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKÉJ A POTRAVINÁRSKEJ TECHNOLOGIE

Evidenčné číslo: FCHPT-5414-58726

PREDIKTÍVNE RIADENIE SYSTÉMU
„GULIČKA NA PLOCHE“

DIPLOMOVÁ PRÁCA

2014

Bc. Erika Szabová

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKÉJ A POTRAVINÁRSKEJ TECHNOLOGIE
ÚSTAV INFORMATIZÁCIE, AUTOMATIZÁCIE A MATEMATIKY

Prediktívne riadenie systému „gulička na ploche“

FCHPT-5414-58726

Študijný program: Automatizácia a informatizácia v chémii a potravinárstve

Číslo študijného odboru: 2621

Názov študijného odboru: 5.2.14 Automatizácia

Školiace pracovisko: Fakulta chemickej a potravinárskej technológie

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

Konzultant: Ing. Juraj Holaza

2014

Bc. Erika Szabová



ZADANIE DIPLOMOVEJ PRÁCE

Evidenčné číslo: FCHPT-5414-58726
ID študenta: 58726
Autorka práce: Bc. Erika Szabová (58726)
Študijný program: automatizácia a informatizácia v chémii a potravinárstve
Študijný odbor: 5.2.14 automatizácia

Vedúci práce: doc. Ing. Michal Kvasnica, PhD.
Konzultant: Ing. Juraj Holaza

Názov témy: **Prediktívne riadenie systému "gulička na ploche"**

Rozsah práce: 50

Špecifikácia zadania:

Cieľom práce je navrhnúť, implementovať a experimentálne verifikovať výkonnosť prediktívneho regulátora aplikovaného na systém objektu pohybujúceho sa na naklonenej ploche. Náklon plochy v oboch smeroch je možné ovládať pomocou motora, pričom aktuálna poloha objektu sa získa rozpoznávaním obrazu.

Úlohy:

- * Vytvoriť a verifikovať matematický model systému.
- * Navrhnúť a implementovať odhad stavu.
- * Navrhnúť a implementovať stavový LQR regulátor bez uvažovania ohraničení.
- * Syntetizovať prediktívny regulátor, ktorý uvažuje s ohraničeniami.
- * Simulačne a experimentálne overiť výkonnosť prediktívneho regulátora a porovnať ho s LQR prístupom.

Zoznam odbornej literatúry:

1. KVASNICA, M. *Real-Time Model Predictive Control via Multi-Parametric Programming: Theory and Tools : Theory and Tools*. Saarbrücken: VDM Verlag Dr. Müller, 2009. 231 s. ISBN 978-3-639-20644-9.

Dátum zadania diplomovej práce: **17. 02. 2014**

Termín odovzdania diplomovej práce: **24. 05. 2014**

Bc. Erika Szabová
šudentka

prof. Ing. Miroslav Fikar, DrSc.
vedúci pracoviska

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

Pod'akovanie

Ďakujem svojmu vedúcemu diplomovej práce, doc. Ing. Michalovi Kvasnicovi, PhD. a taktiež svojmu konzultantovi Jurajovi Holazovi za konzultácie, odborné vedenie a rady pri tvorbe mojej práce.

Abstrakt

Hlavným cieľom práce je riadenie reálneho procesu, ktorým je „Gulička na ploche“. Tento systém je vyrobený firmou Humusoft a je laboratórnym modelom na riadenie pozícií guľôčky. Z toho sa vyplýva, že sa snažíme systém riadiť tak, aby guľôčka bola v strede plochy. Ďalším cieľom je navrhnúť a implementovať optimálne stavové riadenia, tj. LQR a MPC riadenie. Najväčším rozdielom medzi LQR a MPC riadením je, že pri LQR riadení je používaný nekonečný predikčný horizont. Výhodou optimálneho MPC riadenia je, že minimalizujeme účelovú funkciu čím ušetríme náklady a znižujeme opotrebovanie systému. V tomto prípade je používaný konečný predikčný horizont, čo v tomto prípade reálneho zariadenia je rovný piatim. Nato, aby sme mohli navrhnúť dobré LQR a MPC riadenie na reálne zariadenie, musíme poznať korektný model systému a musíme mať čím presnejší pozorovač. Pre všetky typy riadenia sú navrhnuté simulačné schémy, ktoré sú používané pre riadenie reálneho systému.

Kľúčové slová: „Gulička na ploche“, pozorovač, LQR riadenie, MPC riadenie

Abstract

The main aim of this master's thesis is the control of the real process, which is the system "Ball and Plate". This system is manufactured by the company Humusoft. This system is a laboratory model for the control of the position ball. It is clear that we are trying to control the system so that the ball was centered in the plate. Another aim is to design and implement an optimal state control, ie. LQR and MPC control. The biggest difference between LQR and MPC control is that the LQR is used infinite prediction horizon. The advantage of optimal MPC control is that we minimize the objective function, with saving costs and reducing attrition of system. In this case, we used finite prediction horizon, which is equal to five. So if we can design a good LQR and MPC control of the real process, we need to know the correct model of the system and we must have a more accurate observer. For all types of control are designed simulation schemes, which are used for the control of real system.

Key words: "Ball and Plate", observer, LQR control, MPC control

Zoznam symbolov a skratiek

$x(t)$	stĺpcový vektor stavových veličín
$u(t)$	stĺpcový vektor riadiacich (akčných) veličín
$y(t)$	stĺpcový vektor výstupných veličín
T	kinetická energia
V	potenciálna energia
$\dot{q}$	prvá derivácia i-tej súradnice
q	generovaná i-tá súradnica
Q_i	generovaná i-tá sila
y_x	pozícia guličky
$\dot{y}_x$	rýchlosť guličky

Obsah

Zoznam symbolov a skratiek	8
1. Opis zariadenia „gulička na ploche”	14
1.1 Model systému.....	17
1.2 Stavová reprezentácia systému	21
2. Teória	26
2.1 Teória PID regulátorov	26
2.1.1 P zložka PID regulátora.....	27
2.1.2 I zložka PID regulátora	29
2.1.3 D zložka PID regulátora.....	31
2.2 Teória pre návrh pozorovača	32
2.3 Teória pre LQ riadenie	34
2.4 Teória pre MPC riadenie	37
2.4.1 Hustá formulácia	44
2.4.2 Delta u formulácia.....	46
3. Riadenie reálneho systému	48
3.1 Návrh PID regulátora.....	48
3.2 Pozorovač	54
3.3 Návrh LQR regulátora	56
3.4 LQR – tracking	60
3.5 Hustá formulácia MPC	64
3.6 Delta u formulácia MPC	67

Zoznam ilustrácií

Obr. 1 : Zariadenie „gulička na ploche“	14
Obr. 2: Servomotor a podpera.....	16
Obr. 3: Čidlo	16
Obr. 4: Schéma modelu Gulička na ploche [2].....	18
Obr. 5: Linearizovaná dynamika systému gulička na ploche zahŕňajúca servosystém [2]	21
Obr. 6: Bloková schéma URO [3].....	26
Obr. 7: Prechodová charakteristika P zložky [3]	28
Obr. 8: Prechodová charakteristika I zložky [3]	29
Obr. 9: Grafické vyjadrenie definície integračnej časovej konštanty [3]	30
Obr. 10: Schéma pre antiwindup	30
Obr. 11: Prechodová charakteristika zložky D [3].....	31
Obr. 12: Schéma systému riadenia pomocou stavového regulátora s využitím odhadu stavu	33
Obr. 13. Riadenie systému pomocou stavového regulátora.....	36
Obr. 14: Riadenie systému pomocou stavového regulátora so žiadanou hodnotou	37
Obr. 15: Schéma MPC	39
Obr. 16. : Všeobecná štruktúra prediktívneho riadenia s pozorovačom	39
Obr. 17: Princíp algoritmu MPC.....	40
Obr. 18: Schéma pre PID regulátor.....	51
Obr. 19: Diskrétny PID regulátor.....	51
Obr. 20: Schéma na kalibráciu plochy	52
Obr. 21. Škálovanie signálu	53

Obr. 22. Riadenie polohy guľičky s PID regulátorom	53
Obr. 23: Schéma pozorovača	55
Obr. 24: Odhad stavu pre jednu polohu guľičky	55
Obr. 25: Odhad stavu pre druhú polohu guľičky.....	56
Obr. 26: Schéma pre model LQR regulátorom.....	58
Obr. 27: Riadenie jednej polohy guľičky LQR regulátorom.....	59
Obr. 28: Riadenie druhej polohy guľičky LQR regulátorom	59
Obr. 29: Akčné zásahy	60
Obr. 30: Schéma pre LQR riadenie - tracking	62
Obr. 31: Diskrétny filter.....	62
Obr. 32: Riadenie polohy guľičky s LQR – tracking	63
Obr. 33: Schéma pre MPC regulátor.....	65
Obr. 34: Riadenie jednej polohy guľičky MPC regulátorom – hustá formulácia	65
Obr. 35: Riadenie druhej polohy guľičky MPC regulátorom – hustá formulácia	66
Obr. 36: Akčné zásahy pri hustej formulácii MPC.....	66
Obr. 37: Riadenie polohy guľičky MPC regulátorom – delta u formulácia	68
Obr. 38: Riadenie oboch polôh guľičky MPC regulátorom – delta u formulácia	69
Obr. 39: Akčné zásahy pri delta u formulácii MPC.....	69

Úvod

Táto diplomová práca sa zaoberá s problematikou reálneho systému „Gulička na ploche“ vyrobený firmou Humusoft. V prvej kapitole sa čitateľ oboznámi s jednotlivými časťami zariadenia, s modelom systému a so stavovou reprezentáciou systému. Hlavné časti tohto zariadenia sú plocha, na ktorú je umiestnená guľička, kamera a servomotory, ktoré sú podrobnejšie opísané v prvej kapitole práce. Základným cieľom riadenia je poloha guľičky, ktorú chceme riadiť do stredu plochy. Túto vopred zvolenú referenciu dosiahneme pomocou pohyblivej plochy. Plochu dokážeme nakláňať pomocou servomotorov, ktoré ovládame napätím. Kamera dokáže rozpoznať polohu guľičky na základe kontrastu – filtra čierna vs. biela. V reálnom živote s podobným systémom sa môžeme stretnúť napríklad pri živom TV vysielaní pri prenasledovaní auta pred policajtmí. Cieľom je, aby kamera rozpoznala utekajúce auto a prostredníctvom otáčania musí prenasledovať pohyblivujúci objekt. V tomto prípade máme podobný problém, len akurát neotáčame kameru, ale celú plochu.

V dnešnej dobe existuje veľmi veľa prístupov, ako môžeme systémy riadiť. Jedným zo základných prístupov k riadeniu je riadenie pomocou spätnoväzbovej regulácie, kde hlavnú časť riadenia tvorí PID regulátor. Tento spôsob riadenia sa čím ďalej tým viac dáva do úzadia, pretože nie je schopné konkurovať vyspelejším metódam návrhu regulátorov. Tieto vyspelejšie návrhy sa formujú už od minulého storočia, nakoľko veda a technika spravili výrazný krok dopredu. Cieľom je optimalizovať chod prevádzok, zariadení a to z viacerých uhlov pohľadov či už optimalizácie zisku, spotreby paliva či elektrickej energie. Preto v aj systéme „Gulička na ploche“ sa budeme snažiť navrhnúť optimálne riadenie. Najhlavnejší rozdiel od ostatných bežných riadení je, že vieme optimalizovať premenné a vybrať si najlepšie riadenie. Funkcia, pre ktorú sa má stanoviť extrém, sa nazýva účelovou funkciou, ktorú v optimalizácii chceme minimalizovať. Ďalej sa dajú zahrnúť ohraničenia, ktoré sú súčasťou každého reálneho systému. Dôležité sú preto, aby sme mohli zabezpečiť technologické parametre systému, napríklad v našom prípade poloha guľičky nemôže byť väčšia ako 1 a menšia ako -1. Bežné PID regulátory majú jednoduchú štruktúru, ktoré vieme ľahko navrhnúť a implementovať. Z tohto dôvodu sú často využívané aj v priemysle.

V druhej časti bude formulovaný návrh PID regulátora pomocou Matlabovského toolboxu PIDTOOL. PID regulátor je spätnoväzbový regulátor, ktorý podľa žiadanej hodnoty a výstupu meranej veličiny vypočíta regulačnú odchýlku. Základným rozdielom medzi klasickým PID a stavovým regulátorom spočíva v tom, že stavový regulátor využíva pre

výpočet akčných zásahov všetky stavové veličiny. Je teda nevyhnutnou podmienkou poznať všetky informácie o stavových veličinách. To však nie je vždy celkom možné. Nie vždy sme schopní merať všetky stavové veličiny a to buď z fyzikálneho alebo aj ekonomického hľadiska. Tento problém môžeme vyriešiť pozorovaním a odhadom stavu, čo potrebujeme pre presné riadenie. Do pozorovača vstupujú informácie o vstupoch a výstupoch riadeného procesu a z pozorovača vystupujú odhadované stavy systému. Pokiaľ sú známe všetky stavy vieme riadiť reálny systém.

V ďalšej časti bude navrhnutý LQR regulátor, ktorý dokáže vypočítať optimálne akčné zásahy na nekonečnom predikčnom horizonte. Problém pri tomto návrhu riadenia je ten, že nedokážeme vziať do úvahy technologické parametre. Pri návrhu regulátora sa tieto parametre označujú ako ohraničenia. Pre stavový regulátor musíme vyladiť vektor K a váhové matice Q a R . Zvyšovaním matice Q penalizujeme stavy systému a zvyšovaním matice R penalizujeme vstupy systému. Na to aby nám regulátor správne riadil je veľmi dôležité nájsť správny pomer týchto matic a zistiť či je systém so zapojeným stavovým regulátorom asymptoticky stabilný. Ďalej LQR regulátor je rozšírený o integračnú zložku pre odstránenie trvalej regulačnej odchýlky.

Nutnosť zavedenia ohraničení do návrhu regulácie viedlo k návrhu takej metódy, ktorá by toto dokázala zohľadniť. Jedná sa o MPC, ktorý sa stal jedným z najvyspelejších systémov riadenia. Jeho najväčšou výhodou je implementácia už spomínaných ohraničení. V našom návrhu sú ohraničenia dôležité, pretože plocha je ohraničená v každom smere súradnicovej sústavy. Taktiež pomocou MPC môžeme minimalizovať opotrebovanie zariadenia, spotrebu energie. Touto formuláciou riadenia vieme zabezpečiť optimálne akčné zásahy, čím zabezpečujeme, aby servomotory sa neprešmykovali a náhodou sa neroztrhlo lanko. Na systém boli navrhnuté 2 formulácie regulátora – hustá a delta u formulácia .

1. Opis zariadenia „gulička na ploche“

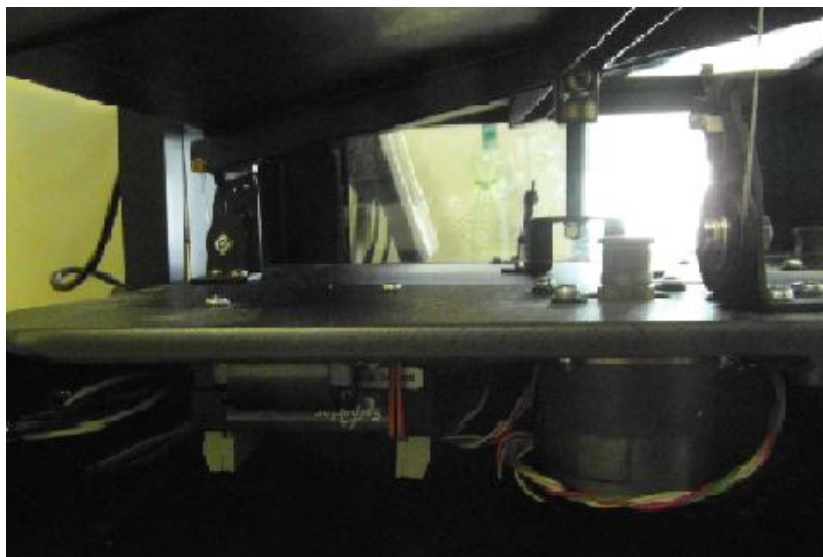
Zariadenie „gulička na ploche“ na obr. 1 vyrobila firma HUMUSOFT spol. s r.o. Najdôležitejším cieľom tohto zariadenia je, aby študenti si mohli vyskúšať ich vlastné teoretické vedomosti na reálnom procese. Toto zariadenie pozostáva z plochy, zo servo motorčekov s lanom, ktoré umožnia pohyb plochy a z kamery, ktorá sleduje, kde sa nachádza guľička. Plochu môžeme nakláňať v dvoch nezávislých smeroch podľa osí x a y . Polohu guľičky môžeme zistiť pomocou CCD kamery. Tento senzor je pripútaný nad pohyblivou časťou plochy. Tento systém je prepojený s PC, čo nám umožňuje riadiť tento systém prostredníctvom vývojového programu MATLAB.



Obr. 1 : Zariadenie „gulička na ploche“

Hlavné časti zariadenia:

- **Kamera** – je optické zariadenie, ktoré slúži na vizuálne zobrazenie polohy guličky. Poznáme dve hlavné skupiny kamier: CCD (Charge Coupled Device) a CMOS (Complementary Metal Oxide Semiconductor). Najväčší rozdiel medzi CCD a CMOS kamier sa nachádza vo výrobnom procese. Výroba CCD kamier je zložitejšia a finančne náročnejšia. Na druhej strane výroba CMOS obvodov je jednoduchšia a lacnejšia. CMOS senzory a ich používanie je výhodnejšie vtedy, keď chceme mať vyššiu rýchlosť pri nižšom rozlíšení. Okrem toho sa dá preferovať aj to, že má oveľa nižšiu spotrebu. Nevýhodou je zlá citlivosť na svetlo. Táto nevýhoda nie je problémom, pokiaľ kamera je používaná v dobre osvetlenej miestnosti. V prípade zle osvetlenej miestnosti je kvalita obrazu horšia. CCD kamery majú lepšiu svetelnú citlivosť, čo sa prejavuje tým, že obraz má lepšiu kvalitu aj pri zlých podmienkach.. Dnes, však vzhľadom k technickým zlepšeniam CMOS senzory ukazujú takmer rovnaké hodnoty parametrov ako CCD kamery. Z toho vyplýva, že na trhu senzorov postupne ich používanie je rovnaké. V našom systéme CCD kamera je umiestnená nad pohyblivou časťou systému. Kamera sníma polohu guličky, čo sníma pomocou kontrastu čierna vs. biela, pretože biela farba odráža svetlo. Časová perióda pre aktualizáciu snímania pozície guličky sa môže nastaviť od 5 – 30 fps (frames per second – počet snímok za sekundu). Je to frekvencia, čo označuje koľko snímok má kamera vyrobiť za sekundu.
- **Servosystém (servo motorčeky)** – servomotor je znázornený na obr. 2. Dnes servomotory sú široko dostupné, majú vysokú presnosť na riešenie rôznych polohovacích úloh. Hlavná charakteristika servomotorov je, že majú malé časové konštanty, čo ukazuje rýchlu dynamiku systému. Servomotor je reprezentovaný ako ideálny integrátor, ktorý vykazuje konštantnú rýchlosť. Používajú sa najčastejšie napríklad na riadenie ramien priemyselných robotov a na riadenie polohy plochy. Musíme dávať pozor, aby sme nedávali moc veľké akčné zásahy, aby servosystém sa nezadrhol a neopotrebovával sa.



Obr. 2: Servomotor a podpera



Obr. 3: Čidlo

- **Plocha** – plocha má výšku 400 mm a šírku takisto 400 mm, ktorá je podopretá na strede plochy. Táto podpera sa pohybuje do 4 strán. Pod plochou sú umiestnené čidlá, ktoré zabráňujú tomu, aby plocha sa pretočila viac ako je maximálna a minimálna hranica. Čidlá vidíme na obr. 3. Maximálna a minimálna limita uhlu plochy v radiánoch je $\alpha_{\max} = -\alpha_{\min} = \pm 0,1878$ rad.

Radián (rad) je v sústave SI jednotka rovinného uhla. Prepočet z uhlu na radián sa dá realizovať podľa vzťahu $1^\circ = \frac{\pi}{180^\circ} = 0.01745 \text{ rad}$. Ak chceme vypočítať maximálnu hranicu uhlu plochy v jednotkách uhlu, tak treba vydeliť tieto údaje $\varphi = \frac{0.1878}{0.17645} = 10.7659^\circ$. Táto hodnota približne reprezentuje ± 300 krokov, čo budeme používať aj v schémach pri kalibrácii plochy. Z toho sa vyplýva, že plocha sa môže otočiť vzhľadom na rovnovážnu polohu maximálne o 21.53176° .

1.1 Model systému

Systém „gulička na ploche“ je dynamický systém 3. rádu v jednej osi. Zahŕňa šesť stavov (pozícia, rýchlosť a naklonenie v každom smere súradníc), 2 vstupy (napätia servomotorov) a 2 výstupy (pozícia v osi x a y). Osi môžu byť riadené nezávisle od seba. Informáciu o polohe guličky dostávame do PC prostredníctvom kamery. Systém je diskretný, pretože servomotory a kamera fungujú v diskretnom čase. Vstupné údaje pre program Matlab sú nasnímané obrázky z kamery, z ktorých sú vypočítané súradnice guličky y_x , y_y . Uhly α a β ukazujú uhly naklonenia plochy. Na obr. 4 je znázornená schéma systému „gulička na ploche“.

Pre odvodzovanie všeobecného modelu systému používame prístup variačného modelu, bez servosystému pre naklonenie plochy, ktorý je modelovaný zvlášť. Všeobecný model je odvodený z Euler – Lagrangeovej rovnice:

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

kde jednotlivé premenné majú tieto význam:

T – kinetická energia systému

$\dot{q}$ – prvá derivácia i-tej súradnice

q – generovaná i-tá súradnica

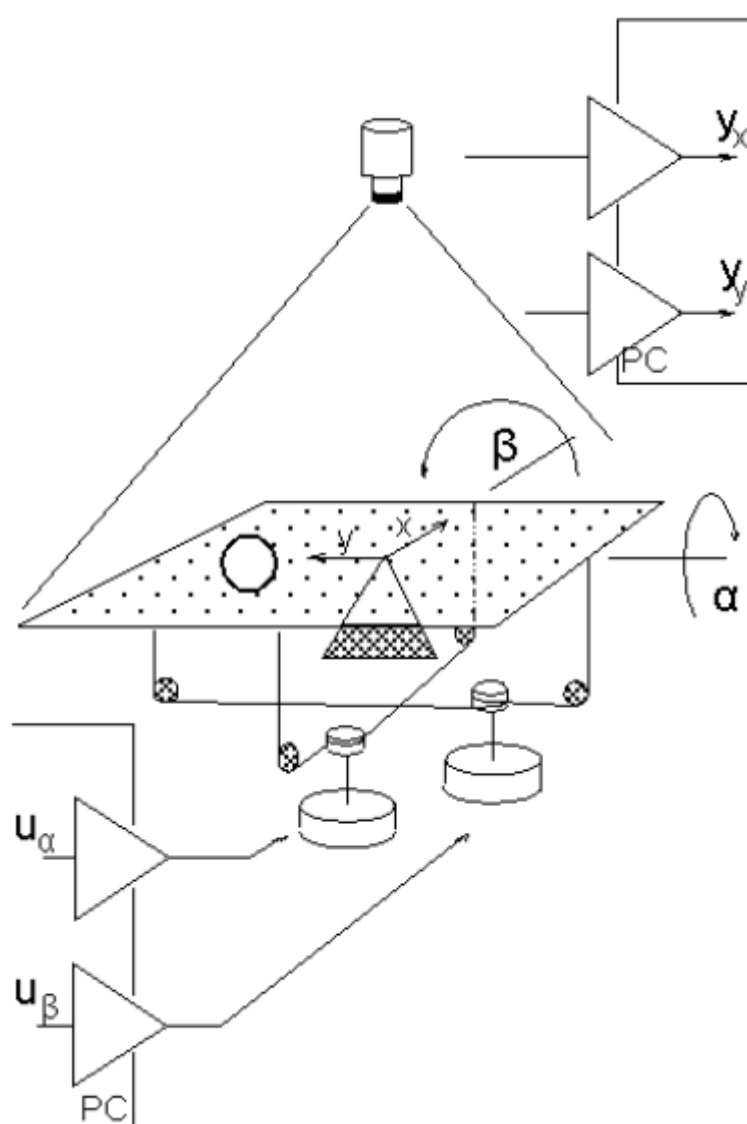
V – potenciálna energia systému

Q_i – generovaná i-tá sila

System má štyri stupne voľnosti (4DoF – Degrees of Freedom), dva v pohybe guľičky a dva pri naklonení plochy. Vygenerované koordináty x a y , ktoré sú určené z pozície guľičky súvisia s uhlami plochy α a β .

Takže vygenerované koordináty systému vychádzajú z týchto premenných:

$$q_1 = y_x \quad q_2 = y_y \quad q_3 = \alpha \quad q_4 = \beta$$



Obr. 4: Schéma modelu Gulička na ploche [2]

To znamená, že môžeme si vybrať z nasledujúcich parametrov:

- x, y – súradnice guľičky na ploche [-] $x, y \in (-1, 1)$
- r – polomer guľičky [m]
- v – rýchlosť guľičky [m/s]
- ω – uhlová rýchlosť pohybujúcej guľičky [rad/s]
- α, β – uhlový sklon plochy [rad].

Potenciálna energia sa vypočíta zo vzťahu:

$$V = mgh = mg(x \sin \alpha + y \sin \beta) \quad (2)$$

kde m je hmotnosť guľičky, g je gravitačné zrýchlenie a h je výška telesa nad podložkou.

Vygenerovaná sila je daná točivým momentom servomotora:

$$Q_\alpha = F_\alpha d \cos \alpha \quad (3)$$

$$Q_\beta = F_\beta d \cos \beta \quad (4)$$

kde d je vzdialenosť medzi stredovým bodom a miestom, kde servosystém dáva akčné zásahy na plochu, F je sila.

Majme lineárny, stabilný, kontinuálny systém opísaný so vstupno – výstupnou diferenciálnou rovnicou s nulovými začiatočnými podmienkami:

$$a_n y^{(n)}(t) + a_{n-1} y^{(n-1)}(t) + \dots + a_1 y'(t) + a_0 y(t) = b_m u^{(m)}(t) + b_{m-1} u^{(m-1)}(t) + \dots + b_1 u'(t) + b_0 u(t) \quad (5)$$

$$y(0), y'(0), \dots, y^{(n-1)}(0) \Rightarrow \text{začiatočné podmienky}$$

Prenos je podiel Laplaceového obrazu výstupnej veličiny a Laplaceového obrazu vstupnej veličiny pri nulových začiatočných podmienkach.

$$G(s) = \frac{Y(s)}{U(s)} \quad (6)$$

Všeobecný tvar prenosovej funkcie je

$$G(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} \quad (7)$$

Prenosová funkcia systému je odvodená linearizáciou modelu, je systém 3. rádu s astatizmom 2. rádu s charakteristickými nelinearitami:

- obmedzovač rýchlosti
- saturácia
- časová konštanta, ktorá závisí od amplitúdy a frekvencie vstupného signálu.

Prenosová funkcia pre tento systém pre obe súradnice:

$$G(s) = \frac{Y_x}{U_\alpha} = \frac{K_\alpha K_b K_x}{s^2 (T_m s + 1)} = \frac{K}{s^2 (T_m s + 1)} \quad (8)$$

$$G(s) = \frac{4.803}{s^2 (0.187 s + 1)} \quad (9)$$

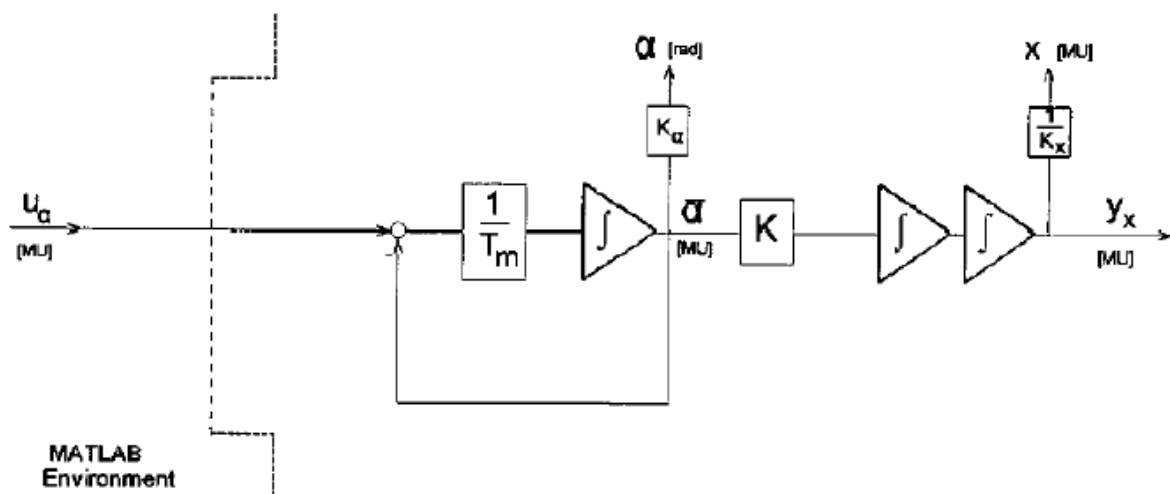
kde jednotlivé premenné symbolizujú tieto konštanty:

- $K_\alpha = 0.1878 \text{ rad/MU}$ – statické zosilnenie servosystému
- $K_b = 4.6 \text{ ms}^{-2}/\text{rad}$ – zosilnenie systému guľička-plocha
- $K_x = 5.56 \text{ MU/m}$ – konštanta senzora polohy guľičky
- $T_m = 0.187 \text{ s}$ – časová konštanta servosystému

Normalizované premenné:

- $K = 4.803 \text{ s}^{-2}$ – zosilnenie
- $\omega = 1.338 \text{ s}^{-1}$ – nominálna rýchlosť servosystému
- $T_m = 0.187 \text{ s}$ – časová konštanta servosystému

Na nižšie uvedenej schéme na obr. 5 je nakreslená linearizovaná dynamická schéma systému „guľička na ploche“ so servosystémom. Na blokovej schéme vidíme, že do systému vchádza nejaká presná hodnota napätia u_α .



Obr. 5: Linearizovaná dynamika systému guľička na ploche zahŕňajúca servosystém [2]

1.2 Stavová reprezentácia systému

Model daného procesu je odvodený a podrobne opísaný v príručke [2]. Systém „Guľička na ploche“ je systém 3. rádu v smere jednej osi, čo je spomenuté v kapitole 1.1. Čiže stavová reprezentácia tohto systému môže byť napísaná ako stavový vektor pre os x a y, čo dokumentujú rovnice (10) a (11). Prvý stav označujeme ako y_x , čo je pozícia guľičky. Táto veličina je bezrozmerná a vieme vypočítať pomocou kamery. Druhý stav označujeme ako $\dot{y}_x$, čo je rýchlosť guľičky. Táto veličina má jednotku s^{-1} a vieme vypočítať z polohy guľičky a to pomocou diskretizácii, čiže vypočítame pozíciu v danom perióde vzorkovania, túto pozíciu oneskoríme, spravíme ich rozdiel a následne vynásobíme $1/T_s$. Tretí stav označujeme ako $\bar{\alpha}_x$, čo je naklonenie plochy. Táto veličina je opäť bezrozmerná. Túto veličinu nevieme vypočítať, preto potrebujeme pozorovač stavov, čo opisujeme v nižšie uvedenej kapitole.

Stavový vektor pre os x:

$$x_x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_x \\ \dot{y}_x \\ \bar{\alpha}_x \end{bmatrix} \begin{bmatrix} \text{pozícia gul.} [-] \\ \text{rýchlosť gul.} [s^{-1}] \\ \text{naklonenie plochy} [-] \end{bmatrix} \quad (10)$$

Stavový vektor pre os y:

$$x_y = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_y \\ \dot{y}_y \\ \bar{\alpha}_y \end{bmatrix} \begin{bmatrix} \text{pozícia gul.} [-] \\ \text{rýchlosť gul.} [s^{-1}] \\ \text{naklonenie plochy} [-] \end{bmatrix} \quad (11)$$

Vstupy z_u sa určujú podľa veľkosti uhla, ktoré sú posielané do riadeného systému. Hodnoty vstupov môžu byť medzi intervalom:

$$z_u \in <-1,1> \quad (12)$$

Výstupy z_y sa zistia podľa pozície guľičky. Hodnoty výstupov môžu byť takisto medzi intervalom:

$$z_y \in <-1,1> \quad (13)$$

Stavová reprezentácia daného procesu teda má nižšie uvedené tvary, čo je opísané v rovniciach (14 - 16). Daný systém je viacrozmerný v zmysle, že riadime zvlášť os x, tak aj os y. Z tohto dôvodu musíme zostrojiť nový model, kde je už táto skutočnosť zahrnutá. Za predpokladu, že doska je rovná a oba motory majú rovnakú charakteristiku, môžeme dôjsť k záveru, že správanie sa systému pre obe osi bude rovnaký. Preto náš model môžeme všeobecne napísať ako v rovnici (21).

Stavový opis pre MIMO systém:

$$\dot{x}_x = Ax_x + Bu_x \quad \dot{x}_y = Ax_y + Bu_y \quad (14)$$

$$y_x = Cx \quad y_y = Cx$$

$$\begin{bmatrix} \dot{x}_x \\ \dot{x}_y \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix} \begin{bmatrix} x_x \\ x_y \end{bmatrix} + \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \quad (15)$$

$$\begin{bmatrix} y_x \\ y_y \end{bmatrix} = \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} x_x \\ x_y \end{bmatrix} \quad (16)$$

Po dosadení jednotlivých matic do všeobecného modelu dostaneme jednotlivé stavové reprezentácie pre os x (17) a pre os y (18). Zosilnenie K dostaneme vynásobením K_a , K_b a K_x , kde K_a je statické zosilnenie servosystému, K_b je zosilnenie systému guľička-plocha a K_x je konštanta senzora polohy guľičky. Ďalej časová konštanta T_m je časová konštanta servosystému. Hodnoty konštánt sú napísané vo vyššie uvedenej kapitole 1.1.

$$\begin{bmatrix} \dot{y}_x \\ \ddot{y}_x \\ \dot{\alpha}_x \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & K \\ 0 & 0 & -\frac{1}{T_m} \end{bmatrix} \begin{bmatrix} y_x \\ \dot{y}_x \\ \alpha_x \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{T_m} \end{bmatrix} u_x \quad (17)$$

$$\begin{bmatrix} \dot{y}_y \\ \ddot{y}_y \\ \dot{\alpha}_y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & K \\ 0 & 0 & -\frac{1}{T_m} \end{bmatrix} \begin{bmatrix} y_y \\ \dot{y}_y \\ \alpha_y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{T_m} \end{bmatrix} u_y \quad (18)$$

V rovnici výstupu matica C má nasledujúcu podobu, ktorá je uvedená v rovnici (19), a to kvôli tomu, lebo na výstupe sme zvedaví na polohu guľičky, čo v matici C je prvý prvok v prvom riadku a v prvom stĺpci.

$$y_x = [1 \ 0 \ 0]x \quad (19)$$

$$y_y = [1 \ 0 \ 0]x \quad (20)$$

$$\begin{bmatrix} \dot{y}_x \\ \ddot{y}_x \\ \dot{\alpha}_x \\ \dot{y}_y \\ \ddot{y}_y \\ \dot{\alpha}_y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & K & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{T_m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & K \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_m} \end{bmatrix} \begin{bmatrix} y_x \\ \dot{y}_x \\ \alpha_x \\ y_y \\ \dot{y}_y \\ \alpha_y \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{T_m} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \frac{1}{T_m} \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \quad (21)$$

Predchádzajúcu veľkú rovnicu môžeme jednoducho označiť ako v rovnici (22), kde máme novú maticu $\tilde{A}$ a maticu $\tilde{B}$.

$$\dot{z} = \tilde{A}z + \tilde{B}z_u \quad (22)$$

Ak dosadíme do zosilnenia a časovej konštanty výsledný stavový opis má takú podobu:

$$\begin{bmatrix} \dot{y}_x \\ \ddot{y}_x \\ \dot{\alpha}_x \\ \dot{y}_y \\ \ddot{y}_y \\ \dot{\alpha}_y \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.803 & 0 & 0 & 0 \\ 0 & 0 & -5.35 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4.803 \\ 0 & 0 & 0 & 0 & 0 & -5.35 \end{bmatrix} \begin{bmatrix} y_x \\ \dot{y}_x \\ \alpha_x \\ y_y \\ \dot{y}_y \\ \alpha_y \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 5.35 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 5.35 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \quad (23)$$

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} z \quad (24)$$

kde jednotlivé premenné sú:

y_x - pozícia guľičky pre os x

$\dot{y}_x$ - rýchlosť guľičky v osi x

α_x - naklonenie plochy v osi x

y_y - pozícia guľičky pre os y

$\dot{y}_y$ - rýchlosť guľičky v osi y

α_y - naklonenie plochy v osi y

Pretože servosystém a kamera pracujú v diskretnom čase, daný výsledný model treba diskretizovať. Pomocou funkcie *ss* v Matlabe spravíme stavový opis už pre veľké matice v rovnici (23) a následne pomocou funkcie *c2d* diskretizujeme s periódou vzorkovania $T_s = 0.15$. Podľa teórie, aby sme nestratili priebeh dynamiky, periódou vzorkovania môžeme zvoliť medzi pätinou a polovicou časovej konštanty systému. Časová konštanta je parameter s rozmerom času, ktorá sa vyskytuje na ľavej strane diferenciálnej rovnice systému.

Po diskretizování jednotlivé matice mají následující podobu:

$$\tilde{A}_d = \begin{bmatrix} 1 & 0.15 & 0.0421 & 0 & 0 & 0 \\ 0 & 1 & 0.4954 & 0 & 0 & 0 \\ 0 & 0 & 0.4482 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0.15 & 0.0421 \\ 0 & 0 & 0 & 0 & 1 & 0.4954 \\ 0 & 0 & 0 & 0 & 0 & 0.4482 \end{bmatrix} \quad (25)$$

$$\tilde{B}_d = \begin{bmatrix} 0.0120 & 0 \\ 0.2251 & 0 \\ 0.5518 & 0 \\ 0 & 0.0120 \\ 0 & 0.2251 \\ 0 & 0.5518 \end{bmatrix} \quad (26)$$

$$\tilde{C}_d = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (27)$$

$$\tilde{D}_d = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (28)$$

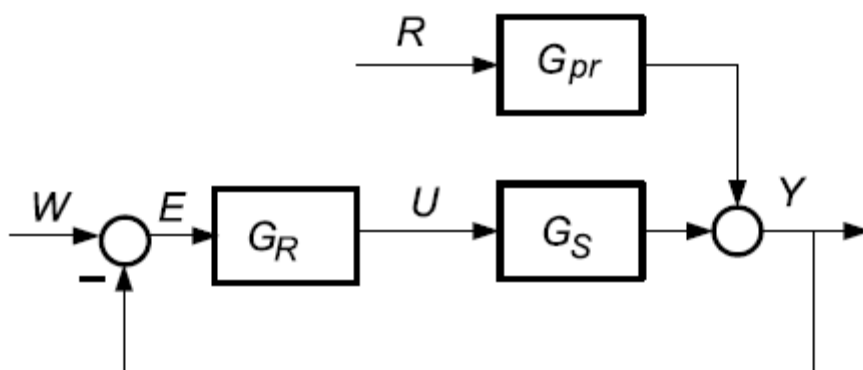
2. Teória

V tejto kapitole práce sa oboznámime s teóriou pre viaceré metódy riadenia. Túto kapitolu čerpáme z literatúry [3]. Najprv je opísaný PID regulátor, potom pozorovač, ďalej LQR a MPC. Tieto riadiace stratégie budú implementované na reálne zariadenie, preto sa musíme najprv oboznámiť s teóriou jednotlivých prístupov.

2.1 Teória PID regulátorov

PID regulátory sú technické zariadenia, ktoré majú svoj matematický model, ktorý sa môže regulovať počítačom. Je jeden z najjednoduchších druhov spätnoväzbového regulátora, ktorý podľa žiadanej hodnoty (w) a výstupu (y) meranej veličiny vypočíta regulačnú odchýlku (e). Regulačná odchýlka sa potom používa na určenie vstupu (u) do systému. Úlohou regulátora je tak vplývať na sústavu prostredníctvom akčnej veličiny ($vstupu - u$), aby sa dosiahol stav $e = 0$, čiže aby regulačná odchýlka bola nulová.

Bloková schéma URO pre syntézu regulátorov je znázornená na obr. 6:



Obr. 6: Bloková schéma URO [3]

PID regulátor sa skladá z troch zložiek – z proporcionálnej, integračnej a derivačnej časti. Zákon riadenia tzv. štruktúry bez interakcie:

$$u(t) = Z_R e(t) + \frac{Z_R}{T_I} \int_0^t e(t) dt + Z_R T_D \frac{de(t)}{dt} \quad (29)$$

$$u(t) = Z_R \left(e(t) + \frac{1}{T_I} \int_0^t e(t) dt + T_D \frac{de(t)}{dt} \right) \quad (30)$$

Prenos v kontinuálnom čase má tvar:

$$G_R(s) = Z_R + \frac{Z_R}{T_I s} + Z_R T_D s \quad (31)$$

Prenos v diskretnom čase má tvar:

$$G_R(z) = Z_R + \frac{Z_R}{T_I (1 - z^{-1})} + Z_R T_D (1 - z^{-1}) \quad (32)$$

kde Z_R je zosilnenie, T_I je integračná časová konštanta a T_D je derivačná časová konštanta.

Regulačná odchýlka sa dá vypočítať zo vzťahu $e(t) = w(t) - y(t)$, čiže dostaneme porovnaním žiadanej veličiny s výstupom. Z toho vyplýva, že trvalá regulačná odchýlka (TRO) je $e(\infty) = w(\infty) - y(\infty)$.

Charakteristická rovnica pre URO:

$$1 + G_S G_R = 0 \quad (33)$$

2.1.1 P zložka PID regulátora

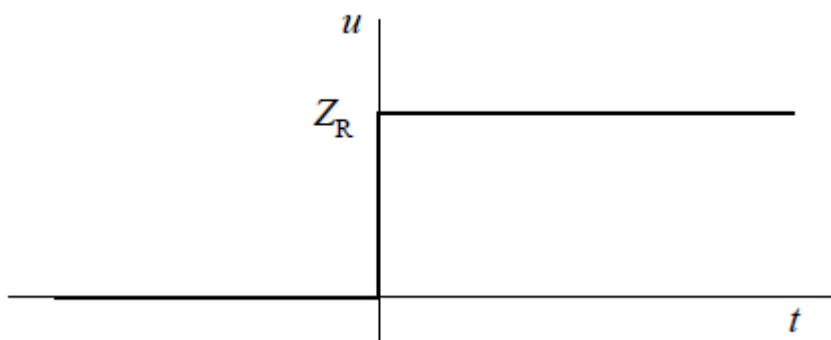
P zložku PID regulátora nazývame ako proporcionálna časť. Zákon riadenia P zložky je $u(t) = Z_R e(t)$, kde Z_R je zosilnenie.

Prenosová funkcia má tvar:

$$U(s) = Z_R E(s) \quad (34)$$

$$G_R = \frac{U(s)}{E(s)} = Z_R \quad (35)$$

Prechodová charakteristika P zložky je vykreslená na obr. 7:



Obr. 7: Prechodová charakteristika P zložky [3]

Z prechodovej charakteristiky vyplýva, že P-regulátor je veľmi rýchly, pretože okamžite zareaguje na skokovú zmenu. Je presný, lebo nedeformuje vstupný signál a následne tento vstupný signál Z_R zosilní. Nevýhoda P-regulátora je, že pracuje s trvalou regulačnou odchýlkou a minimalizuje kolísanie regulovanej veličiny. V porovnaní s ostatnými zložkami regulátora, P zložka má rýchlejšiu odozvu a reaguje oveľa rýchlejšie na zmenu žiadanej veličiny. V prípade, ak systém je MIMO (multi-input multi-output) systém, stáva sa zložitejším a vtedy sa môže akumulovať časový rozdiel. Kvôli tomu systém reaguje rýchlejšie, ale s odchýlkou. Túto odchýlku nazývame posunom a je nežiadajúci v procese. Toto posunutie sa dá vylúčiť s integračnou a derivačnou zložkou. Matematicky sa dá napísať nasledovne:

$$y(t) = Z_R e(t) + b \quad (36)$$

kde y je výstup z regulátora, Z_R je zosilnenie regulátora, e je regulačná odchýlka a b je posun. Z toho vyplýva, že P regulátor vykazuje lineárny vzťah medzi regulačnou odchýlkou systému a výstupom z regulátora systému. Vstupom do regulátora je žiadaná hodnota, skoková zmena a posun.

P zložka regulátora má vplyv na prítomnosť akčného zásahu, ktorý je vygenerovaný. To znamená, že regulátor vidí všetky akčné zásahy v prítomnosti, čiže v danej perióde vzorkovania. Keď Z_R je väčší, tak kladieme väčší vplyv na prítomnosť a keď naopak, keď Z_R je menší, tak kladieme menší vplyv na prítomnosť.

2.1.2 I zložka PID regulátora

I zložku PID regulátora nazývame ako integrálna časť. Zákon riadenia I zložky je

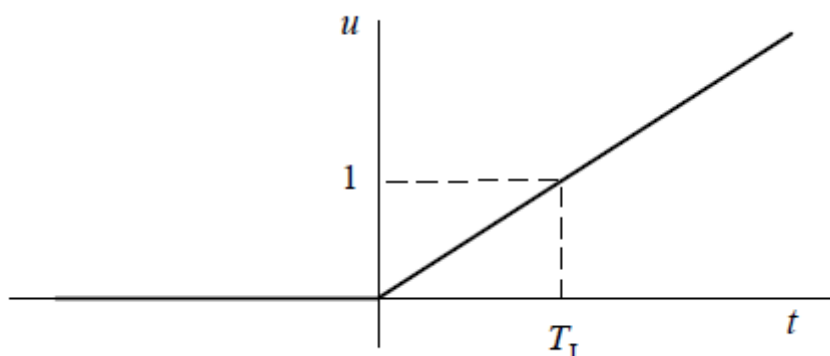
$$u(t) = \frac{Z_R}{T_I} \int_0^t e(t) dt, \text{ kde } T_I \text{ je integračná časová konštanta}$$

Prenosová funkcia má tvar:

$$U(s) = \frac{Z_R}{T_I} \frac{E(s)}{s} \quad (37)$$

$$G_R = \frac{U(s)}{E(s)} = \frac{Z_R}{T_I s} = \frac{\frac{Z_R}{T_I}}{s} \quad (38)$$

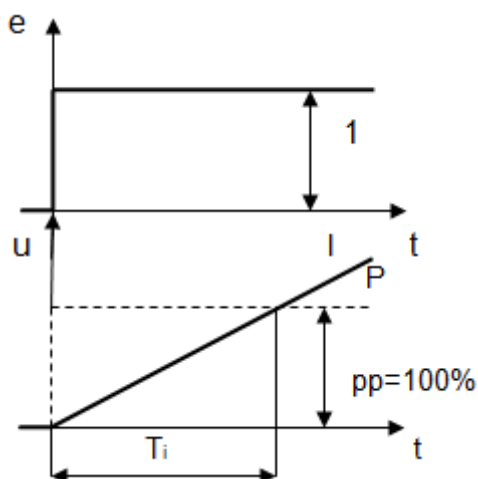
Prechodová charakteristika I zložky so $Z_R = 1$ je znázornená na obr. 8:



Obr. 8: Prechodová charakteristika I zložky [3]

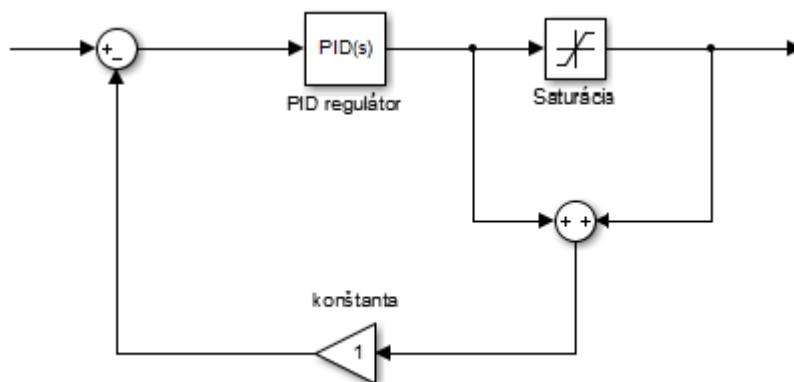
Zo statickej charakteristiky integračného regulátora možno vyčítať, že so znižujúcou sa integračnou časovou konštantou sa zväčšuje citlivosť a presnosť regulátora, zatiaľ čo jeho stabilita sa naopak znižuje. To znamená, že I zložka regulátora je nestabilný systém. Dynamická vlastnosť integračného regulátora sa vyjadruje s prechodovou charakteristikou. Z prechodovej charakteristiky je zrejma nestabilita integračného obvodu. Integračnú časovú konštantu T_I môžeme definovať ako čas, za ktorý výstupná veličina integračného regulátora dosiahne rovnakú hodnotu, akú by dosiahla, keby prenos regulátora bol iba proporcionálny

a pásmo proporcionality by bolo 100%, túto závislosť zobrazuje obr. 9. Jeho najvýznamnejšou vlastnosťou je skutočnosť, že pracuje bez trvalej regulačnej odchýlky.



Obr. 9: Grafické vyjadrenie definície integračnej časovej konštanty [3]

Negatívna odchýlka spôsobí to, že výstupný signál zo systému klesá, zatiaľ čo pozitívna odchýlka spôsobuje, že výstupný signál zo systému stúpa. I zložka regulátora na akčný zásah zareaguje pomalšie ako P zložka, lebo závisí od viacerých parametrov. Najväčšia nevýhoda I zložky je, že môže destabilizovať regulátor.



Obr. 10: Schéma pre antiwindup

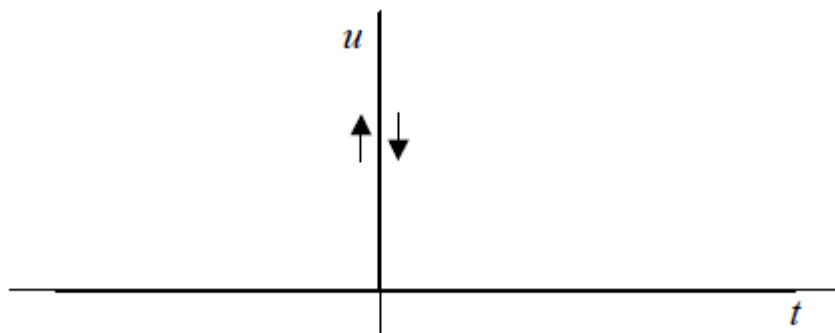
Keď riadenie je dlho na hornom ohraničení, tak pri kladnej regulačnej odchýlke sa integračná časť zvyšuje a z toho dôvodu aj stúpa výstupný signál. V tomto prípade môže vzniknúť windup efekt, pretečenie integrátora. V prípade, ak výstup príde na horné ohraničenie, regulátor by mal začať klesať, ale keď má integračnú časť regulátor si myslí, že je na hornom ohraničení, ale v skutočnosti je oveľa vyššie ako horné obmedzenie. Keď v Matlabe je použitá saturácia, čiže výstup je orezaný, regulátor si myslí, že ešte stále stúpa. Keď sa regulačná odchýlka premení na negatívnu hodnotu, tak výstupný signál z integračnej časti bude postupne klesať, až dovtedy, kým nebude dostatočne malý a regulátor sa nespamätá. Aby tento windup efekt nenastal, je potrebné používať antiwindup schému, ktorá je znázornená na obr. 10. Musíme porovnať výstup z PID regulátora pred a po saturácii a ďalej vynásobiť s konštantou a to odpočítat z regulačnej odchýlky.

I zložka PID regulátora je vlastne integrál z regulačnej odchýlky. Integrál odzrkadľuje minulosť toho procesu, čiže T_I dáva určitú váhu pre minulosť.

2.1.3 D zložka PID regulátora

D zložku PID regulátora nazývame ako derivačná časť. Zákon riadenia D zložky je $u(t) = Z_R T_D \frac{de(t)}{dt}$, kde T_D je derivačná konštanta

Prechodová charakteristika zložky D je znázornená na obr. 11:



Obr. 11: Prechodová charakteristika zložky D [3]

Prenosová funkcia má tvar:

$$U(s) = Z_R T_D s E(s) \quad (39)$$

$$G_R = \frac{U(s)}{E(s)} = Z_R T_D s \quad (40)$$

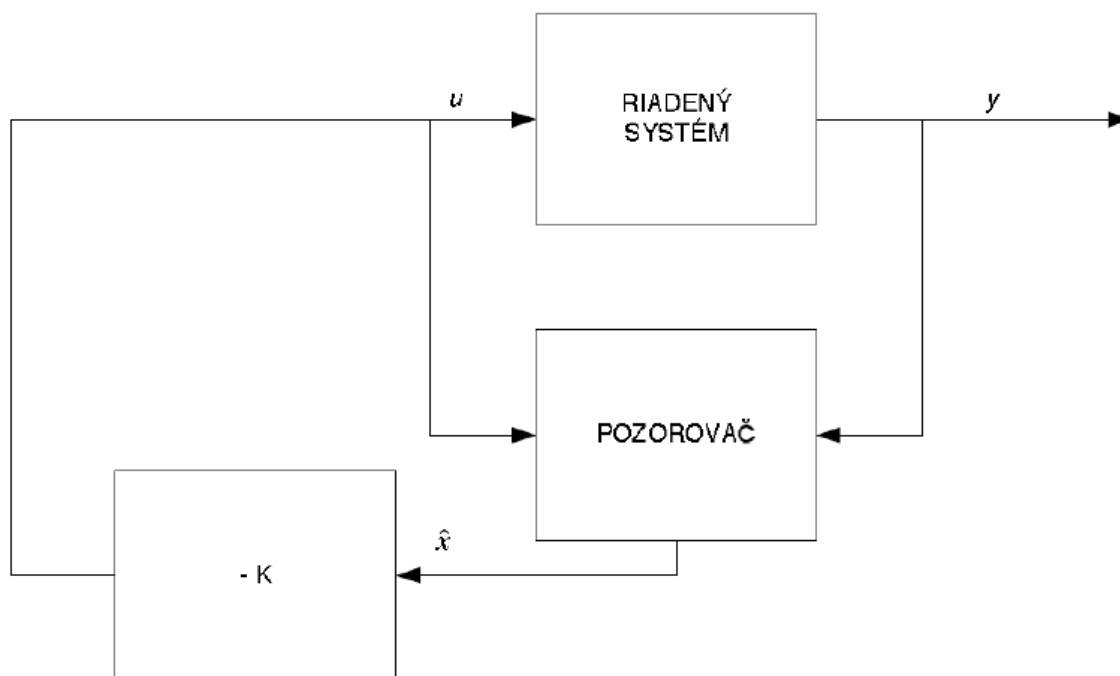
Statické vlastnosti derivačnej časti regulátora môžeme ovplyvňovať nastavením derivačnej časovej konštanty T_d . Z charakteristiky derivačného regulátora môžeme vyčítať, že so zväčšujúcou sa derivačnou časovou konštantou sa zväčšuje citlivosť regulátora, kým jeho stabilita sa naopak znižuje. Dynamické vlastnosti sa vyjadrujú prostredníctvom prechodovej charakteristiky. Derivačná časová konštanta je čas, za ktorý výstupná veličina derivačného regulátora dosiahne rovnakú hodnotu, akú by dosiahla, keby prenos regulátora bol iba proporcionálny a pásmo proporcionality by bolo 100%. Na rozdiel od proporcionálneho a integračného regulátora, derivačná časť neriadi systém do rovnovážneho stavu.

D zložka PID regulátora odzrkadľuje budúcnosť procesu, čiže ako sa bude správať proces v budúcnosti. Derivácia regulačnej odchýlky nám hovorí, či sa regulačná odchýlka znižuje alebo zväčšuje.

2.2 Teória pre návrh pozorovača

V tejto časti práce sa zaoberáme s pozorovačom stavu. Pozorovač stavu je najvhodnejšie používať vtedy, keď sa nedajú namerať stavy systému a je potrebné ich odhadnúť. Tieto neurčitosti stavov generujú takisto neurčitosti budúcich stavov. V prípade, že poznáme perfektne model systému a máme presnú znalosť vstupov systému, tak môžeme navrhnúť efektívny pozorovač stavov. Z toho sa vyplýva, že pozorovač použije model systému a stavy, ktoré sú namerané v minulosti. V praxi sa samozrejme toto nedá vždy zabezpečiť, preto sa snažíme čo najlepšie odhadnúť stavy $x(t)$. Poznáme viaceré pozorovače stavu, napríklad pozorovač plného rádu alebo Kalmanov filter. Pozorovač sa používa na odhad – určenie vektora stavových veličín $x(t)$ na základe nameraného vektora výstupu $y(t)$ a vstupu $u(t)$, kde chceme minimalizovať kvadrát chyby medzi meranými a odhadovanými stavmi.

Na obr. 12 je znázornená schéma s využitím pozorovača pre určenie odhadovaného stavového vektora $\hat{x}(t)$ pri riadení pomocou stavového regulátora K .



Obr. 12: Schéma systému riadenia pomocou stavového regulátora s využitím odhadu stavu

Kalmanov filter znižuje chybu odhadu a šumov. Diskrétny Kalmanov filter sa navrhuje pre pozorovaný systém, ktorý je opísaný stavovým opisom v tvare:

$$x_{k+1} = Ax_k + \xi_x \quad (41)$$

$$x_0 = \bar{x}_0 + \xi_{x0} \quad (42)$$

$$y_k = Cx_k + \xi \quad (43)$$

kde ξ_x je šum so strednou hodnotou 0 a s kovariančnou maticou Q , ξ_{x0} je začiatková podmienka so strednou hodnotou 0 a s kovariančnou maticou P_0 a ξ je šum na výstupe so strednou hodnotou 0 a s kovariančnou maticou R .

Pozorovač možno opísať na základe rovnice:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t)) \quad (44)$$

$$\hat{y}(t) = C\hat{x}(t) \quad (45)$$

Maticu P môžeme počítat pomocou Riccatiho rovnice:

$$\dot{P} = AP + PA^T + Q - PC^T R^{-1} CP \quad (46)$$

Maticu L vieme vypočítať podľa vzťahu

$$L = APC^T (R + CPC^T)^{-1} \quad (47)$$

2.3 Teória pre LQ riadenie

LQ riadenie je stavové spätnoväzbové riadenie. Označenie LQ ukazuje na to, že účelová funkcia je kvadratická a ohraničenia lineárne. Pri hľadaní optimálneho stavového regulátora K riešime LQ problém, ďalej snažíme sa nájsť minimum účelovej funkcie J . Účelová funkcia závisí od počiatkových podmienok x_0 a má 2 tvary, a to v spojitom alebo diskretnom čase:

- pre spojitý systém

$$J(x_0) = \min \left(x_N^T Q_N x_N + \int_0^\infty x^T Q x + u^T R u \right) \quad (48)$$

- pre diskretný systém

$$J_N(x_0) = \min \left(x_N^T Q_N x_N + \sum_0^\infty x_k^T Q x_k + u_k^T R u_k \right) \quad (49)$$

kde N ukazuje na to, že pracujeme v koncovom bode.

Pričom musí byť splnená rovnica stavu v tvare ohraničení v tvare rovnosti:

$$\dot{x} = Ax + Bu \quad (50)$$

Q, R sú váhové matice, pomocou ktorých sa dajú penalizovať buď stavy systému – zvyšovaním matice Q , resp. znižovaním matice R , alebo penalizovať vstupy systému – zvyšovaním matice R , resp. znižovaním matice Q . Vždy ide o nájdenie správneho pomeru

váhových matic Q, R . Matice Q, R sú pozitívne semidefinitné a symetrické, čiže pri invertovaní dostaneme tú istú maticu.

Keďže reálny systém pracuje v diskretnom čase, odvodenie pre stavový regulátor bude takisto v diskretnom čase. Stavový regulátor využíva na výpočet akčného zásahu (vstupu) informácie o všetkých stavových veličinách. Akčný zásah stavového spätnoväzbového regulátora sa dá vypočítať:

- pre spojitý systém: $u = -Kx(t)$
- pre diskretný systém: $u = -Kx(k)$.

Táto rovnica predstavuje zápornú proporcionálnu spätnú väzbu. K je stavový regulátor. Minimálnou požiadavkou je K navrhnuť tak, aby bola zaručená stabilita uzavretého regulačného obvodu. Na analytický výpočet pre K potrebujeme niekoľko krokov. Úlohou dynamického programovania bude nájsť takú optimálnu funkciu pre akčný zásah, ktorá závisí iba od aktuálneho stavu a od K . Aby sme prišli na optimálnu funkciu, musíme minimalizovať účelovú funkciu v čase k a stavovú rovnicu v čase $k+1$.

$$V_k(x) = \min[x^T Qx + u^T Ru + V_{k+1}(Ax + Bu)] \quad (51)$$

Ďalej túto funkciu musíme napísať v čase $N-1$, čiže o jeden krok pred ňou. V tomto prípade do V dosadíme stavovú rovnicu v čase $N-1$.

$$V_{N-1}(x) = \min[x_{N-1}^T Qx_{N-1} + u_{N-1}^T Ru_{N-1} + (Ax_{N-1} + Bu_{N-1})^T P_N (Ax_{N-1} + Bu_{N-1})] \quad (52)$$

Ak roznásobíme tento vzťah musíme na to dávať pozor, že sú to matice.

$$V_{N-1}(x) = \min(x_{N-1}^T Qx_{N-1} + u_{N-1}^T Ru_{N-1} + x_{N-1}^T A^T P_N Ax_{N-1} + x_{N-1}^T A^T P_N Bu_{N-1} + \quad (53)$$

$$+ u_{N-1}^T B^T P_N Ax_{N-1} + u_{N-1}^T B^T P_N Bu_{N-1})$$

Na to, aby sme našli minimum funkcie (53), musíme derivovať podľa u_{N-1} , aby sme dostali extrém danej funkcie.

$$\frac{\partial V_{N-1}}{\partial u_{N-1}} = 2Ru_{N-1} + 2B^T P_N Ax_{N-1} + 2B^T P_N Bu_{N-1} \quad (54)$$

Ďalej už len vyjadríme u_{N-1} , ale musíme na to dávať pozor, že pri maticiach nemôžeme deliť, len vynásobiť s inverznou maticou zľava alebo sprava.

$$u_{N-1} = -(R + B^T P_N B)^{-1} (B^T P_N A x_{N-1}) \quad (55)$$

kde akčný zásah na $N-1$ etape dostaneme spätnou väzbou stavu v $N-1$ etape cez zosilnenie K , ktoré vieme vypočítať zo vzťahu $(R + B^T P_N B)^{-1} (B^T P_N A)$. Toto nazývame ako optimálne riadenie v $N-1$ etape.

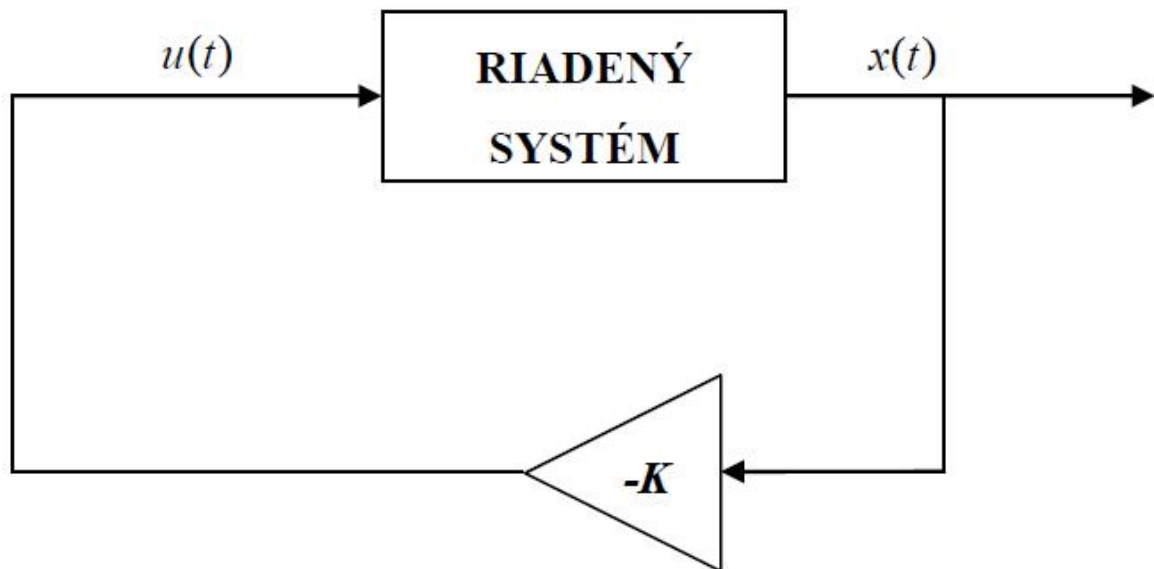
Na výpočet Riccatiho rovnice potrebujeme dosadiť vzťah optimálneho riadenia (55) do rovnice (53), čím dostaneme nasledujúci výraz.

$$V_{N-1} = x_{N-1}^T (A^T P_N A + Q) x_{N-1} + x_{N-1}^T A^T P_N B (B^T P_N B + R)^{-1} B^T P_N A x_{N-1} - 2x_{N-1}^T A^T P_N B (B^T P_N B + R)^{-1} B^T P_N A x_{N-1} \quad (56)$$

Z tohto vzťahu ak si vyberieme matice, ktoré sú pri optimalizovanými premennými, tak dostaneme výslednú rekurzívnu diskretnú algebrickú Riccatiho rovnicu.

$$P_{N-1} = Q + A^T P_N A - A^T P_N B (B^T P_N B + R)^{-1} B^T P_N A \quad (57)$$

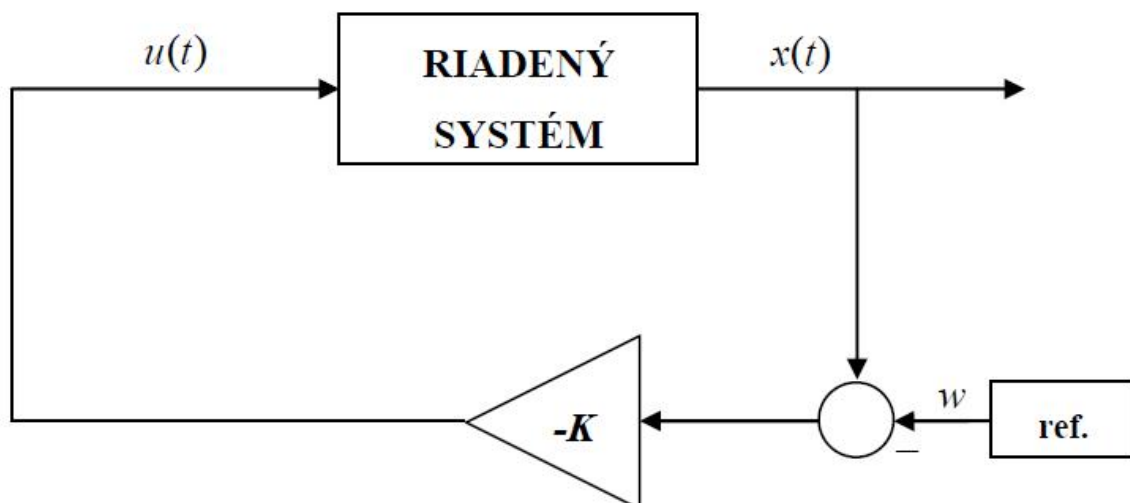
Ak vieme namerať všetky stavové veličiny riadeného procesu, čiže stavy sú merateľné, tak riadenie pomocou stavového regulátora LQR je podľa nasledujúcej schémy, ktorá je znázornená na obr. 13.



Obr. 13. Riadenie systému pomocou stavového regulátora

Ak nepoznáme všetky stavové veličiny (stavy sú nemerateľné) môžeme navrhnúť pozorovač podľa vyššie uvedených podmienkach a podľa vyššie zobrazenej schémy na obr. 13. V tomto prípade sa však namiesto vektorov $x(t)$, $x(k)$ používajú odhadované stavové vektory $\hat{x}(t)$, $\hat{x}(k)$.

Je zrejmé, že stavový regulátor riadi stavové a tým pádom aj vstupné veličiny na hodnotu 0. Preto ak chceme dosiahnuť inú žiadanú veličinu, je potrebné do schémy pridať nejakú inú referenciu. Dá sa to veľmi jednoducho navrhnuť, a to tak, že od $x(t)$, $x(k)$, resp. $\hat{x}(t)$, $\hat{x}(k)$ odrátame referenciu ešte predtým než nám táto hodnota vstúpi do stavového regulátora, viď na obr. 14.



Obr. 14: Riadenie systému pomocou stavového regulátora so žiadanou hodnotou

2.4 Teória pre MPC riadenie

V tejto časti práce sa budeme zaoberať s teóriou pre MPC riadenie. Hlavným cieľom tejto práce je, aby na reálny systém bolo navrhnuté MPC (Model Predictive Control) riadenie v dvoch formuláciách:

- MPC – hustá formulácia
- MPC – delta u formulácia

Prediktívne riadenie je používané v technike hlavne v petrochémií a v rafinérií. Prediktívne riadenie pracuje v diskretnom čase a využíva optimalizáciu v každej perióde vzorkovania, kde sa účelová funkcia minimalizuje. Na dosiahnutie tohto cieľa, proces musí byť s nejakým spôsobom namodelovaný a je implementovaná na konečnom predikčnom horizonte. Existuje na to niekoľko spôsobov, ale v tejto práci sa zaoberáme s vyššie uvedenými formuláciami. Podľa modelu sa dá predikovať, že pri určitých akčných zásahov, aké výstupné signály sa dajú očakávať. Problém je v tom, že optimálny akčný zásah sa musí vypočítať v každom perióde vzorkovania T_s .

Všeobecná matematická formulácia účelovej funkcie prediktívneho riadenia je:

$$J = \min \sum_{k=0}^{N-1} \left(\|Q_x x_{t+k}\|_p + \|Q_u u_{t+k}\|_p \right) \quad (58)$$

kde N je predikčný horizont, p je použitá norma, Q_x a Q_u sú váhové matice, a x a u sú optimalizované premenné. Riadený systém je opísaný nasledovným modelom:

$$x_{k+1} = Ax_k + Bu_k \quad \Rightarrow \text{rovnica predikcie} \quad (59)$$

$$x_t = x(t) \quad \Rightarrow \text{začiatočná podmienka}$$

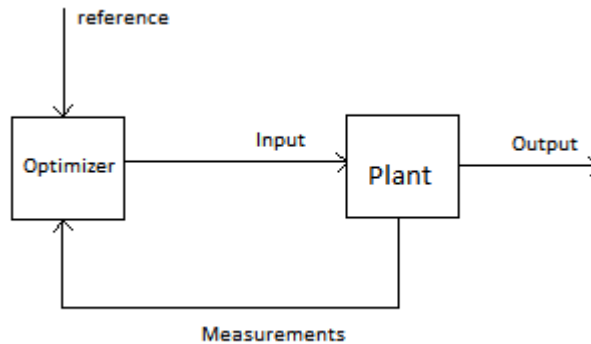
$$x_t \in X \quad \Rightarrow \text{stavové ohraničenia} \quad \Rightarrow Hx_t \leq K$$

$$u_t \in U \quad \Rightarrow \text{vstupné ohraničenia} \quad \Rightarrow Lu_t \leq M$$

Ak kladieme väčšiu váhu na Q_x , tak chceme penalizovať stavy a naopak, ak kladieme väčšiu váhu na Q_u , tak chceme penalizovať reguláciu minimalizácie energie.

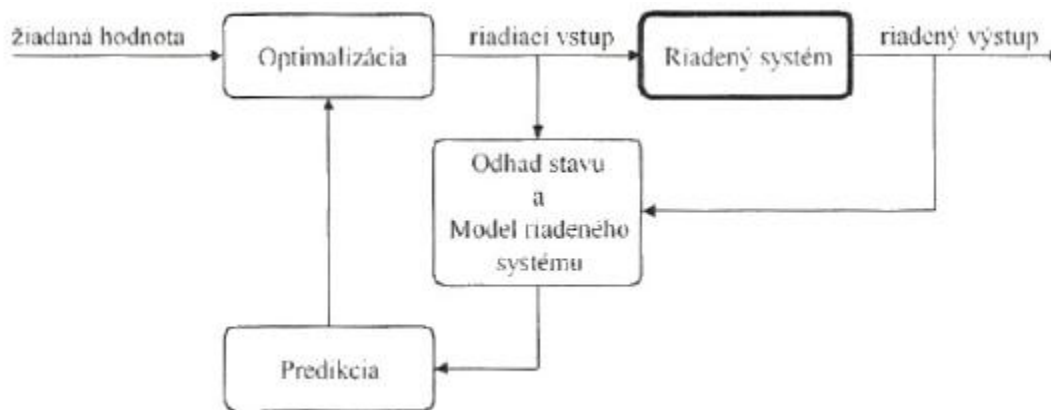
Normy sú indikátormi vzdialenosti od bodu $[0 \ 0]$. Poznáme 1 normu – pohybujeme sa po odvesnách trojuholníka, 2 normu – vzdialenosť od počiatku súradnicovej osi, ∞ normu – vyšetrujeme maximálnu súradnicu nejakého bodu v priestore.

V prípade ak máme elektrické zariadenia, perióda vzorkovania je v mikrosekundách, výpočet optimálneho akčného zásahu je náročné, čiže je potrebné mať výkonnejší hardvér, aby stihol vypočítať všetko načas. Toto je najhlavnejšia nevýhoda MPC. Výhody sú výkonnosť optimalizácie a do výpočtu sa dajú zahrnúť ohraničenia.



Obr. 15: Schéma MPC

Na obr. 15 je znázornená jednoduchá schéma MPC, kde je vidieť, že MPC je spätnoväzbové riadenie. Do optimizera (kde sa počíta optimalizácia) ide nejaká referencia (žiadaná hodnota), ktorú chceme, aby systém dosiahol a ďalej berie namerané stavy (measurements) z riadeného systému (plant). Vstupy (akčné zásahy) sú vygenerované s MPC regulátorom, ktoré sú potom zavedené do riadeného systému a následne sú pozorované jeho výstupy. Na obr. 16 vidíme všeobecnú štruktúru prediktívneho riadenia s pozorovačom.



Obr. 16. : Všeobecná štruktúra prediktívneho riadenia s pozorovačom

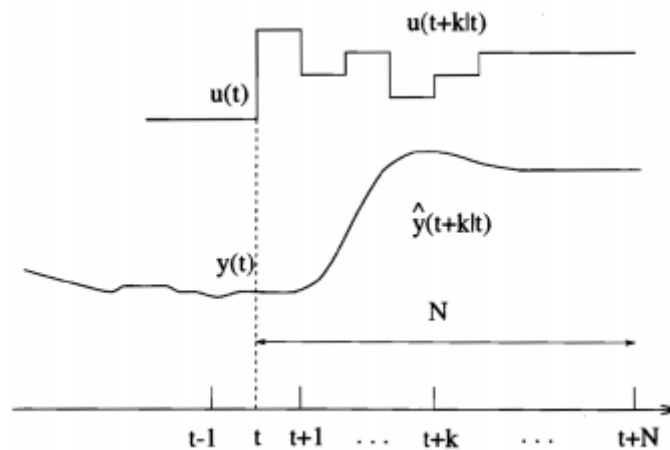
Jednotlivé pojmy plnia nasledovné funkcie:

- *Žiadaná hodnota* – je taká hodnota, ktorú chceme dosiahnuť.
- *Optimalizácia* – blok obsahuje účelovú funkciu a ohraničenia. Hlavným cieľom je vyhodnotiť optimalizáciou danú účelovú funkciu a neprekročiť obmedzenia.

- *Riadiaci vstup* – sú akčné zásahy, ktoré sú vygenerované s MPC regulátorom.
- *Riadený systém* – blok obsahuje matematický model riadeného systému. Riadený systém je systém, ktorý chceme riadiť.
- *Riadený výstup* – výstup z riadeného systému
- *Odhad stavu a model riadeného systému* – pozorovačom sa dá odhadnúť súčasné stavy pre blok predikcie. Je dôležité, aby pozorovač odhadoval presne, aby nenastala nestabilita systému. Tento blok porovnáva skutočné stavy s odhadovanými.
- *Predikcia* – tento blok poskytuje budúce informácie o vstupných údajoch a stavov.

Základy algoritmu MPC:

Je predpokladané, že v každej perióde vzorkovania regulátor pozná trajektóriu y_t, y_{t-1} , čiže regulátor má presné informácie o výstupe. Vieme, že by sme sa chceli dostať na žiadanú hodnotu a u_t je akčný zásah, čo sme doteraz urobili. Výstupy z modelu v budúcnosti sú značené ako $\hat{y}_{t+k}$ a ďalej vstupné údaje sú značené ako $u_{t+1} \dots u_{t+k}$.



Obr. 17: Princíp algoritmu MPC

Môžeme vidieť, že tak ako vo vstupoch, aj ako vo výstupoch je argument t . Preto je dôležitý, lebo algoritmus je vypočítaný v každom čase t . Takže to znamená, že v prípade $\hat{y}_{t+k|t}$

čas t znázorňuje budúci výstup v perióde vzorkovania $t+k$. Z toho vyplýva, že sa snažíme nájsť takú trajektóriu, takú predikciu výstupu, ktorá sa blíži k žiadanej hodnote. K tomu existuje nejaká optimálna trajektória vstupov. Ak je uzavretá slučka (closed loop) používame iba prvý vstup a v čase $k+1$ túto procedúru zopakujeme, čiže opäť vyrátame riadenie u (akčný zásah) do budúcnosti a opäť aplikujeme len prvý vypočítaný element (vstup). Na obr. 17 je znázornený ako je spracovaný algoritmus v čase t .

Výhody MPC:

- Je najvyspelejším návrhom na riadenie
- Dokáže pracovať s obmedzeniami
- Dokáže spracovávať poruchy tým, že v každej perióde vzorkovania zopakuje optimalizáciu stavov, dokáže sa pripraviť na to, čo bude v budúcnosti
- Dobré vlastnosti na ladenie
- Dá sa používať ľubovoľný model (lineárny, nelineárny...)
- Vie riadiť MIMO (multiple input multiple output – viaceré vstupy viaceré výstupy) systémy, vie zohľadniť ekonomiku procesu a vie priamo navrhnuť procesné ohraničenia.

Nevýhody MPC:

- V každej perióde vzorkovania používa optimalizáciu, kvôli tomu je potrebné používať výkonnejší procesor
- Pri zle navrhnutom MPC systém môže byť nestabilný
- Aby sme mohli správne navrhovať a implementovať, musíme naplno pochopiť prediktívne riadenie.

Druhy ohraničenia:

- *Fyzikálne* – parametre môžeme meniť v danom rozsahu a s danou rýchlosťou (napr. otvorenie ventilu)

- *Technologické* – stavové a výstupné veličiny nesmú byť väčšie ako nejaká maximálna hodnota veličiny a takisto nesmú byť menšie ako nejaká minimálna hodnota veličiny (napr. udržiavanie teploty v hraniciach, udržiavanie guľičky na ploche)
- *Bezpečnostné* – tieto veličiny nesmú prekročiť z bezpečnostných dôvodov svoje kritické hranice (napr. udržiavanie tlaku v reaktore, v rektifikačnej kolóne)
- *Stabilizujúce* – tieto obmedzenia garantujú stabilitu riadenia procesu.

Rozdiel medzi otvorenou a uzavretou slučkou:

- *Otvorená slučka* – V prípade ak predikčný horizont je napríklad 5, čiže z toho sa vyplýva, že sa dá získať akčný plán a aktuálne merania pre nasledujúcich 5 sekúnd. Ak je používaná otvorená slučka, tak je postupne implementovaný celý akčný plán, dovtedy, kým sa nevyčerpá. Ak akčný plán je vyčerpaný, tak je potrebné opäť získať nové merania systému, nový optimálny plán na rovnaké časové obdobie, čiže na 5 sekúnd. Potom zase je implementovaný po jednom, až kým dôjdeme na koniec. Toto sa opakuje, až kým sa riadenie nedostane na žiadanú hodnotu. Problém je v tom, že môže dôjsť k poruchám aj medzi tým keď sa ešte len implementuje akčný plán. Napríklad v druhej sekunde môže dôjsť nejaká prekážka, ktorá nebola predvídateľná a v štvrtej sekunde sa dá do nej vraziť a samozrejme v piatej sekunde nemáme čo riadiť. Kvôli neefektívnosti sa musí zaviesť riadenie s uzavretou slučkou.
- *Uzavretá slučka* – Na začiatok: postup je úplne rovnaký. Najprv sa musia zistiť aktuálne hodnoty stavových veličín a musí sa vypočítať optimálny akčný plán na N krokov do budúcnosti, ale namiesto toho aby bol implementovaný celý akčný plán, sa vyberie prvý akčný plán. Tento akčný plán je implementovaný do systému. V ďalšej perióde vzorkovania je vypočítaný nový stav a celý postup je opäť zopakovaný. Znova sa vyráta N krokov do budúcnosti, potom je implementovaný prvý krok plánu a je znovu vypočítaný nový stav. Ak je nejaká prekážka, tak sa dá zahrnúť do optimalizácie a prispôbiť sa k nej. Stále je implementovaná len prvá časť akčného plánu, ostatné neberieme do úvahy. Zahadzovanie nie je zníženie kvality riadenia, iba spôsob ako zaviesť spätnú väzbu.

Ak sa problém dá naformulovať ako konvexný optimalizačný problém, tak ho vieme efektívne riešiť. Najjednoduchšie riešenie konvexného optimalizačného problému je *lineárne a kvadratické programovanie*.

V prípade lineárneho programovania (LP) účelová funkcia je lineárna, ktorú chceme minimalizovať. Ohraničenie sú v tvare nerovnosti a v tvare rovnosti. Ohraničenia v tvare nerovnosti sú v tvare polytopov a v tvare rovnosti sú v tvare nadrovín. LP je taká optimalizačná úloha, v ktorej minimalizujeme lineárnu funkciu vzhľadom na lineárne ohraničenia.

$$J = \min \quad c^T x \quad \Rightarrow \quad \text{účelová funkcia} \quad (60)$$

$$\begin{aligned} s.t. \quad Ax &\leq B \\ Gx &= H \end{aligned} \quad \Rightarrow \quad \text{ohraničenia}$$

Vo všeobecnosti optimum leží na vrchole polytopu, ale ak polytop je zdeformovaný, optimum môže byť aj na hrane polytopu. Vtedy sa nastane, že problém má viaceré možné riešenie, z ktorých stačí vybrať len jedno riešenie.

Ak problém je zložitejší, tak môže byť riešený kvadratickým programovaním (QP). Účelová funkcia je kvadratická. Ohraničenia tak isto ako pri LP sú v tvare nerovnosti a v tvare rovnosti. Ohraničenia v tvare nerovnosti musia vzniknúť ako prienik konečných počtov polpriestorov (polytop) a ohraničenia v tvare rovnosti musia byť lineárne.

$$J = \min \quad x^T P x + 2Q^T x + R \quad \Rightarrow \quad \text{účelová funkcia} \quad (61)$$

$$\begin{aligned} s.t. \quad Ax &\leq B \\ Gx &= H \end{aligned} \quad \Rightarrow \quad \text{ohraničenia}$$

Pre rôzne x účelová funkcia definuje nejakú elipsu, čo sú jednotlivé vrstevnice účelovej funkcie. Pri QP optimum môže byť vo vnútri polytopu. V prípade ak polytop je posunutý, tak optimum môže ležať na hrane alebo na vrchole polytopu. Minimum tak môže byť všade, ale je unikátne, čiže naraz môže mať len jedno riešenie. Matice P , Q a R sú váhové matice, ale na konvexiu má vplyv len matica P . Musí byť pozitívne semidefinitná, čiže vlastné čísla musia byť nezáporné (kladné). Ak by bola záporná, tak účelová funkcia by mala konkávny tvar, k čomu sa ťažko nájst minimum.

2.4.1 Hustá formulácia

Existujú viaceré matematické formulácie MPC. Jedna z nich je hustá formulácia MPC. Účelová funkcia všeobecne je naformulovaná ako:

$$J = \min \sum_{k=0}^{N-1} \left(\|Q_x x_{t+k}\|_p + \|Q_u u_{t+k}\|_p \right) \quad (62)$$

kde N je predikčný horizont, p je použitá norma, Q_x a Q_u sú váhové matice, a x a u sú optimalizované premenné. Riadený systém je opísaný nasledovným modelom:

$$x_{k+1} = Ax_k + Bu_k \quad \Rightarrow \text{rovnica predikcie} \quad (63)$$

$$x_t = x(t) \quad \Rightarrow \text{začiatočná podmienka}$$

$$x_t \in X \quad \Rightarrow \text{stavové ohraničenia} \quad \Rightarrow Hx_t \leq K$$

$$u_t \in U \quad \Rightarrow \text{vstupné ohraničenia} \quad \Rightarrow Lu_t \leq M$$

kde $A \in R^{n \times n}$, $B \in R^{n \times m}$, $x_k \in R^n$ je vektor stavu a $u_k \in R^m$ je vektor vstupu.

Systém „Gulička na ploche“ má 6 stavov, 2 vstupy a 2 výstupy. Tieto premenné stanovia rozmery matíc.

Ak $p = 2$, tak účelová funkcia vyzerá nasledovne:

$$J = \min \sum_{k=0}^{N-1} \left(x_{t+k}^T Q_x x_{t+k} + u_{t+k}^T Q_u u_{t+k} \right) \quad (64)$$

Rovnica predikcie sa dá pretransformovať na nasledujúci tvar:

$$x_{t+k} = A^k x_t + \sum_{i=0}^{k-1} A^{k-i-1} B u_i \quad (65)$$

Na tento systém s predikčným horizontom 5 sa presne dá odvodiť nasledujúca rovnica predikcie a účelová funkcia:

- rovnica predikcie (66)

$$x_1 = Ax_0 + Bu_0$$

$$x_2 = Ax_1 + Bu_1 = A(Ax_0 + Bu_0) + Bu_1 = A^2x_0 + ABu_0 + Bu_1$$

$$x_3 = Ax_2 + Bu_2 = A(A^2x_0 + ABu_0 + Bu_1) + Bu_2 = A^3x_0 + A^2Bu_0 + ABu_1 + Bu_2$$

$$x_4 = Ax_3 + Bu_3 = A(A^3x_0 + A^2Bu_0 + ABu_1 + Bu_2) + Bu_3 = A^4x_0 + A^3Bu_0 + A^2Bu_1 + ABu_2 + Bu_3$$

$$X = \begin{bmatrix} I \\ A \\ A^2 \\ A^3 \\ A^4 \end{bmatrix} x_0 + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 0 & 0 \\ AB & B & 0 & 0 & 0 \\ A^2B & AB & B & 0 & 0 \\ A^3B & A^2B & AB & B & 0 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (67)$$

• účelová funkcia (68)

$$\begin{aligned} J &= (\tilde{A}x_t + \tilde{B}u)^T \tilde{Q}_x (\tilde{A}x_t + \tilde{B}u) + u^T \tilde{Q}_u u = (x_t^T \tilde{A}^T + u^T \tilde{B}^T) \tilde{Q}_x (\tilde{A}x_t + \tilde{B}u) + u^T \tilde{Q}_u u = \\ &= x_t^T \tilde{A}^T \tilde{Q}_x \tilde{A}x_t + x_t^T \tilde{A}^T \tilde{Q}_x \tilde{B}u + u^T \tilde{B}^T \tilde{Q}_x \tilde{A}x_t + u^T \tilde{B}^T \tilde{Q}_x \tilde{B}u + u^T \tilde{Q}_u u = \\ &= u^T (\tilde{B}^T \tilde{Q}_x \tilde{B} + \tilde{Q}_u) u + 2x_t^T \tilde{A}^T \tilde{Q}_x \tilde{B}u + x_t^T \tilde{A}^T \tilde{Q}_x \tilde{A}x_t \end{aligned}$$

• ohraničenia

$$\begin{bmatrix} HB & 0 & 0 & 0 & 0 \\ HAB & HB & 0 & 0 & 0 \\ HA^2B & HAB & HB & 0 & 0 \\ HA^3B & HA^2B & HAB & HB & 0 \\ HA^4B & HA^3B & HA^2B & HAB & HB \\ L & 0 & 0 & 0 & 0 \\ 0 & L & 0 & 0 & 0 \\ 0 & 0 & L & 0 & 0 \\ 0 & 0 & 0 & L & 0 \\ 0 & 0 & 0 & 0 & L \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \leq \begin{bmatrix} K \\ K \\ K \\ K \\ K \\ M \\ M \\ M \\ M \\ M \end{bmatrix} - \begin{bmatrix} HA \\ HA^2 \\ HA^3 \\ HA^4 \\ HA^5 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} x_0 \quad (69)$$

Ak používame túto formuláciu, tak máme znížený počet optimalizovaných premenných, lebo optimalizujeme len vstupy (akčné zásahy u). Takto sa dajú eliminovať substitúciou stavu x . Stratí sa riedka vlastnosť matíc, stanú sa z nich husté matice. V tomto prípade sa zmení aj počet optimalizovaných premenných. V riedkej formulácii počet

optimalizovaných premenných je $(N+1)n_x + Nn_u$, naopak v hustej formulácii počet optimalizovaných premenných je Nn_u .

Riedka formulácia:

- výhoda – málo nenulových prvkov, ľahko sa invertuje
- nevýhoda – počet optimalizovaných premenných je vyššia (u, x)

Hustá formulácia:

- výhoda – menej optimalizovaných premenných (u), trojuholníkové matice
- nevýhoda – matice sa zle invertujú, matice obsahujú v sebe veľmi malé aj veľké čísla

2.4.2 Delta u formulácia

Delta u formulácia prediktívneho riadenia je na odstránenie trvalej regulačnej odchýlky (TRO). Ak výstup je na referencii a delta u je v nule, tak nie je offset.

Offset sa môže nastat' keď je:

- Nekompletná stavová informácia
- Poruchy
- Rozdiel medzi modelom a reálnym procesom.

Účelová funkcia delta u formulácii vyzerá nasledovne:

$$J = \min \sum_{k=0}^{N-1} \left(\|Q_y (y_{t+k} - y_{ref})\|_p + \|Q_{\Delta u} \Delta u_{t+k}\|_p \right) \quad (70)$$

kde N je predikčný horizont, p je použitá norma (1,2 a nekonečno norma), Q_y a $Q_{\Delta u}$ sú váhové matice, a Δu je optimalizovaná premenná.

Ak $p = 2$, tak účelová funkcia vyzerá nasledovne:

$$J = \min \sum_{k=0}^{N-1} \left((y_{t+k} - y_{ref})^T Q_y (y_{t+k} - y_{ref}) + \Delta u_{t+k}^T Q_{\Delta u} \Delta u_{t+k} \right) \quad (71)$$

Na tento systém s predikčným horizontom 5 sa presne dá odvodiť nasledujúca rovnica predikcie a účelová funkcia:

- delta u

$$\Delta u_k = u_k - u_{k-1} \quad (72)$$

$$u_k = \Delta u_k + u_{k-1}$$

- rovnica predikcie

$$\begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} \Delta u_k \quad (73)$$

$$y_k = \begin{bmatrix} C & D \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} + D \Delta u_k$$

- ohraničenia

$$\begin{aligned} H_x x_k &\leq K_x \\ H_u u_k &\leq K_u \end{aligned} \quad \Rightarrow \quad \begin{bmatrix} H_x & 0 \\ 0 & H_u \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} \leq \begin{bmatrix} K_x \\ K_u \end{bmatrix} \quad (74)$$

3. Riadenie reálneho systému

V tejto kapitole budeme riadiť reálny systém „Gulička na ploche“. Najprv navrhujeme a implementujeme riadenie s PID regulátorom, ďalej s LQR regulátorom a naposledy s MPC regulátorom.

3.1 Návrh PID regulátora

Na riadenie je používaný PID regulátor. Riadenie uzavretého regulačného obvodu je zrealizovaný pomocou GUI grafického užívateľského rozhrania *pidtool*. Najprv je potrebné spraviť stavový opis systému pre obe súradnice x a y z modelu jednej súradnice a potom ho diskretizovať:

```
A=[0 1 0;0 0 4.803;0 0 -5.35]
B=[0;0;5.35]
C=[1 0 0]
D=[0]

Ak=kron(eye(2),A)
Bk=kron(eye(2),B)
Ck=kron(eye(2),C)
Dk=kron(eye(2),D)    %spojity

Ts=0.15
sys = c2d(ss(Ak,Bk,Ck,Dk),Ts)

Ad=sys.a
Bd=sys.b
Cd=sys.c
Dd=sys.d              %diskretny

vlastne_c=eig(A)
vlastne_c=eig(Ad)
```

Na odvodenie spojitého stavového opisu pre obe osi x a y naraz je používaný príkaz *kron*. Keď matica A je $m \times n$ a matica B je $p \times q$, tak Kroneckerov produkt z matíc A a B je $mp \times nq$.

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ a_{21}B & a_{22}B & \dots & a_{2n}B \\ \vdots & \vdots & & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{bmatrix} \quad (75)$$

V tomto prípade každú maticu stavového opisu je potrebné zväčšiť o jednotkovú maticu. Takže na diagonále jednotkovej matici sú jednotlivé matice modelu a ostatné prvky sú vyplnené s nulami. Nemôžeme zabudnúť, že na mieste jednotky jednotkovej matici je nová matica, takže aj tie vyplnené nuly budú matice.

$$A=[0 \ 1 \ 0;0 \ 0 \ 4.803;0 \ 0 \ -5.35]$$

$$Ak=kron(eye(2),A)$$

$$A=\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 4.8030 \\ 0 & 0 & -5.3500 \end{bmatrix} \Rightarrow Ak=\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.8030 & 0 & 0 & 0 \\ 0 & 0 & -5.3500 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4.8030 \\ 0 & 0 & 0 & 0 & 0 & -5.3500 \end{bmatrix} \quad (76)$$

Ďalej na návrh PID regulátora je potrebné spraviť zo spojitého systému diskretný systém pomocou príkazu *c2d*. Systém je na hranici stability, jeho vlastné čísla sú:

$$\lambda_1 = 1.0000$$

$$\lambda_2 = 1.0000$$

$$\lambda_3 = 0.5857$$

$$\lambda_4 = 1.0000$$

$$\lambda_5 = 1.0000$$

$$\lambda_6 = 0.5857$$

Schéma na obr. 18 obsahuje tri dôležité časti – časť na riadenie kamery, ktorá je označená s červenou farbou, časť na kalibráciu, ktorá je označená s modrou farbou a PID regulátor, ktorý je označený so zelenou farbou. Prvá časť na riadenie kamery sa používa na to, aby bola známa poloha guľičky. Blok *Reshape* je na zmenu rozmerov vektora alebo vstupného signálu matice. Výstup môže byť ako jednorozmerné pole (vektor), stĺpcový vektor (Mx1 matica), riadkový vektor (1xN matica), matica alebo vektor s rôznymi rozmermi (MxN) alebo

rozmary sa dajú odvodiť z rozmerov referenčného portu. V tomto prípade na výstupe je použitá matica s rôznymi rozmermi, ktorá má dimenziu 14400. Blok *bpsearch* je na rozpoznanie súradníc guľičky. Ďalej výstup z toho zavedieme do tvarovača nultého rádu.

Tvarovač je číslicový riadiaci systém, ktorý počíta akčné zásahy v pravidelných časových intervaloch určených periódou vzorkovania. Tieto číselné hodnoty sú v D/A prevodníku premieňané na postupnosť amplitúdovo modulovaných impulzov, ktoré sú určené len v časoch vzorkovania. Tvarovač zabezpečuje aproximáciu akčných zásahov aj v časoch medzi dvomi časmi vzorkovania. Drží hodnotu akčného zásahu, kým sa nezmení jeho hodnota. Pri tvarovači nultého rádu výstupná veličina je stupňová funkcia.

Výstupný signál z tvarovača vzorkovača sa dá popísať s funkciou

$$\bar{u}(\tau) = u_0 \cdot 1(\tau - t_0) - u_1 \cdot 1(\tau - t_1) \quad (77)$$

Laplaceov obraz výstupného signálu tvarovača nultého rádu je daný s nasledujúcim vzťahom

$$\bar{U}(s) = \frac{1}{s} - \frac{1}{s} e^{-T_s s} \quad (78)$$

Laplaceov obraz vstupného signálu $f(t)$ (Diracov impulz) je

$$\bar{U}^*(s) = 1 \quad (79)$$

Prenos tvarovača nultého rádu je určený ako pomer výstupného a vstupného obrazu signálu

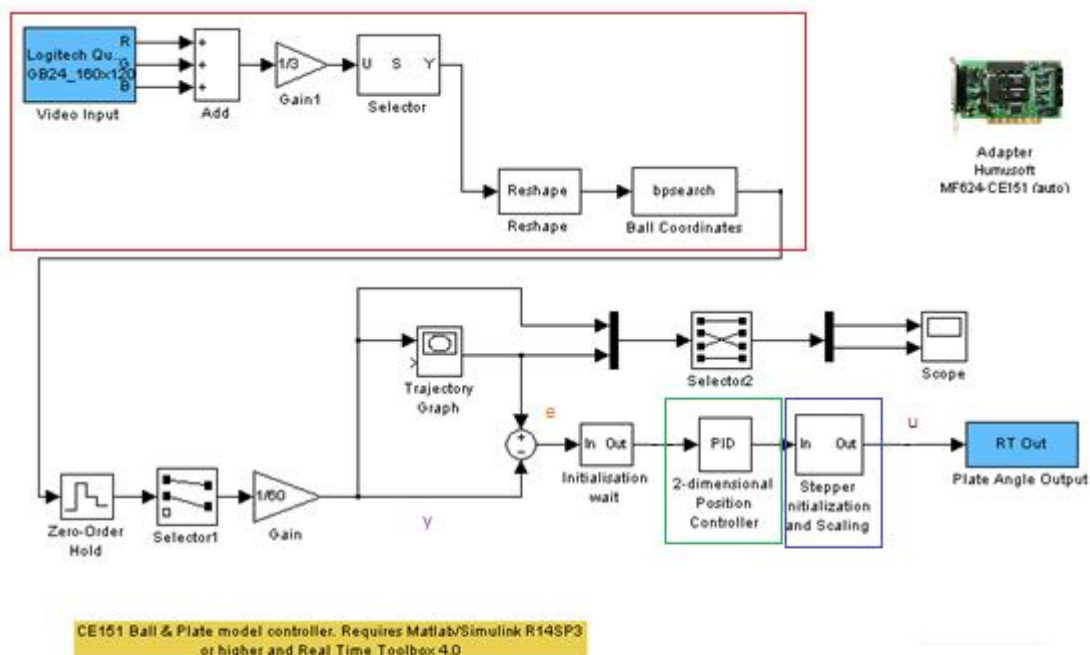
$$H(s) = \frac{\bar{U}(s)}{\bar{U}^*(s)} = \frac{\frac{1}{s} - \frac{1}{s} e^{-T_s s}}{1} = \frac{1 - e^{-T_s s}}{s} \quad (80)$$

Prenosová funkcia vzorkovača tvarovača je

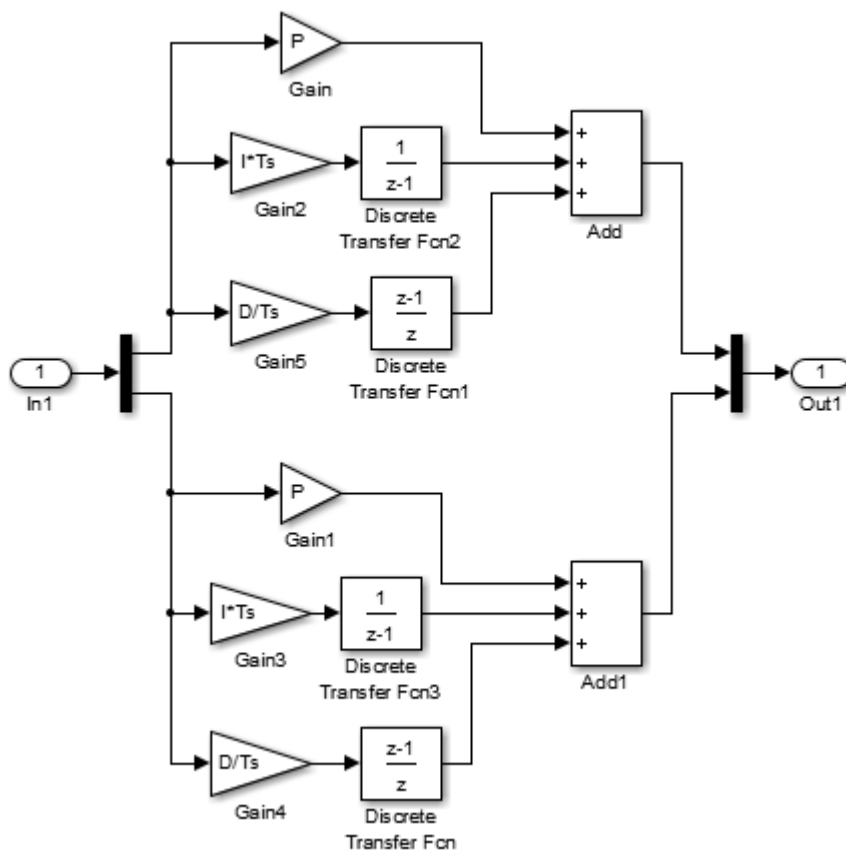
$$G_Z(s) = \frac{1 - e^{-T_s s}}{s} G(s) = (1 - e^{-T_s s}) \frac{G(s)}{s} \quad (81)$$

$$G_Z(s) = (1 - e^{-T_s s}) G_s(s) \quad (82)$$

Navrhnutá schéma na riadenie s PID regulátorom:



Obr. 18: Schéma pre PID regulátor



Obr. 19: Diskrétny PID regulátor

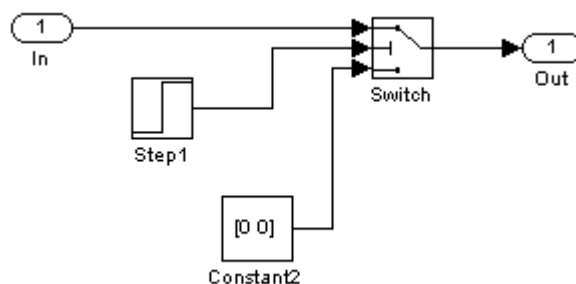
Na riadenie MIMO systémov s PID regulátorom potrebujeme navrhnuť na každú os x a y zvlášť jeden PID regulátor, čo je znázornená na obr. 19. Aby sme dostali I zložku, musíme vynásobiť diskretnou prenosovou funkciou $\frac{1}{z-1}$ a periódou vzorkovania. Následne D zložku musíme vydeliť s periódou vzorkovania a vynásobiť prenosovou funkciou $\frac{z-1}{z}$.

Parametre regulátora:

$$K_p = 0.36195$$

$$T_I = 0.1131$$

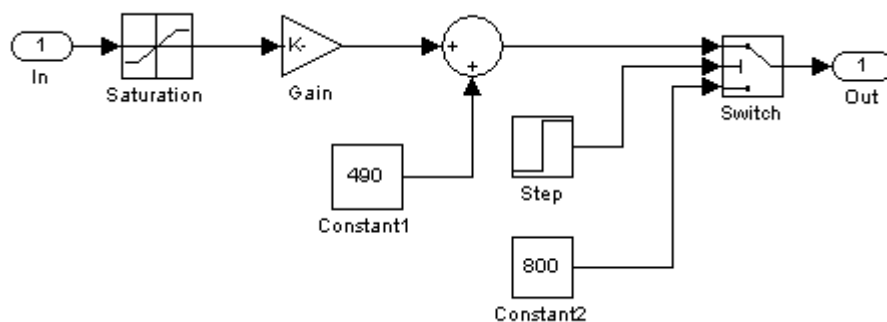
$$T_D = 0.1961$$



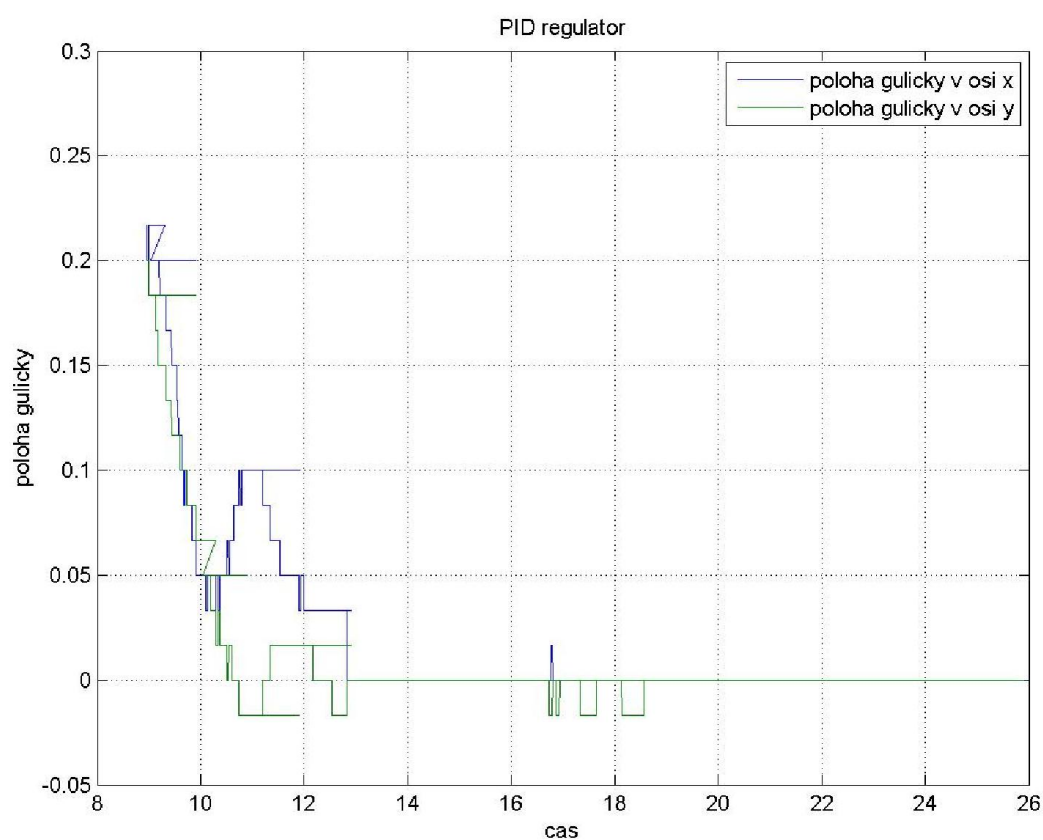
Obr. 20: Schéma na kalibráciu plochy

Kalibrácia plochy je spravená na inicializáciu systému, ktorá je znázornená na obr. 20. Užívateľ pri otvorení subsystemu môže nájsť jeden prepínač, ktorý slúži na to, aby pri rôznych podmienkach mohol prepínať jednotlivé vstupy. Prepínač prejde cez vstup 1 len v prípade, že vstup 2 spĺňa vybrané kritériá. V tomto prípade v čase 4 nulová začiatočná podmienka sa skokovo zmení na hodnotu 1.

Na škálovanie signálu je signál prevedený cez blok saturácie, čo je na obr. 21. Saturácia je nastavená vzhľadom na systémové ohraničenia, ktoré opíšeme v kapitole 1.2 v rovnici (12). Takže ohraničenia na vstupy sú orezané, aby prekročili interval $(-0.8, 0.8)$. Potom je vynásobený so zosilnením 300, čo opäť vyplýva z toho, že najväčšia hranica uhlu plochy je 0.1878 rad a veľkosť kroku pri 1° je 0.000628 rad. Vydelením tých dvoch čísel dostaneme približnú hodnotu zosilnenia 300. Ďalej je pripočítaná konštanta 490. Tento signál je potom privedený do prepínača.



Obr. 21. Škálovanie signálu



Obr. 22. Riadenie polohy guľičky s PID regulátorom

Na obr. 22 je vykreslený graf pre riadenie reálneho systému polohy guľičky s PID regulátorom. Na grafe vidieť, že nie je offset, čiže trvalá regulačná odchýlka, guľička na polohe [0,0] sa ustáli v čase 12.5. Modrá čiara symbolizuje polohu guľičky v osi x a zelená čiara polohu guľičky v osi y.

3.2 Pozorovač

V tomto prípade ide hlavne o overenie, či je pozorovač správne navrhnutý. Pozorovač je správne navrhnutý práve vtedy, ak nám dokáže dostatočne rýchlo a presne odhadovať stavy riadeného systému. Schému s PID regulátorom sme zobrali z toho dôvodu, aby sme dostali dostatočne dlhú dynamiku riadeného systému. Takto dostaneme informáciu o reálnych stavoch, ktoré môžeme následne porovnávať so stavmi, ktoré sme popri riadení odhadovali. Prostredníctvom diferencie týchto stavových údajov sme mohli ladiť pozorovač.

Aby sme mali dobrý pozorovač, najprv musíme navrhnuť maticu L tak, aby potom matica $\bar{A} = (Ae - Ae * L * Ce)$ bola stabilná. Keďže systém je diskretný, tak korene systému musia byť vo vnútri jednotkovej kružnici. Maticu L sme vypočítali pomocou príkazu *dlqe*. Tento príkaz navrhuje Kalmanov filter aj s poruchami. Na základe kovariančných matíc pre vstupy a stavy, ktoré sme zvolili nasledovne môžeme vypočítať s príkazom maticu $L = dlqe(Ae, G, Ce, H, M)$:

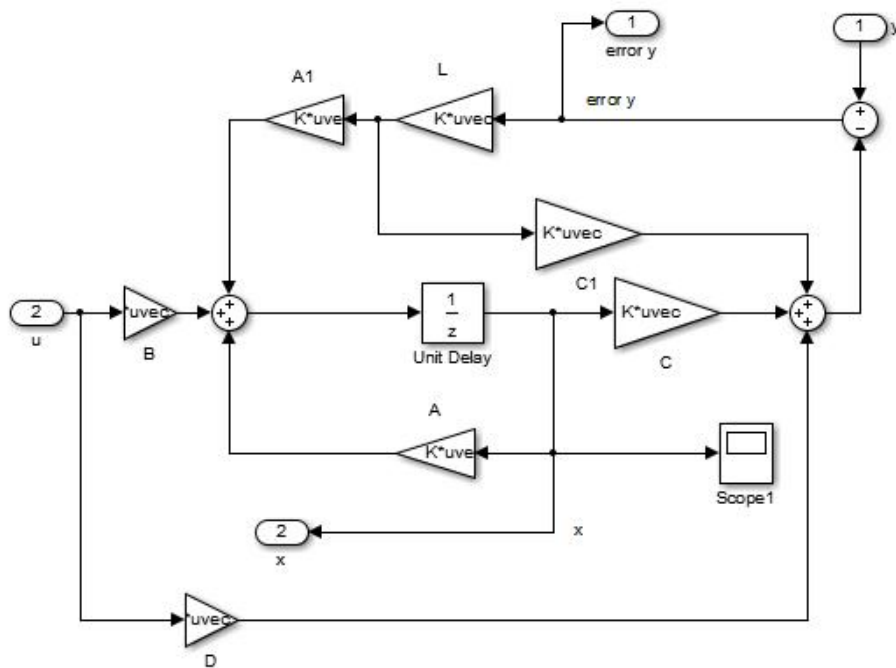
$$H = \begin{bmatrix} 1e1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1e3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1e1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1e1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1e3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1e1 \end{bmatrix} \quad (83)$$

Keďže máme šesť stavov, váhová matica pre stavy má rozmer 6x6 a takisto máme dva vstupy, preto váhová matica pre vstupy má rozmer 2x2.

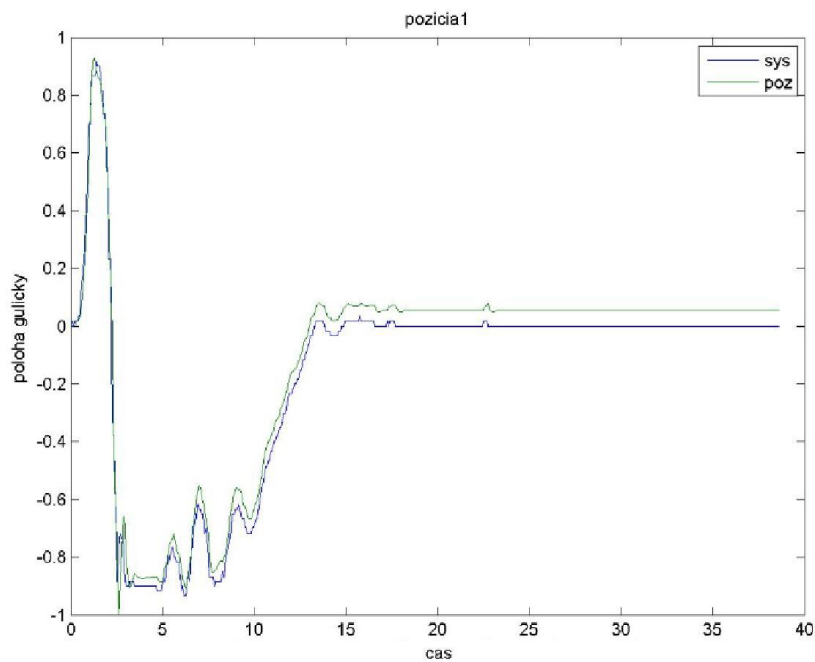
$$M = \begin{bmatrix} 0.5e1 & 0 \\ 0 & 0.5e1 \end{bmatrix} \quad (84)$$

Na obr. 23 vidíme zapojenie Kalmanovho filtra vo vývojovom programe Matlab Simulink. Do pozorovača vstupujú vstupy a výstupy a podľa nich sa dajú odhadovať stavy systému. Tento Kalmanov filter nájdeme ako subsystém pri stavových simuláciách, to znamená, že pri LQR regulátore. Aby sme mohli overiť či máme dobrý pozorovač, musíme

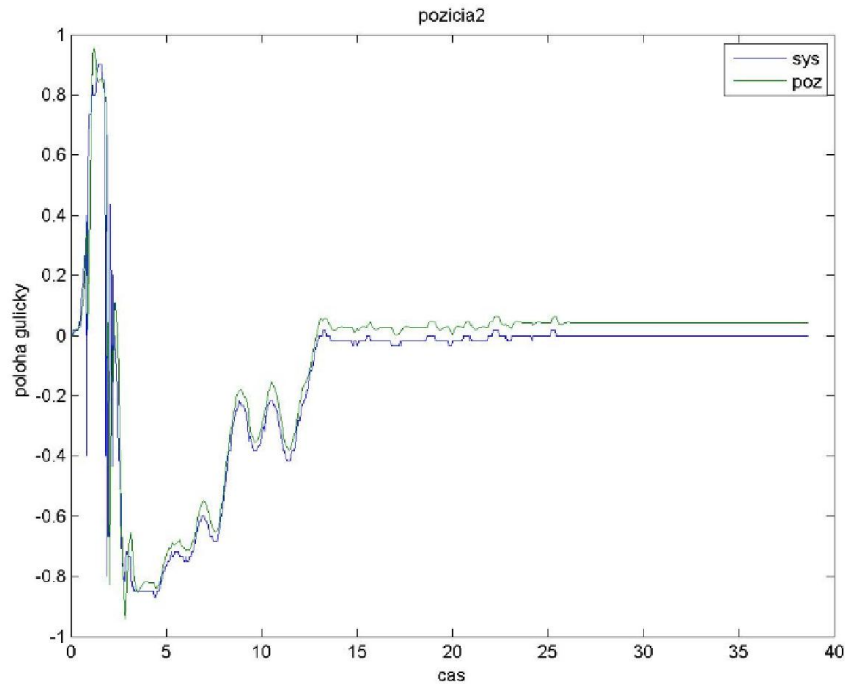
porovnať stavy systému a pozorovača, čo máme uvedené na obr. 24 pozíciu osi x a na obr. 25 pozíciu osi y.



Obr. 23: Schéma pozorovača



Obr. 24: Odhad stavu pre jednu polohu guličky



Obr. 25: Odhad stavu pre druhú polohu guľičky

3.3 Návrh LQR regulátora

V tejto kapitole práce sú prezentované simulačné výsledky systému.

Optimálny stavový regulátor K sme hľadali riešením LQR problému vo vývojovom systéme Matlab a to pomocou príkazu `dlqr`. Na vygenerovanie rôznych K sme spravili Matlab skript, kde sme vygenerovali rôzne náhodné čísla.

M-file:

```
matica = [];
Q_1 = [];
R_1 = [];
for k=1:kf

Q = diag([1000*rand(1,1) 100*rand(1,1) 10*rand(1,1) 1000*rand(1,1)
100*rand(1,1) 10*rand(1,1)]);
R = diag([100*rand(1,1) 100*rand(1,1)]);           %Re - vystupy

[K,S,E] = dlqr(Ad,Bd,Q,R)
sim('lqr_kalman',10)

suma = sum(sum(obs(:,2:end).^2));
%cas kedy sa to ustalilo
```

```

for i=1:length(obs(:,1))
    if sum(abs(obs(i,2:end)))<= 5e-3
        cas = obs(i,1);
        break;
    end
end

matica = [matica;suma cas;];

Q_1 = [Q_1;Q];
R_1 = [R_1;R];
end

```

V tomto prípade váhové matice vyzerajú nasledovne

$$R = \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix} \quad (85)$$

$$Q = \begin{bmatrix} 1000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix} \quad (86)$$

$$K = \begin{bmatrix} 0.7843 & 0.7504 & 0.5357 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.7843 & 0.7504 & 0.5257 \end{bmatrix} \quad (87)$$

Vlastné čísla diskretného systému dostaneme s 2 spôsobmi, buď s príkazom *eig* alebo $[K,P,E]=dlqr(Ad,Bd,Q,R)$, kde vlastné čísla Matlab uloží do premennej *E*.

$$\lambda_1 = 0.4567$$

$$\lambda_2 = 0.4567$$

$$\lambda_3 = 0.7588 + 0.1671i$$

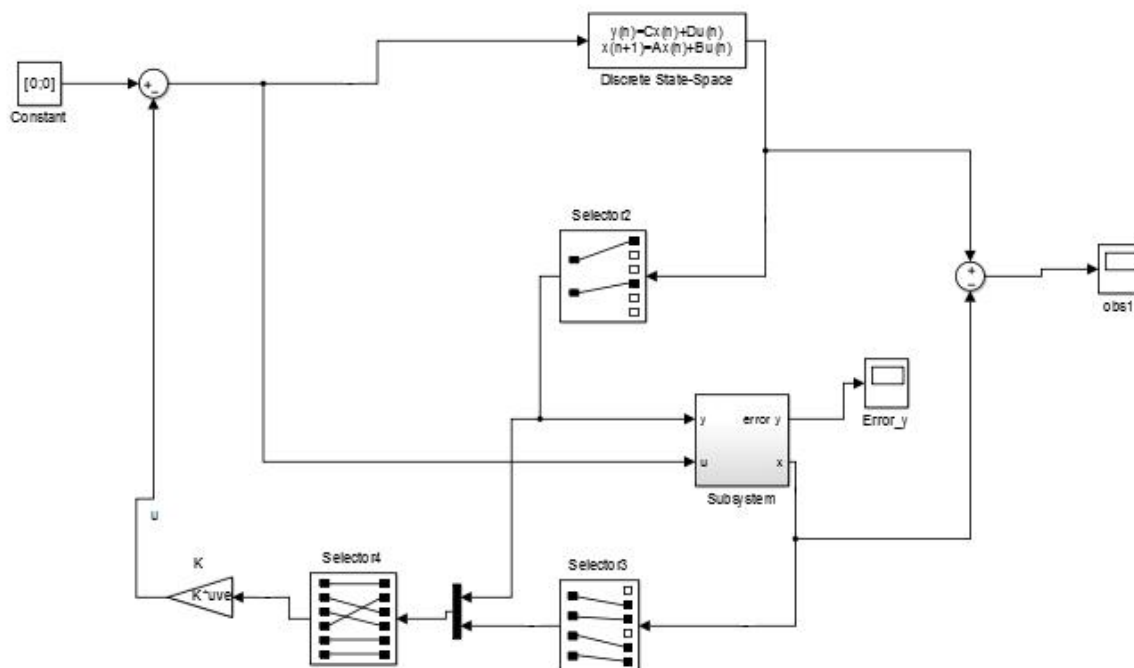
$$\lambda_4 = 0.7588 - 0.1671i$$

$$\lambda_5 = 0.7588 + 0.1671i$$

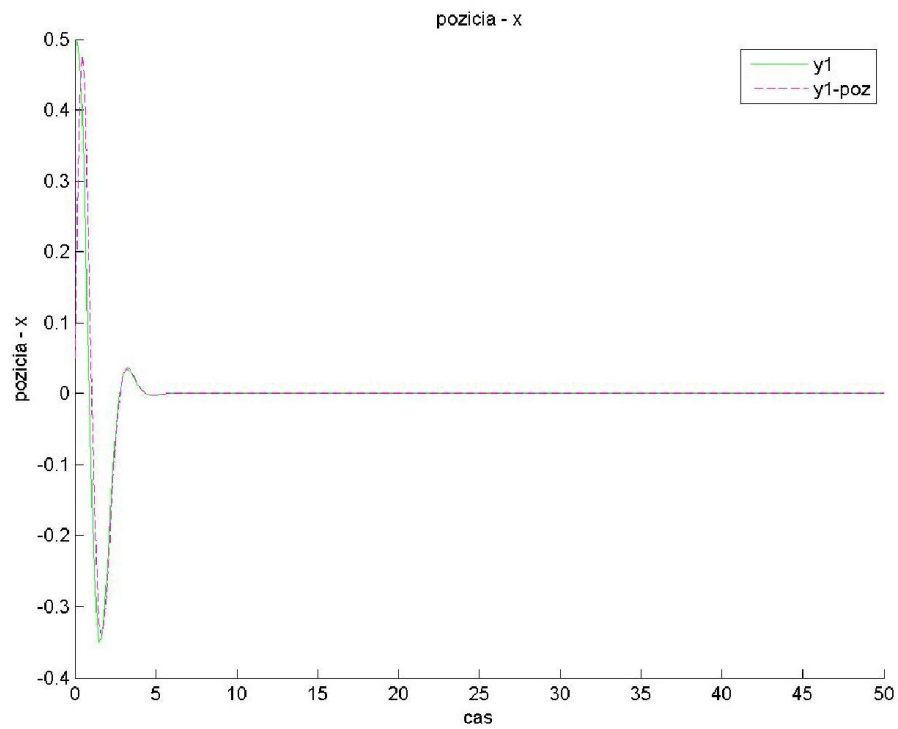
$$\lambda_6 = 0.7588 - 0.1671i$$

Vlastné čísla sú vo vnútri jednotkovej kružnici, čiže systém je stabilný. Na obr. 26 je znázornená simulačná schéma. Selektorom sa dá vybrať, ktoré prvky vstupného signálu chceme používať. Index pre každý prvok je identifikovaný zo vstupného portu alebo tohto dialógu. Môžeme si vybrať spôsob indexovania pre každú dimenziu pomocou parametra „Index Option“. Do prvého selektora vchádzajú stavy zo subsystemu, ale z toho potrebujeme len 2., 3., 5., 6. stav. Tieto stavy idú do druhého selektora, ale predtým sa ešte musia pridať výstupné stavy y , aby do systému išli reálne výstupné údaje. Nové stavy ďalej pokračujú do regulátora K , ktorým výstupom je u . Na obr. 27 je vykreslený graf pre riadenie jednej polohy guličky s LQ riadením a na obr. 28 je vykreslený graf pre riadenie druhej polohy guličky s LQ riadením.

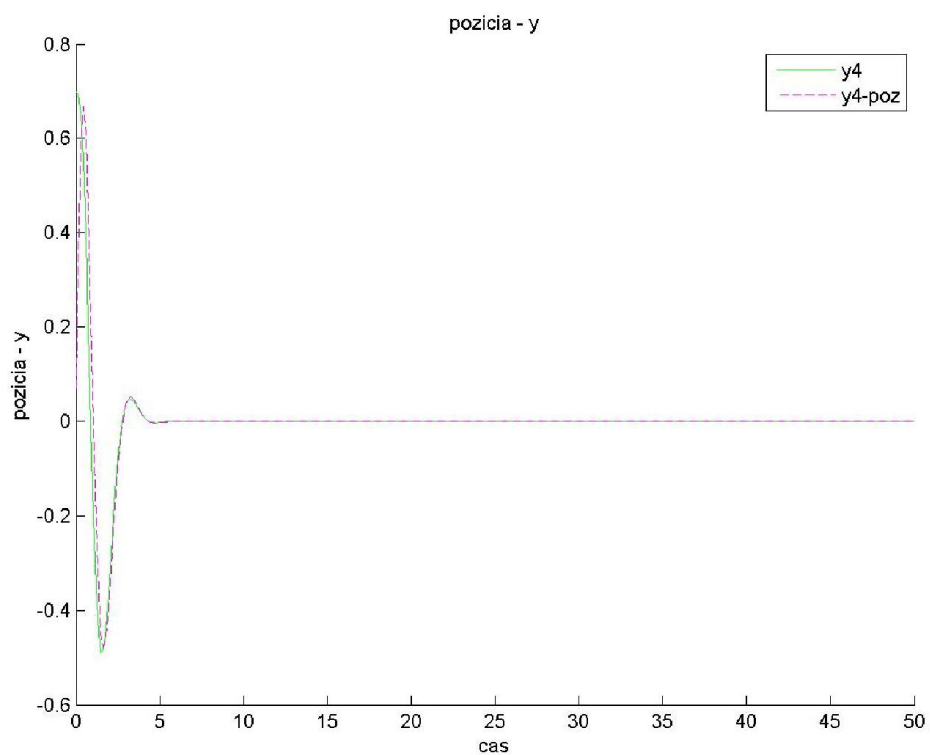
Ťažko sa porovnáva riadenie reálneho systému so simuláciou, lebo máme iné podmienky na riadenie. Na grafoch vidíme, že začiatočná podmienka prvej polohy je 0.5 a druhej polohy je 0.7. V grafe porovnáваме výstup zo systému a výstup z Kalmanovho filtra. Vidíme že priebehy sa ustália v čase 5. Čas maximálneho preregulovania sa nastane v čase 4. Čas maximálneho preregulovania je čas, v ktorom sa nastane maximálne preregulovanie. Ďalej vidíme, že systém predtým, ako pôjde do nuly spraví podkmit a prekmit. Veľkosť podkmitu polohy x je -0.35 a polohy y je -0.5. Na obr. 29 vidíme priebehy akčných členov. Na grafe sú znázornené aj ohraničenia, ktoré sú od -1 po 1, čo sme neprekročili.



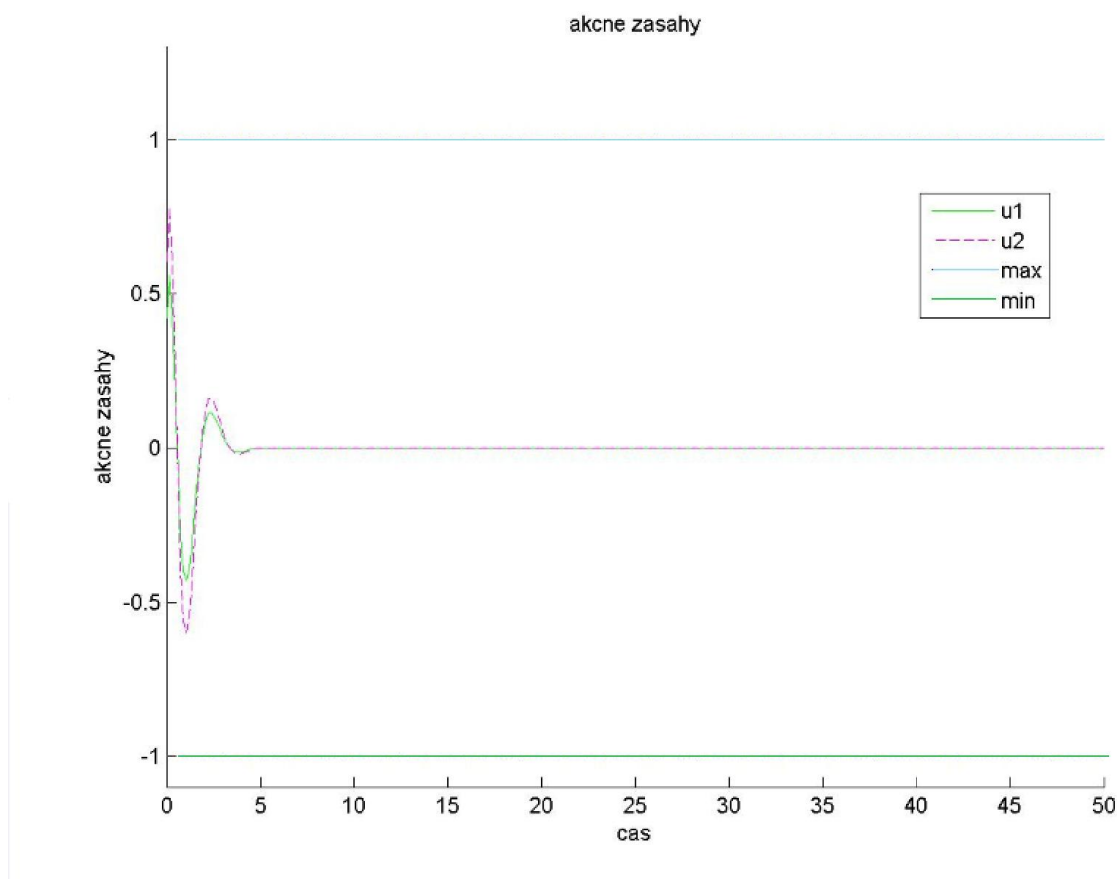
Obr. 26: Schéma pre model LQR regulátorom



Obr. 27: Riadenie jednej polohy guličky LQR regulátorom



Obr. 28: Riadenie druhej polohy guličky LQR regulátorom



Obr. 29: Akčné zásahy

3.4 LQR – tracking

V tejto kapitole práce sú prezentované experimentálne výsledky systému.

Reálny systém, ktorý riadime je v diskretnom čase, z čoho sa vyplýva, že údaje zo senzora sa nám zbierajú s určitou periódou vzorkovania. Perióda vzorkovania v tomto prípade je 0.1. Keďže CCD kamera pracuje v diskretnom čase, tak aj stavový regulátor musí byť v diskretnom čase.

LQR riadenie v podstate je proporcionálna stavová spätná väzba. V priemyselnom živote, aby sme mali garanciu na asymptotickú konvergenciu žiadanej hodnoty, čo znamená, že chceme sledovať žiadanú hodnotu, musíme ešte pridať do stavového spätnoväzbového obvodu integrátor. Čiže v tomto prípade LQR – tracking znamená toľko, že do riadenia je zahrnutý aj integrátor, aby odstránil trvalú regulačnú odchýlku (TRO). Takže, ak chceme mať tú istotu, že stavová spätná väzba (alebo stavový pozorovač) konvergovala asymptoticky

k žiadanej hodnote, tak je potrebné implementovať integračný zásah v spätnoväzbovom obvode. Integrátor má tvar $\frac{1}{F}$. Keď systém je spojité, tak $F = s$, a v prípade ak je diskretný, tak $F = 1 - z^{-1}$.

Vstupno-výstupný integrátor má formu jednotkového skoku:

$$\frac{z}{z-1} = \frac{1}{F(z^{-1})} = \frac{1}{1-z^{-1}} \quad (88)$$

Vstupno-výstupná reprezentácia:

$$x(k+1) = Ax(k) + Bu(k) \quad (89)$$

$$y(k) = Cx(k)$$

Rovnica riadenia:

$$u(k) = u(k-1) + \Delta u(k) \quad (90)$$

kde $u(k-1)$ je staré riadenie a $\Delta u(k)$ je zmena rýchlosti riadenia.

Ak chceme zabezpečiť integračnú činnosť, musíme zaviesť nový stav s integrátorom:

$$\bar{x}(k) = \begin{pmatrix} x(k) \\ u(k-1) \end{pmatrix} \Rightarrow \text{riadenie v minulej perióde vzorkovania} \quad (91)$$

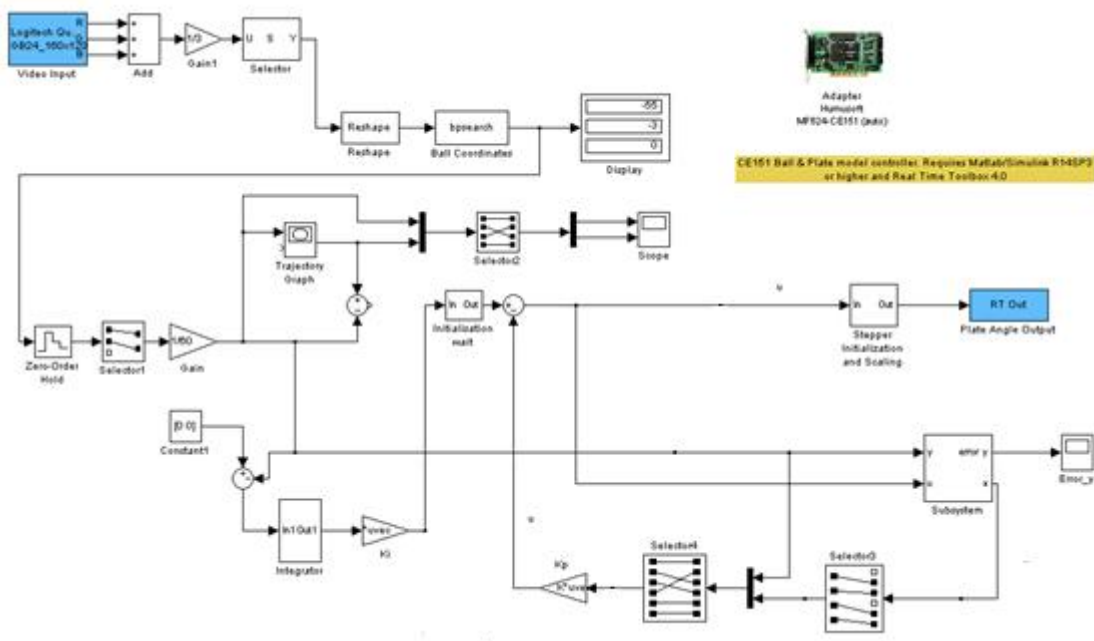
Po dosadení a úprave dostaneme:

$$\bar{x}(k+1) = Ax(k) + Bu(k-1) + B\Delta u(k) \quad (92)$$

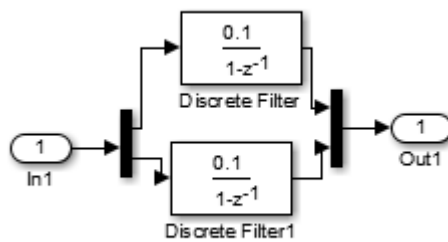
$$\bar{x}(k+1) = \begin{pmatrix} x(k+1) \\ u(k) \end{pmatrix} = \begin{pmatrix} A & B \\ 0 & I \end{pmatrix} \begin{pmatrix} x(k) \\ u(k-1) \end{pmatrix} + \begin{pmatrix} B \\ I \end{pmatrix} \Delta u(k) \quad (93)$$

$$y(k) = (C \quad 0) \begin{pmatrix} x(k) \\ u(k-1) \end{pmatrix} + (0) \Delta u(k) \quad (94)$$

Uvedený systém už má v sebe integrátor. Na obr. 30 je znázornená schéma pre LQR riadenie – tracking, a ďalej je znázornený diskretný filter na obr. 31.



Obr. 30: Schéma pre LQR riadenie - tracking



Obr. 31: Diskrétny filter

M-file:

```
%diskretny s integrotom
Atilda=[Ad Bd;zeros(2,6) eye(2)]
Btilda=[Bd;eye(2)]
Ctilda=[Cd zeros(2)]
Dtilda=[zeros(size(Dd)) ]

r=[0.15 0.17 0.18 0.23 0.22 0.1 0.21 0.3]
[Ktilda,prec]=place(Atilda',Ctilda',r)
Ktilda=Ktilda'

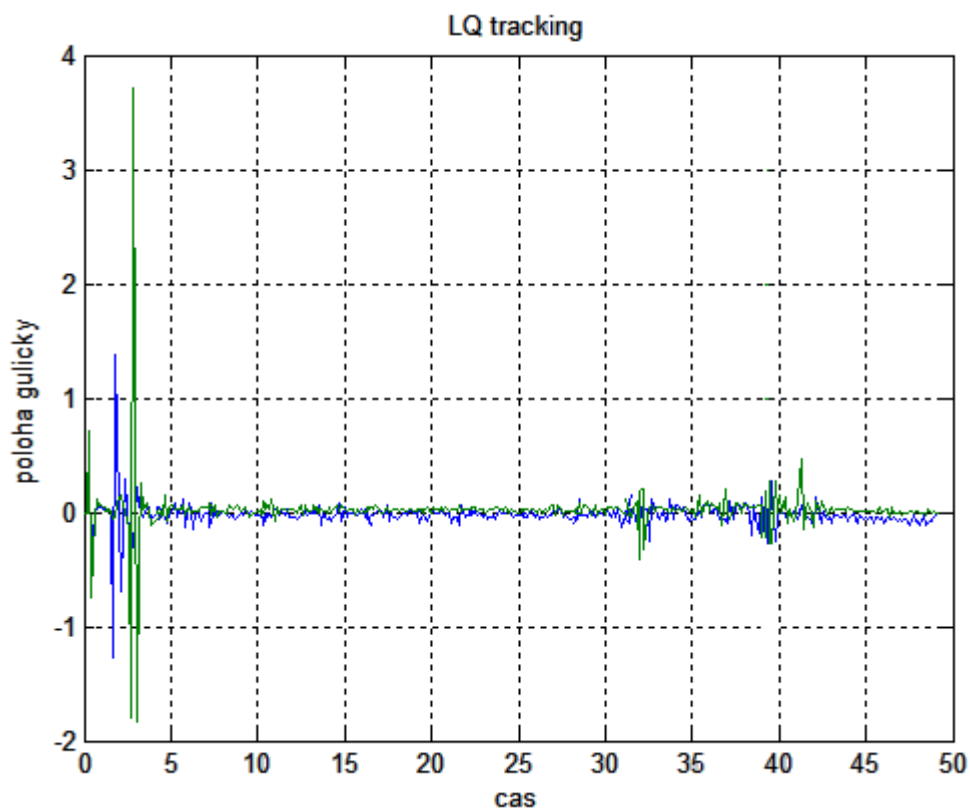
Qtilda=eye(8)*5.43
Rtilda = eye(2)*60
```

```
[Ktilda,Ptilda,Etilda]=dlqr (Atilda,Btilda,Qtilda,Rtilda)
```

```
Kp=Ktilda(:,1:6)
```

```
Ki=Ktilda(:,7:8)
```

Matica zosilnení je vyrábaná s príkazom *dlqr(Atilda,Btilda,Qtilda,Rtilda)*, ktorá je rozdelená do dvoch matíc K_p a K_i . V tejto časti práce opisujeme riadenie reálneho systému. Na obr. 32 je vykreslený graf pre riadenie polohy guľičky s LQR riadením – tracking, kde vidíme, že výstup sa konverguje k žiadanej hodnote $[0;0]$. Maximálne preregulovanie sa vyskytlo v čase 4. Modrá čiara opäť znázorňuje priebeh riadenia polohy v osi x a zelená čiara na druhej strane polohu v osi y.



Obr. 32: Riadenie polohy guľičky s LQR – tracking

3.5 Hustá formulácia MPC

V tejto kapitole práce sú prezentované simulačné výsledky systému. Hustá formulácia prediktívneho riadenia je formulovaná podľa teórie, ktorá je uvedená v kapitole 2.4.1. Naš cieľ je, aby sme riadili výstup, čiže polohy guľičky do nuly. Našou úlohou je minimalizovať danú účelovú funkciu (63), kde berieme do úvahy aj ohraničenia. Ohraničenia v tvare nerovnosti sú na stavy a vstupy, ktoré sú od -1 až po 1. Tieto ohraničenia sú uvedené aj v kapitole 1.2 v rovniciach (12, 13). Nie všetky solvre vedú vypočítať ohraničenia v tvare rovnosti, preto musíme pretransformovať úlohu kvadratického programovania bez použitia ohraničení v tvare rovnosti. V tomto prípade bol používaný solver optimizer a jedna optimalizácia v jednej perióde vzorkovania trvala asi 0.009 s, čo sa dá vypísať s príkazmi *tic* a *toc*.

Číselné hodnoty váhových matic ovplyvňujú celú reguláciu. Nejde len o pomer váhových matic, ale aj o to, že váhovú maticu musíme vynásobiť s rozsahom príslušnej premennej. Napríklad keď máme premennú, ktorá je v miliónoch, tak ho musíme zohľadniť aj pri voľbe váhovej matice.

V našom prípade váhová matica stavov má nasledovnú hodnotu:

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 0 & 10 \end{bmatrix} \quad (95)$$

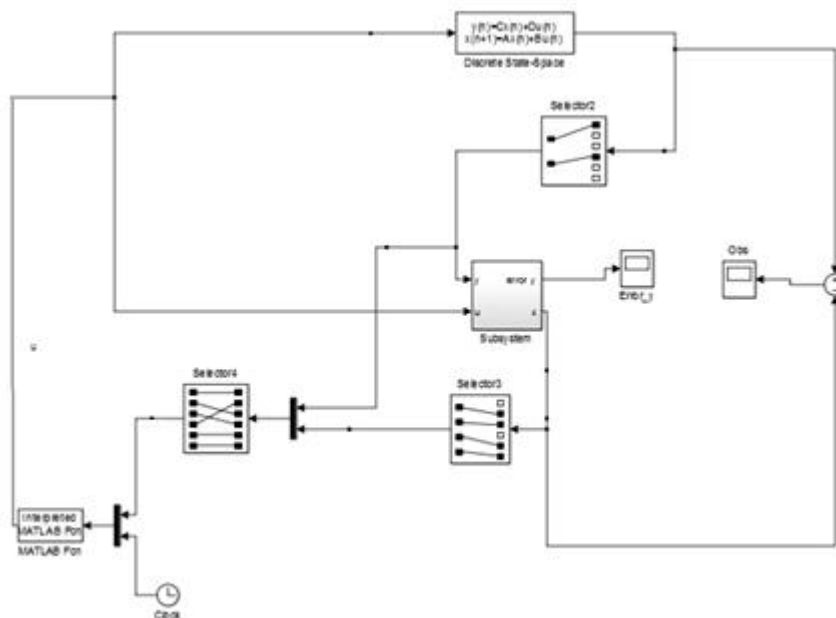
Váhová matica R pre vstupy vyzerá nasledovne:

$$R = \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \quad (96)$$

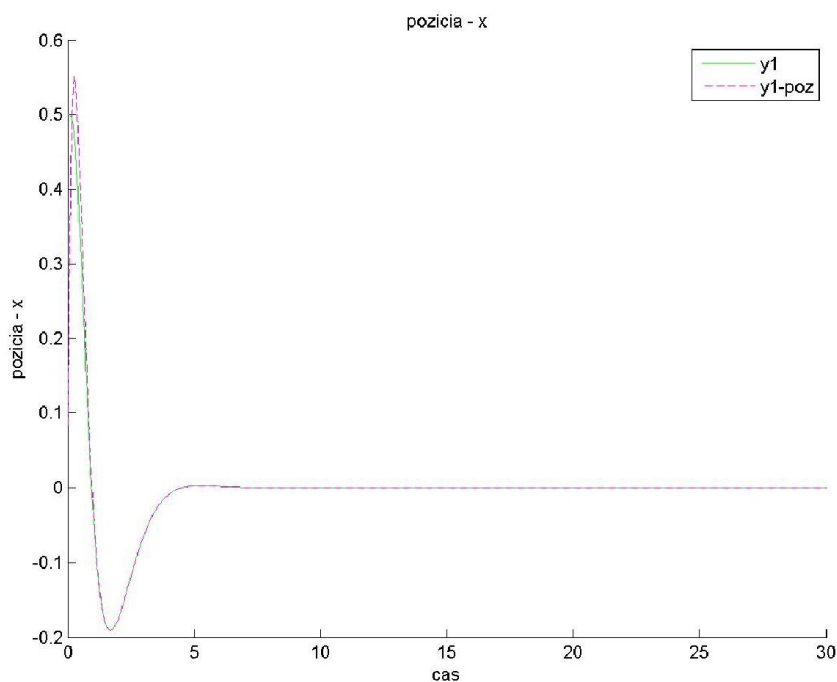
Predikčný horizont v našom prípade je $N = 5$ a perióda vzorkovania je $T_s = 0.05$.

Na obr. 33 je znázornená schéma pre hustú formuláciu MPC. Aby riadenie bolo presnejšie do výpočtu MPC zavedieme stav polohy guľičky z modelu a nie z pozorovača. To môžeme zrealizovať pomocou selektorov. Princíp ich fungovania je opísaný v kapitole 3.3.

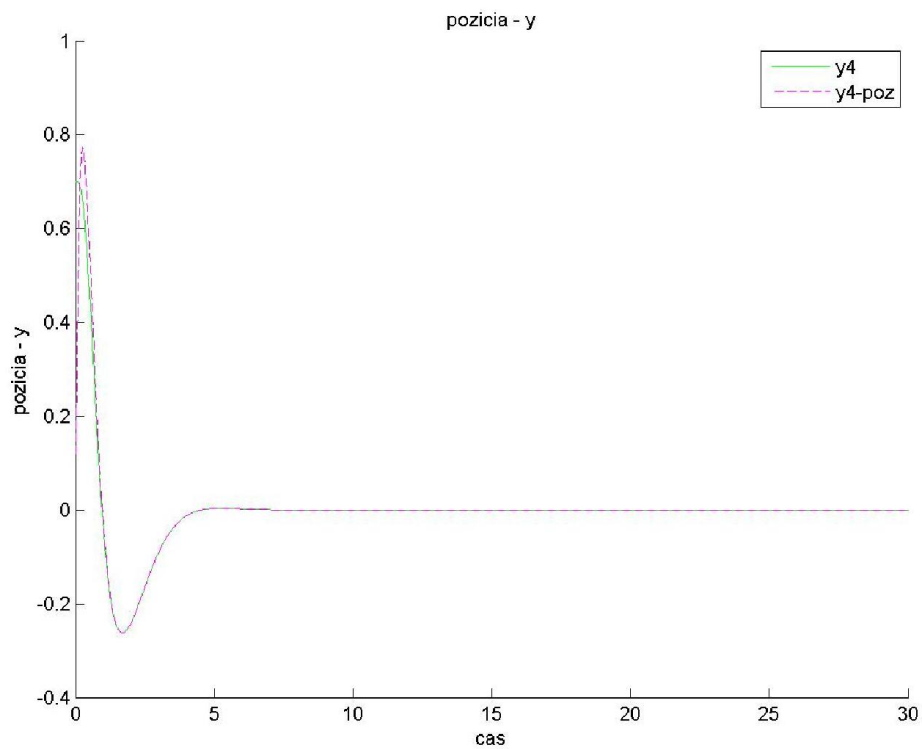
Na obr. 34 je vykreslený priebeh riadenia polohy v osi x, kde sú porovnané reálne výstupy s pozorovanými. Na obr. 35 je opäť vykreslený priebeh riadenia polohy v osi y. V porovnaní s LQR vidíme, že podkmit je oveľa menší (pri jednej polohe -0.195 a pri druhej polohe -0.25) a prekmit skoro je nulový. Akčné zásahy sú znázornené na obr. 36, kde môžeme skonštatovať, že ohraničenia na vstupy sú splnené, ktoré sa môžu pohybovať medzi -1 a 1.



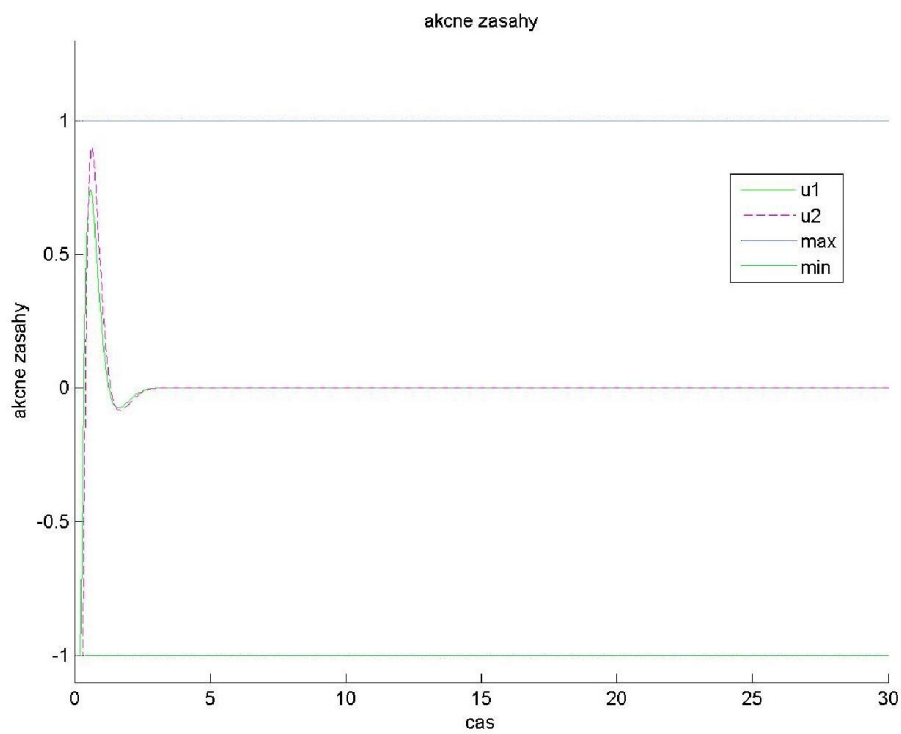
Obr. 33: Schéma pre MPC regulátor



Obr. 34: Riadenie jednej polohy guľičky MPC regulátorom – hustá formulácia



Obr. 35: Riadenie druhej polohy guľičky MPC regulátorom – hustá formulácia



Obr. 36: Akčné zásahy pri hustej formulácii MPC

3.6 Delta u formulácia MPC

V tejto kapitole práce sú prezentované opäť simulačné výsledky systému. V prípade ak chceme používať výstupnú reguláciu a sledovať nenulovú referenciu, tak sa nám môže stať, že máme trvalú regulačnú odchýlku. Tento offset je spôsobený kvôli akčného zásahu, lebo ak sa chceme dostať na nenulovú referenciu, musíme dať aj nenulovú hodnotu akčného zásahu. Delta u formulácia vie eliminovať trvalú regulačnú odchýlku. Pri tejto formulácii keď vstupy sú v ustálenom stave nekolíšu. Táto formulácia je výhodná v tom, že nevedí ak matica systému je neinvertovateľná. Nevýhodou je, že potrebujeme pamätať predchádzajúce údaje akčného zásahu na výpočet Δu . V našom prípade je nulová referencia, aby bolo jednoduchšie porovnať riadenie pri hustej formulácii MPC. V tomto prípade bol takisto používaný solver optimizer a jedna optimalizácia trvala asi 0.08 s. Tu už máme viac optimalizovaných premenných, optimalizujeme y a Δu , ale na výpočet Δu potrebujeme akčné zásahy v danom období vzorkovania a v predchádzajúcom období vzorkovania. Obdobiu vzorkovania v tomto prípade sme zvolili 0.1 s, aby optimizer stihol vypočítať akčné zásahy v každej perióde vzorkovania. Ak by nestihol, mohlo by dôjsť k nestabilite systému. Predikčný horizont v našom prípade je $N = 5$, čo takisto spôsobuje náročnosť výpočtu, lebo v každej perióde vzorkovania musíme vidieť akčné zásahy 5 krokov dopredu do budúcnosti.

V našom prípade váhová matica výstupov má nasledovnú hodnotu:

$$Q_y = \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix} \quad (97)$$

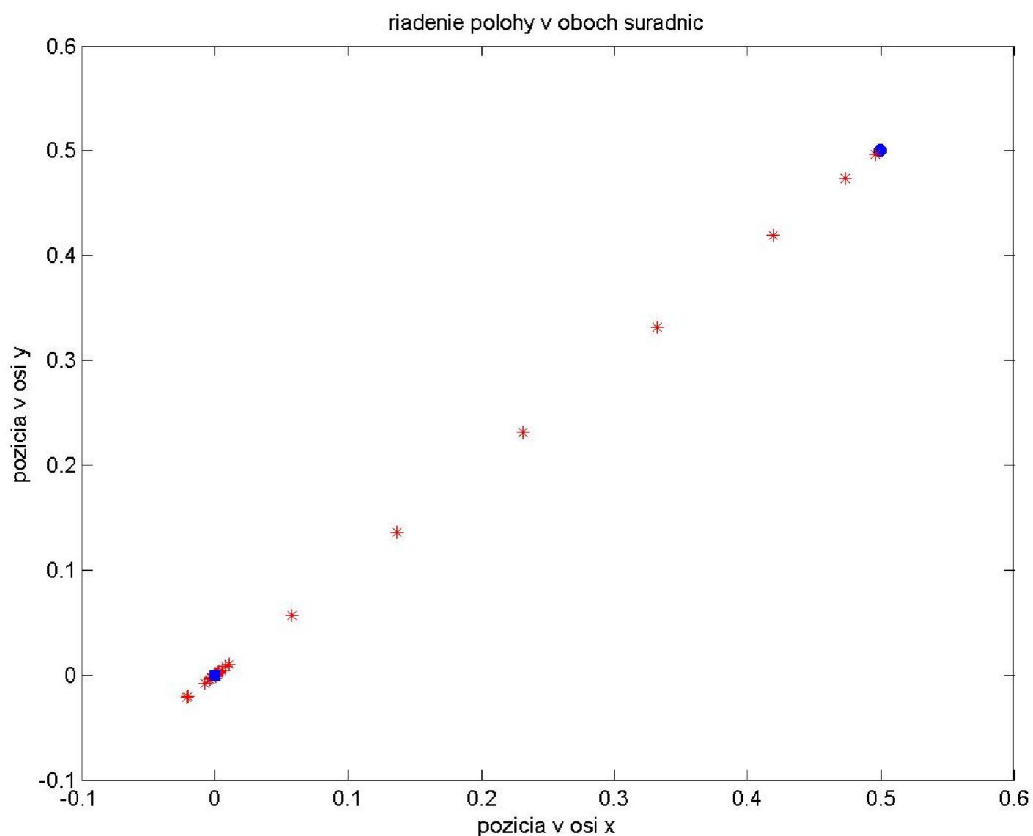
Váhová matica R pre zmeny vstupov vyzerá nasledovne:

$$R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (98)$$

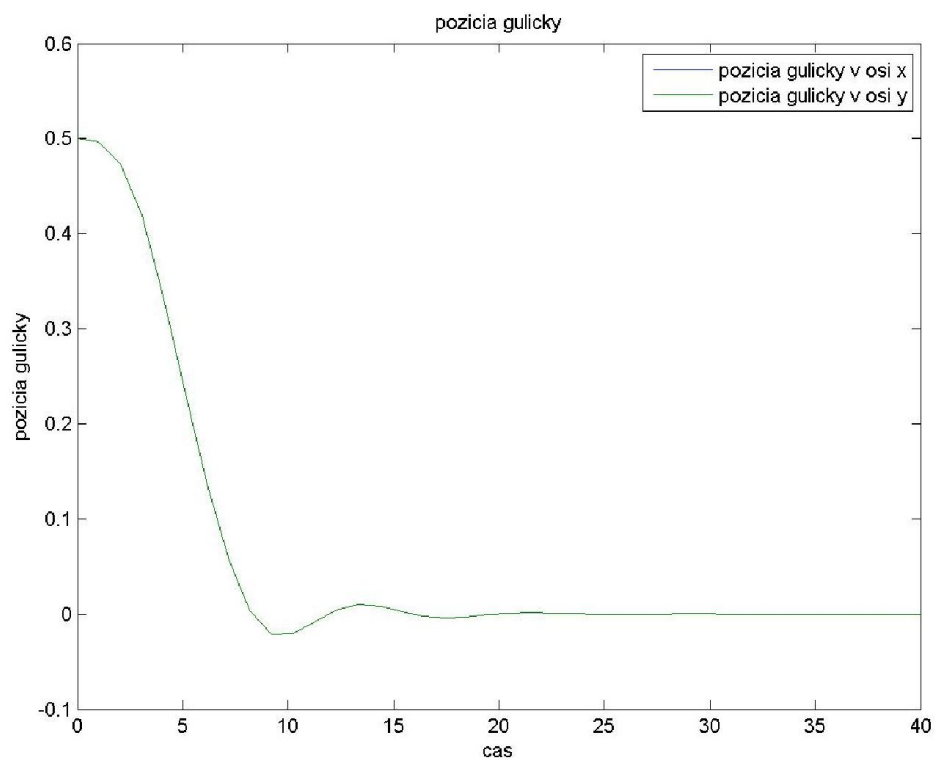
Na obr. 37 je znázornený graf pre riadenie polohy oboch súradníc. Modrým kruhom je vykreslený začiatkový bod $[0.5; 0.5]$, modrým štvorcom zase koncový bod $[0;0]$. Červenými sú vykreslené jednotlivé vypočítané optimalizované výstupy polohy guľičky.

Na obr. 38 sú vykreslené časové priebehy riadenia polohy v osi x a y . V porovnaní s LQR a MPC (hustá formulácia) vidíme, že aj podkmit aj prekmit sú skoro nulové. Ustália sa v čase 20, čo je oveľa väčšie ako pri hustej formulácii. Akčné zásahy sú znázornené na obr.

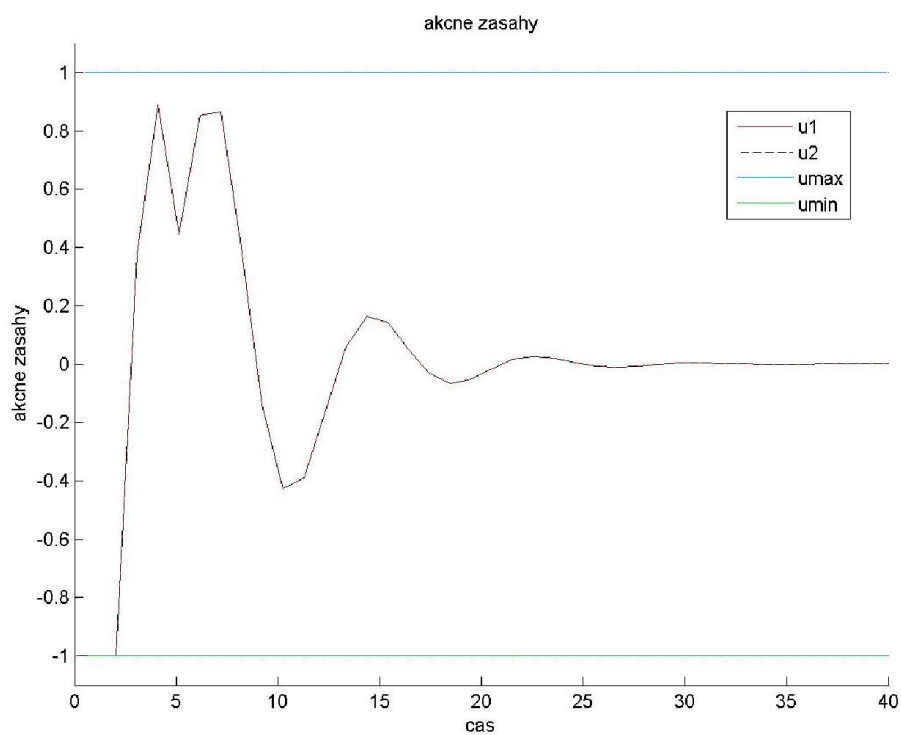
39, kde môžeme skonštatovať, že opäť ohraničenia na vstupy sú splnené, ktoré sa môžu pohybovať medzi -1 a 1. V tomto prípade vstupy sú agresívnejšie, čo by sme mohli vykompenzovať s tým, že by sme pridali ohraničenia aj na delta u. Takto môžeme znížiť zmeny rýchlosti na akčné zásahy.



Obr. 37: Riadenie polohy guličky MPC regulátorom – delta u formulácia



Obr. 38: Riadenie oboch polôch guličky MPC regulátorom – delta u formulácia



Obr. 39: Akčné zásahy pri delta u formulácii MPC

Záver

Hlavným cieľom tejto práce bolo riadenie reálneho systému „Gulička na ploche“ pomocou PID regulátora a optimálneho diskrétného stavového spätnoväzbového LQR a MPC regulátora.

V prvej kapitole sme predstavili reálne zariadenie, jeho matematický model a stavový opis systému. Oboznámili sme sa, že na akom princípe reálne zariadenie pracuje. Zistili sme, že z reálneho zariadenia sa dajú získať informácie iba o dvoch stavových veličinách, a to o polohe guličky na ploche, čo sme získali pomocou CCD kamery a o rýchlosti guličky, ktorú môžeme vypočítať pomocou polohy. Ostatné stavové veličiny sme neboli schopní merať. Akčné zásahy v podobe nakláňania platne sa realizujú pomocou dvoch samostatne pracujúcich servomotorov, ktoré sme schopní riadiť.

Druhá kapitola sa zaoberá s teóriou rôznych spôsobov riadení. Na začiatok sme opísali teóriu pre PID regulátory, čo je druh spätnoväzbového riadenia. Ďalej sme opísali teóriu pozorovača, čo potrebujeme na odhad stavov. Následne sme vypracovali teóriu LQR regulátora a MPC regulátora, ktoré sú optimálne stavové spätnoväzbové riadenia. Najväčší rozdiel je medzi nimi, že LQR regulátor pracuje s nekonečným predikčným horizontom a na druhej strane MPC regulátor pracuje s konečným predikčným horizontom.

V tretej kapitole sme predstavili všetky spôsoby riadenia, všetky naše výsledky. Najprv je navrhnutý PID regulátor a potom diskrétny stavový spätnoväzbový regulátor typu LQR a MPC. Keďže pre riadenie pomocou stavového regulátora potrebujeme poznať informácie o všetkých stavových veličinách rozhodli sme sa na odhad stavu použiť pozorovač. Keď sme získali funkčný pozorovač pomocou ladením váhových matic, mohli sme navrhnúť LQR riadenie. Riadenie pomocou tohto regulátora sme otestovali na vytvorenej simulačnej schéme, kde regulátor pracoval správne. Následne sme sa rozhodli aplikovať na reálne zariadenie, čo sa nám nepodarilo. Ďalej aby bola odstránená trvalá regulačná odchýlka sme rozšírili LQR riadenie o integrátor, ktorý zabezpečuje odstránenie trvalej regulačnej odchýlky. Ďalej sme navrhli prediktívny regulátor pre riadenie pomocou hustej formulácii a delta u formulácii. Keďže sa nepodarilo to implementovať na reálny systém, tak riadenie LQR a MPC regulátora sme implementovali na modeli.

Zoznam použitej literatúry

- [1] MIKLEŠ, J. – FIKAR, M.: Modelovanie, identifikácia a riadenie procesov II.
Bratislava: Vydavateľstvo STU, 1999, ISBN 80-227-1289-2.
- [2] Ball & Plate Apparatus – Technical Manual, Education Manual
User's manual revision 3.2 1996-2008
- [3] BAKOŠOVÁ, M. – FIKAR, M.: Riadenie procesov. STU v Bratislave, 2008. s. ISBN 978-80-227-2841-6.
- [4] KVASNICA, M.: Real-Time Model Predictive Control via Multi-Parametric Programming: Theory and Tools Saarbrücken: VDM Verlag Dr. Müller, 2009. s. ISBN 978-3-639-20644-9
- [5] GREG WELCH & GARY BISHOP: An Introduction to the Kalman Filter
- [6] <http://www.humosoft.cz/>