

Realizace prediktivních regulátorů s kompenzací poruchy v prostředí Simulink

Bc. Jakub Šenkeřík

Diplomová práce
2015



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2014/2015

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jakub Šenkeřík**
Osobní číslo: **A13514**
Studijní program: **N3902 Inženýrská informatika**
Studijní obor: **Automatické řízení a informatika**
Forma studia: **kombinovaná**

Téma práce: **Realizace prediktivních regulátorů s kompenzací poruchy v prostředí Simulink**

Téma anglicky: **The Implementation of Predictive Controllers with Disturbance Compensation in the Simulink Environment**

Zásady pro vypracování:

1. Vytvořte základní simulinkové schéma regulačního obvodu s prediktivním regulátorem založeném na zadaném modelu. Prediktivní regulátor realizujte jako S-funkci.
2. Na základě tohoto základního obvodu vytvořte regulační obvody s prediktivními regulátory, které umožní kompenzaci poruchové veličiny.
3. Implementujte regulátory založené na různém způsobu výpočtu prediktorů.
4. Ověřte simulačně funkčnost vytvořených schémat simulačním řízením různých systémů.
5. Realizujte v prostředí Matlab/Simulink.
6. Regulátory naprogramujte jako S-funkce v programovacím jazyku Matlab.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. V. Bobál, **Adaptivní a prediktivní řízení**, Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2007.
2. J. Mikleš, M. Fikar, **Process Modelling, Optimisation and Control**, Berlin: Springer-Verlag, 2008.
3. F. Camacho, C. Bordons, **Model Predictive Control**, London: Springer-Verlag, 2004.
4. J. Mikleš, M. Fikar, **Modelovanie, Identifikácia a riadenie procesov II.**, STU Bratislava, 2004.
5. J. A. Rossiter, **Model Based Predictive Control: a Practical Approach**, CRC Press, 2003.
6. D. W. Clarke, C. Mohtadi, P. S. Tufts, **Generalized predictive control, part I: the basic algorithm**. Automatica, 23, 1987, 137-148.
7. D. W. Clarke, C. Mohtadi, P. S. Tufts, **Generalized predictive control, part II: extensions and interpretations**. Automatica, 23, 1987, 149-160.
8. M. Morari, J. H. Lee, **Model predictive control: past, present and future**. Computers and Chemical Engineering, 23, 1999, 667-682.

Vedoucí diplomové práce:

doc. Ing. Marek Kubalčík, Ph.D.

Ústav řízení procesů

Datum zadání diplomové práce:

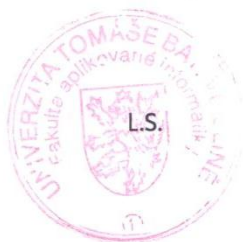
27. února 2015

Termín odevzdání diplomové práce:

20. května 2015

Ve Zlíně dne 27. února 2015

doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

ABSTRAKT

Tato diplomová práce se zabývá tvorbou simulací prediktivního řízení s kompenzací poruchy v prostředí programu MATLAB. V teoretické části jsou popsány dvě metody výpočtu prediktoru, a to za pomoci diofantických rovnic a metoda přímého výpočtu z CARIMA modelu. V praktické části jsou pak popsány jednotlivé simulace pro prediktivní řízení bez kompenzace poruchy, řízení s neznámou měřitelnou poruchou a řízení se známou poruchou. Simulační programy obsahují jako zdroj poruchy funkci sinus, skokovou změnu poruchové veličiny a rampu. Simulační programy jsou vytvořeny v prostředí Graphic User Interface (GUI) softwaru MATLAB. Zdrojové kódy aplikace jsou přiloženy na disku CD-ROM.

Klíčová slova:

MATLAB, CARIMA, prediktivní řízení, kompenzace poruchy, simulace

ABSTRACT

This Master thesis deals with creation of simulations with predictive control compensation of disturbance in MATLAB software. The theoretical part describes two methods of calculating the predictor using diophantine equations and method of direct calculation from Carima model. The practical part describes the individual simulation for predictive control without compensation of disturbance and control with an unknown measurable disturbance and control with known disturbance. Simulation programs contain as a source of disturbance sine, step and ramp function. All simulation programs are made using Graphic User Interface (GUI) in software MATLAB. The source code of the application is enclosed on the CD-ROM disc.

Keywords:

MATLAB, CARIMA, predictive control, compensation of disturbance, simulation

Děkuji tímto svému vedoucímu diplomové práce doc. Ing. Marku Kubalčíkovi, Ph.D. za odborné vedení, příkladnou trpělivost, rady a připomínky, ale také za čas, který věnoval mně i mé práci.

Prohlašuji, že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST.....	10
1 PREDIKTIVNÍ ŘÍZENÍ.....	11
1.1 PRVKY PREDIKTIVNÍHO ŘÍZENÍ	12
1.1.1 Modely	12
1.1.2 Výpočet optimálního řízení.....	13
1.1.2.1 Účelová funkce	13
1.1.2.2 Prediktor.....	13
1.1.2.3 Omezující podmínky.....	16
1.2 METODA VÝPOČTU PREDIKTORU POMOCÍ DIOFANTICKÝCH ROVNIC	17
1.3 METODA VÝPOČTU PREDIKTORU ZALOŽENÁ NA PŘÍMÉM VÝPOČTU Z CARIMA MODELU.....	20
2 MATLAB	21
2.1 PROSTŘEDÍ MATLAB	21
2.2 TOOLBOXY	22
2.3 SIMULINK	22
2.4 GUIDE	22
II PRAKTICKÁ ČÁST	24
3 VÝPOČET PREDIKTORU	25
3.1 MODEL.....	25
3.2 METODA VÝPOČTU PREDIKTORU ZALOŽENÁ NA PŘÍMÉM VÝPOČTU Z CARIMA MODELU.....	26
3.3 METODA VÝPOČTU PREDIKTORU POMOCÍ DIOFANTICKÝCH ROVNIC	29
4 SIMULAČNÍ PROGRAM.....	30
4.1 POPIS PROSTŘEDÍ SIMULAČNÍHO PROGRAMU	30
4.1.1 Popis bloků simulačního programu.....	31
4.2 POPIS SIMULAČNÍCH SCHÉMAT V SIMULINKU.....	35
4.2.1 Simulace řízení bez kompenzace poruchy	35
4.2.2 Simulace řízení s neznámou měřitelnou poruchou	35
4.2.3 Simulace řízení se známou poruchou.....	36
4.2.4 Blok prediktivního regulátoru	36
4.2.5 Blok akční zásah	38
5 SIMULAČNÍ OVĚŘOVÁNÍ.....	39
5.1 SIMULACE ŘÍZENÍ BEZ KOMPENZACE PORUCHY	39
5.1.1 Porucha – Sinus.....	39
5.1.2 Porucha – Skok	40
5.1.3 Porucha – Rampa	41
5.2 SIMULACE ŘÍZENÍ S NEZNÁMOU MĚRITELNOU PORUCHOU	42
5.2.1 Porucha – Sinus.....	42
5.2.2 Porucha – Skok	43
5.2.3 Porucha – Rampa	44

5.3	SIMULACE ŘÍZENÍ SE ZNÁMOU PORUCHOU	45
5.3.1	Porucha – Sinus	45
5.3.2	Porucha – Skok	46
5.3.3	Porucha – Rampa	47
5.4	POROVNÁNÍ RŮZNÝCH HODNOT PERIODY VZORKOVÁNÍ	48
5.4.1	Perioda vzorkování $T_v=1$	48
5.4.2	Perioda vzorkování $T_v=0,5$	48
5.4.3	Perioda vzorkování $T_v=0,1$	49
ZÁVĚR		50
SEZNAM POUŽITÉ LITERATURY.....		51
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....		52
SEZNAM OBRÁZKŮ		54
SEZNAM PŘÍLOH.....		56

ÚVOD

Prediktivní řízení je jednou z metod návrhů řízení, která si získala v posledních letech velkou popularitu. Prediktivní řízení je ve své podstatě založeno na použití diskretních modelech procesů, a proto odvození příslušných řídicích algoritmů je realizováno hlavně v diskretní oblasti. [1]

Jednou z předností prediktivního řízení je možnost uvažovat omezení vstupních a výstupních (případně stavových) veličin přímo při návrhu regulátoru. Z tohoto důvodu je jeho použití vhodné v průmyslových aplikacích. Algoritmy prediktivního řízení jsou při řízení procesů mnohostranně použitelné a robustní. Kvalita řízení je obvykle vyšší ve srovnání s PID regulátory. Jsou aplikovatelné na neminimálně fázové, nestabilní a mnoha rozměrové procesy, a rovněž na procesy s dopravním zpožděním. [1]

Hlavním cílem této práce je tvorba simulací prediktivního řízení s kompenzací poruchy v prostředí programu MATLAB.

Teoretická část se zabývá výpočtem dvou metod výpočtu prediktoru, a to za pomoci diofantických rovnic a metody přímého výpočtu z CARIMA modelu.

Dále je zde popis prostředí programu MATLAB. Popsána je aplikace GUIDE, která je též součástí MATLAB, a pomocí které byly vytvořeny výukové simulace.

Praktická část obsahuje popis jednotlivých simulací pro prediktivní řízení bez kompenzace poruchy, řízení s neznámou měřitelnou poruchou a řízení se známou poruchou. Jako zdroj poruchy je použita funkce sinus, skoková změna poruchové veličiny a rampa.

Zdrojové kódy aplikace jsou přiloženy na disku CD-ROM.

I. TEORETICKÁ ČÁST

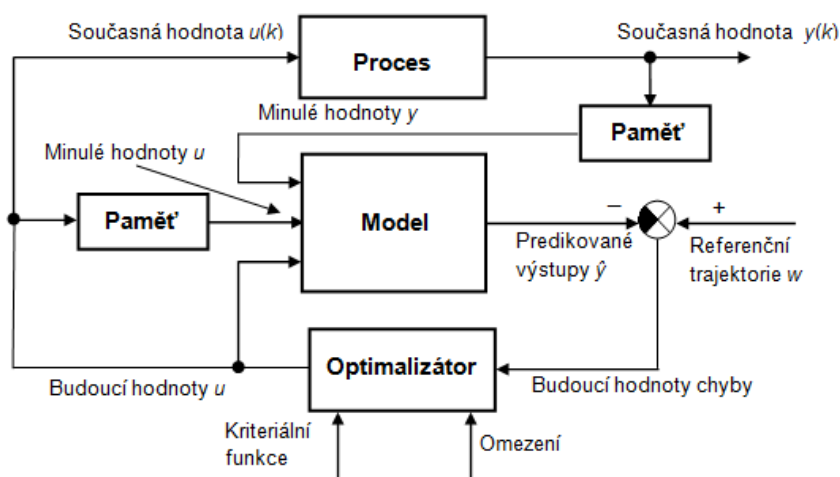
1 PREDIKTIVNÍ ŘÍZENÍ

Prediktivní řízení (Model Predictive Control - MPC) je metoda návrhu řízení. Během posledních let si získalo velkou popularitu, a ukázalo se jako velmi silný nástroj pro řízení různých druhů procesů. Ve své podstatě je založeno na diskretních respektive diskretních modelech procesů. Proto jsou odvození a vztahy prezentovány hlavně v diskretní oblasti. Pojem prediktivní řízení zahrnuje celou škálu řídicích algoritmů. Ty využívají modelu procesu a minimalizují účelovou funkci k určení akčního zásahu.

Prediktivní řízení je vhodné i pro nestabilní systémy, systémy s dopravním zpožděním a neminimálně fázové systémy. Velkou výhodou prediktivního řízení je, že můžeme uvažovat omezení akční a regulované veličiny již při návrhu regulátoru. Kvalita řízení je vyšší než u PID regulátorů.

Prediktivní řízení, jakožto třídu metod řízení, spojují určité charakteristiky:

- Na predikci budoucího řízení výstupu systému je použit matematický model řízeného systému
- Průběh trajektorie žádané veličiny je znám dopředu
- Výpočet posloupnosti budoucích akčních zásahů zahrnuje i minimalizaci vhodné účelové funkce (většinou kvadratické) s budoucími hodnotami přírůstků akčního zásahu a regulační odchylky
- Realizovaný je jen první akční zásah a celý postup minimalizace funkcionálu se opakuje v následující periodě vzorkování – tento princip se nazývá strategie pohyblivého horizontu [1], [2], [4], [5]



Obr. 1.: Základní struktura systému prediktivního řízení [1]

1.1 Prvky prediktivního řízení

1.1.1 Modely

Prediktivní řízení může být založeno na celé řadě druhů modelů. Hodně využívané jsou stavové modely, ale může být využito i modelu vstupně výstupních. Tato práce je založena na vstupně výstupním modelu ve formě přenosové funkce a CARIMA (Controlled Autoregressive Moving Average) modelu.

Přenosová funkce

Tento model je například použitý v metodě GPC (Generalized Predictive Control). Výstup je se vstupem svázán vztahem

$$A(q^{-1})y(k) = B(q^{-1})u(k) \quad (1.1)$$

Výhodou tohoto modelu je jeho platnost i pro nestabilní procesy. Na druhé straně je nutné znát řád polynomů A, B . [1]

CARIMA model

Hodně používaný model v prediktivním řízení je CARIMA model, který můžeme získat tím, že se přidá model šumu jako

$$A(z^{-1})Y(z^{-1}) = B(z^{-1})U(z^{-1}) + \frac{C(z^{-1})}{\Delta} E_s(z^{-1}) \quad (1.2)$$

kde $E_s(z^{-1})$ je neměřitelný náhodný šum, u kterého se předpokládá, že má nulovou střední hodnotu, konstantní kovarianci a $\Delta = 1 - z^{-1}$. Polynom C bude dále uvažován jako $C(z^{-1}) = 1$.

Znamé poruchy mohou být explicitně zohledněny v prediktivním řízení. Tyto poruchy jsou zahrnuty v predikčních rovnicích. V případě modelu (1.2), musí být upraven, aby zahrnoval poruchy [1], [4]

$$A(z^{-1})y(k) = B(z^{-1})u(k) + D(z^{-1})v(k) + \frac{C(z^{-1})}{\Delta} e_s(k) \quad (1.3)$$

kde $v(k)$ je známá porucha a $D(z^{-1})$ je polynom čitatele přenosu poruchy.

1.1.2 Výpočet optimálního řízení

1.1.2.1 Účelová funkce

Výpočet řídicího zákona MPC je převážně založen na minimalizaci následujícího kritéria

$$J(k) = \sum_{i=N_2}^{N_2} [e(k+i)]^2 + \lambda \sum_{i=1}^{N_u} [\Delta u(k+i-1)]^2 \quad (1.4)$$

kde $e(k+i)$ je vektor predikovaných regulačních odchylek a $\Delta u(k+i-1)$ je posloupnost budoucích přírůstků řízení, které mají být vypočítány.

Parametr N_1 je minimální horizont, N_2 maximální horizont a N_u řídicí horizont. N_1 a N_2 určují interval v budoucnosti, kdy je potřeba sledovat trajektorii žádané veličiny. Nastavením dostatečně vysoké hodnoty N_1 se můžeme vyhnout problémům s neminimálně fázovými procesy. Hodnota N_2 by měla pokrývat důležitou část přechodové charakteristiky, obvykle se volí porovnatelná s časem T_{90} (hodnota, za kterou výstupní veličina přejde z 10% na 90% své ustálené hodnoty). Využití řídicího horizontu N_u snižuje výpočtovou náročnost metody. Po úpravě rovnice (1.4) dostáváme

$$J(k) = \sum_{i=N_2}^{N_2} [\hat{y}(k+i) - w(k+i)]^2 + \lambda \sum_{i=1}^{N_u} [\Delta u(k+i-1)]^2 \quad (1.5)$$

kde $\hat{y}(k+i)$ je výstup i kroků v budoucnosti predikovaný vzhledem k informacím dostupných do času k a $w(k+i)$ je posloupnost žádané veličiny.

Implicitně se předpokládají omezení na přírůstky řízení v tvaru

$$\Delta u(k+i-1) = 0, \quad N_u < i < N_2 \quad (1.6)$$

Předpokládá se, že trajektorie žádané veličiny $w(k+i)$ je známa. Pokud to neplatí, nejčastěji se uvažuje konstantní a rovná aktuální hodnotě žádané veličiny.

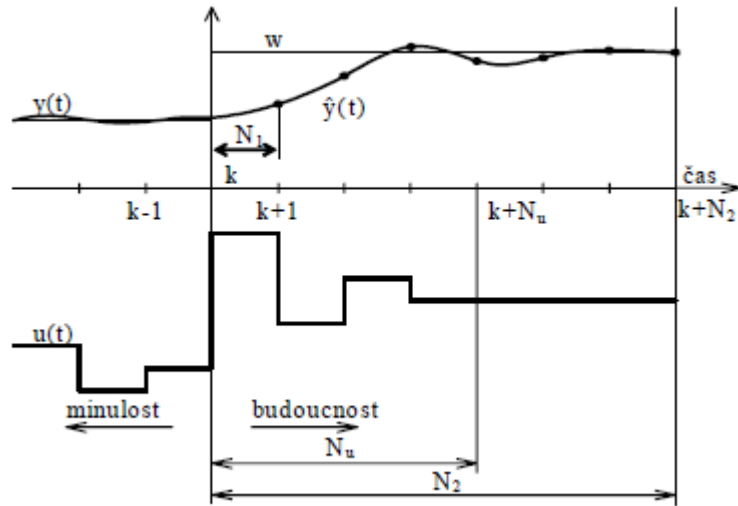
Cílem prediktivního řízení je spočítat sekvenci budoucích hodnot změny akčního zásahu $[\Delta u(k), \Delta u(k+1), \dots]$ tak, aby bylo minimalizováno kritérium (1.5). Pro další práci je třeba kritérium (1.5) převést do maticové podoby. [1], [4]

1.1.2.2 Prediktor

Výstup modelu lze spočítat jako součet volné odezvy modelu y_0 a nucené odezvy y_n . Volná odezva modelu je závislá pouze na minulých hodnotách řízení a výstupu systému, přičemž se předpokládá, že budoucí řízení je konstantní (nulové změny akčního zásahu).

Nucená odezva bere v potaz nenulové budoucí změny akčního zásahu. Pro lineární i obecně nelineární model systému není problém volnou odezvu spočítat.

$$y = y_n + y_0 \quad (1.7)$$



Obr. 2.: Princip prediktivního řízení [4]

Prediktor ve vektorovém tvaru je dán vztahem

$$\hat{y} = G\Delta u + y_0 \quad (1.8)$$

kde \hat{y} je vektor výstupních predikcí systému na horizontu o délce $N_2 - N_1$. Nucenou odezvu lze spočítat jako součin matice dynamiky G (prvky tvoří koeficienty přechodové charakteristiky) a vektoru budoucích změn akčního zásahu Δu , který obecně dopředu neznáme.

Matice dynamiky G je Jacobián modelu (parciální derivace modelu podle jeho proměnných), pro kauzální systémy se jedná o trojúhelníkovou matici, pro lineární systémy je tvar matice G známý

$$G = \begin{bmatrix} g_0 & 0 & 0 & \cdots & 0 \\ g_1 & g_0 & 0 & \cdots & 0 \\ g_2 & g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{N_2-1} & g_{N_2-2} & g_{N_2-3} & \cdots & g_0 \end{bmatrix} \quad (1.9)$$

Rozměry matice jsou $[N_2 - N_1 + 1 \times N_u]$.

Pokud je znám budoucí průběh poruchové veličiny, tak se kvalita řízení zvýší zahrnutím poruchové veličiny do prediktoru. Zvýší se tak míra shody predikovaného výstupu se skutečností. Proto je potřeba rovnici (1.8) přepsat na následující tvar

$$\hat{y} = G\Delta u + L\Delta v + y_0 \quad (1.10)$$

kde Δv je vektor budoucích změn poruchové veličiny. Matice dynamiky L je odezva systému na skokovou změnu poruchové veličiny. Jedná o trojúhelníkovou matici, pro lineární systémy je tvar matice L známý

$$L = \begin{bmatrix} l_0 & 0 & 0 & \cdots & 0 \\ l_1 & l_0 & 0 & \cdots & 0 \\ l_2 & l_1 & l_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & 0 \\ l_{N_2-1} & l_{N_2-2} & l_{N_2-3} & \cdots & l_0 \end{bmatrix} \quad (1.11)$$

Rozměry matice jsou $[N_2 - N_1 + 1 \times N_u]$.

Dále pak můžeme účelovou funkci (1.5) převést do tvaru

$$J = (\hat{y} - w)^T (\hat{y} - w) + \lambda \Delta u^T \Delta u \quad (1.12)$$

$$J = (G\Delta u + y_0 - w)^T (G\Delta u + y_0 - w) + \lambda \Delta u^T \Delta u \quad (1.13)$$

Účelovou funkci (1.13) dále převedeme do maticového tvaru používaného pro metody kvadratického programování

$$J = \frac{1}{2} \Delta u^T H \Delta u + b^T \Delta u + f_0 \quad (1.14)$$

kde

$$H = 2(G^T G + \lambda I) \quad (1.15)$$

$$b^T = 2(y_0 - w)^T G \quad (1.16)$$

$$f_0 = (y_0 - w)^T (y_0 - w) \quad (1.17)$$

Člen f_0 v rovnici (1.14) není závislý na Δu , proto na řešení optimalizační úlohy nemá žádný vliv a z rovnice jej můžeme vypustit. Výsledný tvar kritéria pro optimalizaci je [1], [4]

$$J = \frac{1}{2} \Delta u^T H \Delta u + b^T \Delta u \quad (1.18)$$

1.1.2.3 Omezující podmínky

V reálné praxi najdeme hodně procesů, které vyžadují omezení. Nejčastěji se jedná o omezení na vstupy, které mohou pracovat jen v určitém rozsahu hodnot (průtoky nemůžou být negativní, ventil může být otevřený maximálně na 100%), popřípadě rychlost změny vstupu je omezená. Mimo to, existují i doporučené limity pro výstupy z procesu (z bezpečnostního, environmentálního hlediska). Schopnost pracovat s ohraničeními je jednou z hlavních předností prediktivního řízení, která způsobila jeho velkou rozšířenost a přijatelnost v průmyslu. Je to dané aj tím, že většina procesů je řízena na úrovních blízkých se omezujícím podmínkám nebo přímo na nich. Jeden ze způsobů, jak se vypořádat s omezeními, je nastavit řízení, které je mimo rozsah, na limitní hodnotu. Je však víc důvodů, proč tento způsob není příliš vhodný.

Nejjednodušším a v praxi nejpoužívanějším způsobem aplikace omezení je analytické řešení optimalizační úlohy bez omezení a následná aplikace omezení - saturace na výsledek. Toto řešení je jednoduché, ale nezaručí optimální řízení podle námi zvolených kritérií. Další nevýhodou je také to, že takovéto omezení lze aplikovat pouze na veličiny vystupující z algoritmu optimalizace (akční zásah a změna akčního zásahu).

Vhodnějším řešením je tedy řešit optimalizační úlohu již s danými omezeními. Tento přístup umožní omezit nejen výstupní veličiny algoritmu, ale také samotný výstup soustavy a v případě stavového modelu i jednotlivé vnitřní stavy systému.

Omezení rozlišujeme jako tvrdá (hard constraints) a měkká (soft constraints):

- Tvrdá omezení jsou fyzická omezení procesů, jejichž hranice v prostoru řešení nelze za žádných okolností překročit.
- Měkká omezení jsou ta, která mohou být za určitých podmínek porušena. [1], [4]

Nejčastěji používaná tvrdá omezení:

- Omezení změny akčního zásahu
- Omezení akčního zásahu
- Omezení výstupní veličiny

Pokud se optimalizuje kritérium a je třeba zároveň zohlednit omezení vstupních či výstupních veličin, pak již nelze řešit problém analyticky, ale musí se použít numerické optimalizační metody výpočtu. Je tedy třeba řešit problém optimalizace kvadratické objektivní funkce s lineárními omezujícími podmínkami. To vede na řešení problému kvadratického programování (QP - quadratic programming).

Obecně můžeme řešit problém kvadratického programování různými způsoby. Neexistuje ale žádné univerzální řešení pro všechny případy. Optimalizaci s omezením vstupních či výstupních veličin můžeme v případě omezujících podmínek ve formě rovností převést na řešení optimalizačního problému bez omezení tak, že omezující podmínky začleníme přímo do kritéria. Jednou z dalších možností je převedení problému na řešení sekvence jednodušších optimalizačních úloh.

Při řešení optimalizačního problému v rámci MPC se využívá Optimization Toolbox pro MATLAB, kde pomocí funkce *quadprog* řešíme kvadratické programování. Této funkce můžeme využít i při řízení procesů v reálném čase. [1]

V programu této práce byl výpočet optimalizačního problému proveden za pomoci příkazu *quadprog* v prostředí softwaru MATLAB.

1.2 Metoda výpočtu prediktoru pomocí diofantických rovnic

Často využívaným modelem, který zahrnuje model šumu je CARIMA model, který má tvar rovnice (1.3). V tomto modelu je zahrnuto působení známe poruchy. Pro j -krokovou predikci můžeme rovnici (1.3) napsat ve tvaru

$$\hat{y}(k+j) = \frac{B}{A}u(k+j) + \frac{D}{A}v(k+j) + \frac{C}{\Delta A}e_s(k+j) \quad (1.19)$$

Z posledního členu rovnice (1.2) můžeme oddělit členy s kladnými mocninami z^{-1} , kde E je polynom řádu $j-1$ a F je polynom stejného řádu jako polynom A .

$$\frac{C(z^{-1})}{\Delta A(z^{-1})} = E_j(z^{-1}) + z^{-j} \frac{F_j(z^{-1})}{\Delta A(z^{-1})} \quad (1.20)$$

Po dosazení do rovnice (1.2) získáme prediktor

$$\hat{y}(k+j) = \frac{B}{A}u(k+j) + \frac{D}{A}v(k+j) + E_j e_s(k+j) + \frac{F_j}{\Delta A}e_s(k) \quad (1.21)$$

Z rovnice (1.1) vyjádříme poruchu a dosadíme do rovnice (1.2).

$$\begin{aligned}
\hat{y}(k+j) = & \frac{B}{C} \left[\frac{C}{\Delta A} - z^{-j} \frac{F_j}{\Delta A} \right] \Delta u(k+j) \\
& + \frac{D}{C} \left[\frac{C}{\Delta A} - z^{-j} \frac{F_j}{\Delta A} \right] \Delta v(k+j) + \frac{F_j}{C} y(k) \\
& + E_j e_s(k+j)
\end{aligned} \tag{1.22}$$

Po dosazení dostaneme

$$\begin{aligned}
\hat{y}(k+j) = & \frac{BE_j}{C} \Delta u(k+j) + \frac{DE_j}{C} \Delta v(k+j) + \frac{F_j}{C} y(k) \\
& + E_j e_s(k+j)
\end{aligned} \tag{1.23}$$

Nyní můžeme provést dvě zjednodušení – porucha bude považována za bílý šum a budoucí hodnoty šumu budou dále zanedbány.

$$\hat{y}(k+j) = BE_j \Delta u(k+j) + DE_j \Delta v(k+j) + F_j y(k) \tag{1.24}$$

Můžeme definovat polynomy G_j a L_j jako následující

$$G_j = BE_j \tag{1.25}$$

$$L_j = DE_j \tag{1.26}$$

$$\hat{y}(k+j) = G_j \Delta u(k+j) + L_j \Delta v(k+j) + F_j y(k) \tag{1.27}$$

Pro výpočet j -krokového prediktoru je potřeba řešit následující diofantickou rovnici.

$$1 = E_j \Delta A + z^{-j} F_j \tag{1.28}$$

Dále je potřeba řešit rekurzi diofantické rovnice (1.11). Jednotlivé polynomy diofantické rovnice mohou být rozšířeny následovně

$$\tilde{A}(z^{-1}) = \Delta A(z^{-1}) = 1 + (1 - a_1)z^{-1} + (a_1 - a_2)z^{-2} + a_2 z^{-3} \tag{1.29}$$

$$E_j(z^{-1}) = E_{j,0} + E_{j,1}z^{-1} + E_{j,2}z^{-2} + \dots + E_{j,j-1}z^{j-1} \tag{1.30}$$

$$F_j(z^{-1}) = F_{j,0} + F_{j,1}z^{-1} + F_{j,2}z^{-2} + \dots + F_{j,j-1}z^{j-1} \tag{1.31}$$

Uvažujme diofantickou rovnici odpovídající následující predikci $\hat{y}(k+j+1)$.

$$1 = E_{j+1}(z^{-1})\tilde{A}(z^{-1}) + z^{-(j+1)}F_{j+1}(z^{-1}) \tag{1.32}$$

Odečtením diofantické rovnice (1.11) od diofantické rovnice (1.15) dostaneme

$$0 = \left(E_{j+1}(z^{-1}) - E_j(z^{-1}) \right) \tilde{A}(z^{-1}) + z^{-j} \left(z^{-1} F_{j+1}(z^{-1}) - F_j(z^{-1}) \right) \quad (1.33)$$

Nyní je možné definovat člen

$$E_{j+1}(z^{-1}) - E_j(z^{-1}) = \tilde{R}(z^{-1}) - R_j z^{-1} \quad (1.34)$$

Po dosazení rovnice (1.17) do (1.16) dostaneme

$$0 = \tilde{R}(z^{-1}) \tilde{A}(z^{-1}) + z^{-j} \left(R_j \tilde{A}(z^{-1}) + z^{-1} F_{j+1}(z^{-1}) - F_j(z^{-1}) \right) \quad (1.35)$$

A je zřejmé, že je $\tilde{R}(z^{-1}) = 0$ za účelem získání nulové hodnoty na levé straně rovnice (1.18). Polynom E pak může být rekurzivně vypočten za pomoci následujícího vztahu

$$E_{j+1}(z^{-1}) = E_j(z^{-1}) + R_j z^{-1} \quad (1.36)$$

Tyto vztahy mohou být získány z rovnice (1.18).

$$R_j = F_{j,0} \quad (1.37)$$

$$F_{j+1,i} = F_{j,i+1} - R_j \tilde{A}_{i+1} \quad \text{pro } i = 0 \dots \delta(F_{j+1}) \quad (1.38)$$

Počáteční podmínky pro rekurze jsou následující

$$E_j = 1 \quad F_i = z(1 - \tilde{A}) \quad (1.39)$$

Vytvořením polynomů

$$E_j(z^{-1})B(z^{-1}) = G_j(z^{-1}) + z^{-j}G_{jp}(z^{-1}) \quad (1.40)$$

$$E_j(z^{-1})D(z^{-1}) = L_j(z^{-1}) + z^{-j}L_{jp}(z^{-1}) \quad (1.41)$$

Predikční rovnici lze napsat jako

$$\begin{aligned} \hat{y}(k+j) = & G_j(z^{-1})\Delta u(k+j) + L_j(z^{-1})\Delta v(k+j) \\ & + G_{jp}(z^{-1})\Delta u(k-1) + L_{jp}(z^{-1})\Delta v(k-1) \\ & + F_j y(k) \end{aligned} \quad (1.42)$$

Poslední tři členy rovnice (1.24) jsou závislé na minulých hodnotách výstupu, vstupu a poruch procesu a představují volnou odezvu procesu. První dva členy jsou závislé na

budoucích hodnotách přírůstků řízení a poruchy procesu a představují nucenou odezvu procesu. Rovnice (1.24) může být přepsána do následujícího tvaru.

$$\hat{y}(k+j) = G_j(z^{-1})\Delta u(k+j) + L_j(z^{-1})\Delta v(k+j) + y_{0j} \quad (1.43)$$

Kde následující rovnice představuje volnou odezvu. [3]

$$y_{0j} = G_{jp}(z^{-1})\Delta u(k-1) + L_{jp}(z^{-1})\Delta v(k-1) + F_j y(k) \quad (1.44)$$

1.3 Metoda výpočtu prediktoru založená na přímém výpočtu z CARIMA modelu

Tato metoda je založena na analytickém odvození určitého počtu predikcí a následném rekursivním výpočtu dalších predikcí. Počet predikcí, které jsou nutné, aby byly vypočteny, přímo závisí na řádu systému.

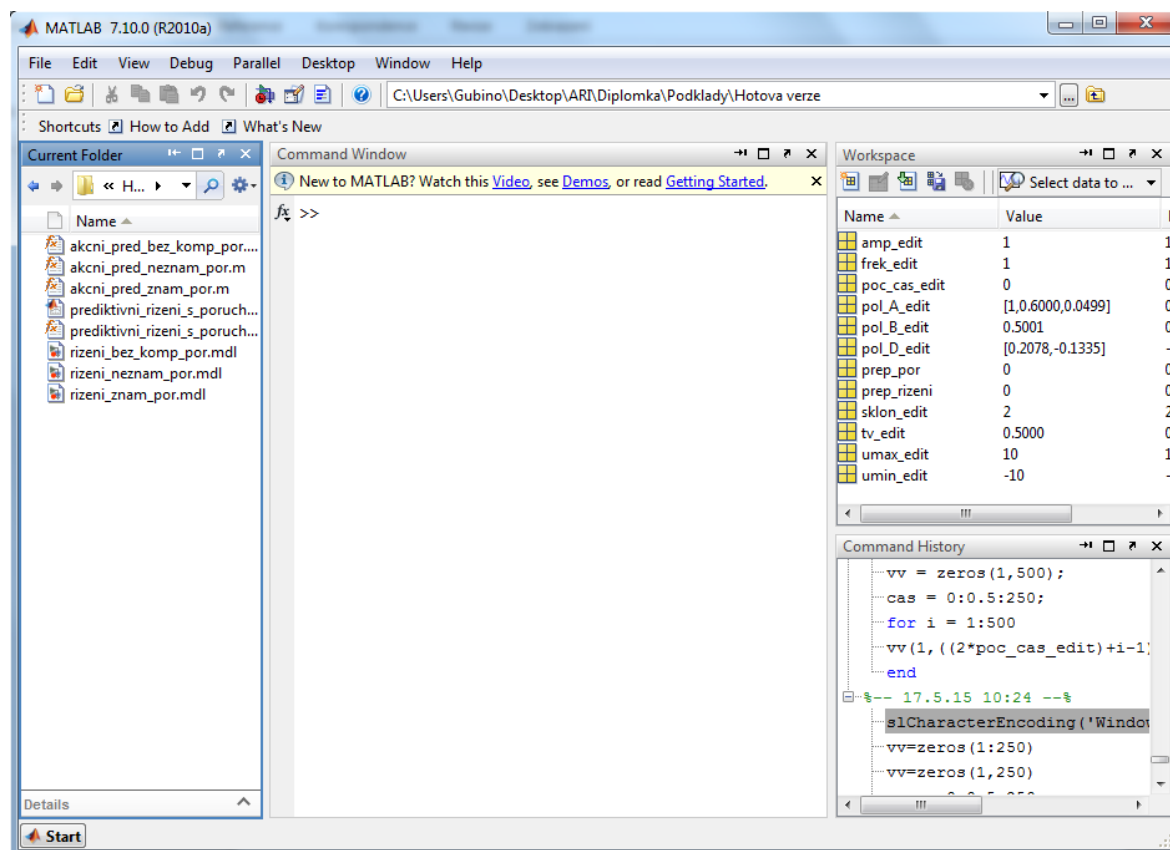
2 MATLAB

MATLAB je integrované prostředí pro vědeckotechnické výpočty, modelování, návrhy algoritmů, simulace, analýzu a prezentaci dat, paralelní výpočty, měření a zpracování signálů, návrhy řídicích a komunikačních systémů. MATLAB je nástroj jak pro pohodlnou interaktivní práci, tak pro vývoj širokého spektra aplikací. [9]

Název MATLAB vznikl zkrácením slov MATrix LABoratory (volně přeloženo „maticová laboratoř“), což odpovídá skutečnosti, že klíčovou datovou strukturou při výpočtech v MATLABu jsou matice. Vlastní programovací jazyk vychází z jazyka Fortran. [10]

MATLAB poskytuje svým uživatelům nejen mocné grafické a výpočetní nástroje, ale i rozsáhlé specializované knihovny funkcí spolu s výkonným programovacím jazykem čtvrté generace. Knihovny jsou svým rozsahem využitelné prakticky ve všech oblastech lidské činnosti. [9]

2.1 Prostředí MATLAB



Obr. 2.1: Prostředí MATLAB

2.2 Toolboxy

Důležitou částí instalace MATLABu jsou knihovny funkcí (adresáře s m a mex soubory), které jsou nazývány toolboxy. Toolboxy obsahují vždy uceleným způsobem, včetně dokumentace a příkladů, zpracovaný určitý obor numerické matematiky, analytické matematiky, statistiky, systémového přístupu k regulacím a další obory, ve kterých nachází Matlab uplatnění. Takovýchto balíků je dnes k dispozici asi 35. [10]

2.3 Simulink

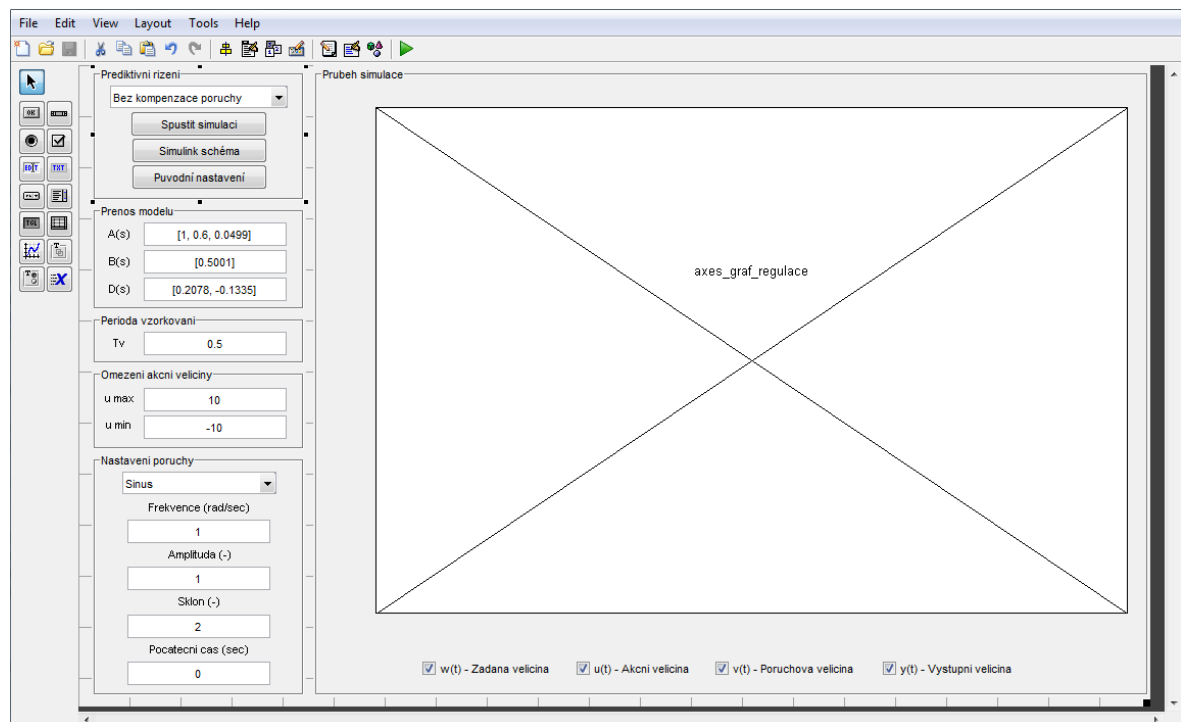
Simulink je nadstavba MATLABu pro simulaci a modelování dynamických systémů, který využívá algoritmy MATLABu pro numerické řešení nelineárních diferenciálních rovnic. Poskytuje uživateli možnost rychle a snadno vytvářet modely dynamických soustav ve formě blokových schémat a rovnic. Pomocí Simulinku a jeho grafického editoru lze vytvářet modely lineárních, nelineárních, v čase diskrétních nebo spojitých systémů pouhým přesouváním funkčních bloků myší. [11]

2.4 Guide

Prostředí, které umožňuje vytvářet aplikace s grafickým rozhraním, se nazývá GUIDE (Graphical User Interface Development Environment). Obsahuje následující vlastnosti:

- umožňuje vytvářet a editovat uživatelské rozhraní pomocí základních komponent (checkbox, sliders, tables apod.)
- všechny komponenty, které jsou vytvořeny v tomto prostředí, lze měnit za běhu aplikace
- vzhled vytvořené GUI aplikace je ukládán do souboru s příponou *.fig a jeho zdrojový kód s příponou *.m

Spuštění průvodce pro tvorbu GUI aplikací je možné více způsoby. Jedním z nich je využít základní menu File/New/GUI nebo zápisem a potvrzením příkazu guide v hlavním prostředí Matlabu. [10]



Obr. 2.2.: Ukázka prostředí GUIDE – program prediktivni_řízení_s_poruchou.fig

II. PRAKTICKÁ ČÁST

3 VÝPOČET PREDIKTORU

V praktické části jsou odvozeny predikční vztahy pro zvolený model systému. Poté byly vypočtené metody naprogramovány v prostředí MATLAB a v prostředí Simulink bylo vytvořeno simulační schéma.

3.1 Model

Přenosová funkce systému a poruchy

Byly zvoleny přenosové funkce druhého řádu, které se osvědčily pro popis a návrh řízení celé řady procesů. Následující model byl zvolen pro systém

$$G_s(z^{-1}) = \frac{B(z^{-1})}{A(z^{-1})} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{Y(z^{-1})}{U(z^{-1})} \quad (1.45)$$

kde

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} \quad (1.46)$$

je polynom $A(z^{-1})$ a

$$B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} \quad (1.47)$$

je polynom $B(z^{-1})$.

A model známé poruchy vypadá následovně

$$G_v(z^{-1}) = \frac{D(z^{-1})}{A(z^{-1})} = \frac{d_1 z^{-1} + d_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{Y(z^{-1})}{V(z^{-1})} \quad (1.48)$$

kde

$$D(z^{-1}) = d_1 z^{-1} + d_2 z^{-2} \quad (1.49)$$

je polynom $D(z^{-1})$ a polynom $A(z^{-1})$ je stejný jako v rovnici (1.46).

CARIMA model

Dále byl v této práci použit CARIMA model, který má rovnici (1.3). Po úpravě této rovnice dostaneme následující model

$$\Delta A(z^{-1})y(k) = B(z^{-1})\Delta u(k) + D(z^{-1})\Delta v(k) + C(z^{-1})e_s(k) \quad (1.50)$$

Jestliže uvažujeme $C(z^{-1}) = 1$ a $e_s(k)$, což je bílý šum, položíme rovno 0. Tak můžeme dále psát rovnici

$$\Delta A(z^{-1})y(k) = B(z^{-1})\Delta u(k) + D(z^{-1})\Delta v(k) \quad (1.51)$$

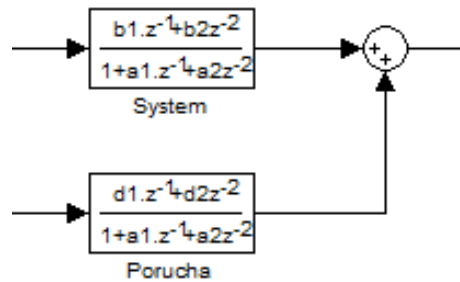
Po dosazení dostaneme

$$\begin{aligned} (1 - z^{-1})(1 + a_1 z^{-1} + a_2 z^{-2})y(k) \\ = (b_1 z^{-1} + b_2 z^{-2})\Delta u(k) + (d_1 z^{-1} \\ + d_2 z^{-2})\Delta v(k) \end{aligned} \quad (1.52)$$

Po dalších úpravách dostáváme výslednou diferenční rovnici

$$\begin{aligned} y(k) = (1 - a_1)y(k - 1) + (a_1 - a_2)y(k - 2) + a_2 y(k - 3) \\ + b_1 \Delta u(k - 1) + b_2 \Delta u(k - 2) + d_1 \Delta v(k - 1) \\ + d_2 \Delta v(k - 2) \end{aligned} \quad (1.53)$$

Blokové schéma



Obr. 3.1.: Blokové schéma

3.2 Metoda výpočtu prediktoru založená na přímém výpočtu z CARIMA modelu

Diferenční rovnice CARIMA modelu bez šumu je vyjádřena v rovnici (1.53).

Bylo nutné přímo analyticky vypočítat predikční rovnice pro tři následující hodnoty. Řád modelu definuje, že výpočet predikcí je založen na třech předchozích hodnotách výstupu systému. Predikční rovnice (1.10), lze po úpravě zapsat ve formě matice.

$$\begin{aligned}
\overbrace{\begin{bmatrix} \hat{y}(k+1) \\ \hat{y}(k+2) \\ \hat{y}(k+3) \end{bmatrix}}^{\hat{y}} &= \overbrace{\begin{bmatrix} g_1 & 0 \\ g_2 & g_1 \\ g_3 & g_2 \end{bmatrix}}^{G\Delta u} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \end{bmatrix} + \\
&\quad \overbrace{\begin{bmatrix} l_1 & 0 \\ l_2 & l_1 \\ l_3 & l_2 \end{bmatrix}}^{L\Delta v} \begin{bmatrix} \Delta v(k) \\ \Delta v(k+1) \end{bmatrix} + \quad (1.54)
\end{aligned}$$

$$\overbrace{\begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} \\ p_{21} & p_{22} & p_{23} & p_{24} & p_{25} \\ p_{31} & p_{32} & p_{33} & p_{34} & p_{35} \end{bmatrix}}^{y_0} \begin{bmatrix} y(k) \\ y(k-1) \\ y(k-2) \\ \Delta u(k-1) \\ \Delta v(k-1) \end{bmatrix}$$

Je možné rozdělit výpočet predikcí na rekurzi volného odezvy a rekurzi matice dynamiky.

Všechny prvky $p(i, j)$ $i = 1 \dots 3, j = 1 \dots 5$ musí být přímo vypočítány pro inicializaci rekurze.

$$\begin{aligned}
p_{11} &= (1 - a_1) & p_{12} &= (a_1 - a_2) & p_{13} &= a_2 & p_{14} &= b_2 & p_{15} &= d_2 \\
p_{21} &= (1 - a_1)^2 + (a_1 - a_2) & p_{22} &= (1 - a_1)(a_1 - a_2) + a_2 \\
p_{23} &= a_2(1 - a_1) & p_{24} &= (1 - a_1)b_2 & p_{25} &= (1 - a_1)d_2 \\
p_{31} &= (1 - a_1)^3 + 2(1 - a_1)(a_1 - a_2) + a_2 \\
p_{32} &= (1 - a_1)^2(a_1 - a_2) + a_2(1 - a_1) + (a_1 - a_2)^2 \\
p_{33} &= a_2(1 - a_1)^2 + (a_1 - a_2)a_2 \\
p_{34} &= (1 - a_1)^2b_2 + (a_1 - a_2)b_2 \\
p_{35} &= (1 - a_1)^2d_2 + (a_1 - a_2)d_2
\end{aligned} \quad (1.55)$$

Další řádek volné odezvy matice je opakovaně vypočítáván na základě tří předchozích predikcí, dokud není dosaženo horizontu predikce.

$$\begin{aligned}
p_{41} &= (1 - a_1)p_{31} + (a_1 - a_2)p_{21} + a_2p_{11} \\
p_{42} &= (1 - a_1)p_{32} + (a_1 - a_2)p_{22} + a_2p_{12} \\
p_{43} &= (1 - a_1)p_{33} + (a_1 - a_2)p_{23} + a_2p_{13} \\
p_{44} &= (1 - a_1)p_{34} + (a_1 - a_2)p_{24} + a_2p_{14} \\
p_{45} &= (1 - a_1)p_{35} + (a_1 - a_2)p_{25} + a_2p_{15}
\end{aligned} \tag{1.56}$$

Nucená odezva v rovnici (1.54), má následující podobu

$$\begin{aligned}
&G\Delta u \\
&= \begin{bmatrix} b_1 & 0 \\ b_1(1 - a_1) + b_2 & b_1 \\ b_1(a_1 - a_2) + b_1(1 - a_1)^2 + (1 - a_1)b_2 & b_1(1 - a_1) + b_2 \end{bmatrix} \tag{1.57} \\
&\begin{bmatrix} \Delta u(k) \\ \Delta u(k + 1) \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
&L\Delta v \\
&= \begin{bmatrix} d_1 & 0 \\ d_1(1 - a_1) + d_2 & d_1 \\ d_1(a_1 - a_2) + d_1(1 - a_1)^2 + (1 - a_1)d_2 & d_1(1 - a_1) + d_2 \end{bmatrix} \tag{1.58} \\
&\begin{bmatrix} \Delta v(k) \\ \Delta v(k + 1) \end{bmatrix}
\end{aligned}$$

Rekurze matic G a L je podobná rekurzi volné odezvy matice. Další prvek v prvním sloupci je opakovaně vypočítáván a zbývající sloupce jsou posunuty. Tento postup se provádí opakovaně, dokud není dosaženo horizontu predikce. V případě, že řídicí horizont je nižší než horizont predikce, počet sloupců v matici G se snižuje. Výpočet nových prvků se provádí následujícím způsobem

$$\begin{aligned}
g_4 &= (1 - a_1)g_3 + (a_1 - a_2)g_2 + a_2g_1 \\
l_4 &= (1 - a_1)l_3 + (a_1 - a_2)l_2 + a_2l_1
\end{aligned} \tag{1.59}$$

3.3 Metoda výpočtu prediktoru pomocí diofantických rovnic

V případě systému druhého řádu, polynom \tilde{A} má následující podobu

$$\tilde{A}(z^{-1}) = \Delta A(z^{-1}) = 1 + (a_1 - 1)z^{-1} + (a_2 - a_1)z^{-2} - a_2 z^{-3} \quad (1.60)$$

Počáteční podmínky rekurze jsou

$$E_1 = 1$$

$$\begin{aligned} F_1 &= z(1 - \tilde{A}) = 1 - a_1 + z^{-1}(a_1 - a_2) + z^{-2}a_2 \\ &= f_0 + z^{-1}f_1 + z^{-2}f_2 \end{aligned} \quad (1.61)$$

Inicializace matice volné odezvy a matic dynamiky je následující

$$\begin{aligned} x &= [f_0 \quad f_1 \quad f_2 \quad b_2 \quad d_2] \\ G &= b_1 \quad L = d_1 \end{aligned} \quad (1.62)$$

Rekurze pak postupuje podle předem zavedených kroků.

$$R = f_0 \quad (1.63)$$

$$f_0 = f_1 - R(a_1 - 1) \quad (1.64)$$

$$f_1 = f_2 - R(a_2 - a_1) \quad (1.65)$$

$$f_2 = R a_2 \quad (1.66)$$

$$E = [E \quad R] \quad (1.67)$$

Rozšíření matic dynamiky a volné odezvy je následující

$$G = \begin{bmatrix} G \\ b_1 E(i+1) + b_2 E(i) \end{bmatrix} \quad L = \begin{bmatrix} G \\ d_1 E(i+1) + d_2 E(i) \end{bmatrix} \quad (1.68)$$

$$x = \begin{bmatrix} x \\ b_1 E(i+1) d_2 E(i+1) \end{bmatrix} \quad (1.69)$$

I když každá metoda využívá jiný postup výpočtu, výsledné predikční rovnice jsou stejné.

4 SIMULAČNÍ PROGRAM

Výpočty provedené v předchozích kapitolách byly pro ověření naprogramovány v softwaru MATLAB a bloková schémata byla vytvořena v prostředí Simulink. Všechn software je dostupný na příloženém CD. Verze programu MATLAB je 7.10.0 (R2010a). Pro tvorbu grafického prostředí simulace byl použit nástroj GUIDE, který je součástí programu MATLAB. Jednotlivé programy jsou popsány níže.

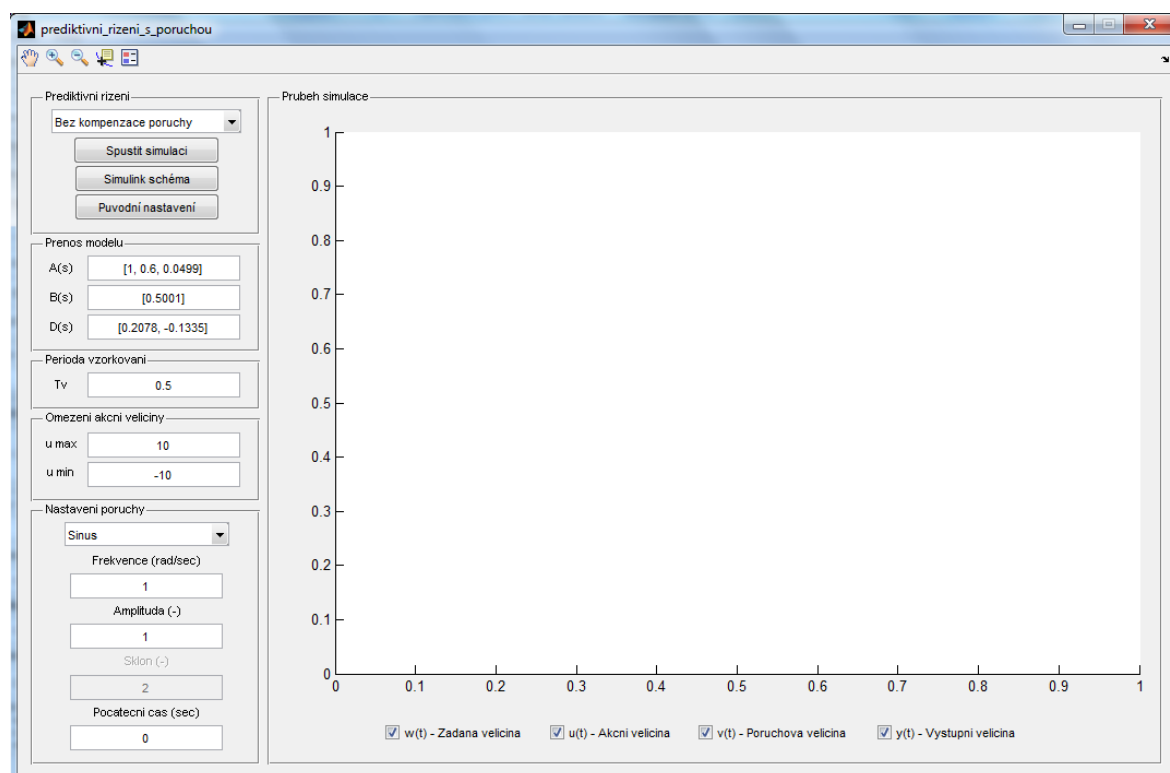
Žádaná veličina a skoková poruchová veličina má přesně daný průběh, který nelze měnit, ale u poruchové veličiny pro sinus a rampu jsou dány vlastnosti, které je možné upravit podle potřeby. Měnit lze amplitudu, frekvenci, sklon a počáteční čas poruchy.

4.1 Popis prostředí simulačního programu

Před spuštěním simulačního programu je nejprve potřeba do příkazového řádku v prostředí napsat a potvrdit následující příkaz (1.70), protože je zde použito jiné kódování znaků.

$$slCharacterEncoding('Windows - 1252') \quad (1.70)$$

Poté můžeme spustit soubor `prediktivni_rizeni_s_poruchou.m`, po jehož spuštění se nám otevře následující okno.



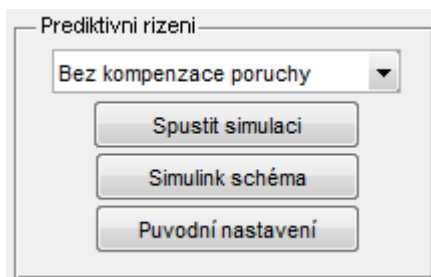
Obr. 4.1.: Prostředí simulačního programu

Na levé straně okna simulačního programu se nacházejí bloky, pomocí kterých lze měnit nastavení simulace. Velkou část simulačního programu tvoří okno, kde se zobrazují průběhy prováděných simulací.

4.1.1 Popis bloků simulačního programu

V následující části jsou popsány funkce jednotlivých bloků simulačního programu

Prediktivní řízení



*Obr. 4.2.: Blok výběru
prediktivního řízení*

Po kliknutí na pop-up menu jsou na výběr 3 verze simulace:

- Bez kompenzace poruchy - Řízení bez kompenzace poruchy
- Neznáma měřitelná porucha - Řízení s neznámou, ale měřitelnou poruchou
- Známa porucha - Řízení s předem známou poruchou

Tlačítko „Spustit simulaci“ slouží ke spuštění simulace v prostředí Simulink, které probíhá na pozadí. Tudiž potom vidíme jen výsledek simulace, který se objeví ve formě grafu v bloku „Průběh simulace“.

Pro zobrazení schéma v prostředí Simulink slouží tlačítko „Simulink schéma“, které se otevře v dalším vyskakovacím okně.

Pokud bychom chtěli vrátit simulaci do původního nastavení, stačí jen kliknout na tlačítko „Původní nastavení“.

Pokud simulace hlásí nějaké chybné hlášení, např. „Perioda vzorkování není zadána!“, tak není možné ji spustit, dokud se chyba neodstraní.

Přenos modelu

Přenos modelu

A(s) [1, 0.6, 0.0499]

B(s) [0.5001]

D(s) [0.2078, -0.1335]

Obr. 4.3.: Blok přenosu modelu

Následující blok slouží k nastavení spojitého přenosu modelu pro systém (1.45) a známou poruchu (1.48). Blokové schéma přenosu modelu je vidět na Obr. 3.1.

Omezení nastavení řádů polynomů:

- Polynom A – 2.řád
- Polynom B – 0. nebo 1. řád
- Polynom C – 0. nebo 1. řád

Perioda vzorkování

Perioda vzorkování

Tv 0.5

Obr. 4.4.: Blok periody
vzorkování

Tento blok slouží k nastavení periody vzorkování pro celou simulaci.

Omezení nastavení periody vzorkování:

- Tv – nesmí být menší nebo rovna 0

Omezení akční veličiny

Omezení akční veličiny

u max 10

u min -10

Obr. 4.5.: Blok omezení akční
veličiny

Tento blok slouží k nastavení omezení akční veličiny. Kde „u max“ je nastavení pro horní mez a „u min“ pro dolní mez.

Omezení nastavení omezení akční veličiny:

- „u max“ musí být větší než „u min“

Nastavení poruchy

Nastavení poruchy

Sinus

Frekvence (rad/sec)

1

Amplituda (-)

1

Sklon (-)

2

Počáteční čas (sec)

0

*Obr. 4.6.: Blok nastavení
poruchy*

Následující blok slouží k výběru typu poruchy, kde je možnost výběru z 3 typů:

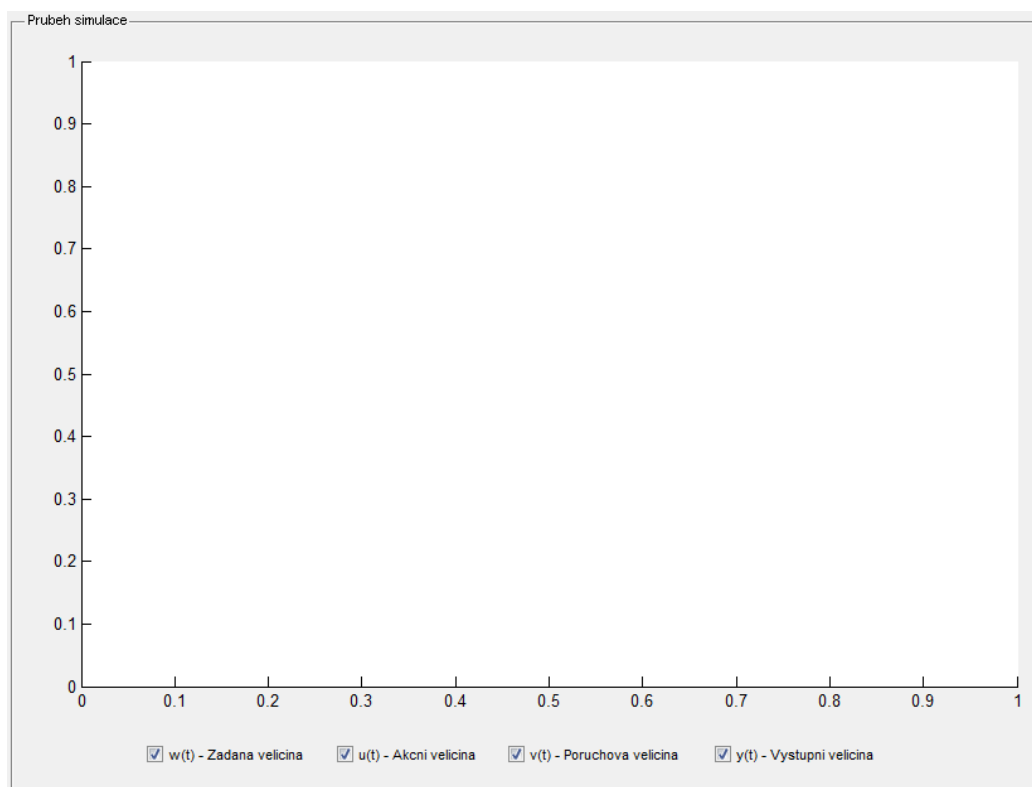
- Sinus – je možné změnit nastavení frekvence, amplitudy a počátečního času
- Skok – je pevně dán průběh poruchové veličiny
- Rampa – je možné nastavení sklonu a počátečního času

Jak je uvedeno, tak každá porucha má nastavitelné parametry. Parametr, který není možné u poruchy nastavit, tak se stane neaktivním (pole zešedne).

Omezení nastavení poruch:

- Frekvence – musí být větší a nesmí se rovnat 0
- Amplituda – bez omezení
- Sklon – bez omezení
- Počáteční čas – musí být v rozsahu 0-200

Průběh simulace

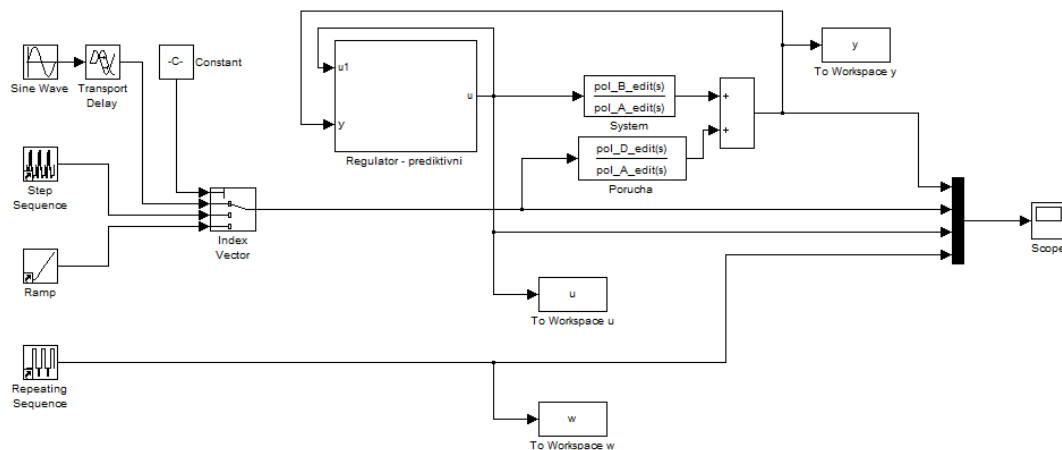


Obr. 4.7.: Blok průběhu simulace

Tento blok slouží k zobrazení výsledků simulace a volbě průběhů veličin, které se mají zobrazit. Součástí grafu je pak zobrazena i legenda, popisy os a název zvolené simulace. Ve spodní části okna je možnost výběru průběhů, které chceme zobrazit. Změna je provedena hned, takže není potřeba znovu spouštět simulaci. Vykreslený průběh je možno i přiblížit nebo oddálit a pokud možno posunout za pomoci ikon z lišty v levém horním rohu.

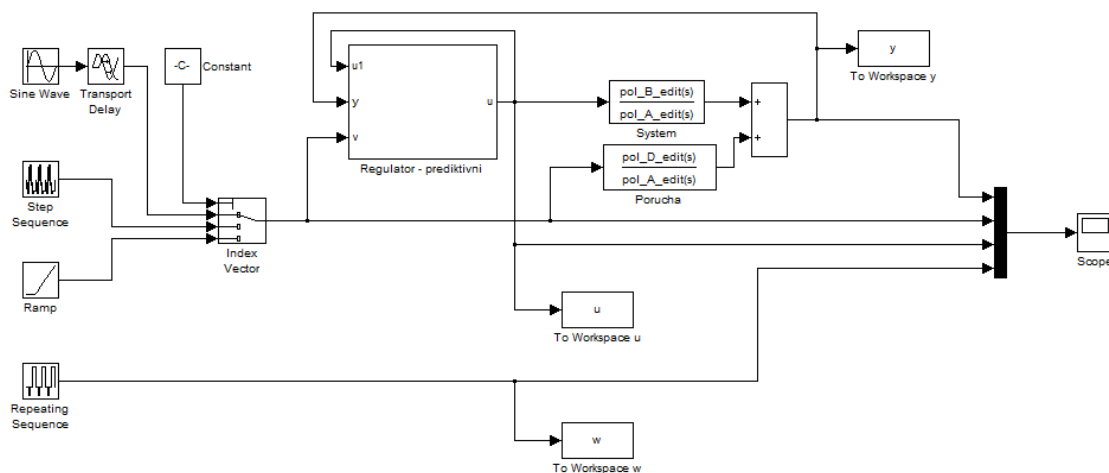
4.2 Popis simulačních schémat v Simulinku

4.2.1 Simulace řízení bez kompenzace poruchy



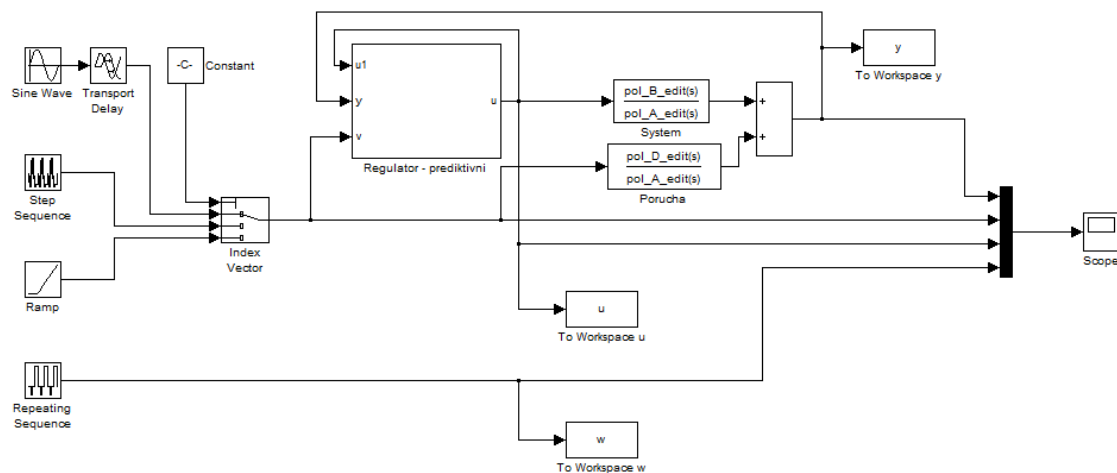
Obr. 4.8.: Schéma simulace řízení bez kompenzace poruchy v prostředí Simulink

4.2.2 Simulace řízení s neznámou měřitelnou poruchou



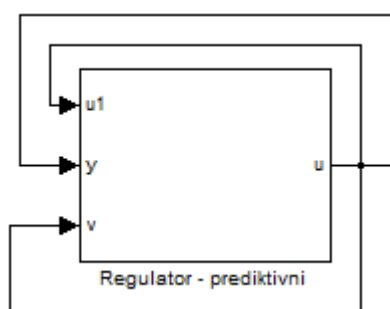
Obr. 4.9.: Schéma simulace řízení s neznámou měřitelnou poruchou v prostředí Simulink

4.2.3 Simulace řízení se známou poruchou



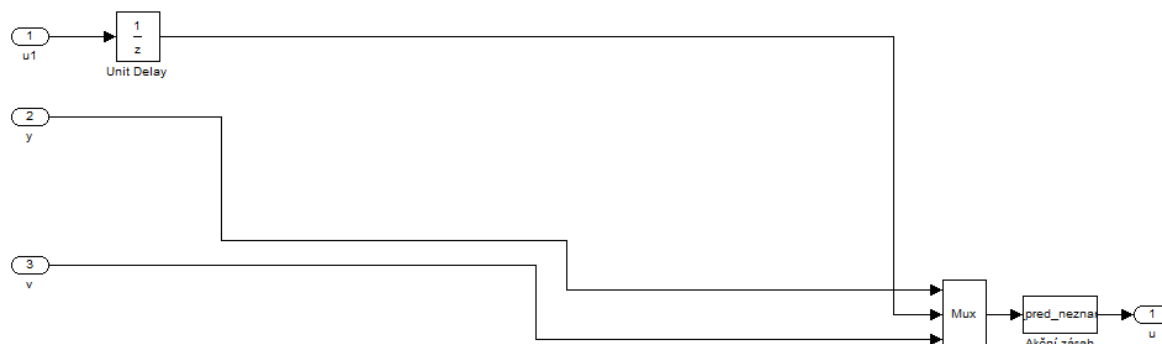
Obr. 4.10.: Schéma simulace řízení se známou poruchou v prostředí Simulink

4.2.4 Blok prediktivního regulátoru



Obr. 4.11.: Blok
prediktivního regulátoru

Blok byl vytvořen v prostředí Simulink za pomoci funkce Subsystem. Tento blok zpracovává hodnoty vstupních veličin a počítá akční zásah.



Obr. 4.12.: Vnitřní schéma bloku prediktivního regulátoru

Pomocí následujícího okna se nastavují parametry bloku prediktivního regulátoru.

Function Block Parameters: Regulator - prediktivni

Pocatek horizontu predikce N1
1

Konec horizontu predikce N2
10

Horizont rizeni Nu
10

Penalizacni faktor lambda
0.1

Zadej sekvenci zadane hodnoty RS_w
[5 5 3 3 5 5 10 10]

Zadej casovou posloupnost RS_t
[0 50 50 100 100 170 170 250]

zadej omezeni [umin umax]
[umin_edit umax_edit]

zadej periodu vzorkovani
tv_edit

Parametry matice A [a1,a2]
[disk_A(2) disk_A(3)]

Parametry matice B [b1,b2]
[disk_B(2) disk_B(3)]

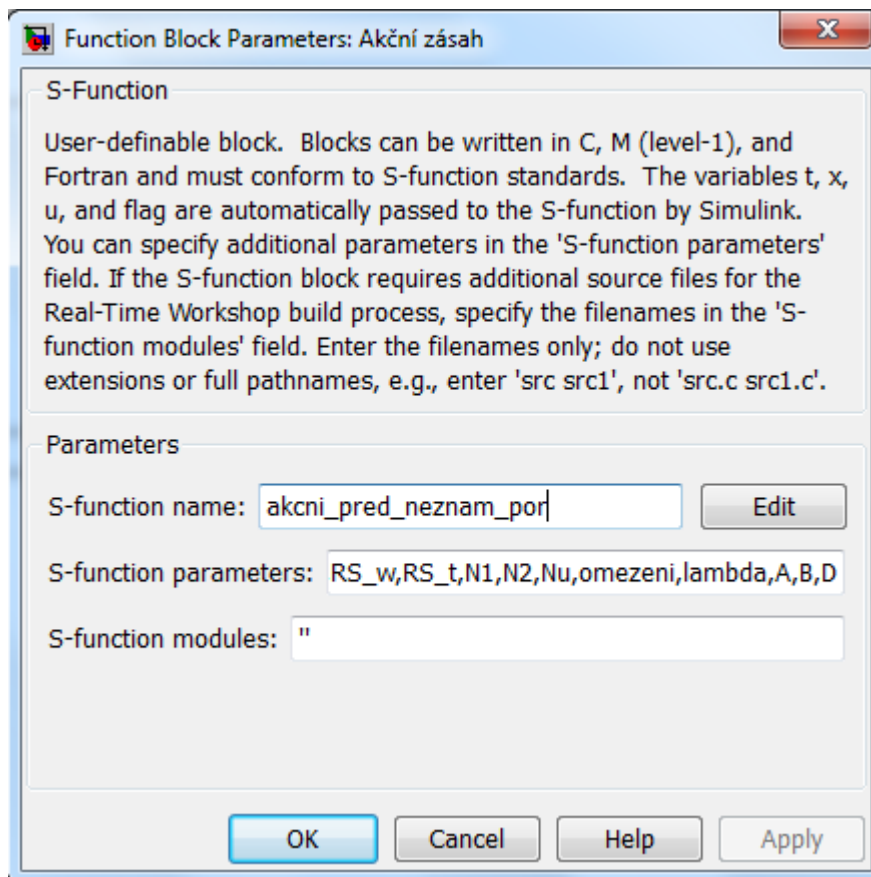
Parametry polynomu D [d1,d2]
[disk_D(2) disk_D(3)]

OK Cancel Help Apply

Obr. 4.13.: Nastavení parametrů bloku prediktivního regulátoru

4.2.5 Blok akční zásah

Nachází se vnitřním schématu bloku prediktivního regulátoru na *Obr. 4.12*. Je vytvořen za pomoci S-funkce.



Obr. 4.14.: Vstupní parametry bloku akční zásah

Kde S-function name odkazuje na m-file, který obsahuje kód pro výpočet akčního zásahu. S-function parameters jsou vstupní parametry do m-filu.

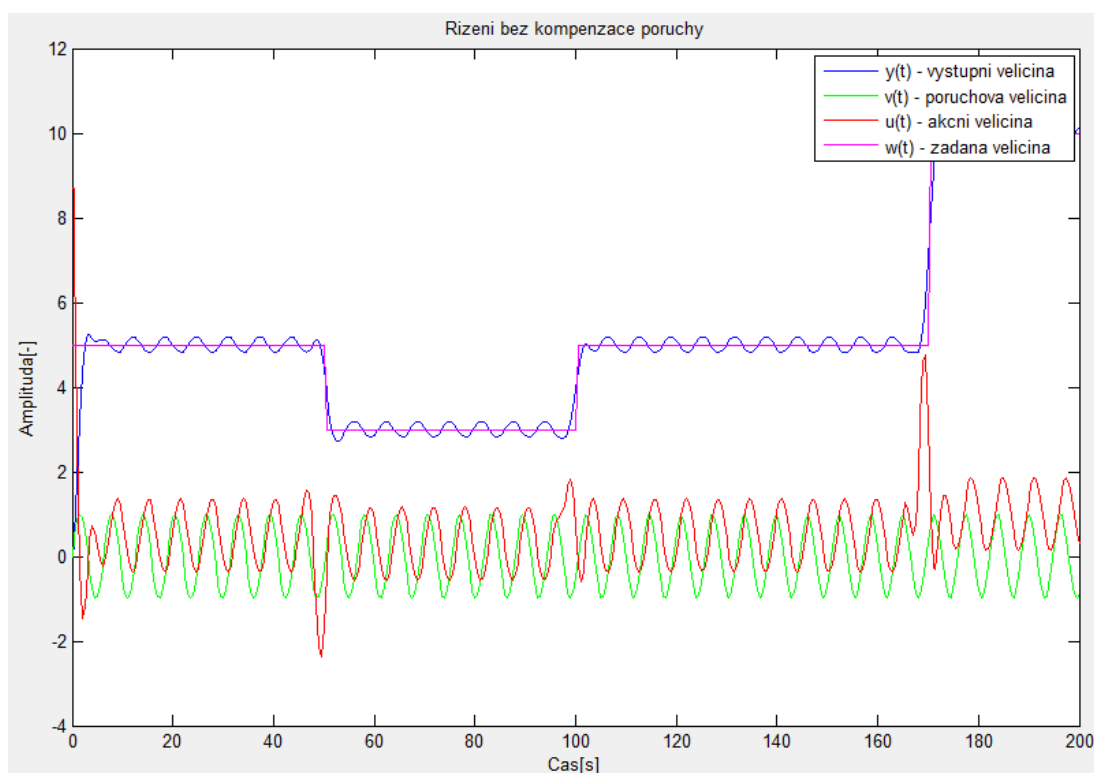
5 SIMULAČNÍ OVĚŘOVÁNÍ

5.1 Simulace řízení bez kompenzace poruchy

V následující simulaci vůbec nepočítáme s poruchovou veličinou. Simulace byla spuštěna s původním nastavením. Schéma simulace je na *Obr. 4.8*.

5.1.1 Porucha – Sinus

Následující simulace byla provedena pro poruchu sinus.

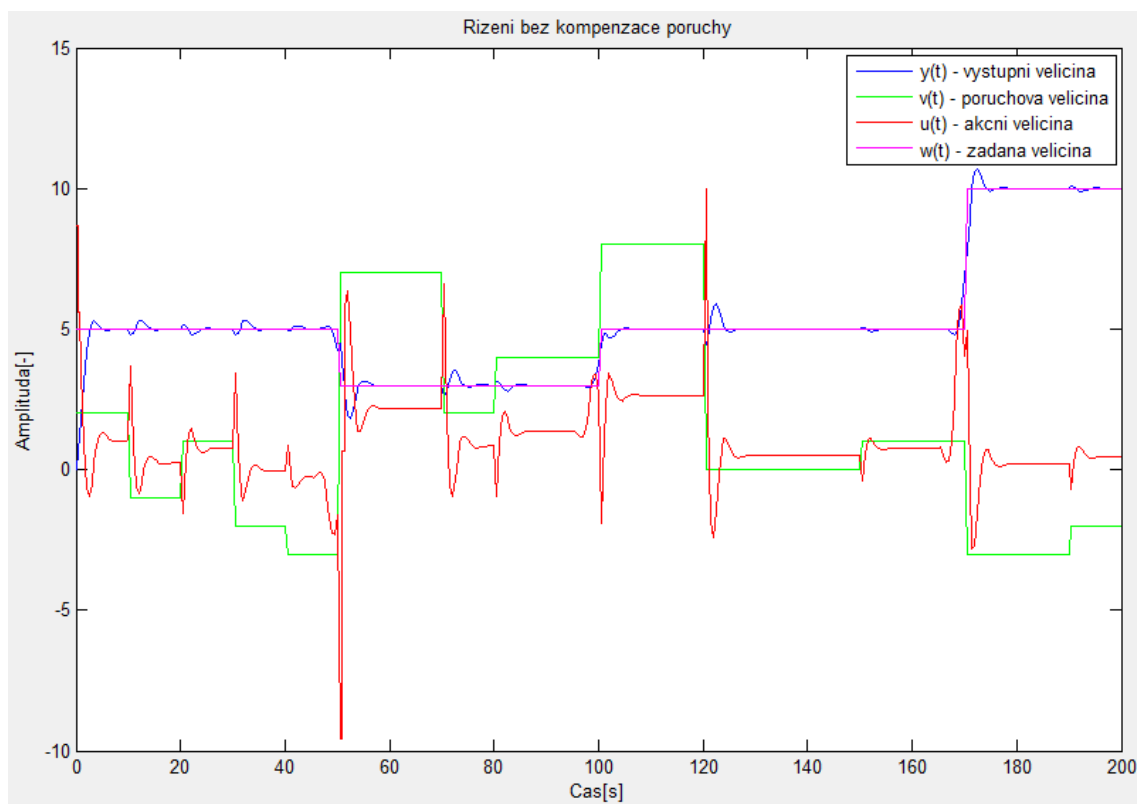


Obr. 5.1.: Výsledek simulace řízení bez kompenzace poruchy pro poruchu sinus

Jak je vidět z grafu, tak poruchová veličina má vliv na akční veličinu, a tudíž se to projeví i na výstupní veličině.

5.1.2 Porucha – Skok

Jako další byla provedena simulace se skokovou poruchou.

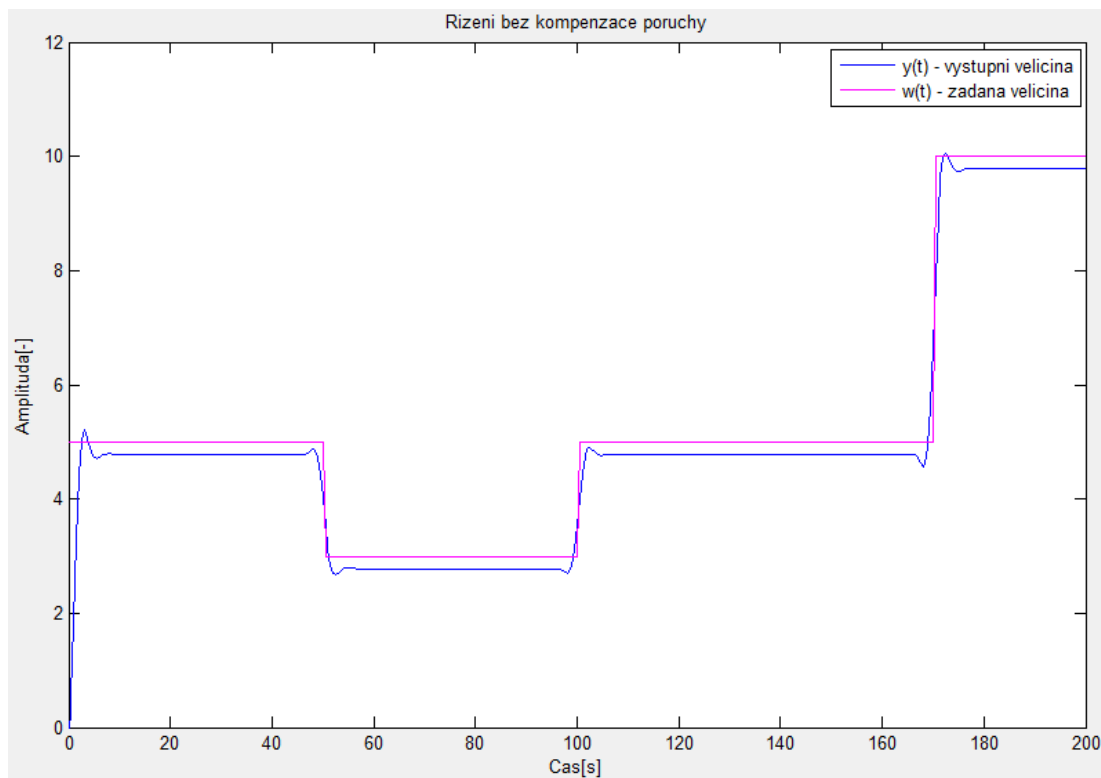


Obr. 5.2.: Výsledek simulace řízení bez kompenzace poruchy pro skokovou poruchu

Jak je vidět z grafu, tak poruchová veličina se více promítla do akční veličiny, která začala ihned reagovat, to se ale ve výstupní veličině projevilo menším rozkmitáním, které se ale po chvíli ustálilo.

5.1.3 Porucha – Rampa

Jako další byla provedena simulace s poruchou rampa. Z následujícího výsledného grafu musely být odstraněny průběhy akční a poruchové veličiny kvůli nepřehlednosti.



Obr. 5.3.: Výsledek simulace řízení bez kompenzace poruchy pro poruchu rampa

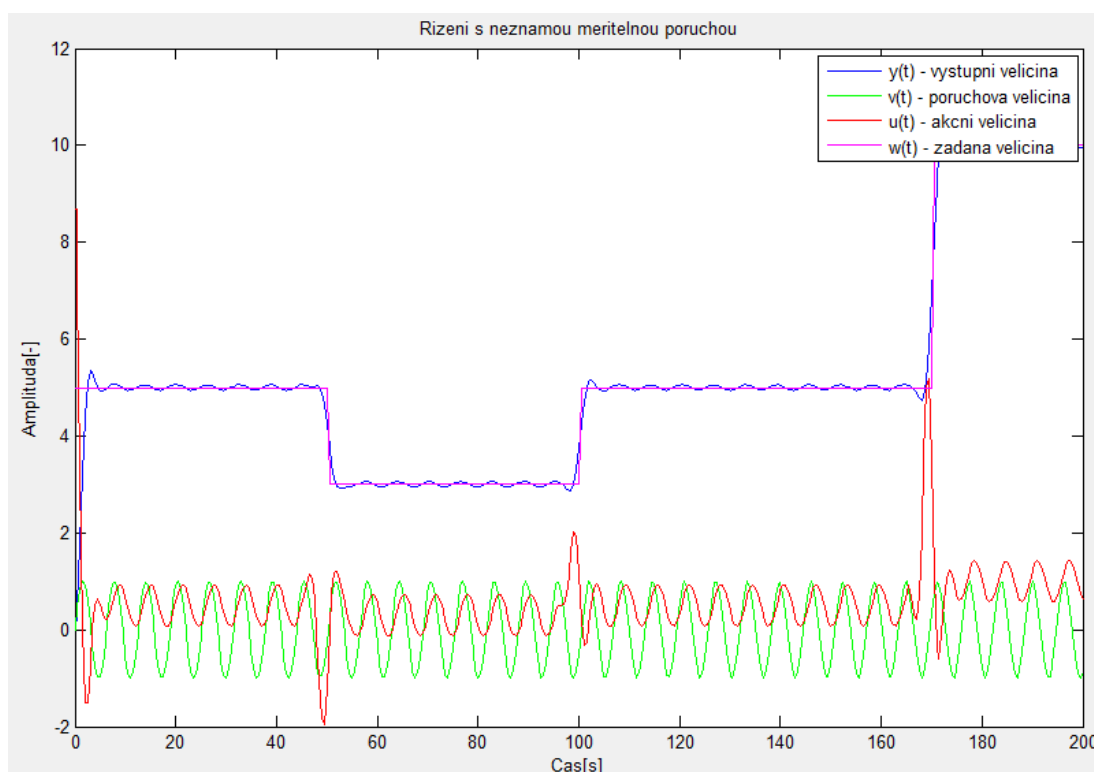
Jak je z grafu patrné, tak došlo k trvalému snížení posunu regulační odchylky.

5.2 Simulace řízení s neznámou měřitelnou poruchou

V následující simulaci počítáme s poruchovou veličinou tak, že ji v celém horizontu řízení uvažujeme jako konstantní a v dalším kroku provedeme tu samou operaci. Simulace byla spuštěna s původním nastavením. Schéma simulace je na *Obr. 4.9*.

5.2.1 Porucha – Sinus

Následující simulace byla provedena pro poruchu sinus.

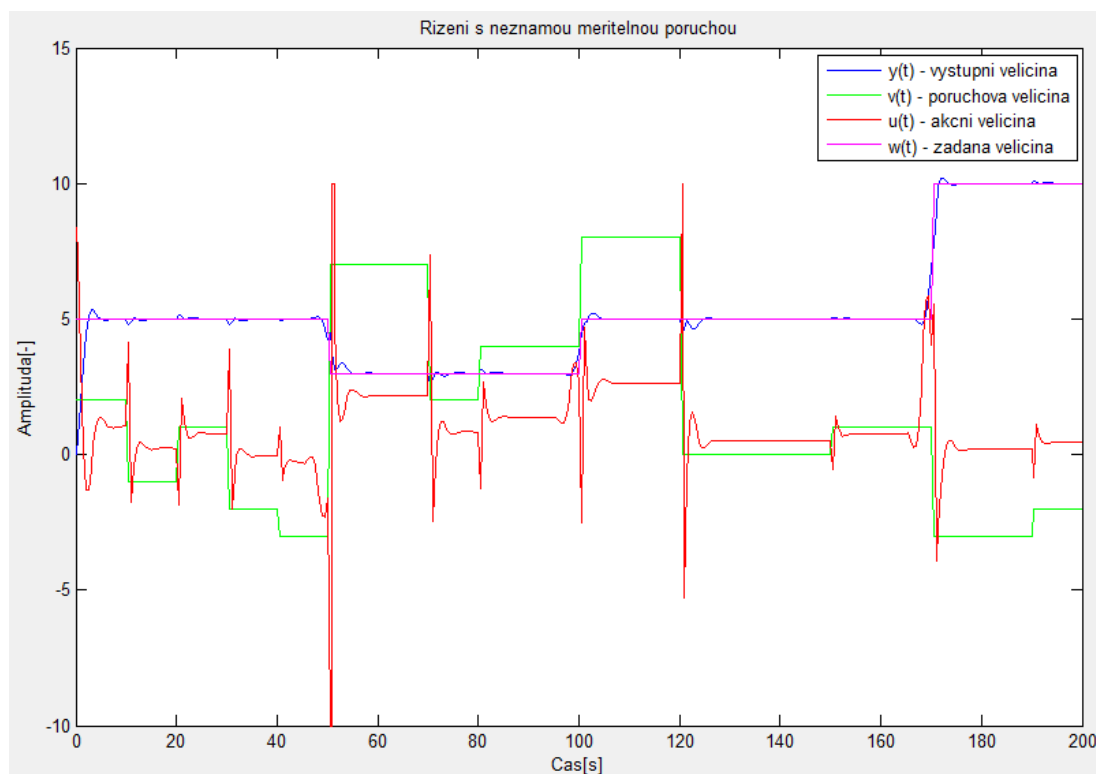


Obr. 5.4.: Výsledek simulace řízení s neznámou měřitelnou poruchou pro poruchu sinus

Zde je vidět, že byla lépe vykompenzována poruchová veličina na rozdíl od předchozí verze řízení bez kompenzace poruchy. Na výstupní veličině je vidět mírné kmitání.

5.2.2 Porucha – Skok

Dále byla provedena simulace, kde porucha byla změněna na skokovou.

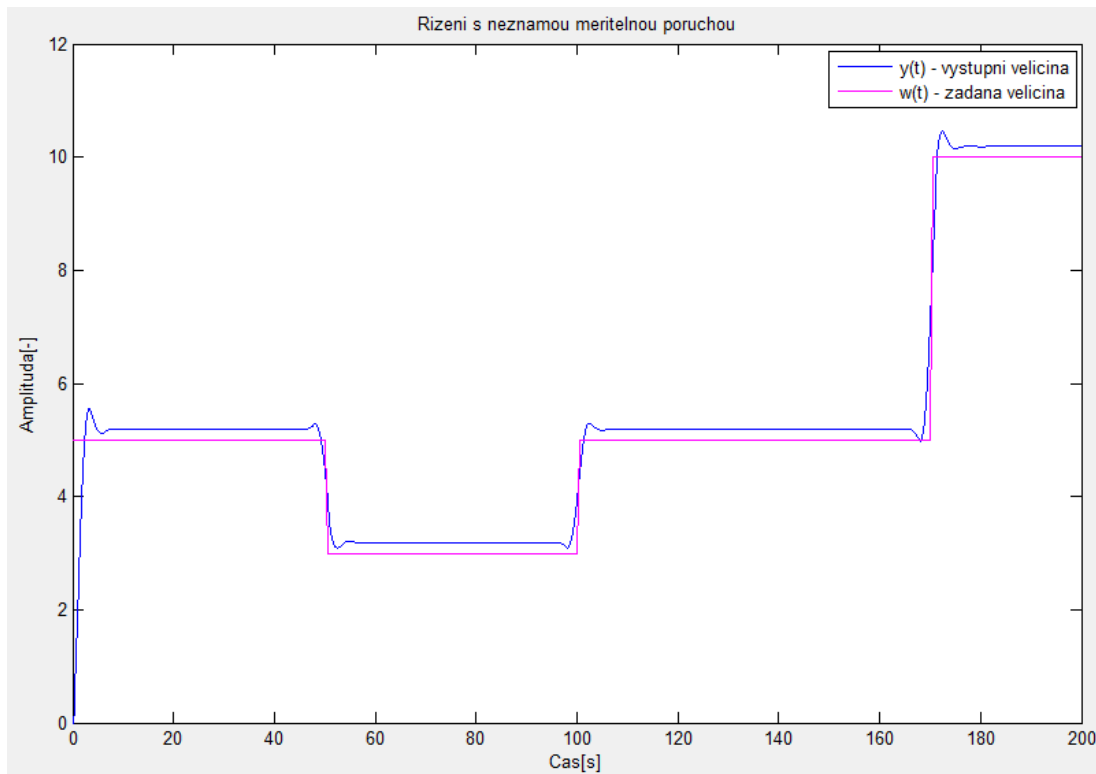


Obr. 5.5.: Výsledek simulace řízení s neznámou měřitelnou poruchou pro skokovou poruchu

Zde je taky vidět zlepšení, na rozdíl od předchozí verze řízení bez kompenzace poruchy. Ve výstupní veličině je vidět menší rozkmitáním, které se ale po chvíli ustálí.

5.2.3 Porucha – Rampa

Jako další byla provedena simulace pro poruchu rampa. Z následujícího výsledného grafu musely být taky odstraněny průběhy akční a poruchové veličiny kvůli nepřehlednosti.



*Obr. 5.6.: Výsledek simulace řízení s neznámou měřitelnou poruchou pro poruchu
rampa*

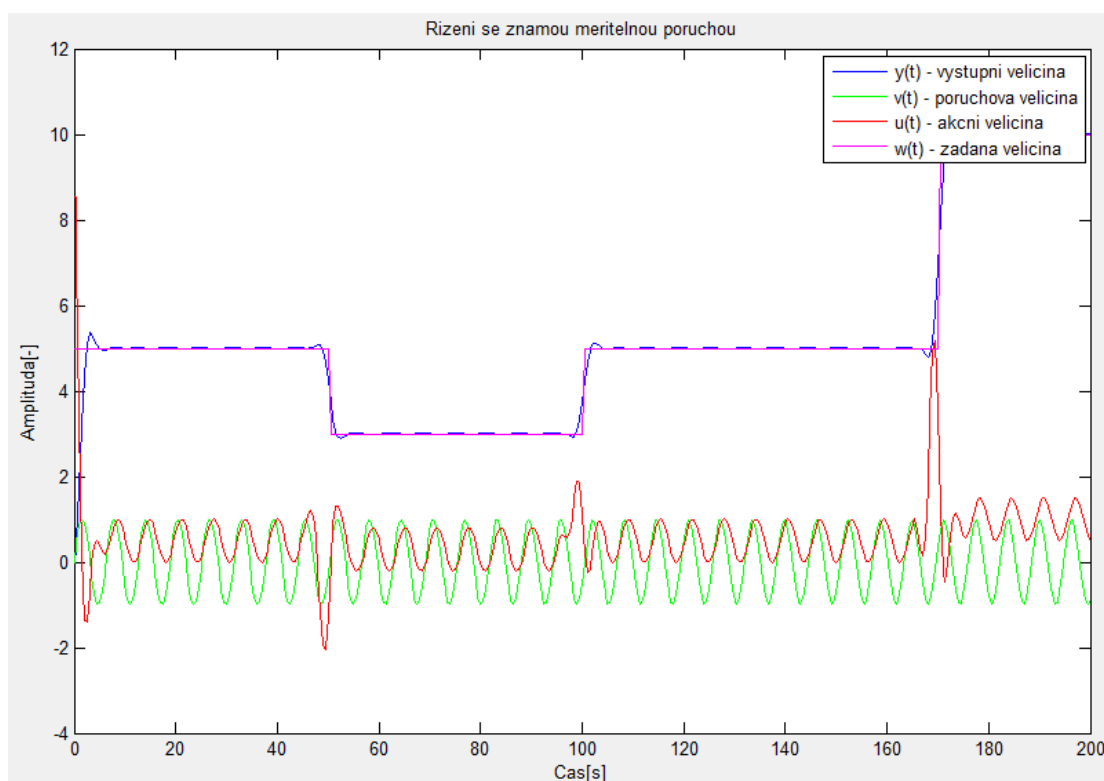
Na rozdíl od verze řízení bez kompenzace poruchy zde došlo ke snížení trvalé regulační odchylky.

5.3 Simulace řízení se známou poruchou

V následující simulaci známe přesně průběh poruchové veličiny a počítáme s ní při výpočtech. Simulace byla spuštěna s původním nastavením. Schéma simulace je na *Obr. 4.10*.

5.3.1 Porucha – Sinus

Tato simulace byla provedena pro poruchu sinus.

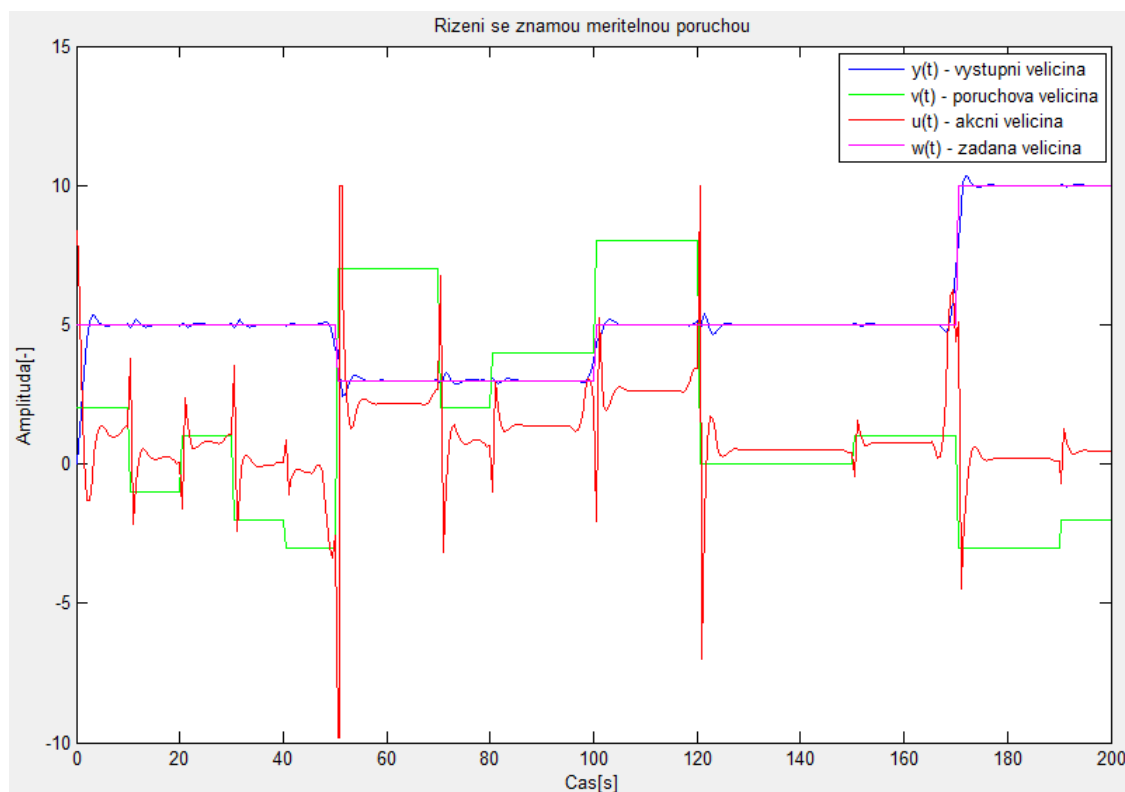


Obr. 5.7.: Výsledek simulace řízení se známou poruchou pro poruchu sinus

Z grafu je vidět, že poruchová veličina je velmi dobře vykompenzována, a tudíž výstupní veličina je téměř bez kmitů, na rozdíl od předchozích dvou verzí řízení.

5.3.2 Porucha – Skok

Jako další byla provedena simulace se skokovou poruchou.

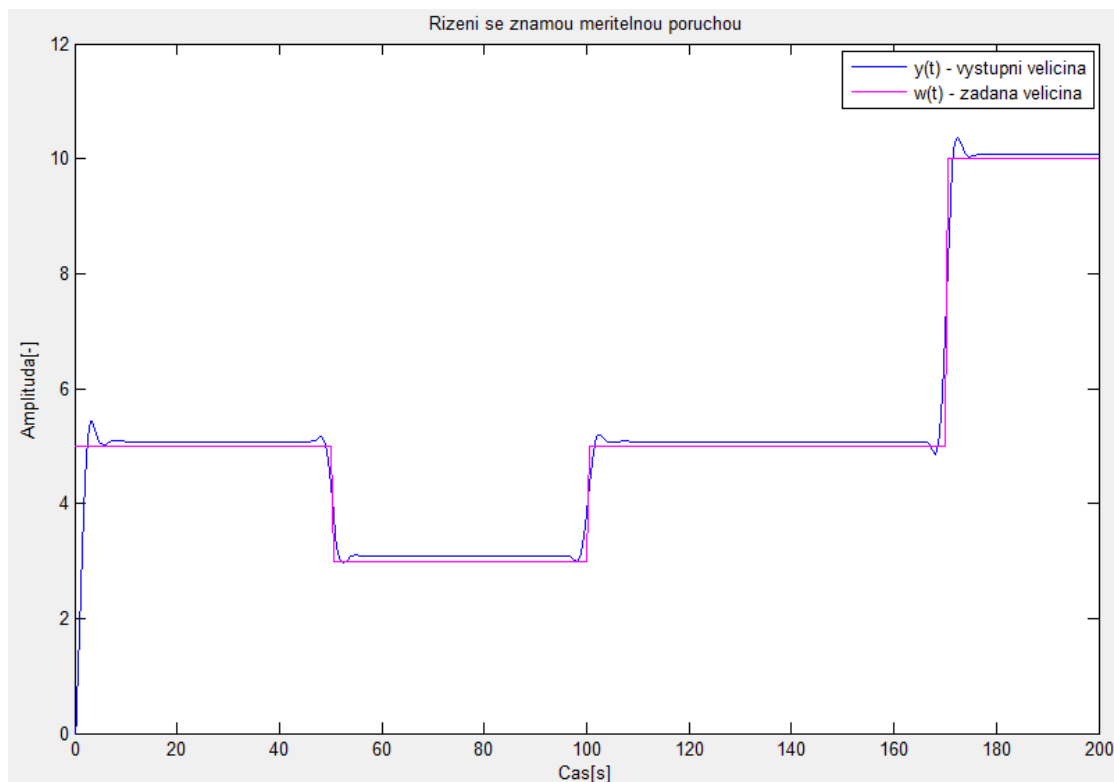


Obr. 5.8.: Výsledek simulace řízení se známou poruchou pro skokovou poruchu

Na rozdíl od předchozího bodu kde došlo ke zlepšení, tak zde jsou průběhy téměř shodné s verzí řízení s neznámou měřitelnou poruchou. I když poruchovou veličinu známe, tak se ji nedokázalo dobře vykompenzovat.

5.3.3 Porucha – Rampa

Jako další byla provedena simulace pro poruchu rampa. Z následujícího výsledného grafu musely být taky odstraněny průběhy akční a poruchové veličiny kvůli nepřehlednosti.



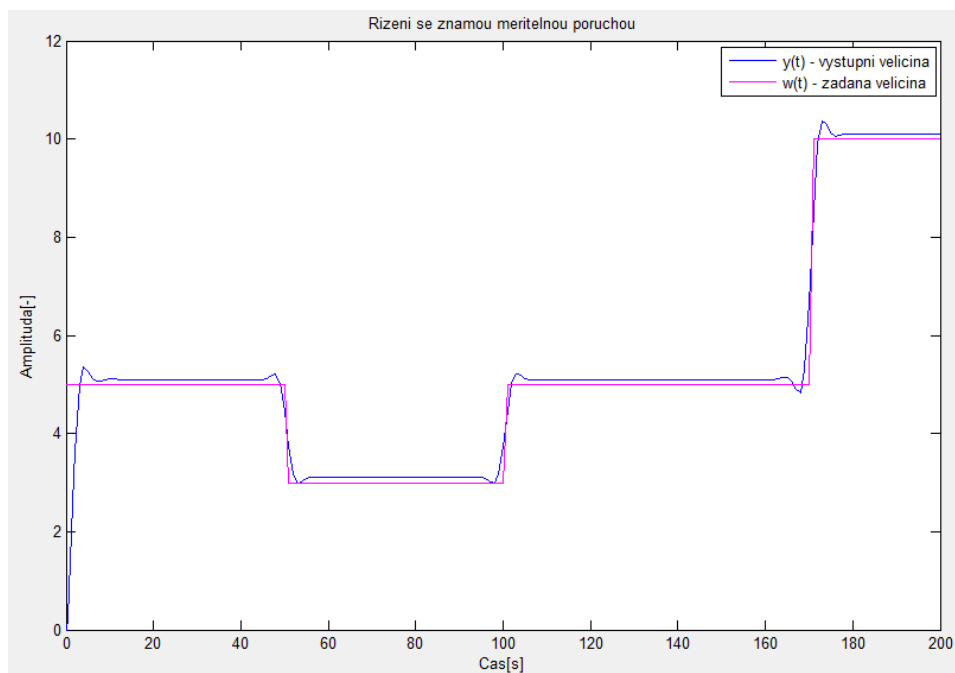
Obr. 5.9.: Výsledek simulace řízení se známou poruchou pro poruchu rampa

I zde je ale vidět jako u verze řízení s neznámou měřitelnou poruchou, že došlo ke snížení trvalé regulační odchylky. Průběh výstupní veličiny se ještě více přiblížil žádané veličině.

5.4 Porovnání různých hodnot periody vzorkování

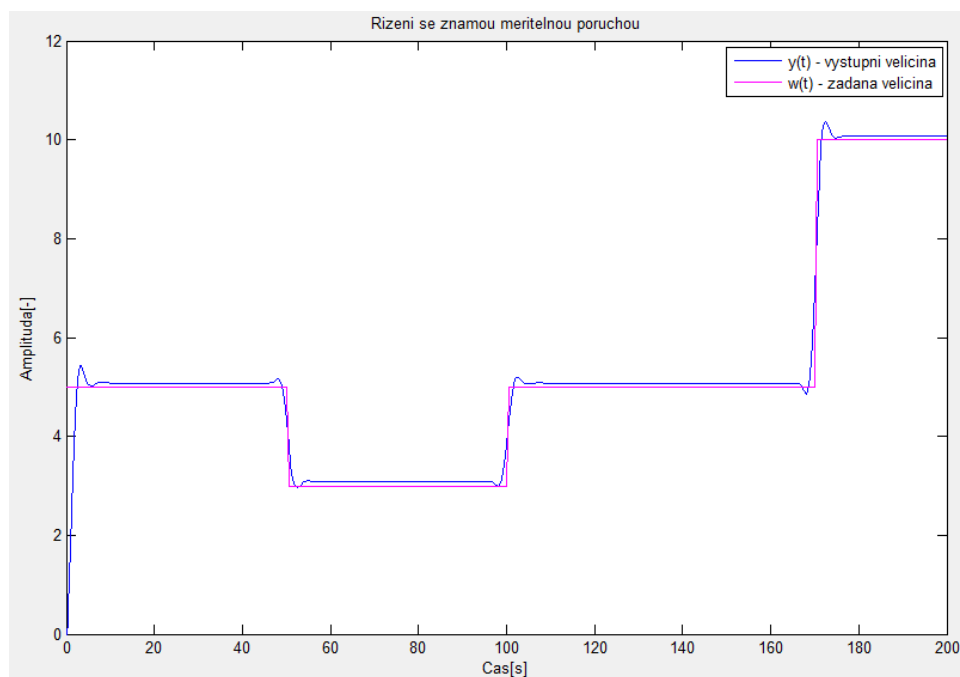
Porovnání bylo provedeno pro simulaci řízení se známou poruchou pro poruchu rampa.

5.4.1 Perioda vzorkování $T_v=1$



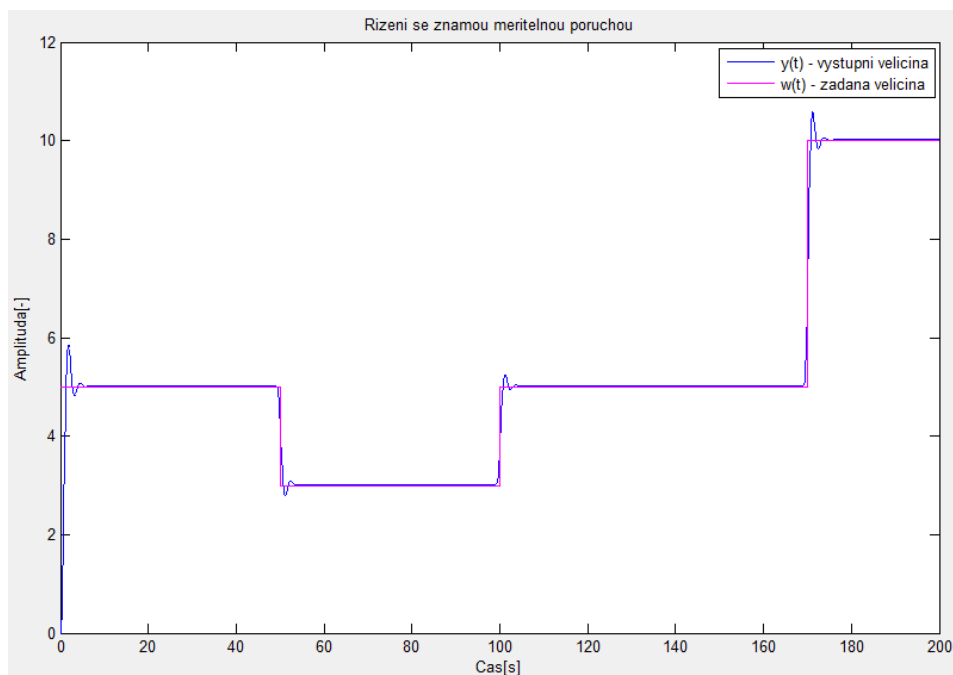
Obr. 5.10.: Simulace pro periodu vzorkování $T_v=1$

5.4.2 Perioda vzorkování $T_v=0,5$



Obr. 5.11.: Simulace pro periodu vzorkování $T_v=0,5$

5.4.3 Perioda vzorkování $T_v=0,1$



Obr. 5.12.: Simulace pro periodu vzorkování $T_v=0,1$

Jak lze vidět grafů, tak s rostoucí periodou vzorkování roste taky trvalá regulační odchylka a naopak. Se snižující se periodou roste schopnost výstupní veličiny, přesně kopírovat žádanou veličinu na úkor toho, že na začátku dojde vždy k většímu překmitu (podkmitu).

ZÁVĚR

Cílem této diplomové práce bylo vytvořit simulace prediktivního regulátoru s kompenzací poruchy v prostředí programu MATLAB. Simulační schémata byla vytvořena v prostředí Simulink, který je součástí programu MATLAB.

Jako model pro systém a známou poruchu byla zvolena vstupně výstupní přenosová funkce a dále byl pak zvolen CARIMA model, který má v sobě zahrnutou i poruchu a šum, se kterým jsme v této práci nepočítali. Výpočet prediktoru regulátoru byl proveden za pomoci dvou metod, a to za pomoci diofantických rovnic a metodou přímého výpočtu z CARIMA modelu. Po provedení výpočtů lze učinit závěr, že i když je každá metoda výpočetně jiná, tak konečný výsledek je vždy stejný. Tudíž v této práci je poté v MATLABu použita jen jedna z metod. Podrobně jsou zde popsány bloky jednotlivých simulací a simulačních schémat.

V práci je porovnáno prediktivní řízení bez kompenzace poruchy, řízení s neznámou měřitelnou poruchou a řízení se známou poruchou. Jako zdroj poruchy zde bylo použito funkce sinus, skokové změny poruchové veličiny a rampy. I když bylo řízení se známou poruchou nejkvalitnější, tak se skokovou změnou poruchové veličiny si nedokázalo tak dobře poradit.

Byl také porovnán vliv změny periody vzorkování na kvalitu řízení. Při snižování periody vzorkování výstupní veličina lépe kopíruje průběh žádané veličiny.

Pro tvorbu simulací byla využita aplikace GUIDE, která je součástí programu MATLAB a slouží k tvorbě grafických aplikací.

SEZNAM POUŽITÉ LITERATURY

- [1] BOBÁL, Vladimír. Adaptivní a prediktivní řízení. Vyd. 1. Zlín: Univerzita Tomáše Bati ve Zlíně, 2008, 134 s. ISBN 978-80-7318-662-3.
- [2] MIKLĚŠ, J a Miroslav FIKAR. Process modelling, identification, and control. New York: Springer, c2007, xxiv, 480 p. ISBN 3540719695.
- [3] CAMACHO, E a C BORDONS. Model predictive control. New York: Springer, c2004, xxii, 405 p. ISBN 1852336943.
- [4] MIKLĚŠ, J a Miroslav FIKAR. Modelovanie, identifikácia a riadenie procesov 2. Identifikácia a optimálne riadenie. Bratislava: STU Press, 2004, 260 pp, ISBN 80-227-2134-4.
- [5] ROSSITER, J. Model-based predictive control: a practical approach. Boca Raton: CRC Press, c2003, 318 p. CRC Press control series. ISBN 0849312914.
- [6] CLARKE, D.W. a MOHTADI, C. a TUFFS, P.S.. Generalized Predictive Control. Part I. The Basic Algorithm. Automatica, 23(2): 137–148, 1987
- [7] CLARKE, D.W. a MOHTADI, C. a TUFFS, P.S.. Generalized Predictive Control. Part II. Extensions and Interpretations. Automatica, 23. 149–160, 1987
- [8] MORARI, M. a LEE, J.H.. Model predictive control: Past, present and future. Computers and Chemical Engineering, 23. 667–682, 1999
- [9] Humusoft: *MATLAB* [online]. [cit. 2015-05-05]. Dostupné z WWW: <<http://www.humusoft.cz/produkty/matlab/matlab/>>
- [10] Wikipedia: *MATLAB* [online]. [cit. 2015-05-05]. Dostupné z: <<http://cs.wikipedia.org/wiki/MATLAB>>
- [11] Humusoft: *Simulink* [online]. [cit. 2015-05-05]. Dostupné z WWW: <<http://www.humusoft.cz/produkty/matlab/simulink/>>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

MATLAB	MATrix LABoratory.
GUIDE	Graphical User Interface Development Environment.
FORTTRAN	FORmula TRANslator.
MPC	Model Predictive Control
GPC	Generalized Predictive Control
PID	proporcionální, integrační a derivační část
$A(z^{-1})$	polynom jmenovatele Z přenosu systému a známé poruchy
$B(z^{-1})$	polynom čitatele Z přenosu systému
$D(z^{-1})$	polynom čitatele Z přenosu známé poruchy
a_1, a_2	koeficienty polynomu A
b_1, b_2	koeficienty polynomu B
d_1, d_2	koeficienty polynomu D
$E_s(z^{-1})$	neměřitelný náhodný šum
$C(z^{-1})$	polynom čitatele Z přenosu
j	krok predikce
w	žádaná veličina
y	výstupní veličina
\hat{y}	predikovaná výstupní veličina
v	poruchová veličina
\hat{v}	vektor budoucích změn poruchové veličiny
e	regulace odchylky
e_s	neměřitelná porucha
$E(z^{-1})$	polynom použitý při výpočtu diofantických rovnic
$F(z^{-1})$	polynom použitý při výpočtu diofantických rovnic

$G_s(z^{-1})$	Z přenos modelu systému
$G_v(z^{-1})$	Z přenos modelu známé poruchy
u	akční veličina
Δu	vektor budoucích změn akčního zásahu
y_n	nucená odezva
y_0	volná odezva
T_v	perioda vzorkování
J	účelová funkce
G	matice dynamiky
L	matice dynamiky
N_u	řídící horizont
H	Hessova matice
λ	váhový koeficient

SEZNAM OBRÁZKŮ

<i>Obr. 2.1: Prostředí MATLAB</i>	21
<i>Obr. 2.2.: Ukázka prostředí GUIDE – program prediktivni_rizeni_s_poruchou.fig</i>	23
<i>Obr. 3.1.: Blokové schéma</i>	26
<i>Obr. 4.1.: Prostředí simulačního programu</i>	30
<i>Obr. 4.2.: Blok výběru prediktivního řízení</i>	31
<i>Obr. 4.3.: Blok přenosu modelu</i>	32
<i>Obr. 4.4.: Blok periody vzorkování</i>	32
<i>Obr. 4.5.: Blok omezení akční veličiny</i>	32
<i>Obr. 4.6.: Blok nastavení poruchy</i>	33
<i>Obr. 4.7.: Blok průběhu simulace</i>	34
<i>Obr. 4.8.: Schéma simulace řízení bez kompenzace poruchy v prostředí Simulink</i>	35
<i>Obr. 4.9.: Schéma simulace řízení s neznámou měřitelnou poruchou v prostředí Simulink</i>	35
<i>Obr. 4.10.: Schéma simulace řízení se známou poruchou v prostředí Simulink</i>	36
<i>Obr. 4.11.: Blok prediktivního regulátoru</i>	36
<i>Obr. 4.12.: Vnitřní schéma bloku prediktivního regulátoru</i>	36
<i>Obr. 4.13.: Nastavení parametrů bloku prediktivního regulátoru</i>	37
<i>Obr. 4.14.: Vstupní parametry bloku akční zásah</i>	38
<i>Obr. 5.1.: Výsledek simulace řízení bez kompenzace poruchy pro poruchu sinus</i>	39
<i>Obr. 5.2.: Výsledek simulace řízení bez kompenzace poruchy pro skokovou poruchu</i>	40
<i>Obr. 5.3.: Výsledek simulace řízení bez kompenzace poruchy pro poruchu rampa</i>	41
<i>Obr. 5.4.: Výsledek simulace řízení s neznámou měřitelnou poruchou pro poruchu sinus</i>	42
<i>Obr. 5.5.: Výsledek simulace řízení s neznámou měřitelnou poruchou pro skokovou poruchu</i>	43
<i>Obr. 5.6.: Výsledek simulace řízení s neznámou měřitelnou poruchou pro poruchu rampa</i>	44
<i>Obr. 5.7.: Výsledek simulace řízení se známou poruchou pro poruchu sinus</i>	45
<i>Obr. 5.8.: Výsledek simulace řízení se známou poruchou pro skokovou poruchu</i>	46
<i>Obr. 5.9.: Výsledek simulace řízení se známou poruchou pro poruchu rampa</i>	47
<i>Obr. 5.10.: Simulace pro periodu vzorkování $T_v=1$</i>	48
<i>Obr. 5.11.: Simulace pro periodu vzorkování $T_v=0,5$</i>	48

<i>Obr. 5.12.: Simulace pro periodu vzorkování $T_V=0,1$</i>	49
---	----

SEZNAM PŘÍLOH

Příloha č. 1: Obsah disku CD-ROM	53
--	----

PŘÍLOHA P I: OBSAH DISKU CD-ROM

- Simulace
 - akcni_pred_bez_komp_por.m
 - akcni_pred_neznam_por.m
 - akcni_pred_znam_por.m
 - prediktivni_rizeni_s_poruchou.m
 - prediktivni_rizeni_s_poruchou.fig
 - rizeni_bez_komp_por.m
 - rizeni_neznam_por.m
 - rizeni_znam_por.m
- fulltext.pdf